4P－9

転送サイクルのパイプライン化による バスのデータ効率向上の一手法

近藤 伸和*　岡澤 宏一*　梅村 雅也*　源馬 英明**

(株) 日立製作所システム開発研究所*　(株) 日立製作所オフィスシステム事業部**

1. はじめに

近年、RISC等のプロセッサの急速な動作周波数向上に伴い、サーバを始めとするハイエンド機では、プロセッサ及び主記憶へのデータ供給能力をプロセッサに比例して向上できるか否かがシステム性能を決定する最大要因の一つになりつつある。本稿では、その中核をなすバス、特にシステムバスに注目し、アーキテクチャ及びプロトコルの両面から、システムトータルの性能向上の手法について述べる。

2. サーバ機のバスアーキテクチャ

2.1 バスの3階層化

サーバ機等では、既存のI/O資源活用の観点から、標準I/Oバスとの接続が必須条件である。

一方、ディスクアレイや高速LAN（FDDI等）など高スループットを要求されるI/Oを接続する場合、高速I/Oをシステムバスに直結し、低速I/Oを下位のI/Oバスに接続することで、装置全体のバランスを取ることも重要になってくる。

これらの理由から、プロセッサバス、高速システムバス、複数の標準I/Oバスによる3階層化が、一般的なサーバ機のバスアーキテクチャとなりつつある（図1）。

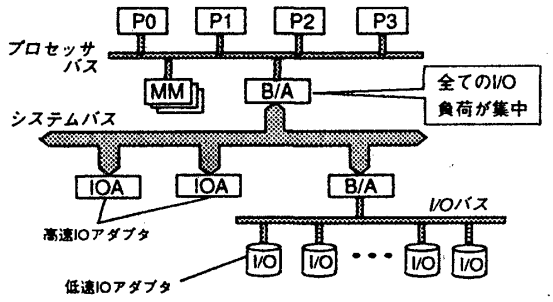

図1.「バスの3階層化」

2.2 システムバスの位置付け

上記バスアーキテクチャにおけるシステムバスは、「全てのI/Oからの負荷を、プロセッサ及び主記憶インタフェース部で束ねる高速インタフェース」という位置付けである。従って、高スループットと同時に高応答性も要求される。

Improvement in data efficiency of the systembus by using pipeline method:

Nobukazu KONDO, Koichi OKAZAWA, Masaya UMEMURA, Hideaki GEMMA　Hitachi, Ltd.

2.3 動作周波数向上のアプローチ

バスの動作周波数向上を容易にするには、全てのバス信号のネット形態を、ラッチ・ツー・ラッチとするのが最も有効である。すなわち転送データを、入出力バッファの直前直後のFFで必ず一旦ラッチすることとし、デコード、エラーチェック、変換等の処理を、転送の前後のサイクルで行う方式である。これは、バス上の伝搬遅延以外の時間をサイクルタイムに反映させないためである（図2）。

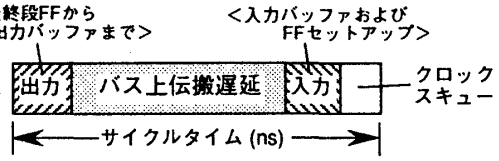

図2.「バスサイクルタイムの内訳」

このような年々向上するバスの動作周波数を最大限に活かすには、データ効率の高いプロトコルが不可欠であり、以下、そのいくつかのアプローチについて述べる。

3. 転送データ効率の向上手法

3.1 転送サイクルのパイプライン化

一般に、バス上の転送は、データが正しく転送できたか否かをアクノリッジ系信号を用いて確認し、そのハンドシェイクによりバスサイクルを終了させる。しかし、ハンドシェイクプロトコルでは、エラーの検出や報告信号の伝搬時間までが転送サイクルに含まれるため、個々の転送時間を短縮できないという問題が生じる。

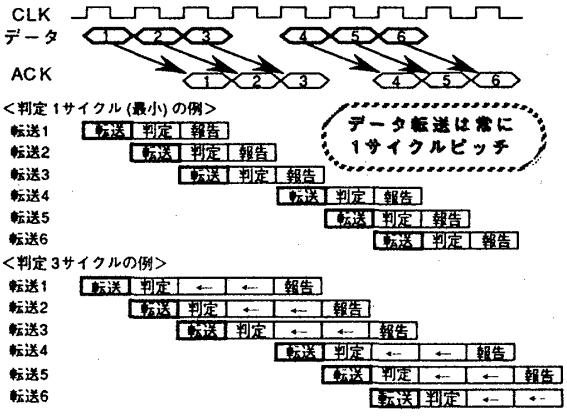

図3.「パイプラインアクノリッジ方式」

そこで、バスの転送サイクルを、「データの伝搬」、「各種判定」、「アクノリッジ報告」の3フェーズに分割し、パイプライン化を図る方式を提案する（図3）。ここでマスタは、データの伝搬に必要な時間でアクノリッジ報告を待たずにバスサイクルを打ち切る（バス権を放棄）ことで、次の転送開始を可能としている。一方、転送とアクノリッジ間の対応の判別は、報告までのディレイ時間を統一することで行う。

3.2 論理ID応答スプリット転送

ハイエンド機等のマルチプロセッサシステムにおいては、個々のプロセッサから独立にPIOアクセス要求が発生する。この場合、一般に、プロセッサバスとシステムバスのインタフェース部に全てのアクセス負荷が集中することになる。従って、システム性能低下を防ぐには、一つのプロセッサがPIOアクセスを起動している間でも、他のプロセッサからのPIOアクセスを並行して起動できる仕掛けが必要となる。そこで、各モジュールに複数の論理的なIDを割り当てる方式を提案する。これにより、マルチシステムにおける複数のPIOアクセスの並列処理化が実現できる（図4）。

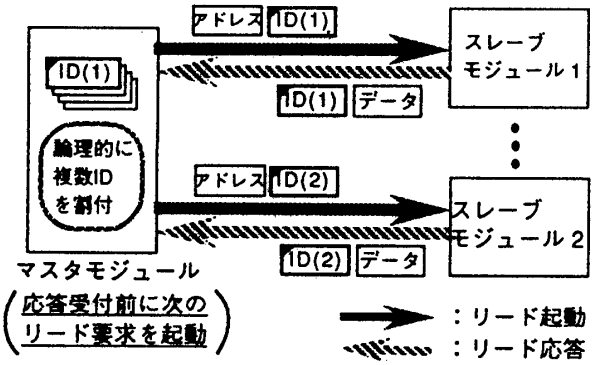

図4.「論理ID応答スプリット転送」

3.3 ビジーリトライプロトコル

近年の高速バスは、スレーブ側のモジュールにおいて、転送データ（アドレス等を含む）用のバッファを十分に用意していることが前提となっている。そのため同一モジュールに対して連続アクセスが発生した場合、転送先モジュールのバッファが満杯になることで、転送受付不可状態に陥る可能性がある。この場合、バスを占有したままマスタが待たされるとバスの使用効率が著しく低下する。そこで、マスタがバッファビジー状態を検知した時点で、一端バス権を放棄し、一定時間後にリトライできるプロトコルを設ける。これにより、同種類のトランザクションのサイクル長を、スレーブ側の転送受付状態にかかわらず一定に保つことができる。

3.4 スプリットバス階層接続方式

サーバ機においては、バスの階層化が必須となるが、その場合のバスのスプリット化の効果について考える。システムの転送経路中に一つでも非スプリットバスが存在する場合、非スプリットバス部分で転送がシリアライズされ、全体の転送速度が決定されてしまう。そこで、全ての階層のバスをスプリットにする方式を提案する（図5）。これにより、プロセッサ及び主記憶装置からI/Oに至るまでの一貫した経路において、完全なパイプライン処理が行なえるため、従来比約2倍以上の転送性能が得られる。

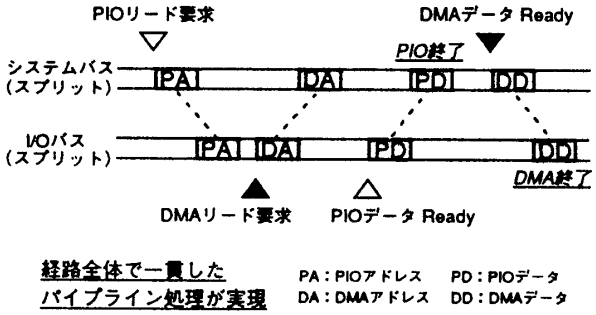

図5.「スプリットバス階層接続の効果」

4. おわりに

サーバ機に適したバスプロトコルおよびバスアーキテクチャを検討した。求められる要件は以下の通りである。

(1) 高速システムバスプロトコル
- (a) バースト転送が基本の同期式プロトコル
- (b)「パイプラインアクノリッジ方式」による高スループットと高信頼性の両立
- (c)「ビジーリトライプロトコル」のサポートによる、同一トランザクションサイクル長の一定化（最大サービス待ち時間の短縮）

(2) 階層型向けバスアーキテクチャ
- (a)「論理ID応答スプリット転送」を階層バスに適用し、マルチプロセッサ対応時の複数プロセッサからのPIOアクセスを並列処理
- (b) スプリットバス同士を階層的に接続し、プロセッサからI/Oまでの転送を一貫したパイプライン処理で実現

参考文献

[1] ISO/IEC DIS 10857.2: Information technology-Microprocessor systems- Futurebus+ -Logical protocol specification (1992)