

ITRON/68K (ITOS68K)

: ITRONを68000にインプリメントした実例

矢部栄一、竹山寛 (日立製作所)

堀越健一 (日立米沢電子)

1. はじめに

超LSI技術が十分使えることを前提とした、ノイマン型コンピュータの新しい体系を作るTRON (The Realtime Operating system Nucleus) プロジェクトが、東京大学坂村健博士により1984年6月開始された。本プロジェクトの最終目的は、コンピュータを構成する全てのコンポーネント、CPUチップから、オペレーティングシステム、マン=マシンインタフェースまでを今一度、従来のものにとらわれずに設計することである。

TRONプロジェクトでは、つぎの3つのシリーズからなる標準モデルを規定している。

(1) ITRON (Industrial-TRON)

機器組み込み型のリアルタイム・マルチタスク・オペレーティングシステムで、各種制御用に使用できる。

(2) BTRON (Business-TRON)

多国語処理、ファイル管理、マン=マシンインタフェース、通信機能を強化したワークステーション用オペレーティングシステムで、応用プログラム間でのデータ交換機能までを含む。

(3) MTRON (Macro-TRON)

環境制御のための分散処理オペレーティングシステムである。

これらのOSのうち、計測・制御システムなど産業用に用いられるITRONを68000上にインプリメントした、高性能リアルタイム・オペレーティングシステムITOS68Kについて、その概要を報告する。

2. ITO S68Kの設計方針

リアルタイムOSの対象分野には、石油・鉄鋼プラント等の各種工業用製造装置の制御、自動車・ロボットの制御、医療機器のデータ収集、レーダーのデータ計算等、数え切れない程の分野がある。とくに図-1に示すように、マイクロプロセッサの発展に伴って今までコスト、スペース的に考えられなかった分野にまで広がって来ている。

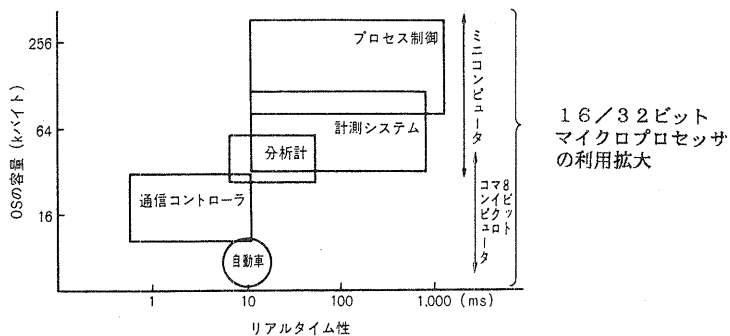


図-1 リアルタイムOSに対する要求

ITOS 68Kはこれらの応用分野を想定し、以下のような設計方針で開発を行った。

- (1) ITRON/68K仕様を完全に満足していること。
- (2) 制御対象から受け取った信号を解析し、制限時間内に処理結果を返さなければならぬため、高い応答性を持つこと。
- (3) 制御機器に組み込まれて使われる場合が多いため、ROM化が可能なこと。
- (4) ユーザシステムによって異なるハードウェア構成に容易に適応でき、拡張や仕様変更に対しても柔軟に追従できること。
- (5) 計測・制御など各種産業用機器に応用されるため高い信頼性を持つこと。

これらの方針のうち(2)の応答性については、割り込みハンドラに対するオペレーティングシステムのオーバーヘッドをなくし、システムコールのサービス中でもなるべく割り込みを可能にした。また高速化のため、ITOS 68Kは68000アセンブリ言語で記述している。(3)、(4)、(5)についてはITRONの設計思想である階層構造化を行い、さらにインプリメント時に各所をモジュール化することにより対応している。

本発表ではITRON仕様の68000へのインプリメント方法のほか、ITOS 68Kのシステム構成、I/Oドライバの実現方法、システムインストール手順などについて述べる。

3. 68000アーキテクチャ

68000は高性能、柔軟性(フレキシビリティ)、信頼性、使い易さを追求した、内部バスが32ビット構成の16ビットマイクロプロセッサである。68000のレジスタは図-2に示すようにつぎのようなレジスタで構成される。

- (1) 32ビットのデータレジスタ × 8本(D0~D7)
- (2) 32ビットのアドレスレジスタ × 9本(A0~A7)
- (3) 32ビットのプログラムカウンタ × 1本
- (4) 32ビットのステータスレジスタ × 1本

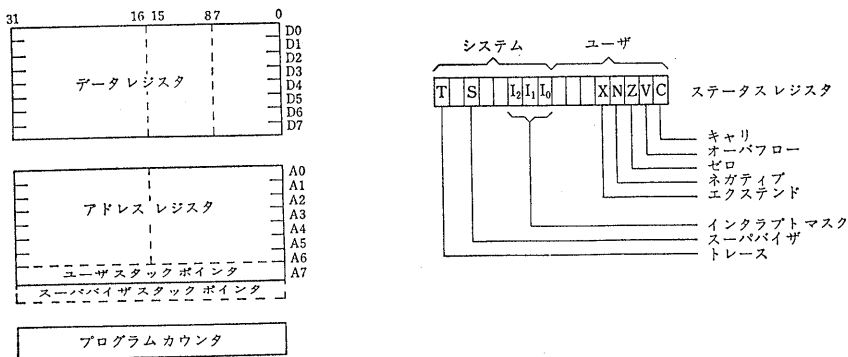


図-2 68000レジスタ構成

アドレスレジスタA7はスタックポインタとして使えるようになっており、ユーザ状態にあるときは図の上側のユーザスタックポインタUSPが選択され、スーパーバイザ状態にあるときは下側のスーパーバイザスタックポインタSSPが選択される。ステータスレジスタはキャリーC、オーバーフローV、ゼロZ、ネガティブN、エクステンDの各フラグで構成されるコンディションコードCCの部分と、システム用のT、S、I₁~I₂の各フラグとで構成される。スーパーバイザSのフラグが1であることはスーパーバイザ状態にあることを示す。I₀~I₂は割り込みマスクビットであり、マスクすべき割り込み優先レベルを示す。

4. 68000へのインプリメント

I TRONは、複数のマイクロプロセッサを対象とした標準オペレーティングシステムである。しかし、I TRONはリアルタイム・オペレーティングシステムとしての性能を重視しているため、I TRONをインプリメントする際は対象となるプロセッサの特性を生かして効率の向上を目指す必要がある。とくにリアルタイム、組み込み用のオペレーティングシステムにおいて、性能を要求される入出力制御などの部分では、ソースプログラムやオブジェクトコードの互換性がなくてもあまり支障はないので、ユーザがその実行環境に合わせて最適なI TRONシステムを構築できることが重要である。

4.1 特権モード

68000は実行モードとして、ユーザモードのほかにスーパーバイザモードと呼ばれる特権モードを持っている。

I TRON 68Kでは一般のユーザタスクをユーザモードで実行し、リセット処理、例外処理、外部割り込み処理、トラップ処理等のシステムプログラムの処理をスーパーバイザモード実行している。これにより、I TRON 68Kのシステムプログラムの処理をユーザタスクから保護することができる。たとえば、ステータスレジスタのモディファイは特権命令になっており、ユーザモードで実行されるユーザタスクからはステータスレジスタの割り込みマスクレベルを変更することはできない。また、68000 CPUからは表-1のようなプロセッサステータスを表わすファンクションコードが出力されるので、メモリ側にハード的な保護機構を設けることにより、ユーザタスクからのシステムプログラムへのアクセスを禁止することができる。さらに、ユーザモードからスーパーバイザモードへ移る際には、スタックポインタ(レジスタA7)もユーザスタックポインタUSPからスーパーバイザスタックポインタSSPに切り替わるので、プログラム・ワークエリアとしてのスタック領域を保護することができる。

表-1 68000プロセッサステータス

ファンクションコード			参照分類
FC2	FC1	FC0	
0	0	0	(未定義)
0	0	1	ユーザデータ
0	1	0	ユーザプログラム
0	1	1	(未定義)
1	0	0	(未定義)
1	0	1	スーパーバイザデータ
1	1	0	スーパーバイザプログラム
1	1	1	割り込みアクトリッジ

ユーザモードからスーパーバイザモードへの特権状態変更は、システムコールに使われるTRAP命令によって行われる。またスーパーバイザモードからユーザモードに戻す特権状態変更はRTE命令によって行われる。この模様を図-3に示す。

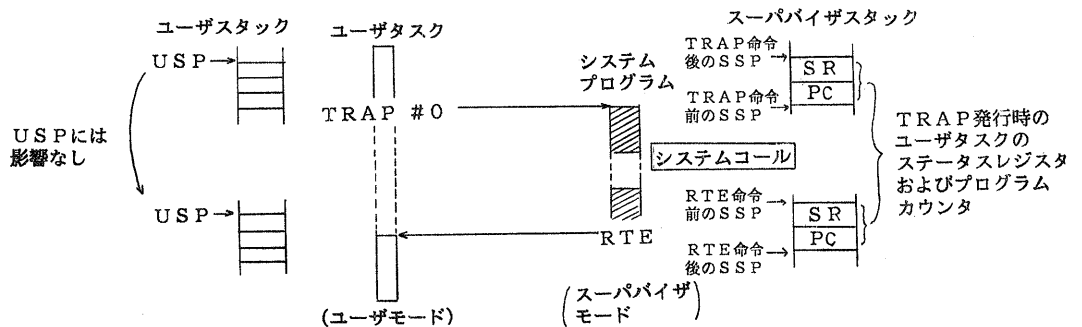


図-3 特権状態の変更

4. 2 割り込み処理

68000のもうひとつの特長は、192個のユーザ割り込みベクタを備えた協力的な割り込み管理機構を持っていることである。

外部割り込みはレベル1からレベル7までの割り込みレベルと割り込みベクタによって管理される。スーパーバイザモードのシステムプログラムは、ステータスレジスタの $I_2 \sim I_0$ ビットをモディファイすることにより、設定した割り込みレベルおよびそれ以下のレベルの割り込みをマスクすることができる。ただし、割り込みレベル7はノンマスクابلで、電源ダウンなどの非常時に使われる。

ITRON68Kでは68000の割り込み管理機構を利用し、def-int (define interrupt handler) システムコールにより割り込みベクタを設定する以外、OSは割り込みに全く介入しない。したがって、割り込み発生時にはユーザの割り込みハンドラが直接起動され、割り込みに対するOSのオーバーヘッドは基本的にゼロである。またCPUは、割り込みを受け付けると直ちにステータスレジスタの $I_2 \sim I_0$ ビットを当該割り込みのレベルにセットするので、よりレベルの低い割り込みは自動的にマスクされ、ユーザの割り込みハンドラでは割り込みマスク処理を行う必要がない。ただし割り込みハンドラは、割り込み終了時のハードウェアコンテキストを割り込み発生時前の状態に戻す必要があるため、使用するレジスタ類の退避や復帰を行わなければならない。

ITOS68Kにおけるユーザタスクとシステムサービス、割り込みハンドラの関係を図-4に示す。

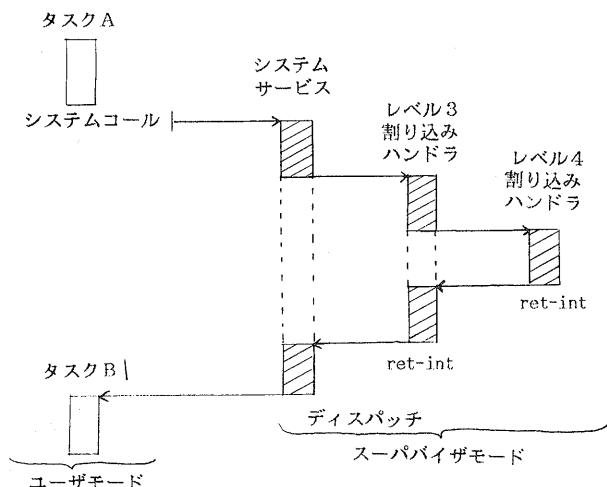


図-4 ユーザタスクとシステムサービス、割り込みハンドラの関係

タスクAから発行されたシステムコールのサービス中にレベル3割り込みが発生し、def-int システムコールによって定義された割り込みハンドラに制御が移った。この割り込みハンドラは割り込みマスクレベル3で実行されるので、レベル4割り込みが発生した時点で中断され、レベル4のベクタに対応する割り込みハンドラが実行される。レベル4割り込みハンドラ内でタスクBを起動し、ret-int システムコールにより処理を終了すると、ITRON68Kは多重割り込みであることを認識し、ディスパッチを行わずにレベル3割り込みハンドラを中断した時点から実行再開する。レベル3割り込みハンドラからret-int システムコールが発行された際も、ITOS68KではタスクAのシステムサービスを最後まで実行してからディスパッチを行い、よりプライオリティの高いタスクBへのタスクチェンジを行う。

4.3 例外処理

68000が管理する例外ベクタの一覧を表-2にしめす。ITOS68Kはこの中でベクタ番号2のバスエラーからベクタ番号11のライン1111エミュレータまでをCPU例外とし、ユーザがdef-excによって定義した例外処理ハンドラに制御を渡す。この際、CPU例外が発生したベクタ番号をスタック上に積んでパラメータとして渡すので、例外処理ハンドラでは容易に原因を解析することができる。

また、ITOS68Kでは#4から#15までのソフトウェアトラップに対して、CPU例外と別の例外処理ハンドラを設定できる。これらの例外処理ハンドラは例外が発生したタスクの一部として実行されるので、トラップサービスルーチンとして使うことができる。

例外処理ハンドラはタスク部と非タスク部にそれぞれ独立に定義できる。さらにタスク部に対しては、全タスク共通またはタスク毎に固有の例外処理ハンドラを定義できる。

表-2 68000例外ベクタ一覧

Vector Number(s)	Address			Assignment
	Dec	Hex	Space注3	
0	0	000	SP	Reset:Initial SSP
	4	004	SP	Reset:Initial PC
2	8	008	SD	Bus Error
3	12	00C	SD	Address Error
4	16	010	SD	Illegal Instruction
5	20	014	SD	Zero Divide
6	24	018	SD	CHK Instruction
7	28	01C	SD	TRAPV Instruction
8	32	020	SD	Privilege Violation
9	36	024	SD	Trace
10	40	028	SD	Line 1010 Emulator
11	44	02C	SD	Line 1111 Emulator
12*	48	030	SD	(Unassigned, reserved)
13*	52	034	SD	(Unassigned, reserved)
14*	56	038	SD	(Unassigned, reserved)
15	60	03C	SD	Uninitialized Interrupt Vector
16~23*	64	04C	SD	(Unassigned, reserved)
	92	05C		
24	96	060	SD	Spurious Interrupt
25	100	064	SD	Level 1 Interrupt Autovector
26	104	068	SD	Level 2 Interrupt Autovector
27	108	06C	SD	Level 3 Interrupt Autovector
28	112	070	SD	Level 4 Interrupt Autovector
29	116	074	SD	Level 5 Interrupt Autovector
30	120	078	SD	Level 6 Interrupt Autovector
31	124	07C	SD	Level 7 Interrupt Autovector
32~47	128	080	SD	TRAP Instruction Vectors
	188	0BC		
48~63*	192	0C0	SD	(Unassigned, reserved)
	252	0FC		
64~255	256	100	SD	User Interrupt Vectors
	1020	3FC		

5. ITOS68Kのシステム構成

ITRONは将来のVLSI化を考慮した階層構造になっている。リアルタイム・マルチタスク処理の基本となるつぎの部分をチップ核とし、比較的高速性を必要としないチップ核周辺と区別している。

	タスク関連操作機能
	同期・通信機能
チップ核	メモリ管理機能
	割り込み処理機能
	例外処理機能

図-5にITOS68Kのシステム構成を示す。ITOS68Kではさらにシステムコールのパラメータチェック部をモジュール化し、取り外し可能にしている。また、各システムコールの処理とキュー操作などのサブルーチン群を明確に分離することにより、システムの信頼性を向上させている。

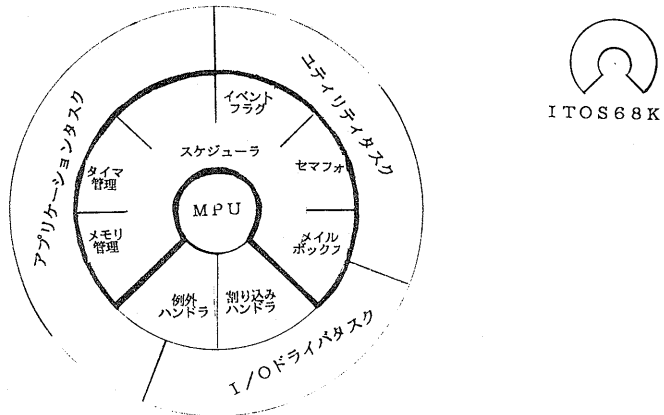


図-5 ITOS68Kのシステム構成

6. I/Oドライバ

一般にI/Oドライバは、ユーザタスクからの入出力要求を処理するドライバ・タスク、入出力デバイスからの割り込みを処理する割り込みハンドラ、および入出力制御情報が格納されるUCB (Unit Control Block) と呼ばれるテーブルから構成される。ユーザタスクとI/Oドライバの関係を図-6に示す。

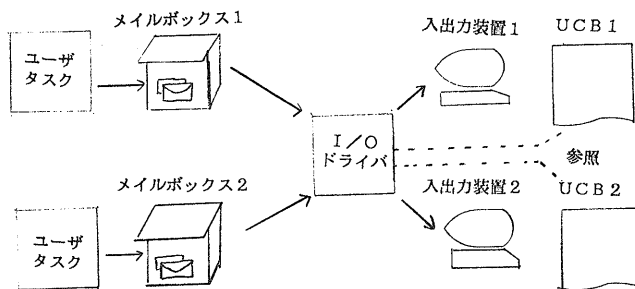


図-6 ユーザタスクとI/Oドライバの関係

UCBは各入出力デバイスごとに定義され、ドライバ・タスクにはタスク起動時の初期情報 (initcode) としてUCBの先頭アドレスが渡される。したがってリエントラントに作成されたドライバ・タスクは、各入出力デバイスに対応するUCBを参照しながら、ユーザからの入出力要求を実行することができる。

ITOS68Kのコンソール・ドライバを例に、I/Oドライバが入出力に際して行う処理を以下に説明する。

- ・システムスタート時に、入出力デバイスをイニシャライズする。
- ・ユーザタスクからの入出力要求メッセージを受け取り、入出力デバイスを起動する。
- ・入出力デバイスからの割り込みを処理する。
- ・ファンクションキーが入力された場合は、各ファンクションの処理を行う。
- ・入出力エラーが発生したときは、入出力を中断し、ユーザタスクにエラーコードを報告する。
- ・入出力終了時にエラーコード、送受信文字数、受信データをユーザタスクに報告する。

7. システムインストール

ITOS68Kは標準ハードウェアシステムをVMEバス仕様の日立製作所製シングルボードコンピュータファミリ (H68K SBCシリーズ) としているが、他の68000搭載のハードウェアでも容易に稼働させることができる。

ユーザアプリケーションプログラムの作成とITOS68Kシステムのジェネレーションは、汎用オペレーティングシステムCP/M-68K上で行う。図-7にITOS68Kシステムの作成手順を示す。ユーザアプリケーションプログラムは、C言語で記述することができるで、生産性の向上が図れる。また、68000用リアルタイムエミュレータを用いることにより、ハードウェアの環境も含めたユーザシステム全体の総合的なデバッグが可能となる。

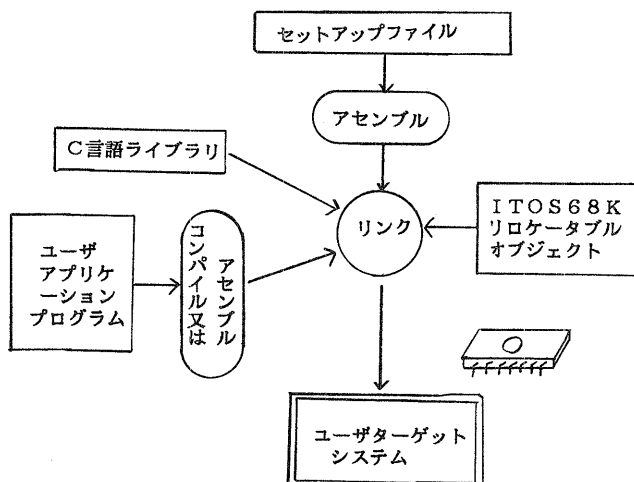


図-7 IRON68Kのシステムの作成手順

8. おわりに

以上説明したように、ITOS68KはITRONのアーキテクチャおよび外部仕様を68000CPU上にインプリメントした、高性能リアルタイム・オペレーティングシステムである。インプリメントに当たっては、68000の機能・性能を最大限に生かせるように種々の工夫を図り、Wakeup Task (wup-tsk) システムコール発行からターゲットタスク実行開始まで184 μ secの性能を実現している。

今後の課題としては、教育環境および教育体制の整備が考えられるが、もともとTRONプロジェクトでは教育を重視しており、TRONプロジェクトのメンバーの一員として教育環境の整備を図って行きたい。

最後に、ITOS68Kの開発において御指導・御援助いただいた、東京大学の坂村健博士およびTRON研究グループの皆様には感謝の意を表します。