

連載講座



~~~~~  
 計算機の記憶システム—Ⅲ  
 ~~~~~

メインフレームの記憶システム

—高速化、大容量化、高信頼化を目指して—†

竹田 克己†† 桐生 芳雄†††

1. はじめに

近年のメインフレームの利用分野は、オンライントランザクション処理や大規模データベースの参照処理など、多量のデータを高速転送する必要がある分野が多く、そのデータ転送を制御する記憶システムの性能はメインフレームの性能に大きく影響する。この稿では、メインフレームの記憶システムの高速化技術と高信頼化技術について解説する。

2. 高速化技術

記憶システムの高速化とは、主記憶 (Main Storage, 以下 MS) からのデータのアクセス時間を短縮することと、単位時間に転送可能なデータ量を増やすことであるが、この実現手法としては、記憶装置の階層化技術が多く用いられている。階層化技術は、命令プロセッサ (Instruction Processor, 以下 IP) と MS の間に小容量だが高速のキャッシュ^{†)}を置く方式である。すなわち、参照頻度が高い MS データの一部をキャッシュに置き、必要なデータがキャッシュにあるときは MS の代わりにキャッシュをアクセスすることにより、実効的なアクセス時間を短縮するものである。キャッシュには、実装位置が IP に近いほうから 1 次キャッシュ (バッファ記憶とも言う) と 2 次キャッシュ (ワーク記憶、システムキャッシュ、グローバル記憶などとも言う) の 2 種類ある。メインフレームには、1 次、2 次の両キャッ

シュを備えたものと、1 次キャッシュのみのものがある。メインフレームの記憶システムの例を図-1 に示す。

この記憶システムは、1 次キャッシュ、2 次キャッシュ、主記憶の 3 階層の記憶装置より構成される。各記憶装置の特徴を以下に説明する。また、MS と入出力装置の間に位置する記憶装置として拡張記憶装置があるが、これについては次章で述べる。

キャッシュは、1 次、2 次両キャッシュ共に MS のデータブロックを一時的に保持しておく記憶装置であり、MS よりも高速なアクセスが可能である。キャッシュ内のデータブロックを書き変えるときは、対応する MS のブロックも書き換えなければならない。この MS の更新時期には、下記の 2 方式がある。

(a) ストアスルー (Store-through) 方式

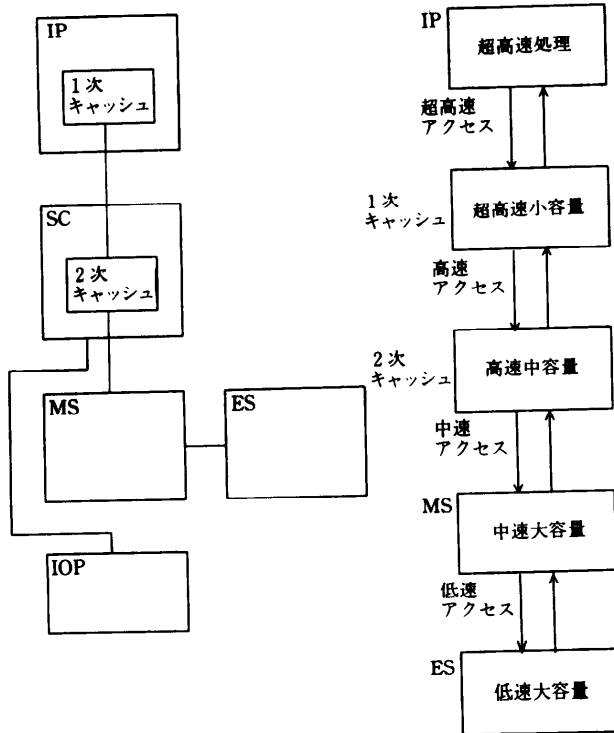
IP がストアを行うときは、MS へは必ずストアするが、キャッシュへはヒット時にのみ、すなわちストアするデータブロックがキャッシュに存在するときのみストアする方式である。この方式では、ストアするたびに MS へのアクセスを必要とするため、MS のアクセス時間を小さくする必要がある。この時間が大きいと、MS 性能がボトルネックとなって、1 次キャッシュの性能が十分発揮できないことになる。この方式では最新のデータが MS にあるので、IP のデータアクセス時には MS のみアクセスすれば良く、マルチプロセッサ構成でも、他の IP のキャッシュの内容検索が不要で制御がしやすいなどの利点がある。

(b) ストアイン (store-in) 方式

IP がストアを行うときに、キャッシュにのみ書き込む方式である。MS に書き込む時点は、このデータブロックがキャッシュ内で新たなデータ

† Storage Systems of Mainframes—Aiming at High Performance, Large Capacity and High Reliability by Katsumi TAKEDA (Hitachi Ltd. General Purpose Computer Division RISC Development Department) and Yoshio KIRIU (Hitachi Ltd. General Purpose Computer Division Computer Development Department II).

†† (株)日立製作所汎用コンピュータ事業部 RISC 開発部
 ††† (株)日立製作所汎用コンピュータ事業部開発第 2 部



MS: 主記憶 (Main Storage) ES: 拡張記憶 (Extended Storage)
 IP: 命令プロセッサ (Instruction Processor)
 SC: システム制御装置 (Storage Control)
 IOP: 入出力プロセッサ (Input Output Processor)

図-1 記憶システム

ブロックに置き換えられる時点である。書き込むときにキャッシュにデータが存在しないときは、書き込み対象となるデータブロックを MS からキャッシュに転送した後にキャッシュに書き込みを行う。すなわちこの方式では、書き込みごとに主記憶をアクセスする必要はないので、MS に対する書き込みの頻度を減らすことができる。ただし、最新のデータはキャッシュ上にあるため、マルチプロセッサのデータアクセス時は、他の IP 内のキャッシュの内容検索が必要となり、ストアスルー方式に比べて制御が複雑になる。

(1) 1次キャッシュ

容量最小だが、最高速の記憶装置で IP 内に置かれる。容量は、大型メインフレームで数 100 キロバイト程度である。容量の拡大と、命令とオペランドの同時アクセスを目的として、命令格納用とオペランド格納用の 2 種類持つこともある。主記憶更新の方式は、機種によりストアスルー、ストアイン方式のいずれかが採用されている。

(2) 2次キャッシュ

容量、アクセス速度とも 1 次キャッシュと MS の中間に位置する記憶装置で、大型メインフレームでの容量は数メガバイト程度である。マルチプロセッサシステムでは、複数の IP で共有される形態が多いので、バンクを複数設けて複数 IP からのアクセス要求を同時処理可能とするなどの性能向上策を実施している。

2 次キャッシュの役割は、1 次キャッシュミスヒット時のペナルティ (データ転送時間の増加分) を小さくするものであるが、この 2 次キャッシュを持つか持たないかは、次のような点を考慮して決定する。

- ① 1 次キャッシュミスヒット時のペナルティの目標値
- ② 2 次キャッシュがあれば、MS は低速で安価な DRAM で構成してよいが、ないときは高速だが高価な SRAM で構成する必要が生じることがある。したがってコストパフォーマンスの点からの検討が必要となる。

(3) 主記憶装置 (MS)

大型メインフレームではギガバイト級の容量を持つ。性能向上方式としては、複数バンクを設けてアクセスのコンフリクトを低減するのが一般的である。記憶素子としては、性能と経済性のバランスを考えて、DRAM または SRAM が選択される。

3. 拡張記憶

3.1 概要

現行メインフレームの拡張記憶 (Extended Storage, 以下 ES) の容量は、最大数ギガバイトであり、MS とディスク装置の間にある。データ転送速度では、ES は数 100 MB/S であり、MS より遅く、ディスク装置より速い。また、単位容量当たりの価格についても MS より安くディスク装置より高い。

ES は、このような特徴を活かして、次のように使用される。

(1) 高速な MS の実効的な容量拡張

ES を MS の容量限界から生じるページングや

スワッピング用の記憶装置として利用することにより、ほぼ MS プラス ES の容量に近い性能を実現する。MS を増設した場合に比べて、コストパフォーマンスを改善することができる。

(2) 入出力時間の短縮

ディスク装置は、チャンネルを介して CPU に接続するので、データ転送速度は 20 MB/S を越えない。ディスク装置上に置いたファイルを ES 上に置くことにより、入出力時間を短縮することができる。

ES の接続方式を図-1 に示す。

3.2 制御方式

ES のデータ転送は、MS との間で行う。現行メインフレームの ES のデータ転送方式には、同期型と非同期型の 2 通りがある。

(1) 同期型転送

命令語で転送を起動し、データ転送を終了するまで命令語を終了しない。すなわち、データ転送中は次命令に進めないで短いデータの転送に適している。

(2) 非同期型転送

命令語はデータ転送を起動するのみで終了し、データ転送の終了は割込みで報告される。すなわち、データ転送を起動したら命令の終了を待たずに次の命令に進むので、データ転送中の待ち時間をなくすることができるため、長いデータの転送に適している。

ES には、CPU 筐体に内蔵するもののほかに、独立した筐体を持って、以下のように使用できるものもある。

(1) 複数システム間で共有できる。

(2) 不揮発化機構（停電時にバッテリーにより、メモリ内容を保存するなど）を持つ。

(3) 二重書き機構により、同じデータを二つの ES に同時に書き込み、片方の記憶装置で障害が発生しても、正常な ES を使用してメインフレームの運転を続行する。また、運転中に障害がある ES を修理し、再度二重書き運転に復帰させることができる。

この形式の ES は、上記のような特徴を活かしてたとえば次のような利用が考えられる。

(1) ディスクに準ずる高速な入出力装置として使用する。

(2) 複数システム間を高速に結合し、従来型

の疎結合マルチプロセッサより性能、信頼性、運用性を向上させた複合システムを実現する。

ES は、半導体の高集積化、高速化を利用していくことができるので、今後発展していく装置であると考えられる。

4. マルチプロセッサの記憶システム

メインフレームは約 5 年ごとに新機種が出荷されるが、ユーザが新機種に対して要求する性能向上は、平均して旧機種の 3~4 倍とされている。これに対してシングルプロセッサで達成できる性能向上は、約 2 倍であり、ユーザ要求を満たすことができない。したがって、複数のプロセッサを備えたマルチプロセッサを開発し性能を向上させることが必要となる。マルチプロセッサは、複数の IP で MS を共有するハードウェア構成であり、一つの OS で制御されるので、シングルプロセッサのイメージを保ったまま処理能力の向上を図ることができる。しかし複数の IP で MS を共有することから、MS のアクセスで競合が生じるなどのハードウェアに起因する性能低下要因とその他のソフトウェアに起因する性能低下要因により、IP の台数に比例した性能を実現することが困難という問題がある。

現在の大型メインフレームのマルチプロセッサは、IP 8 台からなるものが最大であるが、今後の性能向上要求に応じていくためにも、マルチプロセッサの性能向上は重要な課題である。

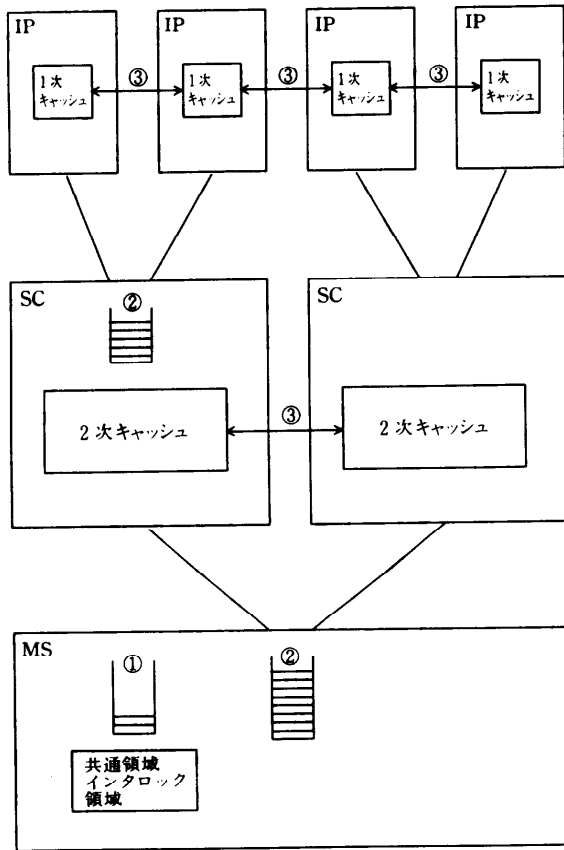
4.1 マルチプロセッサの性能向上

4.1.1 性能低下要因

マルチプロセッサの性能は、記憶装置（1 次キャッシュ、2 次キャッシュ、MS）からのデータアクセス時間に大きく依存する。マルチプロセッサでは、複数の IP で記憶装置を共有するので、記憶装置でのアクセスリクエスト競合などシングルプロセッサにはなかった性能低下要因が存在する。この性能低下要因は図-2 に示すが、主に以下の三つである。

(1) 複数の IP に、同じメモリ領域が割り当てられているとき、IP が同時にその領域をアクセスすることによる競合。複数 IP が同じメモリ領域を排他的にアクセスするための排他制御もこれに含まれる。

(図-2 の①)



IP: 命令プロセッサ SC: システム制御装置 MS: 主記憶
 ① 複数命令プロセッサに同じメモリ領域を割り当てたことから生じるメモリのアクセス待ち (排他制御を含む)
 ② MS, 2次キャッシュアクセス待ち
 ③ 1次キャッシュ, 2次キャッシュ一致制御

図-2 マルチプロセッサの性能低下要因

(2) 複数の IP が共有する記憶装置 (MS, 2次キャッシュ) でアクセス競合。

(図 2 の②)

(3) 複数の キャッシュ間での内容の一致制御。

(図 2 の③)

(1)の問題を解決するためには、プログラムをマルチプロセッサ向きに改良することが有効である。すなわち、IP 間で共有する必要がないデータ領域はできるだけ IP 固有に割り当てるのが、対策となる。

またマルチプロセッサでは、複数の IP が同一の記憶領域に書き込みを行うと、結果を誤ることがある。これを防止するために、複数 IP の書き込みを排他的に、すなわち一つのプロセッサの書き込みが終わったあとに次の書き込みを行うとい

う処理をする。具体的には、IP が同一記憶領域をアクセスするときは、その領域を確保するためのインタロック命令 (TEST AND SET 命令, COMPARE AND SWAP 命令など) を発行するが、この命令は他の命令に比べて実行すべき内容が複雑で多く、実行時間が長くなる傾向がある。排他時間に要する時間を短縮するために、特別のハードウェアを用意してインタロック命令を高速にすることも効果がある。

(2)の問題は、複数の IP が、IP と記憶装置間のデータバスを共有することから生じる。この問題を解決するには、次のようなハードウェア方式が有効である。

① キャッシュと MS 間、キャッシュとキャッシュ間で発生するデータ転送の高速化。

② キャッシュの制御方式改良によるヒット率の向上。これにより、キャッシュと MS 間のデータ転送を低減する。

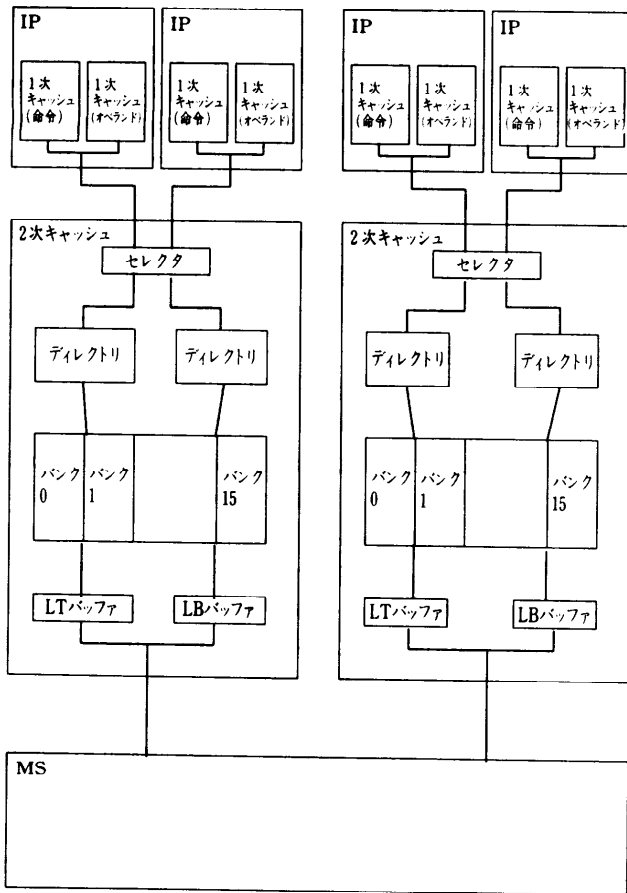
③ キャッシュ, MS の制御方式改良により、アクセス競合による待ち時間を低減する。複数の IP で共有される 2次キャッシュは、アクセス競合の低減が特に重要である。

図-3 の具体例を参照しながら、マルチプロセッサの記憶システムの高速化方式を解説する。

4.1.2 高速化方式

まず 1次キャッシュの高速化について説明する。1次キャッシュは、IP の命令実行に追従できる高速アクセスが必要なほか、命令、オペランドそれぞれのアクセス間の競合による待ち時間を低減する必要がある。(シングルプロセッサでも事情は同じである。)このためには、命令格納用とオペランド格納用の 2種類の 1次キャッシュを用意してアクセス競合を低減する方式が有効である。

1次キャッシュのブロックサイズを変更する方法もある。一般的には、ブロックサイズを大きくすると、ヒット率の向上が望める。これは、2次キャッシュ (2次キャッシュがない場合は、MS) から転送されるブロックの中には、IP がアクセスしたデータのほかにそれに隣接したデータも含まれているからである。しかしブロックサイズを大きくすることによるマイナス要因も存在する。



IP: 命令プロセッサ MS: 主記憶 LT: ライン転送
LB: ラインバック

図-3 マルチプロセッサの記憶システム

すなわち、大きなブロックを2次キャッシュ（2次キャッシュがない場合はMS）から読み出す時間が長くなることと、キャッシュに登録したブロックの中のデータで使用しないものが増えてくることである。ブロックのサイズについては、メインフレームの応用分野もよく考慮して決定する必要がある。

次に2次キャッシュの高速化方式について説明する。2次キャッシュは、複数IPで共有されるものであるから、これがボトルネックにならないように特に注意する必要がある。

2次キャッシュは、ディレクトリ部とキャッシュ本体からなる。ディレクトリは、あるデータがキャッシュに登録されているか否か、またキャッシュのどこに登録されているかを示す。2次キャッシュをアクセスするときは、まずディレクトリ

を参照してデータの有無を確認し、データがある場合のみキャッシュ本体をアクセスする。すなわち、キャッシュ本体に対するアクセス頻度よりもディレクトリに対するアクセス頻度のほうが高い。したがって、ディレクトリに対するアクセス能力を高める必要があるが、これは図-3に示すように、ディレクトリを複数持つて、複数のディレクトリアクセスを同時に可能とするなどの方法が取られる。

2次キャッシュ本体の高速化については、次のような方式が取られる。すなわち、2次キャッシュを複数のバンクに分割して1度に多くのデータのアクセスを可能とすることである。図では、8バイト幅のバンクを16用意しているので、1度に $16 \times 8 = 128$ バイトのデータアクセスができる。

その他、図-3の方式では2次キャッシュとMSのデータ転送速度の差を解消するために、2次キャッシュ側にライン転送(LT)、ラインバック(LB)用のバッファを設けている。すなわち、MSから2次キャッシュにライン転送する場合、また逆に2次キャッシュからMSにラインバックする場合は、2次キャッシュのデータ転送速度のほうが大きいので、

キャッシュ側でいったんバッファする。バッファと2次キャッシュの間は、2次キャッシュの転送速度でデータ転送し、2次キャッシュがビジーとなる時間を短縮する。

4.1.3 キャッシュコヒーレンス制御

(3)の問題は、複数存在するキャッシュの内容を一致させるための制御であり、キャッシュコヒーレンス制御とも呼ばれる。

キャッシュは、MSの一部データのコピーであるから、MSブロックが、複数のキャッシュ内に存在しているときは、それらの内容は一致していなければならない。さもなければ、あるIPがロード命令を実行したときに読み出す値と、他のIPが同じメモリ番地に対して実行したロード命令で読み出す値が違うということがおきてしまう。

通常のキャッシュの制御では、読み出しについては、同一のブロックが複数のキャッシュに存在することを許すが、書き込み時には、ただ一つのキャッシュにブロックが存在するようにする。すなわち、キャッシュコヒーレンスは、書き込みはただ一つのキャッシュを行うことで保証する。メインフレームのコヒーレンス制御は、以下の3方式が代表的なものである。

①書き込みアドレス送信方式

キャッシュのブロックへ書き込みを行うとき、そのブロックのアドレスをマルチプロセッサ内の他のキャッシュに送信する方式である。これを受け取ったキャッシュ側では、そのブロックがキャッシュ内にあるか否か調べ、あればそのブロックを無効にする。

別の方式としては、ブロックのアドレスとともにデータも他のキャッシュに送信する方式がある。これを受け取ったキャッシュ側では、ブロックがキャッシュ内にあるか否か調べ、あれば、そのブロックに送られてきたデータを書き込む。この方式は、単なるブロック無効化に比べて制御が複雑であり、一般には、あまり用いられていない。

この書き込みアドレスの送信方式は、通常はストアスルー方式のキャッシュと組み合わせて用いられる。これは、ストアスルー方式では、最新のデータはMSにあるため、キャッシュのブロックを無効化しておけば、次にそのブロックをアクセスしたときにブロック転送が発生し、最新のデータがMSからキャッシュに書き込まれるからである。この方式の短所は、IP台数が増えるにつれて、アドレスを送信すべきキャッシュの台数が増え、ブロックの無効化に要する時間も増えていくことである。したがって、大規模なマルチプロセッサへの適用は難しく、メインフレームでは、IP2台程度のマルチプロセッサに適用されている。

②システムの全キャッシュのディレクトリ情報を統括する統括ディレクトリ（グローバルディレクトリ、コピーフラグなどと呼ばれる）を使用する方式

統括ディレクトリを使用したコヒーレンス制御を図-4により、ストアスルー方式のキャッシュについて説明する。

(i) IP0がキャッシュ内のデータAをデー

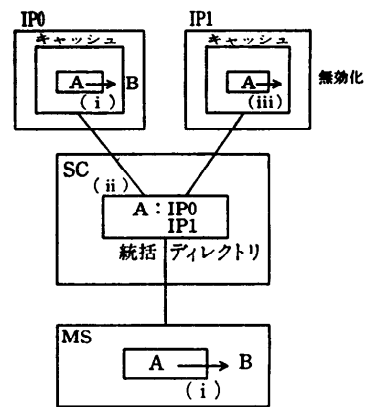


図-4 統括ディレクトリ方式

タBに書き替える。このとき、時間的なズレはあるが、MS内のデータAもデータBに書き替えられる。

(ii) システム制御装置内の統括ディレクトリを検索し、データAがIP0のほかにIP1のキャッシュにもあることを知る。

(iii) IP1のキャッシュを無効化する。これ以降、IP1がデータAを読もうとすると、キャッシュにデータが存在しないため、MSを参照して正しいデータBを得ることができる。

すなわち、統括ディレクトリを検索すれば、あるデータがどのIPのキャッシュに存在するかが分かるので、各IPのキャッシュのディレクトリを検索してデータの有無を判定する時間が省けるほかに、この検索のために、各IPのキャッシュ参照が待たされることがないので、IPのキャッシュ参照時間短縮にも効果がある。

以上は、ストアスルーキャッシュについて説明したが、この統括ディレクトリ方式はストアインのキャッシュにも使用できる。ストアインキャッシュでミスヒットしたときは、MSに最新データがあるとは限らないので、統括ディレクトリを検索してデータを保持しているキャッシュを知り、そこからブロックを要求元キャッシュに転送する。書き込み時にキャッシュでヒットしたときも、他のキャッシュにデータが存在している可能性があるため、統括ディレクトリを検索する必要がある。しかし、現在のメインフレームでは、次のような方式で、キャッシュのディレクトリ検索により、統括ディレクトリ検索の要不要を判定することも行われている。すなわち、キャッシュの

ブロックごとに読み出し専用状態、排他状態という属性を示すフラグを設ける。読み出し専用状態のブロックは、複数のキャッシュに存在している可能性がある。排他状態のブロックは、一つのキャッシュのみに存在する。したがって、読み出し専用状態のブロックに書き込みを行うときは、そのブロックを排他状態とするとともに、統括ディレクトリを検索して他のキャッシュに存在するブロックを無効にする。排他状態のブロックに書き込みを行うときは、たんに書き込みを行うのみで良い。

統括ディレクトリ方式は、現在のほとんどの大型メインフレームに採用されている。

③スヌープキャッシュ方式

図-5 に示すように、IP と MS 間をバスで結合するマルチプロセッサに適する方式である。

②のように統括ディレクトリを持つ替わりに、各 IP がバス上のアクセス要求を監視して、キャッシュコヒーレンス制御を行う。この方式は、バス上のトラフィックが小さくなるストアインキャッシュに適している。この制御は、概略次のように行う。

(i) 読み出しでキャッシュヒット時
バス上へのアクセス要求は、不要である。

(ii) 読み出しでキャッシュミスヒット時

読み出し要求をバス上へ送り出す。他の IP はバスを監視していてこの要求を知り、自キャッシュ内に最新データがあれば、それをバスに乗せて要求元のキャッシュに送る。最新データがキャッシュになければ、MS よりデータを読む。

(iii) 書き込みでキャッシュヒット時

自キャッシュに書き込むとともに、バス上に書き込み要求を送り出す。他の IP はバスを監視していてこの要求を知り、自キャッシュ内にそのデータがあれば、それを無効とする。

(iv) 書き込みでキャッシュミスヒット時

(ii)と同様に、他のキャッシュまたは MS よ

りデータを自キャッシュに転送してから書き込みを行う。他のキャッシュ中のデータは、無効にする。

上記は、基本的な動作原理であり、実際にはバストラフィックを減らす工夫をすることが多い。この方式は、IP と MS をバスで接続するマルチプロセッサに適合する。したがって、サーバ、ワークステーションなどのマイクロプロセッサを結合するタイプのマルチプロセッサには適しているが、大型メインフレームのようなバス結合でないマルチプロセッサには適用できない。

5. RAS 技術

メインフレームの記憶システムは、高速・小容量から低速・大容量までの各種の記憶装置を階層化し、高速化と大容量化を両立させる多階層化が進んでいる。たとえば、高速・小容量な記憶装置には超高速・高速な論理内蔵メモリ LSI を使い、また、中低速・大容量な記憶装置は大容量な DRAM を使うことによりコストパフォーマンスの良い高速・大容量記憶システムを実現している。

このように各種の記憶装置で構成される記憶システムでは、DRAM を採用した大容量な記憶装置に注目するだけでなく、記憶システム全体を考慮したバランスの良いチェック・リカバリ方式が重要である。

また、DRAM を大量に使用している中低速・大容量な記憶装置では、スループット向上のためのウェイ数の増加傾向と、半導体技術の急激な進歩による DRAM の大容量化の中で、システムの処理量に応じた広範囲の記憶容量レンジをカバーするため、入出力を多ビット化した DRAM の採用が進んでいる。

こういった背景の中で、最近のメインフレームの記憶システムの RAS 技術では

- (1) 各階層の記憶装置の特質を活かしたチェック・リカバリ方式
- (2) DRAM の入出力多ビット化に対応したハード構成や誤り訂正符号化の採用
- (3) 2重障害の防止を考慮したソフトウェアの回復処理などの工夫がなされている。

たとえば、図-6 に示すように、1次キャッシュ、

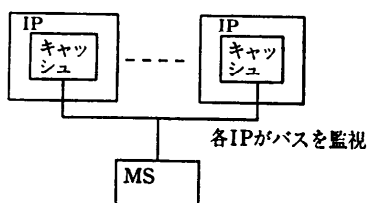
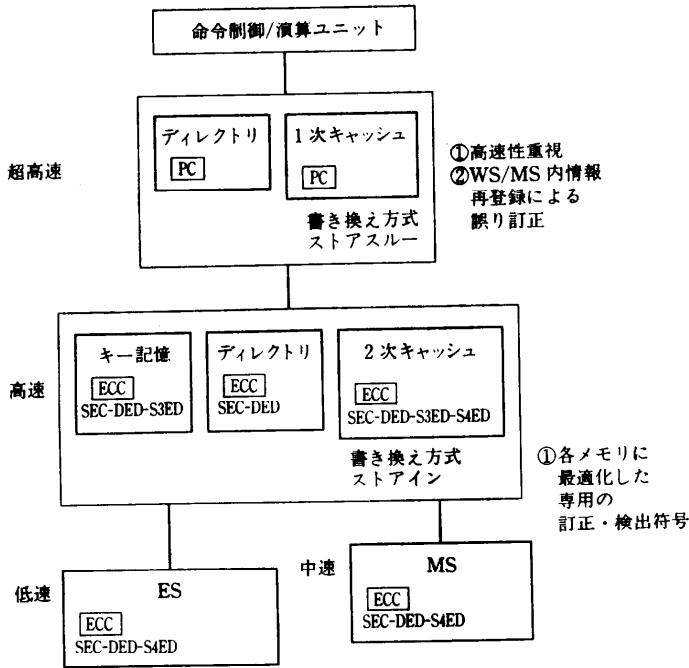


図-5 スヌープキャッシュ方式



PC (Parity Check)
 ECC (Error Checking and Correction または Error Correcting Code)
 SEC-DED (Single bit Error Correcting-Double bit Error Detecting)
 符号 1ビット誤りを訂正し、2ビット誤りを検出することができる符号
 SEC-DED-S b ED (SEC-DED-Single b-bit byte Error Detecting) 符号
 SEC-DED 符号に、1カ所のバイト誤り (bビット幅) の検出機能を持たせた符号

図-6 各階層の記憶装置の特質を活かしたチェック・リカバリ方式の例

2次キャッシュ、MS、ES の多階層の記憶装置を採用したメインフレームの例では、高速性や書き換え方式、RAM のビット構成、経済性など各記憶装置の特質を活かしたチェック・リカバリ方式と誤り訂正・検出符号^{3)~5)}を採用している。

DRAM の入出力多ビット化に対応したハード構成の工夫としては、記憶装置の読み書き単位が

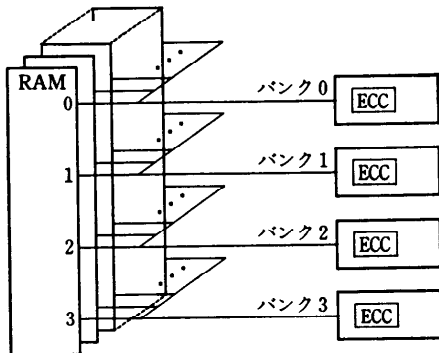


図-7 RAM 出力のバンク間スクランブル割付けの例

大きい超大型・大型メインフレームでは、図-7のように、RAM 出力の多ビット誤りが個々のバンクの誤り検出・訂正 (ECC) 回路では1ビット誤りにみえるように、各出力を別々のバンクにスクランブルした割り付けを行っている。

一方、読み書き単位が小さい小形メインフレームでは、DRAM の入出力ビット数 b に対応した b ビットのかたまりの誤りを訂正できる $SbEC-DbED$ 符号⁶⁾を採用し始めている。

また、DRAM の内部構造についても、図-8に示すように、入出力ごとにワード線を分割したアレイ構成とするなど、障害を局所化する工夫を取り入れている。今後、大容量化とともにさらに進むと予想されるDRAM の多ビット化に対応するには、こういったDRAM の内部構造にまで踏み込んだ標準的な障害の局所化が重要になると考えられる。

さらに、高集積化・高密度化にともなう α 線などの影響によるソフト

エラーについては、書き直すことにより回復するという特徴を利用した障害の累積防止策や、固定障害との重複障害対策などに種々の手法を取り入

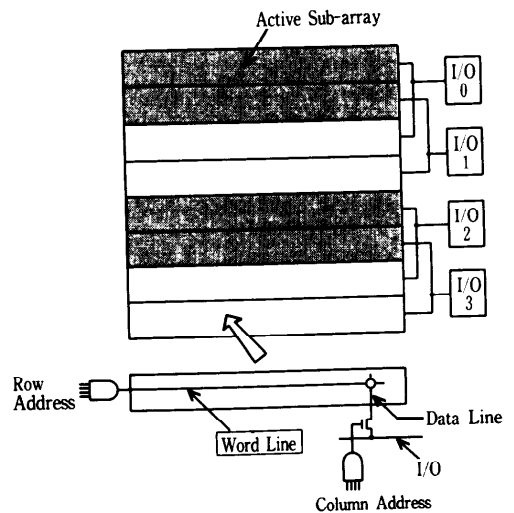


図-8 入出力ごとにワード線を分割したDRAMのアレイ構成例

れ代表的な例として、訂正可能な障害を検出したとき、正しい記憶情報に直しておく再書き込み、重複障害を検出したとき、数種のテストパターンにより固定障害位置を求めて誤りを訂正する消去訂正、また、訂正可能な障害を固定的に検出したとき、予備に切り替えるメモリチップ代替などがあげられる。

6. おわりに

本稿ではメインフレームの記憶システムを、現行機を例にして具体的に解説した。誌面の都合上、記憶システム全体にわたる解説はできなかった。現在の記憶システムの重要技術であるマルチプロセッサ技術を中心に解説した。読者諸氏の、メインフレームのハードウェアに対する理解の助けになれば幸いである。

参考文献

- 1) 大型汎用コンピュータの全て、日経コンピュータ、No. 237 (1990. 10.15).
- 2) 大型コンピュータ M-880 の処理方式とハードウェア技術、日経エレクトロニクス、No. 515, pp. 210-225.
- 3) Hamming, R. W.: Error Detecting and Error Correcting Codes, Bell Syst. Tech. J., Vol. 26, pp. 147-160 (Apr. 1950).

- 4) Hsiao, M. Y.: A Class of Optimal Minimum Odd-Weight-Column SEC-DED Codes, IBM J. Res. Develop., Vol. 14, pp. 395-401 (July 1970).
- 5) Kaneda, S.: A Class of Odd-Weight-Column SEC-DED-SbED Codes for Memory System Applications, IEEE Trans. Comput., Vol. C-33, No. 8, pp. 737-739 (Aug. 1984).
- 6) Kaneda, S. and Fujiwara, E.: Single Byte Error Correcting-Double Byte Error Detecting Codes for Memory Systems, IEEE Trans. Comput., Vol. C-31, No. 7, pp. 596-602 (July 1982).
- 7) 朴 泰祐: 計算機の記憶システム「キャッシュと仮想記憶の技術」, 情報処理, Vol. 33, No. 11, pp. 1348-1357 (1992).

(平成4年8月25日受付)



竹田 克己 (正会員)

昭和23年生れ。昭和46年東京大学工学部電子工学科卒業。同年(株)日立製作所入社。以来、大型計算機の開発に従事。



桐生 芳雄 (正会員)

1948年生。1969年長岡工業高等専門学校卒業。同年(株)日立製作所入社。大形汎用コンピュータ、スーパーコンピュータ記憶装置の開発に従事。現在に至る。

事、現在に至る。

