Vol. 30 No. 1 情報処理学会論文誌 Jan. 1989

V60/V70 マイクロプロセッサと高信頼化システム†

河本恭彦†† 矢野陽一†† 鈴木奈利子††
藤井卓哉†† 椎葉忠明††

V60/V70 は日本電気の開発した32ビットマイクロプロセッサであり，高級言語，オペレーティングシステム，高信頼性システムをサポートするための機能を備えている．高級言語サポートとしては手続きのコール/リターンを効率的に行うための命令が用意されている．オペレーティングシステムのサポートとしてはいろいろな機能を備えているが，本論文ではメモリ管理機構，コンテキストスイッチ機能および非同期トラップ機能について述べている．さらに，60V/V70は高信頼化システムを構築する上での基本的な要素となるFRM (Functional Redundancy Monitoring) 機能を備えている．FRM 機能は複数個の V60 (またはV70)を pin-to-pin 接続するするだけで冗長プロセッサ構成を採用するシステムの構築を可能にする．このようなシステムでは1つのプロセッサは通常モードで動作し，他のプロセッサは監視モードで動作する．監視モードのプロセッサは通常モードのプロセッサの出力と自分自身の出力を比較し，不一致が検出されるとそれを外部回路に通知する．本論文では FRM 機能を利用した高信頼化システムの実現例として3個の V60 を用いた FRM ボードについて述べる．V60 FRM ボードは 2M バイトのメモリを持っており，2枚のボードを通常系，スタンバイ系として用いることでメモリの高信頼化を実現することもできる．

1. はじめに

半導体製造技術の進歩によるマイクロプロセッサの高性能・高機能化に伴い，その応用も従来はミニコンピュータや汎用コンピュータが利用されていた分野にまで拡大しつつある．そのような分野においては高度な信頼性を要求されるが，従来のマイクロプロセッサには高信頼化システムをサポートするための機能はそれほど十分とは言えなかった．このため，高信頼化システムのサポートは外部回路による実現を余儀なくされていた．このような背景のもと，V60/V70 マイクロプロセッサはチップ内に故障検出をサポートする機能を備え，高信頼化システムを容易に実現することを目指した．

本論文では V60/V70 マイクロプロセッサのアーキテクチャの特徴を述べた後，V60 を利用した高信頼化システムの実現例について述べる．

2. V60/V70 のアーキテキチャ[1)~3)]

V60/V70 のアーキテクチャの特徴を，レジスタセット，仮想記憶管理，命令セットに分けて説明する．

2.1 レジスタセット

V60/V70 は32ビットの汎用レジスタを32本持つ汎用レジスタアーキテクチャのプロセッサである．32本という大量の汎用レジスタは，最適化コンパイラによる変数のレジスタ割り付け数を増やし，メモリアクセスの回数を減らすとともに，処理の高速化に寄与する．**図1**に V60/V70 のレジスタセットを示す．V60/V70 のレジスタセットは，一般のプログラムが活用することのできるプログラム・レジスタセットと，通常はオペレーティングシステムのみが参照することのできる特権レジスタセットに大別される．すべてのレジスタ長は32ビットである．

(1) プログラム・レジスタセット

プログラム・レジスタセットは32本の汎用レジスタ，PC (プログラムカウンタ) および PSW (プログラム・ステータス・ワード) からなる．

汎用レジスタは，32本がすべてデータ，ポインタおよびインデックス用として利用することができ，浮動小数点データも整数と同様におくことができる．レジスタ R29 から R31 の3本は，高級言語における手続き呼出し/戻りのために特別な用途を割り当ててある．

R29 はアーギュメントポインタ (AP) として手続き呼び出し時のパラメータ領域のベースアドレスを保持する．R30 はフレームポインタ (FP) として手続きが呼び出されたときのスタックフレームのアドレスを保持する．R31 はスタックポインタ (SP) として，スタックの先頭アドレスを保持する．

PC は現在実行中の命令の先頭番地を保持する．

PSW は分岐条件に係わる条件フィールドのほか

† V60/70V Microprocessor and Highly-reliable Systems by YASUHIKO KOUMOTO, YOICHI YANO, NARIKO SUZUKI, TAKUYA FUJII and TADAAKI SHIIBA (Microcomputer Division, NEC Corporation).

†† 日本電気(株)マイクロコンピュータ事業部

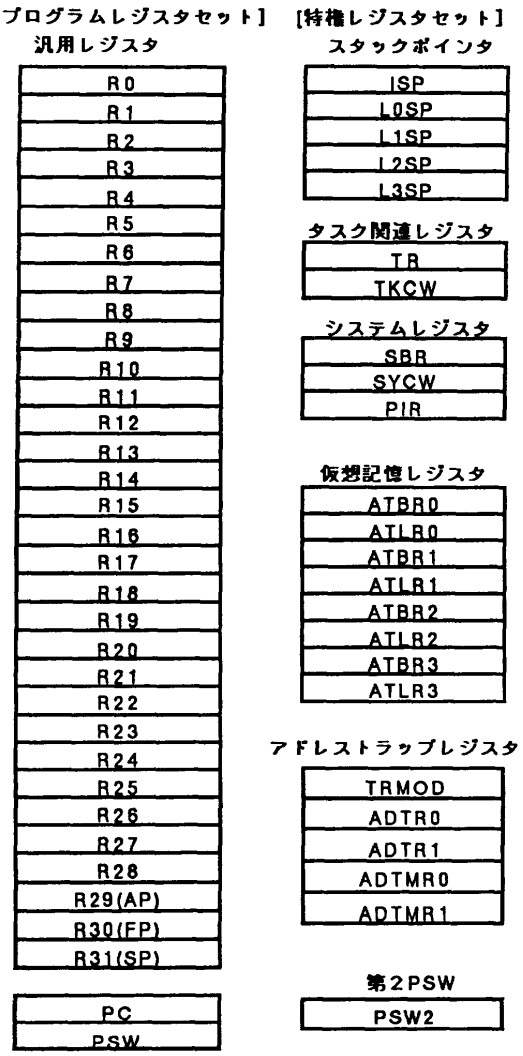

図 1 V60/V70 のレジスタセット
Fig. 1 V60/V70 register set.

に，浮動小数点の例外状態を示す浮動小数点フィールド，シングルステップや割込みの許可などのプログラムの実行状態を規定するコントロールフィールドおよび実行レベルなどのシステムの状態を示すステータスフィールドの4つのフィールドからなる．

（2） 特権レジスタセット

特権レジスタは，スタックポインタ群，タスク管理，システム管理，仮想記憶管理，アドレストラップ・レジスタ群およびエミュレーション用 PSW に分類される．

スタックポイント群は4つの実行レベルと割込み処理用の5本のスタックポインタからなる．タスク管理のレジスタ群はタスクの，システム管理のレジスタ群はシステムの状態をそれぞれ規定する．アドレストラップ・レジスタ群は2組存在する．

2.2 仮想記憶管理

V60/V70 は内蔵するメモリ管理ユニットにより，ページング方式で仮想アドレス空間を生成，管理，維持する．仮想アドレス空間は 4G バイトであり，各タスクごとに 4G バイトのアドレス空間を持つことのできる多重仮想空間の機能がある．

2.2.1 仮想アドレス空間の構成

V60/V70 の仮想アドレス空間の構成を図2に示す．4Gバイトの仮想アドレス空間は4つのセクションに，1G バイトの各セクションは 1,024 個のエリアに，1M バイトの各エリアは 256 個のページに分割される．各ページのサイズは 4K バイトである．

2.2.2 多重仮想空間

通常のマルチタスキングを行っている仮想記憶システムでは，各々のタスクごとに独立した仮想アドレス

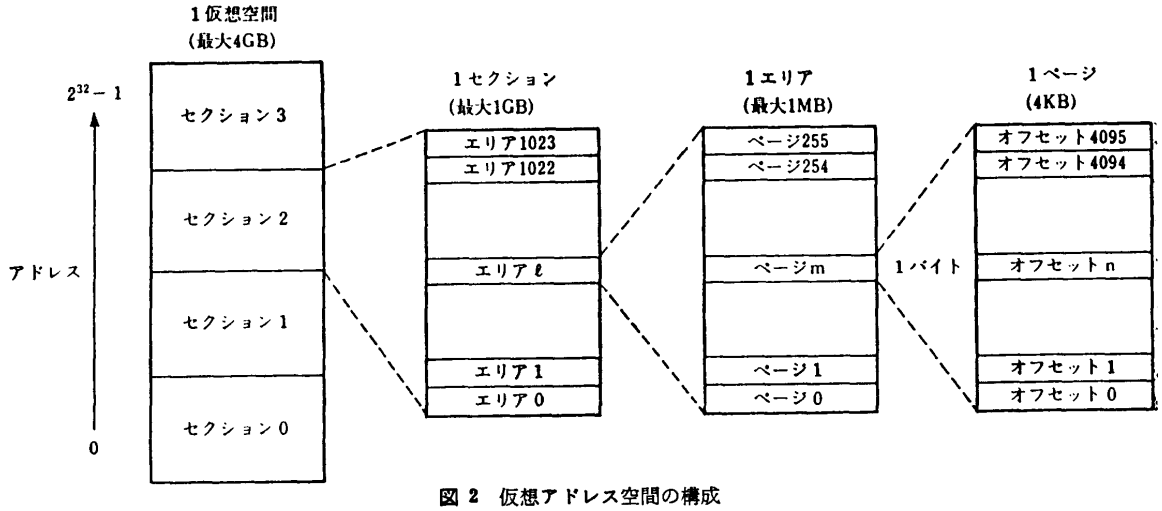

図 2 仮想アドレス空間の構成
Fig. 2 Configuration of address space.

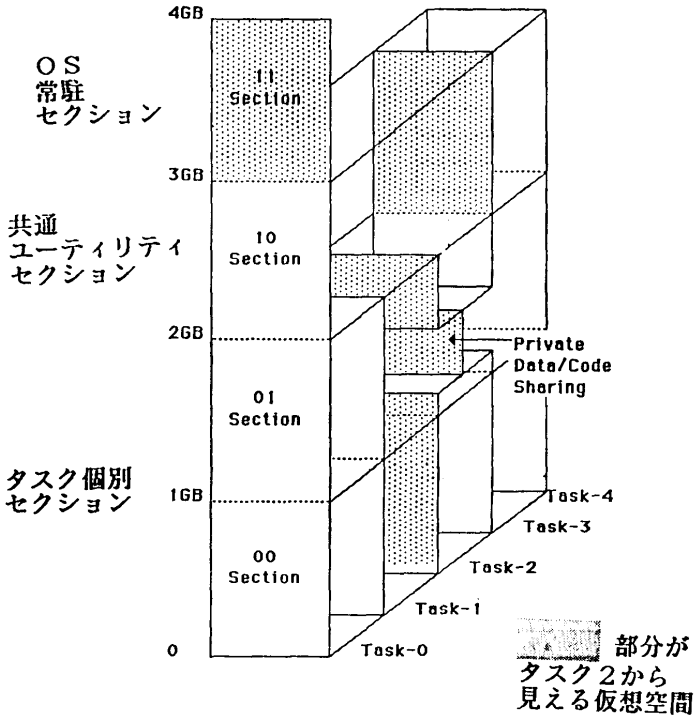

図 3 多重仮想空間の構成例
Fig. 3 Multiple virtual space configuration example.

空間を与えるのが普通である．すなわち，各タスクごとに別々の仮想アドレス空間が存在する．

V 60/V 70 では，このような多重仮想空間構成をとることが可能になっている．

多重仮想空間構成を採用するシステムでは，各々のタスクに独立なアドレス域だけでなく，すべてのタスクに共通したサービスを提供するプログラム（オペレーティングシステムなど）を使用するために，各々のタスクに共通のアドレス域を提供する必要がある．

V 60/V 70 では，仮想空間をタスク間で共有するか独立にするかを任意に指定することができる．例えば，共通アドレス域として11セクションの1Gバイト，独立アドレス域として00セクションから10セクションまでの3Gバイトを割り振る仮想空間構成をとることができる．また，システム全体での仮想空間の共有のほかにも，複数の仮想空間の間でのプライベートな共有ができる．これはエリア単位の共有である．**図3**に多重仮想空間の構成例を示す．

2.2.3 アドレス変換

V 60/V 70 は仮想アドレス空間を実現するために必要となる，仮想アドレスから実アドレスへのアドレス変換機構をオンチップに内蔵している．通常の変換はファームウェアにより，オンチップのレジスタ（エリアテーブル・レジスタペア）およびメモリ上のアドレス変換テーブル（エリアテーブル，ページテーブル）を参照して行われる．アドレス変換で参照するレジスタペアと変換テーブルを**図4**に示す．ひとたびアドレス変換が行われると，その結果はオンチップ内のTLB（高速アドレス変換機構）にキャッシングされ，以後のアドレス変換はTLB上で高速に行われる．TLBは16エントリを持つフルアソシアティブ方式であり，エントリの入換えは疑似LRUで行われる．**図5**にアドレス変換の機構を示す．

2.2.4 アクセス保護機構

V 60/V 70 では2段階の保護機構を実現している．プログラムの実行時には，レベル0からレベル3までの4つの実行レベルのうちのひとつがとられる．実行レベルは数的に小さいものほど特権性が強くなり，レベル0では通常の命令に加えて特権命令の実行が可能になる．通常，レベル0ではオペレーティングシステムのカーネル部が実行され，レベル1ではデバッガなどのシステムタスクが実行される．ユーザのアプリケーションタスクはレベル3で実行される（**図6**）．第1の保護機構はエリアに対するものである．エリ

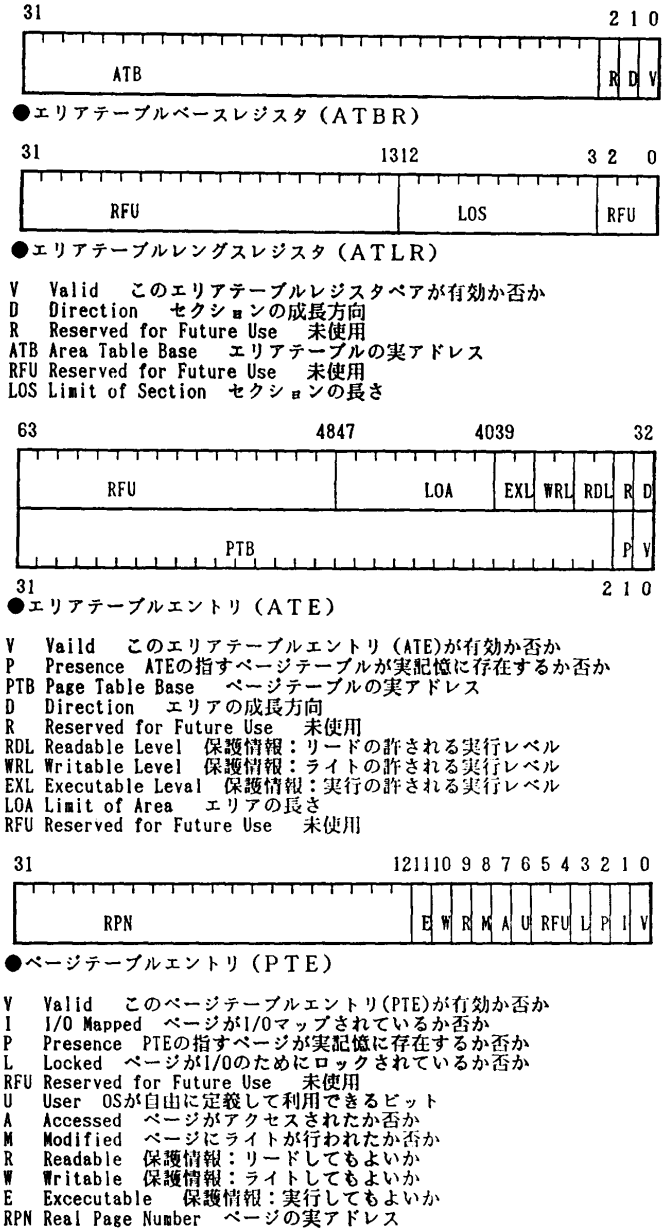

図 4 アドレス変換用のレジスタとテーブル
Fig. 4 Registers and tables for address translation.

アの保護はエリアテーブル・エントリによって規定される．エリアテーブル・エントリ内ではリード/ライト/実行の各アクセスタイプごとに独立に，そのエリアをアクセスできる最低位の実行レベルが指定される．第2の保護機構はページに対するものである．ページの保護はページテーブル・エントリによって規定される．ページテーブル・エントリ内ではリード/ライト/実行に関するアクセスができるか否かを指定する．

すべてのメモリアクセスは，エリア/ページの両方で許可された場合のみ可能になり，許可されていないアクセスに対しては例外が発生する．

2.3 命令機能

（1） 命令セット

V60/V70 の命令セットはソースオペランドとデスティネーションオペランドに対し，自由なアドレッシ

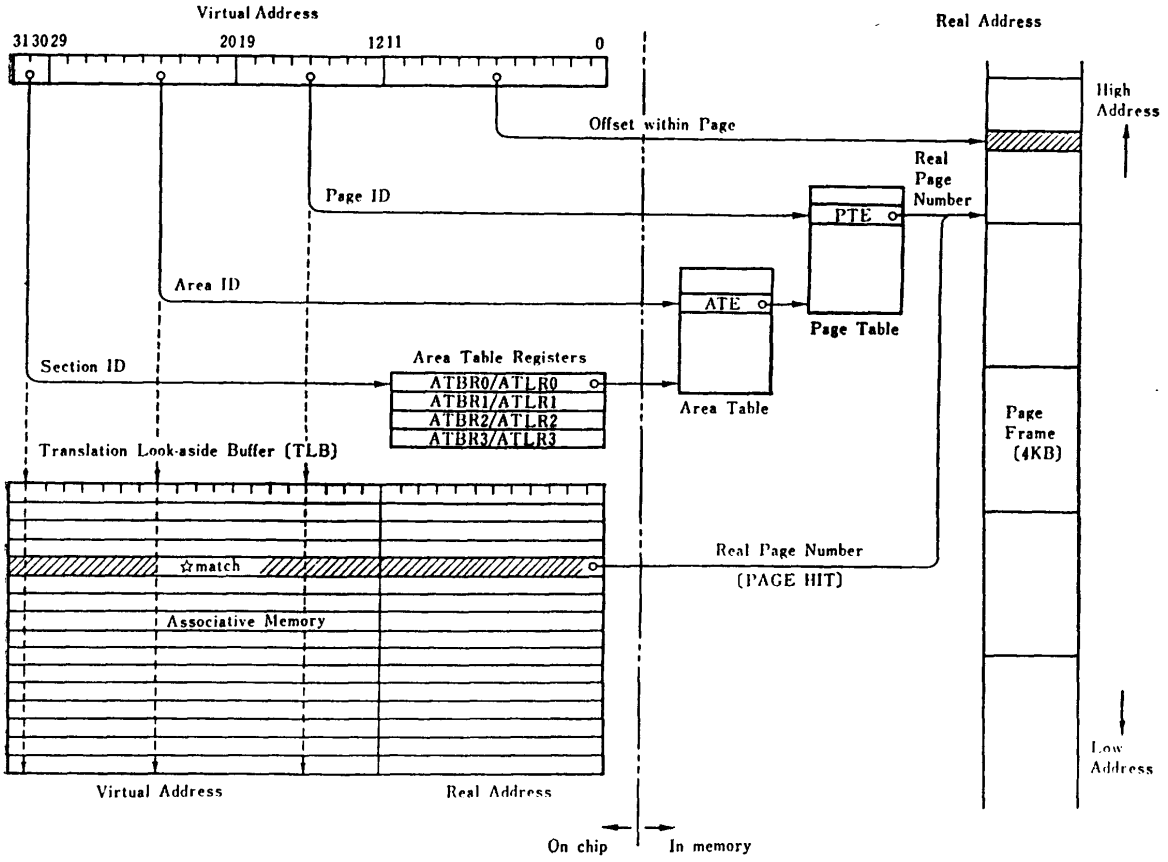

図 5 アドレス変換の機構
Fig. 5 Virtual-to-actual address translation.

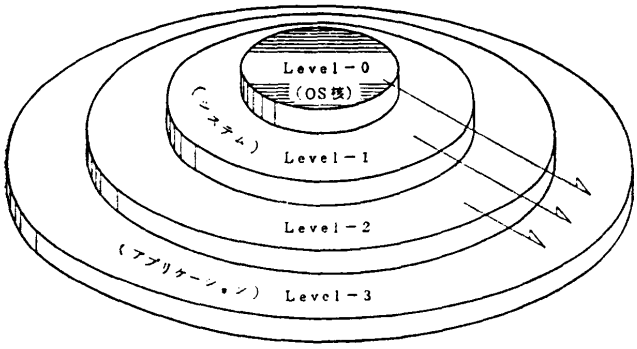

図 6 メモリの保護機構
Fig. 6 Memory protection function.

ングモードを使用できる2アドレス方式を採用している．このため，メモリとメモリ間の演算も自由に記述できる．

V60/V 70 には 119 種 273 個の命令が用意され，命令セットは**表 1** に示すように 12 のカテゴリーに分類できる．オペレーティングシステムの行う仮想記憶管理やタスク管理をサポートする命令は特権命令に含まれる．

表 1 V60/70 の命令の分類
Table 1 V60/V70 instruction set summary.

番 号	命 令
1	転送命令
2	整数算術演算命令
3	論理演算命令
4	シフト・ローテート命令
5	ビット演算命令
6	10 進演算命令
7	文字列操作命令
8	浮動小数点演算命令
9	制御移動命令
10	高級言語サポート命令
11	特権命令
12	その他の命令

（2） 高級言語サポート

V 60/V 70 では高級言語の手続き呼出し/戻り を効

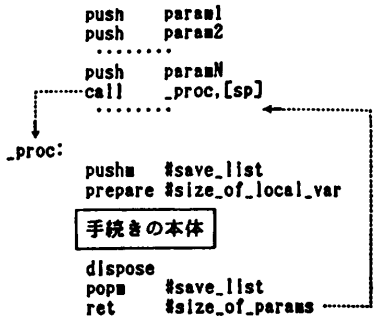

図 7(a) 手続き呼出しのシーケンス
Fig. 7(a) Procedural calling/receiving sequence.

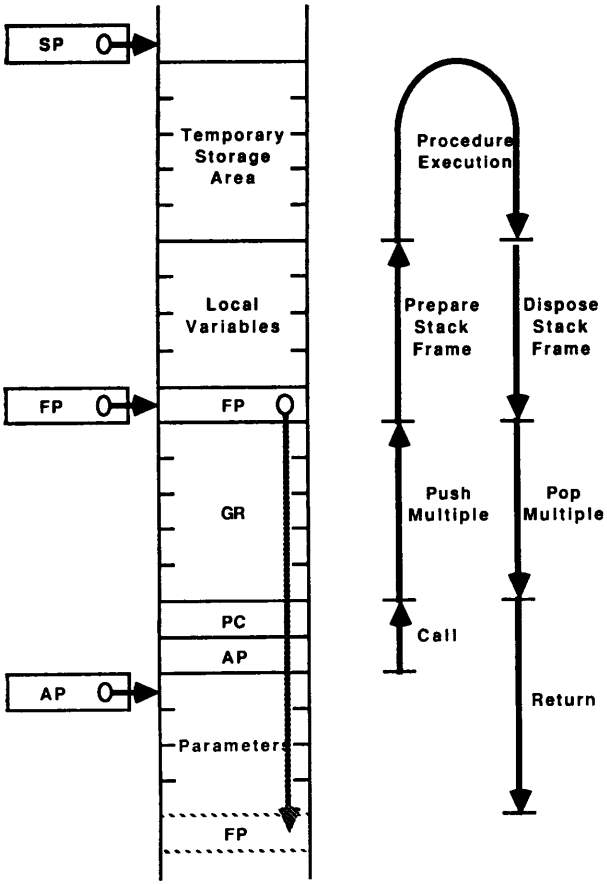

図 7(b) 手続き呼出しとスタック
Fig. 7(b) Stack operation during procedural call/return.

率的に実現するための命令が用意されている．ここでは，V60/V70 における手続き呼出しの1例を示す．**図7(a)**は，呼出し例と手続きのコード例，**図7(b)**は，スタックの動きである．このとき，汎用レジスタのうち R29 から R31 の3本（それぞれ，アーギュメントポインタ (AP)，フレームポインタ (FP)，スタック・ポインタ (SP) を操作する．以下が，その操作の手順である．

① パラメータをスタックに積む．

② 手続き呼出し命令 (CALL) により，戻り先 PC と旧 AP がスタックに退避され，新 AP が設定された後，手続きへ分岐する．

③ 手続きで使用(破壊)する汎用レジスタを退避する．

④ スタックフレーム生成命令 (PREPARE) により，旧 FP を退避した後に新 FP を設定し，手続きに必要なローカル変数領域を確保する（すなわち，SP を進める）．

⑤ この時点で手続きの処理本体が開始される．パラメータのアクセスは AP を，ローカル変数のアクセスは FP をベースとして行われる．

⑥ 手続きからの戻りには，スタックフレーム削除命令(DISPOSE) により，FP, SP の値を復帰し，ローカル変数領域を開放する．

⑦ 退避していた汎用レジスタを復帰させる．

⑧ 帰還命令 (RET) により，呼出し側に制御が戻るとともに AP の値が復帰され，同時にスタック上のパラメータ領域を開放する（すなわち，SP を戻す）．

以上の例では手続きのパラメータ領域をスタック上に確保したが，言語によっては別のデータ領域にパラメータを置くことがある．この場合は単に CALL 命令のオペランドで指定する新 AP 値が変るのと，RET でパラメータ領域の開放を行わないだけで同様な命令列を使用する．

（3） オペレーティングシステム・サポート

V60/V70 がオペレーティングシステムをサポートする命令としては，仮想記憶管理に関するもの，タスクのコンテキスト切替えに関するものなどがある．これらはすべて実行レベル0のみで使用できる特権命令である．

（a） 仮想記憶管理

仮想記憶管理命令にはアドレス変換に用いるテーブル（エリアテーブル，ページテーブル）を参照/更新する命令，TLB のエントリをクリアする命令，仮想アドレスを実アドレスに変換する命令がある．

（b） コンテキスト切替え

V60/V70 においてタスクコンテキストは次に示すレジスタ群と仮想記憶を管理するメモリ上のテーブルにより定義される．これらのレジスタ群がタスク切替え時に入れ替わる対象となり，各タスクごとに TCB（タスクコントロールブロック）領域としてメモリ上に格納されている．

① 仮想アドレス空間環境
　エリアテーブル・レジスタペア群
② プログラム実行環境
　汎用レジスタ
　各実行レベル用スタックポインタ
③ タスク固有の状態情報
　タスクレジスタ
　タスク・コントロール・ワード

V 60/V 70 ではタスクコンテキストを入れ替えるための命令を用意している．それが LDTASK 命令，STTASK 命令である．TCB の大きさは可変であり，それに含まれるレジスタは LDASK 命令，STTASK 命令のオペランドおよび SYCW（システム・コントロール・ワード）によって指定する．TCB 領域のベースアドレスは TR（タスクレジスタ）が保持する．LDTASK 命令を実行すると，第2オペランドで指定した TCB 領域から，第1オペランドで指定した V 60/V 70 のレジスタ群に環境を設定する．同時に，TR にTCB 領域のベースアドレスをセットする．逆に STTASK 命令では，オペランドで指定した V 60/V 70 上のレジスタから TR が保持する TCB 領域に環境を退避する．V 60/V 70 においてコンテキスト切替えは STTASK（旧コンテキストのストア），LDTASK（新コンテキストのロード），RETIS（ハンドラからの復帰）の3命令で記述できる．

図3の多重仮想空間において，コンテキスト切替えでは変化しないオペレーティングシステムなどの共通のアドレス域のエリアテーブル・レジスタペア群は TCB に含めない．すなわち，タスク切替えが起きても仮想空間のそのセクションは変化させない．一方，各タスク固有のアドレス域のエリアテーブル・レジスタペア群は TCB に含めておいて，タスク切替え時に仮想空間も切替える．

（c） 非同期トラップ

V 60/V 70 には，オペレーティングシステムの実現を簡素化するために，非同期トラップの機能がある．非同期トラップとは，事象に対応する処理ルーチンを事象発生の時点とは異なった時点まで遅延して起動するためのトラップ機構である．非同期トラップには，オペレーティングシステムに対して事象発生の通知を行う非同期システムトラップ(AST)と，タスクへの事象の通知を行う非同期タスクトラップ（ATT）の2種類がある．

●非同期システムトラップ（AST）

AST の使用例として，オペレーティングシステム内のタイマ割込み処理ルーチンとタスクディスパッチがある．

タイマ割込み処理ルーチンでは，待ち時間が終了したタスクが存在するなら，そのタスクを実行可能状態に移しスケジュール対象として登録しなければならない．このとき，タスクディスパッチャに制御を移し，そのタスクをスケジュールの対象とするように要請する．ディスパッチャへの制御の移し方としては，タイマ割込み処理ルーチン内から直接ディスパッチャにジャンプしたり，そこへの手続き呼出しによるものが考えられる．しかし，そのような設計をするとオペレーティングシステムの内部は手続き間の移行が絡み合う複雑な構造になる．また，タイマ割込み処理ルーチンやディスパッチャ内は割込みが禁止されている言わば不可分領域であり，手続き呼出しによるタスクディスパッチャでは不可分領域（タイマ処理）から他の不可分領域（ディスパッチャ）へと切れ目なく制御が渡ることになり，リアルタイム性が失われてしまう．

AST はこの問題を解決するためにはある．まず，AST 発生時に起動される AST ハンドラのアドレスをディスパッチャに設定しておく．割込み処理中にディスパッチャに制御を移す要請が生まれると，AST を発生させる旨を SYCW の AST フィールドに設定し，プログラムを通常に終了する．すなわち，割込みからの復帰命令 (RETIS) までを実行する．RETIS 命令は，割込んだ元の状態に戻るか，他の割込み要因による別の割込みハンドラに制御を移すか，あるいは，AST の発生要求があるかの判断を行う．別の割込み要因がなければ AST ハンドラ（ディスパッチャ）に制御が移り，そこでディスパッチが行われる．この機構を図8に示す．

AST の利点としては，以下の点が挙げられる．

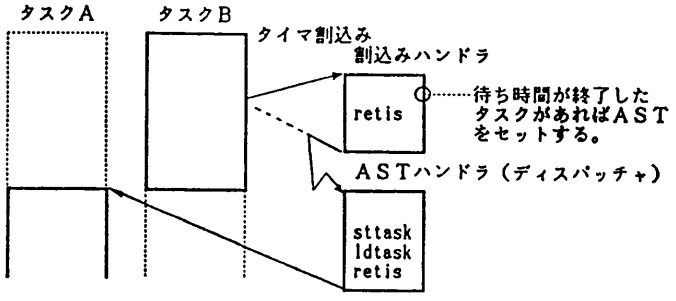

図 8　タスクディスパッチと AST
Fig. 8　Task dispatch and AST.

① 2つの不可分領域間の遷移がマイクロプロセッサによって自動的に行われるため，遷移をソフトウェアで陽に記述する必要がない．

② 2つの不可分領域の遷移の際に割込み要求があるか否かのチェックを自動的に行うために，不可分領域が長くならない．

このため，ソフトウェアの構造が簡単になり，オペレーティングシステム作成の手間を削減できる．

●非同期タスクトラップ（ATT）

ATT は，トラップが発生した後に制御の移る先が，オペレーティングシステムではなくユーザレベルのタスクである点に特徴がある．タスクの実行過程では，各種の例外の発生への対応，終了時の処理，他のタスクからの要求への対応，などの多様な処理を行わなければならない．これらすべてに対して判断や制御の移行を陽にソフトウェアで記述すると複雑なものになる．

ATT の使用例としては，ユーザレベルで行う割込み処理がある．例えば，会話型アプリケーションプログラムにおけるコマンド中断（コントロール+C の入力によるキーボード割込み）で，制御を再びアプリケーションプログラムに戻す場合が考えられる．

ユーザのアプリケーションプログラムでは，まず，システムコールなどで ATT ハンドラのアドレスを設定する．キーボード割込みで割込みハンドラが起動されると，ハンドラ内ではシステムの既定処理が行われ，最後に ATT を発生する旨を TKCW 内の ATT フィールドに設定する．その後は，AST の時と同様に，割込みハンドラの最後で実行する RETIS 命令で ATT の有無が判断され，別の割込み要因がなければ ATT ハンドラが起動される．

ユーザ固有の ATT ハンドラからの復帰には，ユーザレベルで使用できる（すなわち，特権命令ではない）割込みからの復帰命令 (RETIU) を用いる．この機構を**図 9** に示す．

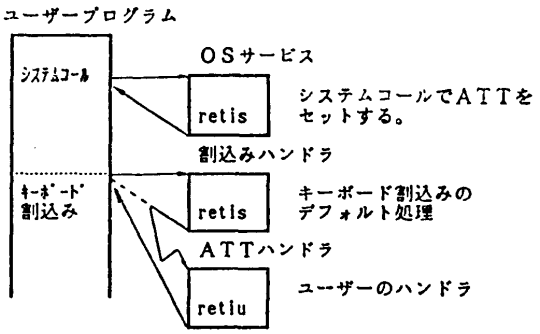

図 9 キーボード割込みと ATT
Fig. 9 Keyboard interrupt and ATT.

以上のように，V60/V70 では AST，ATT を用いることにより，事象発生に対する処理とそれに付随する処理を分割することができるため，従来のオペレーティングシステムの内部設計で煩雑になっていた部分を簡素化，モジュール化でき，事象応答性も向上する．

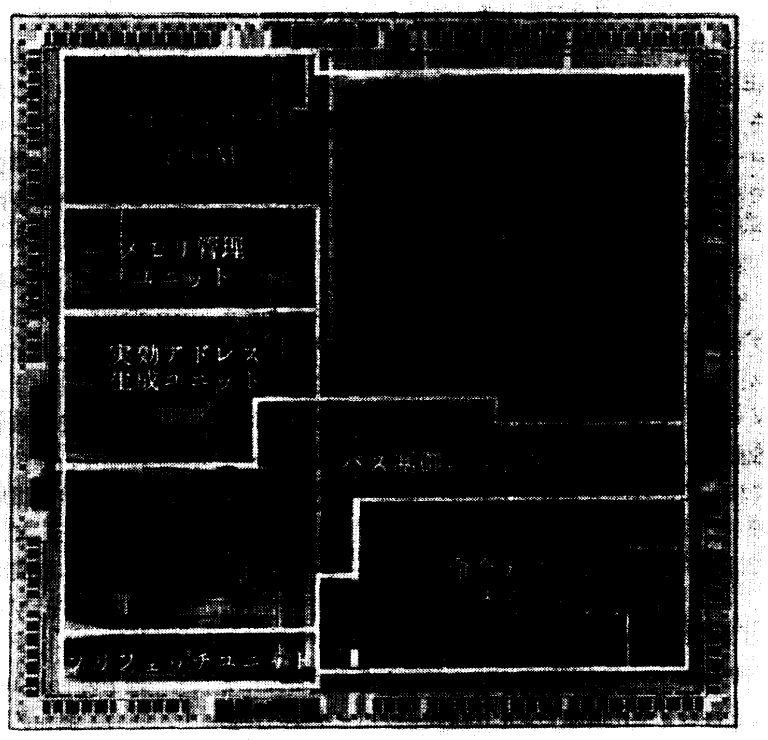

図 10 V70 (μPD 70632) のチップ写真
1.5 μm ルールの CMOS アルミ2層プロセスにより約 385,000 トランジスタを 14.35 mm×14.24 mm のチップ上に集積している．V70 は写真に示すように 6 つのユニットから構成され，各ユニットはパイプライン動作する．
Fig. 10 Microphotograph of the V70 (μPD 70632).

表 2 V60/70 の相違点
Table 2 V60/V70 comparison.

項　　目	V60	V70
アドレスバス (外部)	24 ビット	32 ビット
(内部)	32 ビット	32 ビット
データバス (外部)	16 ビット	32 ビット
(内部)	32 ビット	32 ビット
最小バスサイクル	3 クロック	2 クロック
バスサイジンク	な　し	I/O のみあり
性　能	3.5 MIPS*	6.6 MIPS**

* 16 MHz 動作時
** 20 MHz 動作時

3. V60/V70 のハードウェア

(1) パイプライン

V60/V70 のブロック構成を示すチップ写真を図 10 に示す．図 10 では V70 について示してあるが，V60 も基本的には同じである．V60/V70 は 6 つの独立した機能ユニットから構成され，各ユニットはパイプライン処理を行うように結合されている．機能ユニットの内訳はプリフェッチユニット (PFU)，命令デコードユニット (IDU)，実効アドレス生成ユニット (EAG)，メモリ管理ユニット (MMU)，バス制御ユニット (BCU)，実行ユニット (EXU) である．マイクロコード ROM は実行ユニットに含まれる．V60/V70 では，このような 6 段パイプライン方式を採用したことにより，最大 4 命令を同時実行できる．

(2) V60 と V70 の相違点

V70 と V60 はオブジェクト互換の 32 ビットマイクロプロセッサであり，V70 は V60 のアドレスバス，データバス拡張版として位置づけられる．表 2 に V60 と V70 の相違点を示す．V70 は単にバス幅を拡張しただけでなく，基本バスサイクルを 2 クロックにして性能向上を図っている．また，それ以外にも細かい改良を各ユニットに対して行っているが，ここでは説明を省略する．

4. 高信頼化システムの試作[4]

V60/V70 は，高信頼化システムを実現する際に必要となる 2 重化構成や 3 重化構成をサポートするために，FRM (Functional Redundancy Monitoring) と呼ばれる機能を備えている．

ここでは，FRM 機能の応用例として，3 個の V60 を使用し，多数決論理を用いて高信頼化システムを実現するための FRM ボードの設計について報告する．

4.1 FRM 機能

本論文で想定する高信頼化システムとは，ハードウェア等の故障によりシステム内のある構成要素は動作不可能になるが，その故障ユニットを多重化構成をとる別のユニットにより代替することで，システムの性能は低下するがシステムダウンすることなく実行を継続することのできる計算機システムである．一般に，このような高信頼化システム構成を実現する時には，

- 故障の検出
- 故障部分の隔離
- 故障部分の修理・取替
- 回復処理
- 再統合

というステップを踏むことで故障からの回復を図ることになる．

現在の VLSI 時代においては，マイクロプロセッサを用いて高信頼化システム構成をとるとき，プロセッサの周辺に問題を限る（メモリ系，入出力系を除く）ならば，故障および故障部分の修理・取替はマイクロプロセッサチップの故障・修理・取替ということになる．このとき，故障の検出とはマイクロプロセッサが期待どおりの動作を行うかどうかを監視し，誤動作を検出することになる．したがって，マイクロプロセッサのチップ単位で故障検出を助ける機能がマイクロプロセッサ自身にない限り高信頼化システム構成の実現は困難である．なぜなら，複数のマイクロプロセッサからなる冗長系で，チップのすべての入力/出力の一致回路を外部に接続することで故障検出を行わなければならないからである．V60/V70 には高信頼化システムを実現するために必要となる機能をサポートしており，その機能を FRM とよんでいる．その第 1 の機能は，他のプロセッサの動作を監視する機能である．第 2 の機能はプロセッサの外部動作の凍結機能である．そして，第 3 はプロセッサに対する異常ないしシステム構成の変更の通知機能である．

図 11 に V60/V70 を 3 個用いた 3 重化システム構成の例を示す．V60/V70 では 3 つのプロセッサの各端子の pin-to-pin 接続だけで冗長プロセッサ構成をとり，動作の不一致を検出できるように設計してある．図 11 のシステム構成において，3 個のプロセッサの内の 1 つが通常モード，他の 2 つの監視モードに設定され，お互いに同期して並列動作している．このとき，通常モードの V60/V70 は通常の命令実行を行っている．一方，監視モードの V60 (V70) は，外

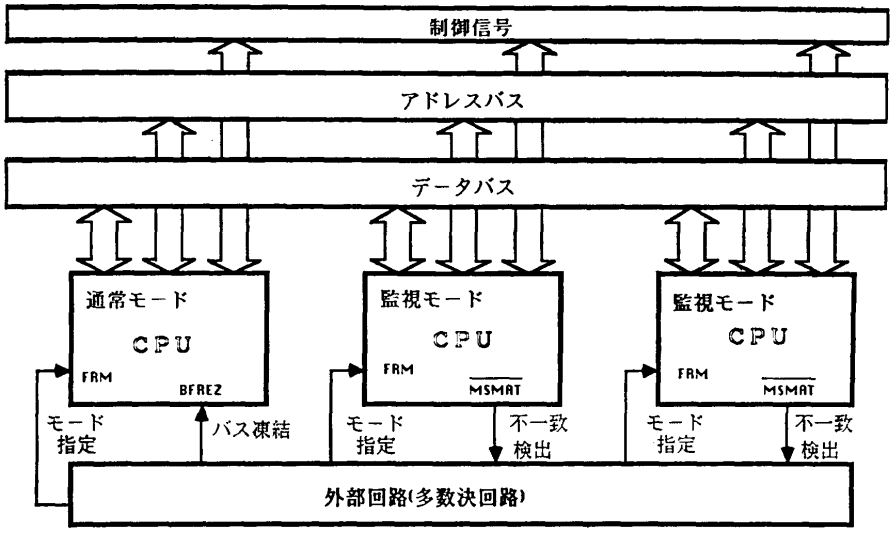

図 11 3 重 化 シ ス テ ム 構 成
Fig. 11 Triple mode redundancy configuration.

部バスを駆動せずに，背中合せに接続された通常モードの V60 (V70) の出力信号（アドレスバス，データバス，ステータス信号）と自分自身の出力が一致しているか否かの比較を各バスサイクルごとに行う．比較が一致している限り，3つのプロセッサは正しい動作を行っていると仮定できるが，もし一致しないときはいずれのプロセッサに何らかの故障が生じたと考えることができる．不一致が検出されると，監視モードの V60 (V70) は専用端子を通じ，故障処理関連の外部回路に不一致を通知する．外部回路は不一致が通知されると動作をしていたプロセッサのペアを外部より強制的に停止させる．これがバスの凍結 (バスフリーズ) 機能である．バスフリーズ状態になったプロセッサは，アドレスバスとデータバスはハイインピーダンスとなり各種の信号はインアクティブになるため，外部バスから切り離された状態になる．プロセッサのペアがバスから切り離されると，不良診断はシステムから独立に行うことができる．このとき，故障であると診断されたプロセッサはバスから切り離されたままであるが，故障でないと診断された残りのプロセッサはバスフリーズ状態を解除して再スタートさせる．バスフリーズ要求と同時にバスサイクルのリトライを要求することによって，V60/V70 は故障が起こったときの状態からそのまま実行を継続することができる．このため，2つのプロセッサのペアの上で動作するソフトウェアにとってはあたかも異常が何もなかったように見える．

なお，V60/V70 において以上の機能を実現するために要したハードウェアの量はチップ面積の約 0.24% である．

4.2 FRM 機能を応用したシステム試作

今回試作したボードの外観を図 12 に示す．このボードは3個の V60 を搭載し，2M バイトのメモリを持つ．また，FRM 機能の試験用に故意に不一致を発生させるための回路を持つ．

図 12 FRM ボードの外観
Fig. 12 Appearance of V60 FRM board.

（a） システムの概要

本システムで使用するボードでは，図11で示すように3重化システム構成をとり，1つのプロセッサが通常モードで，他の2つのプロセッサが監視モードで並列動作する．もし，プロセッサの動作に不一致が生じると，外部の多数決回路によって不具合プロセッサが決定され，自動的にそのプロセッサをバスから切り離す（以後，システムは2個のプロセッサで動作する）．しかし，このときの不一致はプロセッサの故障に無関係な一過性のノイズによる場合もある．その場合は切り離されているプロセッサを再統合し，システムは再び3個のプロセッサで動作する．

また，2枚のボードを使用して，通常系/スタンバイ系の構成がとれるようになっており，通常系が完全に動作不能になったときは，通常系からスタンバイ系にシステムの運行を引き継ぐ機能がある．

さらに，メモリとCPUとのデータ転送に対して高信頼性を確保するために，ライトバックアップやスタンバイ系のメモリへのリード/ライトがソフトウェアから制御できるようになっている．

メモリのライトバックアップとは通常系のプロセッサがメモリにライトをする場合は，通常系のボードのメモリだけでなく，スタンバイ系のメモリにもライトをする動作のことである．これにより，通常系のボードのメモリに異常が生じた場合にも，リードを行うメモリをスタンバイ系のものに切り替えればプログラムの実行に支障をきたさない．

（b） 通常系ボード内における高信頼化

3個のV60は，最初1つが通常モードで動作し，残りの2つが監視モードで動作している．この状態で，監視モードのV60から不一致が通知されるとハードウェアは，多数決により不具合を起こしたプロセッサを決定し，そのプロセッサをバスから切り離す（バスフリーズをかける）．このとき，通常モードで動作していたプロセッサが不具合プロセッサと判断される場合は，監視モードで動作していたプロセッサを通常モードに変更する再構成をしてから，不一致を起こしたバスサイクルからプログラムの実行を継続する．この時点から通常系のV60は2個で動作することになる．

いま，3個のプロセッサをそれぞれCPU 0，CPU 1，CPU 2と呼ぶことにする．CPU 0が通常モード，他の2つが監視モードで動作しているとすると，不一致が発生した場合の不具合プロセッサは，表3の方式によって決定することができる．

表3 多数決の方式
Table 3 Mismatch detection and reconfiguration.

CPU 1	CPU 2	不具合	処置
一致	一致	なし	通常実行
一致	不一致	CPU 2	CPU 2 を切り離す
不一致	一致	CPU 1	CPU 1 を切り離す
不一致	不一致	CPU 0	CPU 0 を切り離す CPU 1 を通常モード

表4 各プロセッサの状態
Table 4 State of each processor.

状態	CPU 0	CPU 1	CPU 2	遷移の要因
①	通常モード	監視モード	監視モード	リセット直後から正常動作をしている間.
②	通常モード	監視モード	×	CPU 2 のみ不一致を出力した時.
③	通常モード	×	監視モード	CPU 1 のみ不一致を出力した時.
④	×	通常モード	監視モード	CPU 1, CPU 2 共に不一致を出力した時.

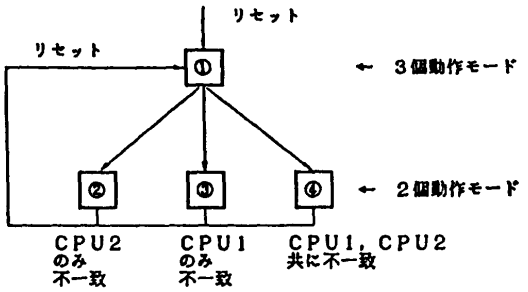

図13 プロセッサの状態遷移
Fig. 13 State transition of processors.

プロセッサが3個で動作している状態で不一致が発生すると，プロセッサ2個で動作する状態に遷移する．このとき，3個のプロセッサの状態には表4で示す4つの状態があり，遷移の様子は図13に示すとおりである．状態①から状態②～④への遷移はハードウェアにより自動的に行われる．プロセッサが2個で動作している状態で不一致が発生すると，もはや多数決による不具合プロセッサの決定はできない．その時点で通常系のシステムの運行は停止される．このとき，HALT命令により停止しているスタンバイ系にハードウェア割り込みがかけられ，制御はスタンバイ系に移る．

（c） メモリの高信頼化

ここで想定しているシステムには，通常系とスタンバイ系の2枚のボードがあり，同一空間にメモリが存

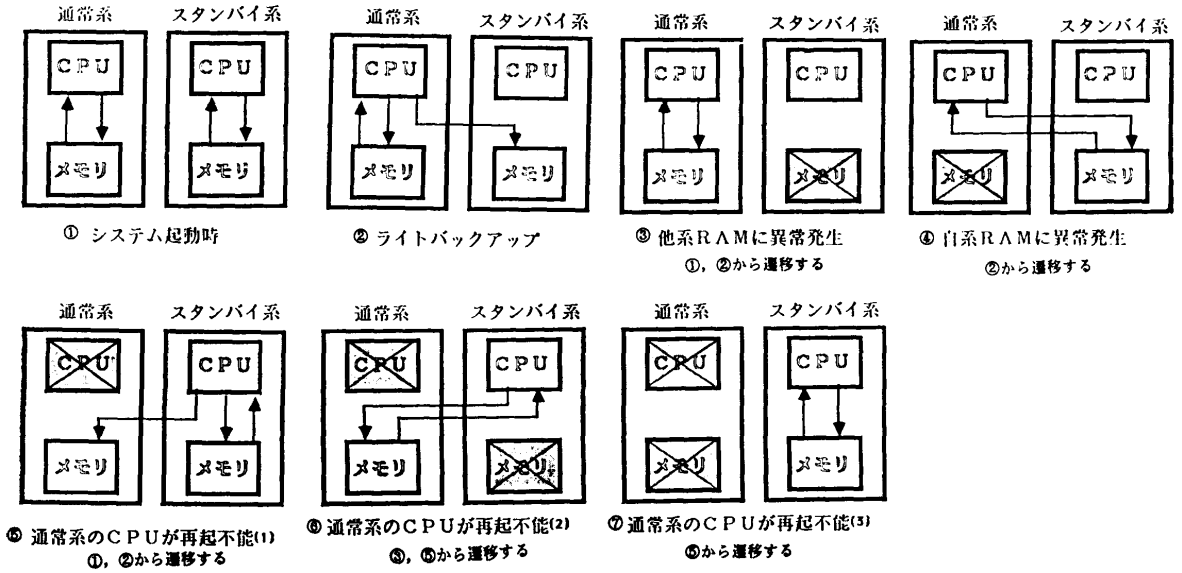

図 14 メモリアクセスの形態
Fig. 14 Dual system and memory access.

在する．これを利用してメモリの FRM 機能を実現することができる．システムが起動されたとき，通常系，スタンバイ系の各プロセッサは自系のボード内のメモリをアクセスしている．その後，ライトバックアップの設定がされたシステムは運行を開始する．もし，システムの運行中にメモリ系からの異常が通知されると，ハードウェアはプロセッサをバスフリーズで停止させる．そして，あらかじめ設定されているようにメモリの切替えを行った後，バスフリーズを解除し，異常が通知されたバスサイクルから処理を続行する．以上の動作はハードウェアで自動的に行われるため，ソフトウェアからは知ることができない．メモリの切替えはハードウェア割込みによって，ソフトウェアに通知される．

通常系とスタンバイ系のメモリへのアクセスについて起こり得る形は図 14 に示す 7 通りである．これら 7 つの状態のうちどの状態でメモリへのアクセスを行うかはソフトウェアによって設定される．

また，メモリ系の異常時に，どの状態にメモリを切り替えるかもソフトウェアによって設定される．本来なら，メモリ系に異常が発生した場合，どちらのメモリに異常があるのかを判定して，正常なメモリ系だけで処理を実行しなければならない．しかし，本システムではメモリエラーを疑似的に発生させるため，どちらのメモリに異常を起こさせるかが，どちらのメモリに切り替えるかを決定してしまう．

4.3 オペレーティングシステムでのサポート

以上で述べたように不具合と判断されたプロセッサの分離やメモリの切替えはハードウェアによって自動的に行われる．しかし，プロセッサの不一致はプロセッサの故障のほかに，一過性のノイズによっても引き起こされる．もし，不一致が一過性のノイズによるものであれば，システムをプロセッサ 2 個で動作する状態から 3 個で動作する状態に戻さねばならない．

ソフトウェアは I/O ポートをポーリングすることにより各プロセッサの状態を知る．それにより，ソフトウェアはシステムがプロセッサ 3 個で動作する状態から 2 個で動作する状態になったことを知ることはできる．しかし，その原因がプロセッサの故障によるものか一過性のノイズによるものかを判断することはできない．そこで，I/O 命令でシステムをリセットし，強制的にプロセッサ 3 個で動作するモードに復帰させることができるようになっている．この操作を何回か繰り返しても同じプロセッサが不具合と判断されるとき，そのプロセッサは故障したものとみなすことができる．このときソフトウェアは，不具合プロセッサを除いた 2 個でシステムを再スタートさせる．どの 2 個のプロセッサを使用するかは I/O 命令で設定できる．

これまで述べてきたように，V60/V70 は高信頼化システム構築のために必要な機能の一部を FRM 機能としてプロセッサ自身が提供している．このため，これまでに述べた 3 重化システム構成をとるプロセッサ上で動作するオペレーティングシステムにおいても

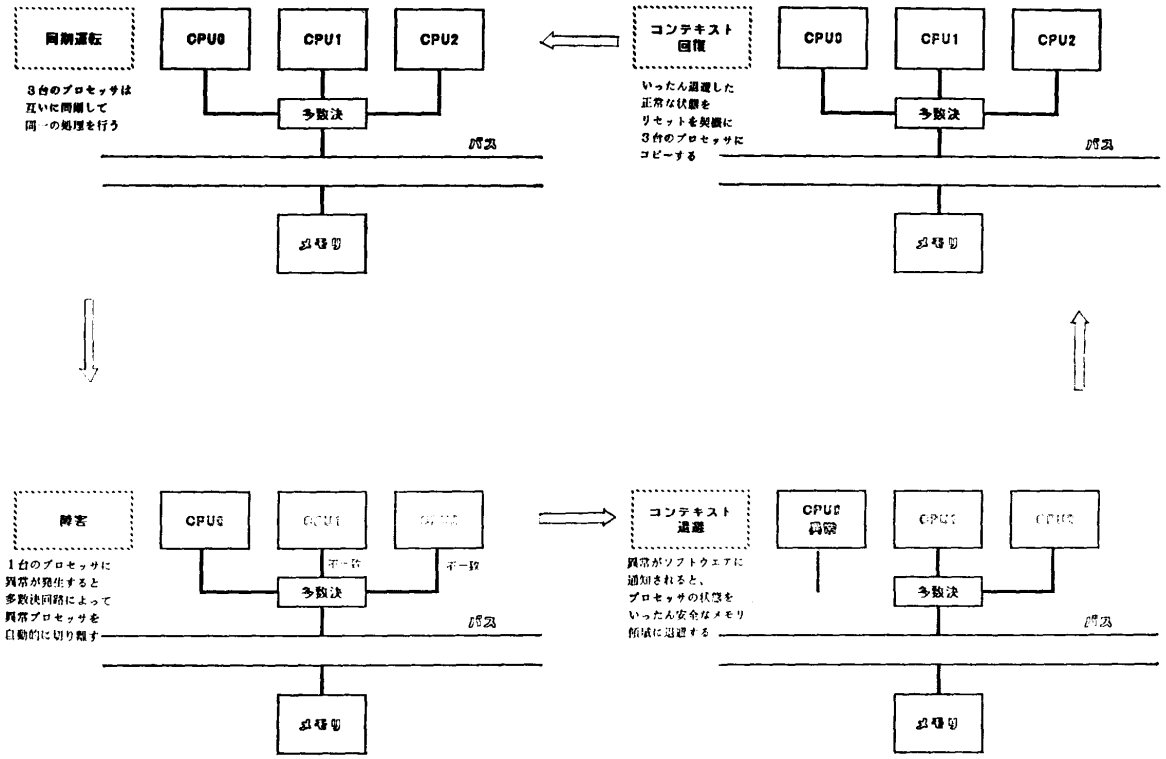

図 15 障害からの自動復旧
Fig. 15 Auto-recovery from accident.

高信頼化システムを構築するための基本機能を容易に提供できるようになる．具体的には，障害から自動復旧するための処理の再開機能[5]である．この機能により，自動復旧可能な障害に対しては，ユーザタスクは障害の発生には関知せず処理を続けることができる．実際，この機能は V60 リアルタイム OS (RX 616 の改造版) で実現されているものである．この機構を**図 15** に示す．

図 15 に示すように，3 重化構成をとっているプロセッサからの電源雑音，放射線などによる一時的誤動作は，プロセッサ状態やメモリ状態は正常であると考えられるため，完全再開が可能である．このとき，オペレーティングシステムはシステムコールにより，多数決回路の判断によって分離されているプロセッサを再統合して実行を再開する．

5. むすび

V60/V70 の提供する FRM 機能を用いて高信頼化システムを実現する例を紹介した．

V60/V70 の応用分野としては，スーパパソコン，EWS，大規模 PBX，シングルボードコンピュータシステムなどが考えられるが，応用システムの多様化，高性能化に伴って，マイクロプロセッサに対する高速化の要求が今後一段と高まることが予想される．現在，我々は V60/V70 に引続き 1 チップ上に約 90 万トランジスタを集積化し 10 MIPS 以上という処理能力を有した分岐予測機構とキャッシュメモリ搭載型の V80 を開発中である．

謝辞 最後に，発表の機会を与えていただいた可児事業部長，松本部長に感謝いたします．また，ボード作成にあたり，協力いただいたネオローグ電子(株)の方々にも感謝します．

参考文献

1) 佐藤ほか：仮想記憶管理機構と浮動小数点機構を内蔵した 32 ビットマイクロプロセッサ V60，日経エレクトロニクス， No. 391, pp. 199-240, Mar. 24 (1986).
2) V60/V70 アーキテクチャ・マニュアル，日本電気 (1987).
3) 山畑ほか：マイクロプロセッサ V60 のアーキテクチャ，情報処理学会マイクロコンピュータ研究会，43-2 (1987).
4) 野原ほか：32ビット・マイクロプロセッサ・V60

の FRM 機能と高信頼化システムの構成，電気通信学会技術研究報告 FTS 86-9～16 (1988).
5) 古城ほか：耐故障機構とランデブ機能を備えるV60 リアルタイム OS，日経エレクトロニクス，No. 417, pp. 173-201, Mar. 23 (1987).

(昭和63年 4 月 8 日受付)
(昭和63年11月14日採録)

河本 恭彦

昭和34年生．昭和58年東京大学工学部計数工学科卒業．昭和60年同大学大学院工学系研究科計数工学専門課程修士課程中退．同年日本電気(株)入社．以来V60, V70, V80マイクロプロセッサの開発に従事．現在マイクロコンピュータ事業部システム部に勤務．

矢野 陽一

昭和29年生．昭和53年京都大学工学部数理工学科卒業．昭和55年同大学院数理工学専攻修士課程修了．同年日本電気(株)に入社．昭和58年～59年，カナダ・ウォータールー大学電気工学科 Research Associate. 現在，マイクロコンピュータ事業部に勤務．ハイエンド・マイクロプロセッサ，グラフィクスなどのシステム設計に従事．IEEE 会員．

鈴木奈利子

昭和 59 年，東京農工大学工学部数理情報工学科卒業．同年，日本電気(株)入社．V60, V70, V80マイクロプロセッサの開発に従事．現在マイクロコンピュータ事業部システム部勤務．

藤井 卓哉

1961 年生．1985 年，関西大学工学部電気工学科卒業．同年，日本電気(株)入社．マイクロコンピュータ事業部において，32 bit マイクロプロセッサの開発，特に，プロセッサの性能評価に従事．

椎葉 忠明

昭和 56 年同志社大学工学部電子卒業．同年日本電気(株)入社．以来，マイクロコンピュータの開発に従事．