CMP アーキテクチャのもとで実現する CS 7000 シリーズの基礎技術

Basic Technology for CS 7000 Series System under CMP Architecture

櫻 井 成 一, 齋 藤 雅 樹

要 約 Unisys は ClearPath HMP（Heterogeneous Multi-Processing）シリーズでメインフレームと Windows 環境との融合性の高いシステムを提供してきた．CMP（Cellular Multi-Processing）アーキテクチャを採用した大規模エンタープライズサーバ ES 7000 を基礎として，メインフレームとオープンエンタープライズサーバに共通のプラットフォーム（CMP プラットフォーム）を導入することで，更にメインフレームとオープンシステムの融合を進めることができる．

ClearPath Plus サーバ CS 7000 シリーズはその実現方法により大きく二分される．ひとつは従来からのメインフレーム OS 環境をメインフレーム専用プロセッサにより実現した CS 7800/7400 シリーズであり，もう一つは Windows OS 環境でハードウェア・エミュレーション技術によりメインフレーム OS を稼働させる CS 7100/7200 シリーズである．どちらもオープンサーバとの共存を容易に実現できる．

本稿ではこれら CS シリーズの基礎技術を次の 2 部に分けて紹介する．

その 1 : CS 7800/7400 メインフレーム・システム

その 2 : CS 7100/7200 メインフレーム・エミュレーション・システム

Abstract Unisys has been offering the high harmonization of the Windows environment with the mainframe system in the ClearPath HMP (heterogeneous multi-processing) series. Based upon the ES 7000 technology adopting the CMP architecture, the common hardware platform could make harmonize easily both enterprise servers and mainframe systems. The ClearPath Plus Server CS 7000 series could be theoretically classified by their implementation into two categories. One is the CS 7800/7400 series system on which the mainframe OS environment is implemented with previous unique processor, and the other is the CS 7100/7200 series system on which the MCP environment on the Windows OS is implemented with an emulation technology of mainframe hardware. Both series systems can make an open server system available in their system. This paper introduces the fundamental technology for the CS series system, focusing on two topics, " CS 7800/7400 mainframe system with CMP architecture ", and " CS 7100/7200 mainframe emulation system with CMP architecture ".

1. はじめに

Unisys はこれまで ClearPath HMP シリーズとして IX シリーズと NX シリーズの 2 種類のメインフレーム製品群を市場に提供してきた．IX シリーズは OS 2200，NX シリーズは MCP（Master Control Program）と呼ばれるそれぞれ異なるメインフレーム OS（オペレーティングシステム）で動作する．Unisys は開発期間の短縮による迅速な後継メインフレーム商品の提供を目指し，それぞれの CPU や OS の特徴を生かしつつ，共通のプラットフォームを使用したメインフレームを生み出した．メモリ機構と I/O 機構を統一し，CPU のみの交換により異なる OS 環境での稼働を目指したプラットフォームが CMP プラットフォームである．

本稿その 1 では，CS 7800/7400 シリーズの第 1 弾である CS 7802 システムの開発に携わった筆者が，その経験をもとに，CMP プラットフォームで実現するメインフレームの優位性とその技術について紹介する．

また，Unisys には Windows　OS のもとで稼働する A/NX シリーズのハードウェア・エミュレーションの技術がある．ClearPath Plus Server CS 7100/7200 シリーズは，CMP アーキテクチャの優位性と発展性を背景に，10 年を悠に超えるエミュレーション・プロダクトでの実績を併せ持ったメインフレーム・エミュレーション・システムである．

本稿その 2 として，MCP 環境で稼働する CS 7101/7201 システムのハードウェア・エミュレーション技術を紹介する．

その 1 : CS 7800/7400 メインフレーム・システム

2.　CMP プラットフォームの概要

2.1　CMP アーキテクチャの概要

CMP アーキテクチャでは，プロセッサとメインメモリ間のクロスバー技術を発展させ，Intel CPU バスアーキテクチャと結合している．マルチ・プロセッサ環境で，より高い性能を得るためには，プロセッサ自身の高速化に加えて，メモリデータの高速アクセスが必要である．プロセッサの高速化技術の進み具合に比べ，現在でもメインメモリのアクセス速度は決して速いとは言えない．このギャップを埋めるために，CMP プラットフォームでは Dual Intel FSB（Front Side Bus，以下 FSB と呼ぶ）の採用とプロセッサとメインメモリ間に 3 段階のキャッシュメモリを採用し，できるだけプロセッサとメインメモリ間の直接のアクセスを抑えて高速化を実現している．

CMP アーキテクチャは 2000 年に ES 7000 に初めて採用され，最大 32 のプロセッサを Intel CPU バスアーキテクチャに合わせて結合し，マルチ・プロセッサ構成を実現した．その構成を図 1 に示す．

2.2　CMP プラットフォームの構成要素

CMP プラットフォームは次の主要ハードウェアにより構成されている．

1）　Crossbar Intraconnect（以下，CI）

Main Storage Unit（以下，MSU）と一対一に接続するための 4 本の MI Pipe，入出力装置と接続するための 2 本の MIO Pipe，及び TLC（Third Level Cache：以下，TLC）3 次キャッシュと接続するための 2 本の MT　Pipe と呼ばれるインタフェースを持つ．TCM（Third-level Cache Memory interface）と呼ばれる ASIC で制御され，各 CPU 及び I/O（DIB : Direct I/O Bridge 経由）からすべての MSU へのアクセスを可能としている（図 2）．

2）　MSU

最大 4 の MSU 構成が可能で，64 GB（1 MSU 当り最大 16 GB）のメモリ容量を搭載できる．四つの CI と MI Pipe によりそれぞれ一対一に接続される．

3）　TLC

Intel 互換の FSB に接続され，最大 4 の CPU により使用される共有（Shared）

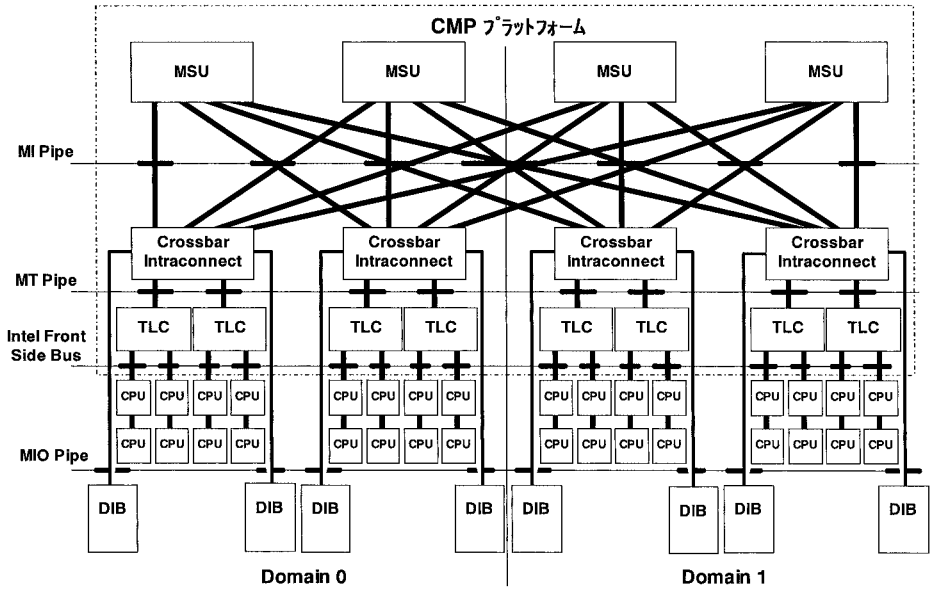

図 1　CMP プラットフォームの構成

キャッシュであり，MT Pipe を介して CI 経由で MSU に接続される．16 MB あるいは 32 MB の容量を持つ SRAM と TCT（Third-level Cache Tag）と呼ばれる ASIC により制御される（図 2）．

ハードウェアの構成単位として，一つの TLC とそれに接続される 4 CPU を含むモジュールをサブポッド（SubPOD）と呼ぶ．また，一つの CI とそれに接続される二つの SubPOD をポッド（POD）と呼ぶ．従って，システムの最大構成は四つの POD と八つの SubPOD になる．図 2 に POD と SubPOD の関係を図示する．

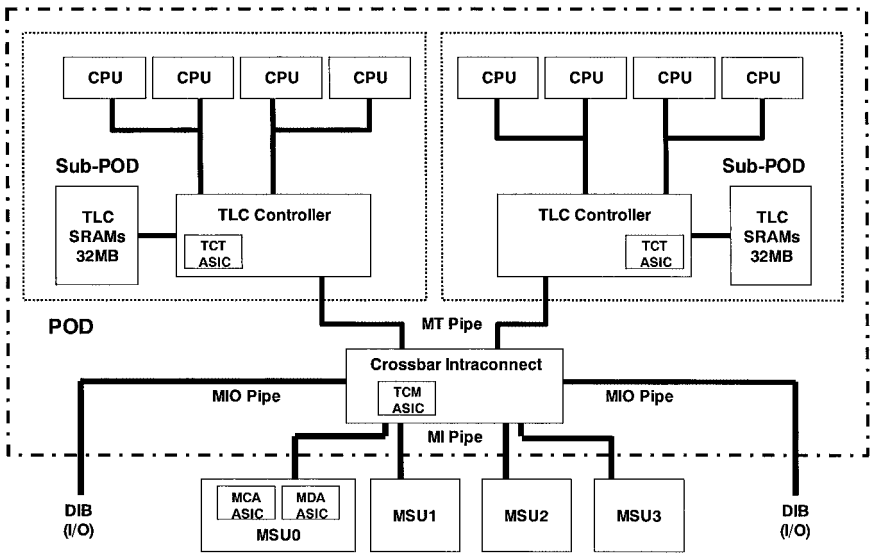

図 2　POD と SubPOD

なお，更に詳細な CMP プラットフォームのアーキテクチャについては技報通巻 66 号[1]に記述されているので参照されたい．

3. CMP プラットフォーム上でのメインフレーム環境の実現

3.1 パーティション分割（パーティショニング）

パーティショニングとは 1 筐体の中の構成要素を分割し，それぞれの独立したオペレーティングシステムを稼働させるための機能である．OS 2200 環境で稼働するパーティションを 2200 ノードと呼び，MCP 環境で稼働するパーティションを MCP ノードと呼ぶ．また，このほかに CS 7800/7400 シリーズでは Intel プロセッサ Chip による IA（Intel Architecture）サーバ環境も構成でき，この Windows 環境で稼働するパーティションを NT ノードと呼ぶ．

このように CS 7800/7400 シリーズでは，あるパーティションは OS 2200 環境（あるいは MCP 環境）で稼働し，他のパーティションは Windows OS で稼働するというように，異なったオペレーティングシステムが共存できる．パーティションは，CPU は SubPOD 単位で，I/O は MIO Bus 単位で，MSU は 32 MB の領域単位で構成可能となっている．

現在 CS 7800/7400 シリーズでは，OS 2200 及び MCP を稼働させるノードが最大 4 パーティション，Windows OS を稼働させる NT ノードが最大 4 パーティションの計 8 パーティションをサポートしている．将来は任意の組み合わせのパーティションでメインフレーム環境と Windows 環境が混在して稼働するようになる．

3.2 メインフレーム専用（OS 2200&MCP）プロセッサ

CS 7800/7400 シリーズはそれぞれのメインフレーム OS （OS 2200 あるいは MCP）専用プロセッサをもっており，それぞれの専用プロセッサは 2 本の Intel FSB に接続され，1 本の FSB Bus 上に 2，1 SubPOD 当り 4 のプロセッサの搭載が可能で，最大 4 SubPOD で 16 のマルチ・プロセッサが構成できる．ハードウェア構成上は最大 32 までのマルチ・プロセッサが搭載可能であり，将来は最大 8 パーティション 32 プロセッサのサポートが予定されている．

3.3 マルチ・プロセッサ環境での考察

CMP プラットフォームで動作する OS 2200 専用プロセッサを開発するにあたって，マルチ・プロセッサ構成においてのメモリデータのアクセスを高速化する課題があった．従来の IX シリーズでは，キャッシュメモリ構造が 2 段階で構成されていたが，CS 7802 システムではプロセッサ内部に 2 段階のローカル FLC（First Level Cache）と SLC（Second Level Cache），4 CPU によるシェアード TLC という計 3 段階のキャッシュ構造となる．

キャッシュの階層を増やせばキャッシュでの Hit 率が向上し，MSU へのアクセス頻度は減り，効率面でのメリットは大きいが，大規模なマルチ・プロセッサ環境でのキャッシュメモリのアクセス速度の高速化が課題とされた．3 章ではそのメカニズムの解析とそれに対応する方策について記述する．

3.3.1 マルチ・プロセッサ環境でのキャッシュ構造とレスポンス・タイム

CMP プラットフォームでは，各々の SubPOD のプロセッサ群から CI 経由で 4

MSU とのアクセスラインをもっているが，マルチ・プロセッサ環境では，それぞれの CPU が使用しているキャッシュ上のデータの正当性を常に管理しなければならない．あるキャッシュメモリでデータが更新されると，同じ MSU アドレスのデータを持つ他のキャッシュメモリはデータの正当性を確保するためにそれまで持っていたデータを無効（Invalidate）にする必要がある．これをキャッシュ・コヒーレンシ（Cache Coherency）と呼び，コヒーレンシを保つには，同じデータの存在するすべてのキャッシュメモリの場所とそれを更新する権利（オーナーシップ）を持っているプロセッサを管理しなければならない．この管理は，マルチ・プロセッサ環境が大きくなるほど，またキャッシュメモリの階層が増えるほど，複雑で時間のかかる作業となる．

キャッシュ上のデータは，大別するとシェアード状態かエクスクルーシブ状態になるよう定義される．シェアード状態とは参照のみが許され，複数のキャッシュ上に同じデータが複数存在できる．また，エクスクルーシブ状態とは，データを更新するためのオーナーシップを持っている状態で，データは常に一つのキャッシュ上のみに存在することが許される．プロセッサ数が多いシステムになればなるほど，これらの状態のキャッシュデータが方々のキャッシュメモリ上に散在する可能性が高くなり，キャッシュ上のデータを参照したり，更新するためのコヒーレンシの管理動作が無視できなくなる．

ここで，図 3 のキャッシュ構造の簡易図を使い，あるプロセッサ（図 3 中の IP-A）がデータを要求してからデータを受け取るまでの時間，いわゆるレスポンス・タイムについて考えてみる．

プロセッサのデータ・リクエストが FLC でミスした場合，データを上位のキャッシュから読み込む必要がある．このためレスポンス・タイムはデータが存在するキャッシュの場所が遠くなるほど長くなり，同じ CPU 内にあるローカル SLC（図中①）でデータがヒットした場合が一番短くなる．

以下最新データの存在する場所が，

1） 同じ FSB に接続されるもう一方の CPU 上の SLC（図中②），
2） 同じ SubPOD 内にある TLC（図中③），
3） 同じ SubPOD 内のもう一方の FSB に接続される CPU 上の SLC（図中④），
4） MSU（図中⑤），
5） 同じ CI 配下のもう一方の SubPOD 内にある TLC（図中⑥），
6） 同じ CI 配下のもう一方の SubPOD 内にある CPU 上の SLC（図中⑦），
7） 別の CI 配下の SubPOD 内にある TLC（図中⑧），
8） 別の CI 配下の SubPOD 内にある CPU 上の SLC（図中⑨）

の順にレスポンス・タイムは長くなっていく．

以上の事を考えると，シングル・プロセッサや小規模なマルチ・プロセッサ環境では問題とならないが，大規模なマルチ・プロセッサ環境を 1 パーティションで稼働させるような場合に，多数のプロセッサによる同じキャッシュデータの参照や更新が頻発するとレスポンス・タイムの遅延が起こり得る．

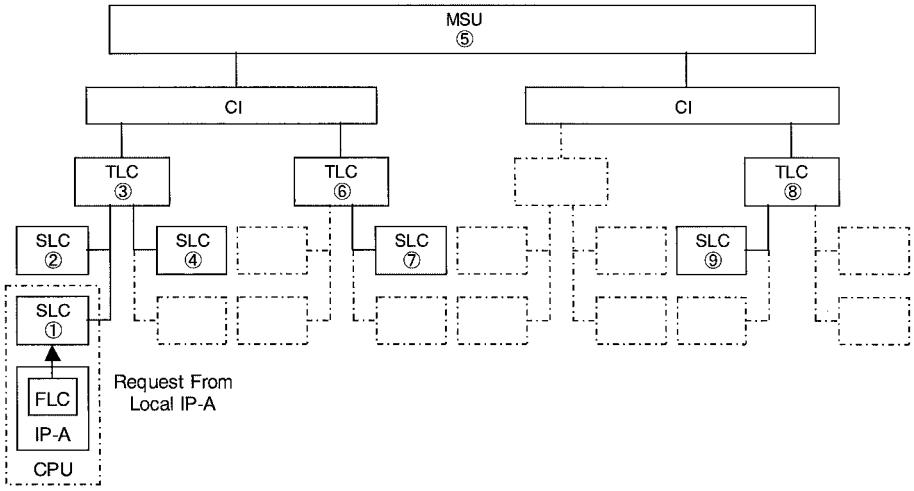

図 3　CS 7800/7400 シリーズキャッシュ構造簡易図

3.3.2　ハードウェアによるプロセッサ効率を向上させる技術（Leaky Write Cache）

CS 7802 システムではハードウェアとソフトウェア両面からキャッシュメモリの使用環境を最適化するよう工夫し，大規模なマルチ・プロセッサ環境でも効率の低下を防ぐことに成功している．

ここではハードウェアによりプロセッサ効率を上げるために考えられた Leaky Write という機能について説明する．通常，CPU のキャッシュ上のデータを非常駐にするため，上位キャッシュに書き戻す処理は LRU（Least Recently Used）アルゴリズムに基づいて行われる．このアルゴリズムは，頻繁に使用されるデータはなるべくキャッシュ上に残すように配慮されており，書き戻す対象となったデータの内，参照された時間が一番古いものを書き戻す．但し，前項で述べたように，複数のプロセッサが頻繁に参照するデータの場合は，あるプロセッサが更新した後，なるべく早く上位に書き戻した方が他のプロセッサのレスポンス・タイムを短くすることになり，効率的である．そこで考えられたのが，この Leaky Write という考え方である．

この Leaky　Write 機能を使用してプロセッサがデータを更新する場合，そのプロセッサのローカル SLC でデータがヒットした時は，データを更新後 LRU のテーブルを一番古いものにし，なるべく早く上位キャッシュに書き戻すように配置する．また，データがローカル SLC 上でミスした場合は，読み込みデータを SLC 上に常駐させることなく，CPU　ASIC 内部にある Storage　制御用バッファ上でデータを更新し，そのまま上位キャッシュに書き戻すようにしている．

この機能を使用してデータの更新を行うか否かは，プロセッサの命令語によりマイクロコードで制御する．命令語の特性によって使用の有無を決めることが可能で，複数のプロセッサから頻繁に参照される可能性の高いセルを参照するのに使用される命令語はこの機能を使用している．

3.3.3　ソフトウェアによるプロセッサ効率を向上させる技術（Affinity）

プロセッサ効率を向上させるための工夫はハードウェアのみならずオペレーティングシステムによっても行われている．その中でキャッシュメモリのヒット率を向上さ

せるソフトウェア的な機能を Affinity 機能と呼ぶ．メモリ領域はアクティビティ毎に割り当てられるため，キャッシュの効率を考えると一つのアクティビティは同じプロセッサで実行するのが効率的である．従って Affinity 機能では，あるプロセッサが割り込みによってアクティビティの処理を中断した時に，実行中のアクティビティを憶えておき，割り込み処理終了後，なるべく同じアクティビティを実行させるようにしている．また，該当プロセッサが忙しい場合はなるべく近くのプロセッサに実行させるようにプライオリティを付けている．

このプライオリティはレスポンス・タイムの早い順，すなわち；

① 同じ FSB のプロセッサ

② 次に，同じ SubPOD 内のプロセッサ

③ 次に，同じ CI 内のプロセッサ

の順になる．

さらに，シェアード・データを効率よく使用するため，アプリケーションをなるべく隣接しているマルチ・プロセッサ群（アフィニティ・グループと呼ぶ）に分けて割り当て，実行させるという機能も試されている．このアフィニティ・グループに参加するプライオリティも同様に，同じ FSB のプロセッサが一番高く，以下同じ SubPOD 内のプロセッサ，同じ CI 内のプロセッサの順になる．

4. CMP プラットフォームで CS 7800/7400 シリーズを構築する効果

CS 7800/7400 シリーズは，CMP プラットフォームを使用して，異なったメインフレーム OS 環境を，それぞれの専用プロセッサを開発することにより実現した．異なった OS 環境を CMP プラットフォームで実現することによって得られる効果には様々なものが挙げられる．ユーザにとっては，メインフレームと同等の RAS 機能を持ったオープンサーバと OS 2200/MCP メインフレーム環境の共存により，オープン化に対応した異種システム間マルチ・プロセッサ（HMP：Heterogeneous Multi-Processor）システムの構築が可能となった．メーカとしても，共通プラットフォームの採用による開発期間の短縮と製造の一元化によるメリットがある．

また，CS 7800/7400 シリーズでは，現行の I/O を接続するために，I/O 制御部に特別な工夫が必要であった．しかし，将来は業界標準の共通インタフェースをもつ I/O（Commodity　I/O）に統一される方向なので，これからは I/O 制御部についても CMP プラットフォームで共通化できる．将来，メインフレーム環境で Commodity I/O のサポートが主流となれば，オープンシステムとの SAN（Storage Area Network）環境での共存や，ストレージのコンソリデーションもスムーズに行えるようになる．

従って今後はそれぞれ OS 2200 と MCP と互換性のあるプロセッサを開発すれば，後継メインフレーム商品を提供できることになる．CMP プラットフォーム自身の Bus データ幅の拡張や Bus Clock の高速化などで能力向上が図られ，またメインフレーム専用プロセッサ自身の能力向上と機能拡張により，さらにフレキシブルなマルチ OS 環境のシステムが構築できるようになる．

その 2 : CS 7100/7200 メインフレーム・エミュレーション・システム

5. CS 7101/7201 システムの概要とハードウェア構成

ClearPath Plus サーバ CS 7100/7200 シリーズは, A/NX シリーズのハードウェア・エミュレーション技術を CMP プラットフォームで実現した，メインフレームシステムである．CMP プラットフォームにて構成された，プロセッサ部，メモリ部，I/O 部を含むシステム全体は，パーティションと呼ばれる論理的な領域に分割することができる．それぞれオペレーティング・システムとして，Windows 2000 Advanced Server が搭載されている．

本稿その 2 では，MCP 環境で稼働する CS 7101/7201 システムのハードウェア・エミュレーション技術を紹介する．標準構成では二つのパーティションが提供され，パーティションの一つは，Windows OS の 1 アプリケーションとして，常時エミュレーション・ソフトウェアが起動しており，MCP アプリケーションがその上で稼働している．もう一つのパーティションでは Windows アプリケーションが稼働する．図 4 に 4 分割したパーティション構成を示す．

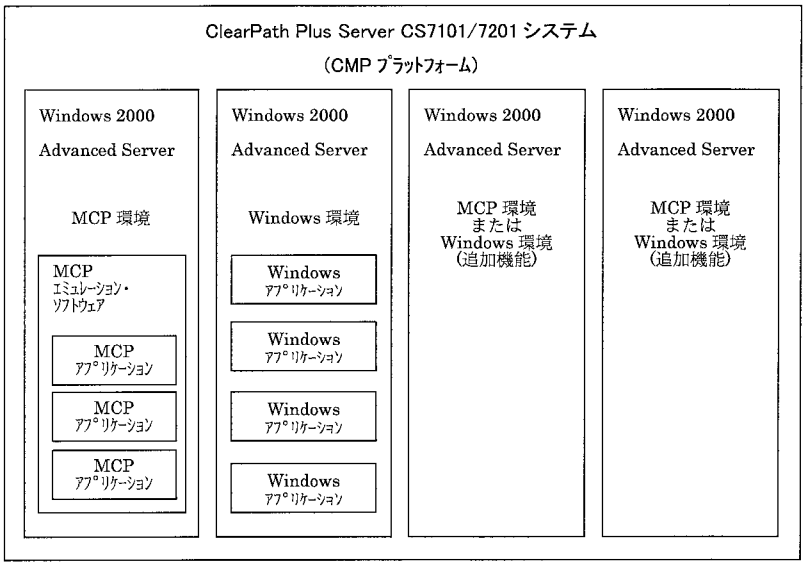

図 4 CS 7101/7201 システムのパーティション構成図

MCP 環境には一つの CPM（Central Processor Module）をエミュレーションするシングル CPM タイプと，二つの CPM をエミュレーションするマルチ CPM タイプがあり，処理能力の違いによりモデルを選択することができる．また CS 7201 システムでは，必要な時期に，必要とする処理能力に性能アップできるソフトウェア・キーの提供が計画されている．この機能は CoD（Capacity on Demand）機能と呼ばれ，MCP 環境を停止することなく，一時的または恒久的に処理能力を向上させることができ，短期の繁忙期や業務の増加によるシステム能力増強の必要性に対応できる．CMP プラットフォームのハードウェアは拡張性に優れているが，加えて CoD 機能は

MCP 環境のパフォーマンス変更要求に対応する観点でも拡張性に優れているといえる．

6. ハードウェア・エミュレーション技術の変遷

CS 7101/7201 システムは，Windows 上で NX シリーズのハードウェアをエミュレーションすることにより MCP 環境を実現している．このハードウェア・エミュレーション技術は 1989 年に発表された．当時の超小型メインフレームである，マイクロ A で初めて部分的に採用され，十数年の歳月を経て現在の CS 7101/7201 システムで集大成したといえる．

マイクロ A は中央処理装置を SCAMP（Single Chip A Series Mainframe Processor）と呼ばれる一つの VLSI チップ（大規模集積回路）で実現し，これまで A シリーズ固有のハードウェアで構成されていた I/O プロセッサである MLIP（Message Level Interface Processor）と I/O 制御装置である DLP（Data Link Processors）を OS/2 オペレーティング・システム上でエミュレーションした．これら I/O プロセッサと I/O 制御装置のエミュレーションは，メインフレーム OS である MCP から入出力の要求を受け，OS/2 がサポートしている入出力装置へのアクセスを可能にした．1993 年に発表された A 7 シリーズでは，機能拡張された SCAMP チップによる MCP 環境の性能向上を図るとともに，I/O プロセッサをエミュレーションする技術が UNIX 上で受け継がれた．1996 年には中央処理装置と主記憶装置もエミュレーション化に成功し，固有ハードウェアに依存することなく，Windows サーバ上に A シリーズアーキテクチャを実現した A 2100 シリーズが発表された．その後，Windows アプリケーションとシームレスに統合した HMP　NX 4200 シリーズへと進化し，1998 年には，汎用大型機 NX 5000/6000 シリーズで採用されている CPM と I/OM（Input/Output Module）のエミュレーションを実現し，オペレーティング・システムも汎用大型機と共通化された．この結果，現在は CS 7101/CS 7201 システムを含めて LX 5000 シリーズから最上位機種 NX 6800 シリーズまで同一 OS 上で稼働し，アプリケーション・プログラムはマシン・コード（機械語）レベルでの完全互換が保証されている．

6 章では現在のエミュレーション・ソフトウエアの構成の概要を述べ，その構成要素から MCP メモリ・エミュレーションと MCP 論理ディスクを例にとり，仕組みの一端を紹介する．

7. MCP 環境を実現するエミュレーション・ソフトウェア群

エミュレーション・ソフトウェアは，エミュレーション・システムでない NX シリーズのハードウェアと同じ働きをする複数のプロセスで構成されており，MCP コネクションサービスと呼ぶ Windows のサービスとして稼働する．エミュレーションは図 5 に示すとおり，いくつかのハードウェア・エミュレーション・モジュールから構成されている．CS 7101/7201 システムのハードウェア・エミュレーションでは，NX シリーズのハードウェアのモジュールやユニット単位に，全く同じ働きをする Windows 上のプロセスとして構成し，Windows メモリ間の高速データ転送を可能にしている．MCP や MCP 環境の利用者プログラムはハードウェアがエミュレーションさ

れていることを意識することなく利用することができ，NX シリーズとの互換性を保っている．

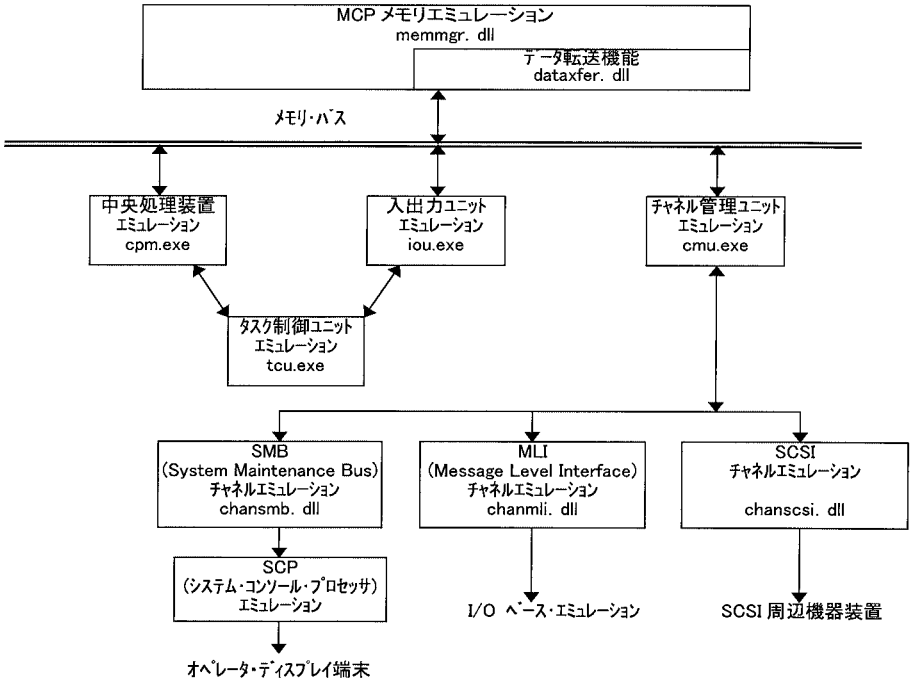

図 5　CS 7101/7201 エミュレーション・ソフトウェア構成図

以下に MCP エミュレーションのモジュール技術の例として MCP メモリと MCP 論理ディスクについて記述する．

7.1 MCP メモリエミュレーション

MCP 環境の Windows 2000 Advanced Server には 2 GB の物理メモリが用意されているが，CPM をエミュレーションするプロセスである CPM.EXE は Windows サーバのメモリ空間の一部である 1 GB を使用する．MCP メモリは 48 ビットを 1 ワード（語）として扱い，これにタグと呼ばれる 4 ビットが付加され 52 ビットで構成されている．Windows 2000 は 32 ビットを 1 ブロックとして扱うため，MCP メモリの 1 ワード（48 ビット＋4 ビット・タグ）は 2 ブロック（64 ビット）を使ってエミュレーションする（図 6）．

このメモリ・エミュレーションにより，Windows から CPM.EXE プログラムに与えられた 1 GB のメモリ領域のうち 75%（48/64 ビット）すなわち 768 MB を MCP 環境として使用することができる．この方式は，Windows メモリブロック内に若干の未使用ビットが存在することになるが，MCP オペレーティング・システムが 1 ワードを 48 ビットとして扱うことを可能にしており，メモリ管理においても NX シリーズとの完全互換性を維持している．

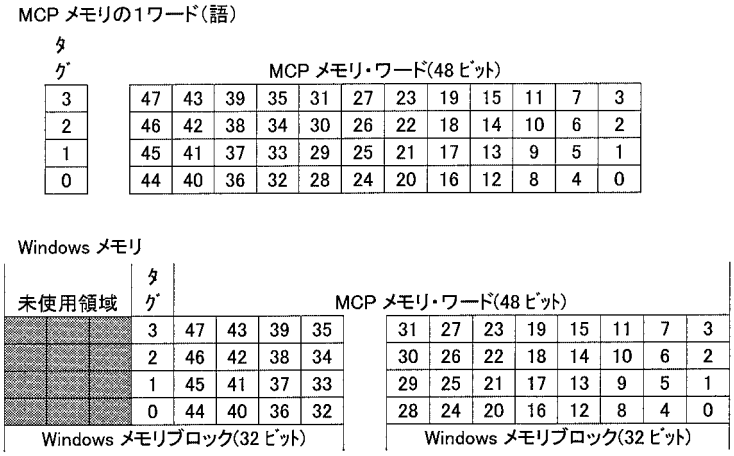

図 6　MCP および Windows メモリの構成

7.2　MCP 論理ディスク

MCP 環境のディスク・サブシステムへの物理的な入出力処理は，ディスク・ドライブ上のセクタ単位におこなわれており，1 セクタは 180 バイトである．しかし，Windows で使用するディスク・ドライブは業界標準の 1 セクタあたり 512 バイトで構成されている．CS 7101/7201 システムでは MCP 環境と Windows 環境の二つのオペレーティング・システムが稼働できるので，どちらからも入出力を可能にする必要がある．そこで Windows で制御可能なファイルシステム（NTFS）でフォーマットされたドライブ上に「diskxxxx.asd」と名付けられたファイルを作成し，MCP 環境のディスク・サブシステムとして論理的に扱う方法を考え，これを『MCP 論理ディスク』としている．

図 7 に Windows ディスク内に格納する MCP 論理ディスク配置例を示す．

MCP 論理ディスクは，ドライブの空領域に 419 MB から 48 GB まで，1 MB 単位に任意の大きさで作成することができる．この方式は，必要な容量を必要な数だけ MCP 環境のディスク・サブシステムとして構築することができ，ドライブに空領域がある限り，ハードウェアを増設することなしに容易に MCP 論理ディスクを追加することができる．

この技術の最大の利点は，オペレーティング・システムである MCP の物理的な入出力処理を変更することなく，また多種多様なディスク装置のハードウェア特性などを意識することなく，対応できるという点である．今後は MCP 論理ディスクの領域を拡張できるリサイズ（Resize）と呼ばれる機能が計画されており，ディスク・サブシステムの構築において拡張性・柔軟性がさらに高まる．

8.　ハードウェア・エミュレーション化の優位性

長い歳月を経て開発を積重ね，発展してきたハードウェア・エミュレーション技術は，プラットフォームの技術進歩により，現時点で MCP 環境の処理能力として，汎用 NX シリーズシステムの中型機の領域までもカバーするに至っている．今後は Intel

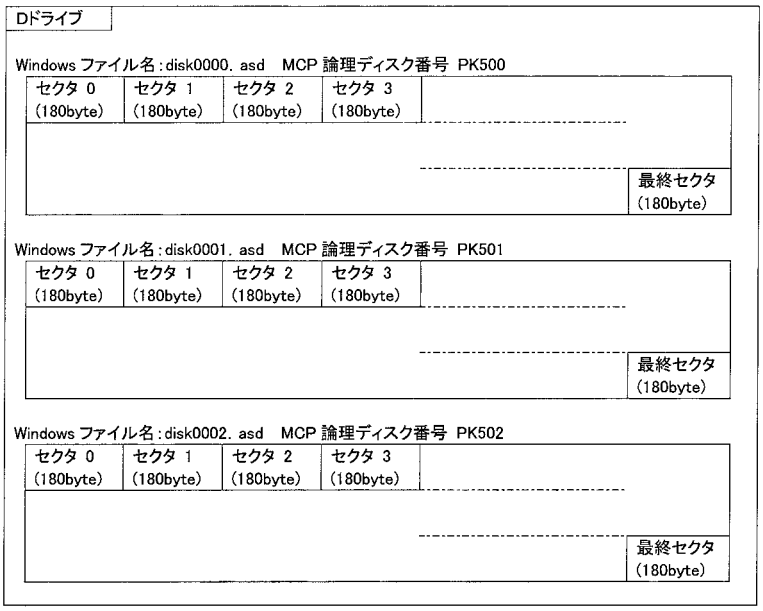

図 7　MCP 論理ディスク配置例

プロセッサ自身の高速化と CMP プラットフォームの技術進歩により Windows の処理能力が更に増していく．エミュレーション・ソフトウェアにより Windows 上で稼働する MCP 環境はこれらの技術進歩に伴い，飛躍的な性能向上が見込まれている．これはエミュレーション・マシンの優位性であり，ハードウェア・エミュレーション化の狙いでもある．

また，先に述べてきたように，CS 7101/7201 システムで実現したハードウェア・エミュレーション技術は，NX シリーズの固有のハードウェアを Windows 上で個々のプロセスとして構成することにより，MCP オペレーティング・システムの変更を必要とせず，NX シリーズとの互換性を維持し，保守性と信頼性をも向上させている．MCP オペレーティング・システム上で稼働するアプリケーション・プログラムについても，ハードウェアがエミュレーションされていることを一切意識することなく，既存のシステム資産を継承することを可能としている．

9. おわりに

「CMP アーキテクチャのもとで実現する CS 7000 シリーズの基礎技術：その 1」では，CMP プラットフォームを使用して，異なったメインフレーム OS 環境を，それぞれの専用プロセッサを開発することにより実現した技術を紹介した．「その 2」では，CMP プラットフォームとハードウェア・エミュレーション技術を採用することで，メインフレーム OS 環境と Windows 環境の共存させる技術を紹介してきた．将来はハードウェア・エミュレーション技術により，OS 2200 環境と Windows 環境の共存も実現する計画もある．基幹システムを稼働させながら，敏速に Windows アプリケーションと連携することが容易にできるようになる．CMP プラットフォームの多彩なパーティショニング機能を活かし，メインフレームの現在までの情報資産を継

承しつつ，新しい情報資産を形成することが可能となる．これらの面で CS 7000 シリーズはメインフレーム OS 環境を含めたサーバ統合をいち早く実現できるプロダクトである．

参考文献

（その 1）［1］ 小川康博, CMP アーキテクチャに基づくエンタープライズサーバ, ユニシス技報, 通巻 66 Vol.20 No.2, 2000 年 8 月.
［2］ Voyager New Product Information Transfer, Unisys Corporation, 06/2001.
［3］ Robert Malek, Voyager Platform Specification, Unisys Corporation 06/2001.
（その 2）［4］ CS 7101 Capabilities Overview, Unisys Corporation, 04/2001.
［5］ LX 5000 New Product Information Transfer, Unisys Corporation, 05/1998.
［6］ 室木　通, 坂本　徳明, HMP　NX シリーズ実現の基礎技術, ユニシス技報,通巻 56, 1998 年 2 月

執筆者紹介

櫻　井　成　一（Seiichi Sakurai）
（その 1）
1987 年東海大学工学部電子工学科卒業．同年日本ユニシス(株)入社．2200 シリーズのハードウェア保守サービスを経験後に 2200 シリーズのフィールドサポート，プロダクトサポートに従事し，XPC,CS 7802 の開発に参加．現在，ユニアデックス(株)CMP プロダクト部に所属．

齋　藤　雅　樹（Masaki Saito）
（その 2）
1985 年法政大学工学部経営工学科卒業．同年日本ユニシス(株)入社．A/NX シリーズのハードウェア保守に従事．その後，A/NX シリーズのフィールドサポートを経て，プロダクトサポートに従事し，LX/CS シリーズの受入, 保守およびコンソールソフトウェアの日本語化を担当．現在，ユニアデックス(株)　CMP プロダクト部に所属．