

## 5H-9 新ELIS用LSIのリアルタイムAIエンジンへの応用

渡辺和文

NTT ヒューマンインタフェース研究所

## 1 はじめに

新ELISでは、AIの実応用を指向し、リアルタイムAIを実現しているが[1]。これを様々な分野に応用するためには、開発したAI処理プログラムを組み込むためのエンジン(実行機、ボードなど)、さらにこれらのための一貫した開発環境が必要である。本報告では、新ELIS用LSIを用いてモジュール性の高いAIエンジンボード設計できること、およびその開発環境について述べる。

## 2 AIエンジンの構成

## 2.1 LSIの特徴と汎用性

図1に構成例を示す。LISPエンジン部分はワークステーションとほぼ同一の構成となる。CPU-LSIに制御記憶(RAMまたはROM)とスタックの2つの専用メモリ、さらに周辺LSIを接続する[2]。周辺LSIには主記憶用DRAMチップアレイの他、インタフェース論理(外付け)を介して組み込み先システム(以下MAINと呼ぶ)へ(例えばバス接続で)接続する。

新ELISでは汎用性を重視し、組み込み先として幅広いシステムに対応できるように周辺論理LSIに種々の機能を搭載し、モジュール性の高いエンジンボードの実現が可能である[3]。それらの機能を次に述べる。

## (1) I/O インタフェース

CPU-LSI内部の制御/状態レジスタおよびDMA制御レジスタをMAINのI/O空間へ配置し、MAINからこれらのレジスタを読み書きできるようにするインタフェースである。特定のI/Oバスを対象とした専用のバスインタフェース制御論理はチップ内部に組み込んでいない。したがって外部に簡単な回路を組むだけで汎用的なバス(例えばMulti-Bus, VME-Busバス, NuBus, PC98バスなど)に対してAIエンジンを接続することができる。

## (2) メモリ制御

DRAMチップで構成されたメモリアレイのリフレッシュ、およびメモリアクセスに対する制御を行う。1ビットエラー訂正、2ビットエラーの検出を行い、メモリの信頼性を向上させている。

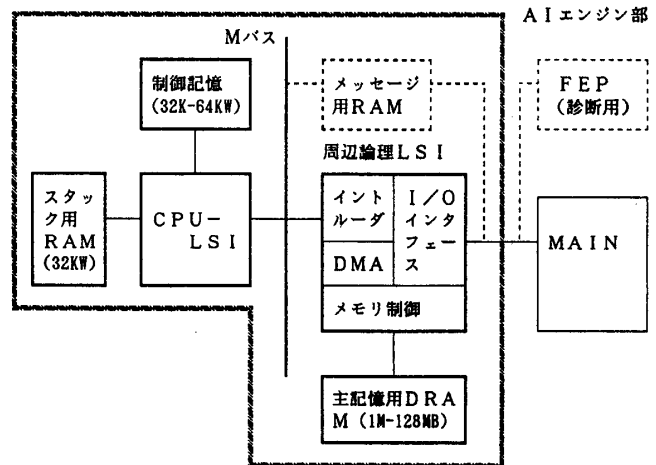


図1 AIエンジンの構成例

## (3) ダイレクトメモリアクセス(DMA)

DMAは自律型(コマンドの受理によってコントローラが自動的に転送を開始、終了する)で、エンジン内主メモリとMAIN側のメモリとの間の転送をサイクルスチールモードで直接行う。転送データの並びは、MAIN側システムがインテル系のプロセッサとモトローラ系のプロセッサではバイトの並びの上下の順序関係が逆であるので、どちらのプロセッサでもMAINとして用いることができるようにバイトスワップの機能を設けている。さらにMAIN側のバス幅が16ビットと32ビットの両方に対応できるようにしている。

## (4) イントルーダー

CPU側からMAIN側のメモリを直接アクセスできる機能である。これをイントルーダーと呼ぶ。イントルーダーは、CPUからの特定アドレスに対するアクセス命令で動作する。マッピングレジスタにより、MAINメモリ空間の全てに対してアクセスできる。

## (5) ユーザデバイス接続用外部バス(Mバス)

CPU-LSIと周辺論理LSIの間では、多重化Mバス(Multiplexed Bus-32ビット幅)を用いて64ビット幅のメモリデータ転送などを行うが、このバスにユーザデバイスを接続してCPU-LSIから直接アクセスすることができる。

## 2.2 メッセージ通信領域の付加

AIエンジンを組み込み、処理対象データの転送と処理内容を指示し、システムと協調させて動作させる場合には、一般的によく使われるメッセージ通信

を利用する。ELIS ボードは相手先からの要求に基づいて処理を行い、その結果を返す。メッセージ領域を設ける方法はELISの場合以下のように2通り考えられる。ひとつはイントルダを用いた共有メモリ方式である。これは、MAIN側のメモリの一部をELISと共有し、ELIS側からはイントルダを用いてアクセスする。専用のハードウェアを設ける必要はないが、MAIN側にそのためのメモリ領域を確保しなければならない。もうひとつは専用のRAMを用意し、MバスおよびMAIN側の両方からアクセス可能な領域を設ける方法である。組み込み先システムのメモリ構成に依存しないため、モジュール性を高めるという観点からは有利である。ひとつの具体例ではこの方法を取っている[4]。

### 2.3 割り込みの生成

CPUはMAINから7レベルまでの割り込みを受けられる。割り込みはマイクロコードで検出する。ELISからMAINへの割り込みには、Mバス上で割り込み発生回路が必要である。

## 3 開発環境

### 3.1 ICEカードによるワークステーションへの接続

ELISでは、CPU内部に診断機能を備えており、ワークステーション版ではこの機能を利用し、フロントエンドプロセッサ(FEP)側の支援プログラムによりシンボリックにかつ会話的に、CPU内部の状態、レジスタの値、各メモリの内容の読みだし/更新、またマイクロプログラムの実行制御が行える。新たにボードの開発を行う場合でも、ワークステーションのFEPを用いて同じ環境を使いたい。このために専用のICE(In Circuit Emulator)カードを作成した。このICEカードは周辺論理LSIを搭載しており、ボードとワークステーション内ELIS-CPUボードの各周辺論理LSIのソケットの間をフラットケーブルで接続する。接続の関係を図1に示す。I/OインタフェースがMAINの他FEPとも接続されFEPから開発ボードの診断機能を利用することができる。ボードでは使われないコンソールI/O等のための制御線はFEP側へ接続できるので、ボードにシステムソフトウェアを載せればあたかもボードをワークステーションのCPUとして動作させることができる。DMAについては、相手先としてMAINとFEPの切り替えができる。

### 3.2 デバグの進め方

ハードウェアデバグは、FEPを用いてマイクロ命令を逐一実行させて回路を動作させその結果を確認しながら進める。また新たに組み込むマイクロプログラム(例えばメッセージ通信用関数など)のデバグも同じよ

うに、マイクロプログラムのステップ実行と内部状態を観測しながら効率よく進めることができる。ボードの動作が安定した段階でシステムを載せれば、LISPのトップレベルを利用することができる。MAIN上でプログラムを走らせELISとの協調動作の試験を、LISPの実行環境の下でコンソールから行える。つまりデバグを利用した応用ソフトの実機でのデバグが行える。

### 3.3 応用プログラムのエンジンボードへの移植

エンジンボードに組み込む応用ソフトウェアは、ワークステーションを用いて開発する。ハードウェアは、実装規模に応じて制御記憶は32K~64K語、また主記憶は1M~128MBまで容量を変えることが可能である。この場合、限定されたハードウェア資源で動作するソフトウェアを作成する必要がある。このためワークステーションでは、決められた範囲内で適切なサイズのマイクロプログラムファイルおよび主記憶イメージファイルを生成できるよう、ソフトウェア構成を関数単位にモジュール化し、必要なモジュールのみを切り出すことができるようにしている。このシステムはCommonLISPのサブセットとなり、この環境の下でのプログラム開発が可能である。

作成した応用プログラムはシステムプログラムとともに、主記憶イメージファイルに格納する。これとマイクロプログラムファイルをMAINへ持っていき、マイクロプログラムは制御記憶へ書き込み、イメージファイルはDMAを起動して主記憶にロードする。立ち上げ後応用プログラムの実行となる。

## 4 おわりに

新ELIS用LSIによりモジュール性の高いAIエンジンボードが容易に実現できる。またICEカードとワークステーションを組み合わせることで一貫した開発環境が構築でき、ハードウェアからソフトウェアまで効率良く開発を進めることができる。最後に日頃御指導いただき日比野靖グループリーダーに感謝いたします。

## 参考文献

- [1] 鈴木、菅原、杉村、"リアルタイムAI用OSカーネル-新ELIS用OSの開発-", NTTR&D, No.9, Vol.39, pp.1327-1336, (1990).
- [2] 渡辺、川村、日比野、"新ELISのCPU-LSIの開発", 1989年信学会秋季全国大会論文集, 5-117 (1989).
- [3] 渡辺、川村、日比野、"AIエンジン用LSIの開発-新ELISのハードウェアについて-", NTTR&D, No.9, Vol.39, pp.1337-1344, (1990).
- [4] 渋谷、小平、鈴木、"機器組み込み用ELIS-VMEボードのハードウェア", 第42回情報処学会全国大会論文集 (1991).