

パーソナルロボット用ハードウェアアーキテクチャASPIREへの 7B-3 RISCプロセッサの応用*

澤村 省治 山崎 信行 安西 祐一郎†

慶應義塾大学‡

1 はじめに

将来、現在のパーソナルコンピュータのようにロボットが利用されることが想定される。このような「パーソナルロボット」は、パーソナルコンピュータに対するものとは異なった要求を満たす必要がある[anzai92]。そこで、それらの要求を満たすアーキテクチャとしてASPIREアーキテクチャが提案されている。

ここで、人間、ロボット、コンピュータ間で複雑なインタラクションを実行することは計算負荷を大きく増大させる。つまり、ASPIREアーキテクチャ上で複雑なアプリケーションを実行するには、高速な処理系が不可欠である。

本研究では、ASPIREアーキテクチャ上にRISCプロセッサの一つである μ SPARCを実装することによって、その速度を以前の実装と比較する。

2 背景

自律ロボットは、センサからの情報をロボットが処理をし、必要な行動を算出してから行動するため、敏捷な行動を取ることが難しい。しかし、実際にロボットが受け入れられるためには、敏捷性、緊急処理能力、リアルタイム性、ロバスト性をもつ必要がある。

そのためには、ロボットに搭載されている処理装置(PU)が十分にニーズに見合った処理速度を持つ必要がある。現在、高速な処理装置はRISC型のCPUを用いて設計されることが多い。しかし、組み込み用途に用いる場合には問題が多かった。その問題は、

- 分岐、割り込み時のパイプラインの処理
- 割り込み時のレジスタの退避

である。RISC型のPUは、パイプラインとキャッシュで命令やデータを高速に処理することによってその速度を得ている。しかし、割り込み駆動を用いることの多い組み込み用途では、パイプラインが寸断されたり、多くのレジスタをメモリに退避するため、その高速性を発揮できなかった。後述のように、割り込みに対しても高速性を維持しようとするPUに μ SPARCがある。本研究では、その μ SPARCを採用し、割り込みを多用しながらも高速化を試みた。

*Application of RISC processor for personal robot hardware architecture ASPIRE

†Shoji Sawamura, Nobuyuki Yamasaki and Yuichiro Anzai

‡Keio University

2.1 ASPIREアーキテクチャ

ロボットに必要な敏捷性、緊急処理能力、リアルタイム性、ロバスト性を与えるためのアーキテクチャとしてASPIREが提案されている。ASPIREでは、パーソナルロボットの各構成要素を機能別のモジュールに分割する。各モジュール間は、コミュニケーションメモリと割り込み線によって接続される。各モジュールはその割り込み線を用いて他のモジュールにイベントを伝達することが出来る。割り込みをイベントの伝達に用いることによって高速に状況の変化に対応することが可能となり、敏捷性を得ることが出来る。また、各モジュールは、独立した処理系をもっており、各モジュールは抽象的な命令を受けとり、各々が独立に行動することができる。さらに、緊急時には中央の処理装置を介することなく対応することが可能となっている。また、各モジュールは独立に動作するため、モジュールが異常になっても、システム全体のダウンは避けられ、ロバスト性を得ることが出来る。

このようにASPIREは並列に動作するが、ロボット全体のプランニングを行ったり、総括するモジュールが必要である。そのモジュールは、他のモジュールから情報を受けとり、ロボット全体に共通な情報として蓄積、供給するような能力を持つ必要がある。そのような能力を持ったモジュールが存在し、各モジュールがその情報を用いることによって、ロボット全体の行動を決定し、コンシステンシを維持し、全体を統括することが可能となる。

現在、ASPIREはVMEバス上に実装されている。また、メインボードとして、MC68030をCPUとしたボードを用いているが、このボードでは、大規模なプランニングなどを実装する時に必要な処理能力を提供できない。そのため、より処理能力の大きなメインボードを実装する必要がある。

2.2 μ -PULSER

ASPIREアーキテクチャ上には、マルチタスク・マルチスレッド・リアクティブOSである、 μ -PULSERが実装されている[yakoh94]。

コンテキストスイッチの際にはコンテキストを保存する必要があるが、コンテキストをメモリに退避せずに行なうことが出来れば、高速にコンテキストスイッチを行なうことが可能である。このことから、数個のコンテキストスイッチの際には、コンテキストをメモ

りに退避する必要のないアーキテクチャの CPU を選択することが望ましい。

さらに、他のモジュールからの信号は割り込みを用いて伝達されるため、割り込み処理が高速でなければならない。割り込み処理の際には必ずレジスタを退避する必要があるが、RISC 型の処理装置は多数のレジスタを用いるために、多くのメモリアクセスを生じ、それが割り込み処理を遅くする要因となっていた。不可避なレジスタの退避をメモリに行なわなければ、高速に割り込み処理を行なうことが出来ると考えられる。

以上の点から、本研究では μ SPARC を実装する。 μ SPARC は、8 組のレジスタウィンドウを持ち、8 組以下のコンテキストスイッチや割り込み処理をレジスタをメモリに退避せずに高速に実行できる。つまり、スタック操作のような大きなオーバーヘッドを負うことなくコンテキストスイッチや割り込み処理を行なうことが出来る。

3 設計

3.1 μ SPARC と SBUS

μ SPARC は図 1 に示されるような外部バスを有する。つまり、全ての外部デバイスは SBUS 規格に従って接続されることが求められる。 μ SPARC では、最大 5 枚の SBUS モジュールをサポートできるが、実際には 1 枚は占有されるために 4 枚までしかサポートできない。VME とのインターフェースは SBUS モジュールを 1 つ占有する。

以下に SBUS 規格の重要な点をあげる。

- 割り込み線はモジュール間共有で 8 本
- 同期式バス
- 32 ビットデータバス

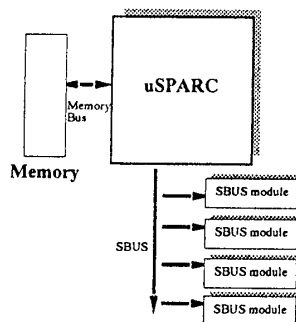


図 1: μ SPARC

3.2 ASPIRE

ASPIRE は通信手段として、

- 割り込みによるイベントの伝達
- コミュニケーションメモリによる通信

を持つ。割り込みによるイベントの伝達をサポートするためには、モジュール間、またはモジュール内部

で、十分な数の割り込み線が使えるようになっていないなければならない。現在の ASPIRE は VME バス規格上に実装されている。そのため、メインボードは VME バスからの割り込み線を受け付けなければならない。

また、コミュニケーションメモリを用いてモジュール間の通信を行なうために、各モジュールには分散共有メモリを搭載する。しかし、他のモジュールが μ SPARC のバス (SBUS) を占有することを防ぐために、VME バスから SBUS へのアクセスパスは設けない。しかし、メインボードは他のモジュールの分散共有メモリにアクセスする必要があるため、SBUS から VME バスへのアクセスパスは設ける必要がある。

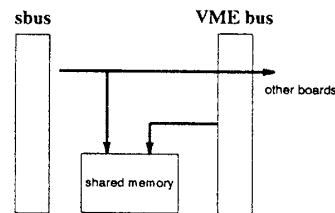


図 2: SBUS と VME バスの接続

4 実装及び評価

6U サイズの VME ボード上には全ての回路を搭載することは難しいため、 μ SPARC 及び周辺回路を独立したワンボードコンピュータとし、SBUS \leftrightarrow VME バスのインターフェースを 6U サイズの VME モジュールとして実装中である。

大規模なプランニングなどに対応できるように、 μ SPARC は最大 32MB のメモリを搭載できる。共有メモリは、512KB 搭載する。また、直接外部へアクセスするためのインターフェースとして、SCSI, EtherNet インターフェース、RS-232C 通信ポートを 2 つ搭載している。

5 おわりに

本研究では、パーソナルロボット用のアーキテクチャ ASPIRE におけるメインボードの処理能力を向上させる設計を行ない、それを実装中である。これによって、より複雑なアプリケーションにも耐えられるようになると思われる。

参考文献

- [anzai92] Y. Anzai. Towards a new paradigm of human-robot-computer interaction. Proc. of IEEE Int'l Workshop on Robot and Human Communication, pages 11-17, 1992.
- [yakoh94] 矢向高弘, 菅原 智義, 安西 祐一郎. μ -pulsar: パーソナルロボットを構築するためのオペレーティングシステム. 電子情報通信学会論文誌, Feb 1994.