

364

ALPS / I のバルクメモリ・アクセス機構と LISP インタプリタ

井田昌之, 森芳喜, 間野浩太郎

(青山学院大学 理工学部 経営工学科)

1. バルクメモリアンタフェイス

ALPSシステムの構成に不可欠なバルクメモリ(64kw, 1w=36 bit + 1パリティ)は仮想メモリではなく, DMAチャンネルを介して接続されている。

このインタフェイスは, RAM(1μs / byte)とバイトとバルク(1μs + refresh)1語の間を約15μsで転送する。

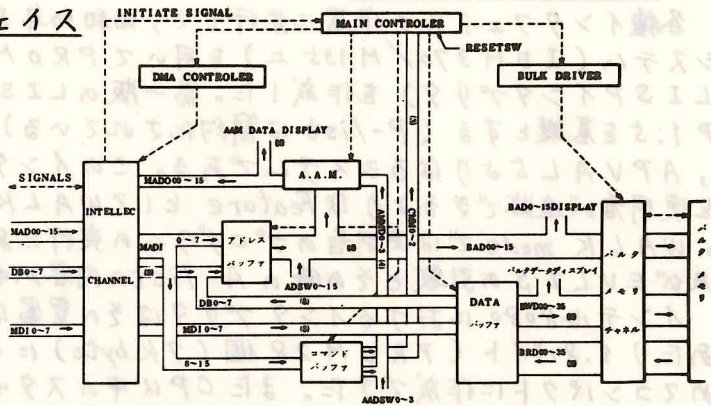


図1 メモリアンタフェイスブロックダイアグラム

1.1. データバッファ(図2)

語長変換(8 bit x 5 ⇒ 36 bit + 1パリティ)のためにインタフェイスはデータバッファを持っており, RAM→バルク時はパリティ生成, バルク→RAM時はパリティ・チェックを行なっている。パリティ・エラー時にはインタフェイス操作パネル上にエラー表示を行なう。この時CPUに対するホールド要求を解除しないのでCPUは停止し, プログラムは暴走しない。

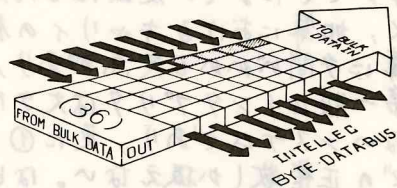


図2 INTELLEC DATA BUS V.S. BULK DATA LINE

1.2. AAM (Address Association Memory: 16個のレジスタ)

転送に必要な情報のうちRAMアドレス16bitはインタフェイス上のAAMレジスタ(16bit x 16, アクセス30ns)に記憶し, 転送時にはレジスタ番号を指定する間接方式をとる。このAAM機構の採用により, 転送に必要なアドレス情報がアドレス16bit・データ8bitで充足され, ユコンの1byte命令をバルク参照命令として代用する事が可能となった。

1.3. 転送の制御(図3)

チャンネルスタート命令がユコンの命令セットにはないので, それに相当する命令として32768番地以降に対する「MOV M, A」(Move A register to Memory)を用いる。この時, インタフェイスはMemory Write cycleで, アドレスバスの最上位ビットの1をバルク参照信号と認識し, CPUにホールド要求をかけバス上の情報をラッチする。ラッチされた23bitは次の情報に分割され, 対応する機能が実行される。

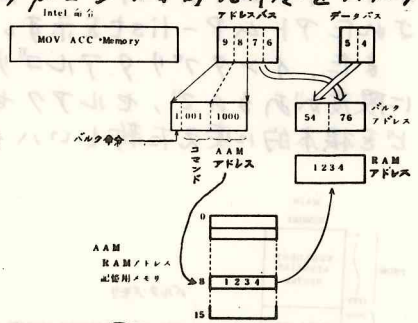


図3 メモリ転送制御の例

- ① コマンド (3 bit) : RAM⇔バルクの転送方向の指示及びSET・AAMの3種。複合コマンドは存在せず操作パネル上にエラー表示されCPUは停止する。
- ② AAMアドレス (4 bit) : インタフェイス中のAAMレジスタアドレス(0

~15)の指示。RAM側アドレスの間接指定用レジスタとなる。

③**バルクアドレス**, または SET・AAM命令の場合は指定されたAAMLリストに入れられるRAMアドレス (16bit)

2. LISPインタプリタ

各種インタフェースの作成に並行して, 昭和50年8月よりクロスソフトウェアシステム (IBM370/M135上) を用いて PROMに書き込むソフトウェア (LISPインタプリタ) を作成した。第一版のLISPインタプリタは, LISP 1.5を基礎とする (P-listは簡約化されている) SUBR 48, FSUBR 9, APVAL 5よりなるシステムである。このインタプリタには万能関数の動作を使用者が追跡できるような Feature として WALK Feature が用意されている。WALK mode では使用者のプログラムの実行にあたっては通過した EVAL 及び EVALIS の引数とその時の A-list の内容が順に印刷される。

インテル8080におけるインタプリタはその豊富な1バイト命令により1命令あたり1.8バイト (PROM 28個 (7K byte) に4Kステップ格納) と, きわめてコンパクトに作成できた。また CPU 中のスタックポインタをもとにした, PUSH・POP 命令によりスタック操作は高速になり, 簡潔なプログラムにすることができた。その反面16bit演算は加算・Increment・Decrementの3種しかなく, 加算におけるキャリーの検出しかないのど, アドレスの比較・16bit四則演算に手間がかかる欠点があり, この辺に8bitμコンの限界が感じられる。

第一版のインタプリタは大きなフリーストレージを頼みとしたがバジコレクタのないシステムであり, 更に①PRGインタプリタを持たない。②数は $2^{16}-1$ までの正整数しか扱えない。などの欠点があり, 実用的なLISPをもととする高級言語のインプリメントなどは不可能である。そのため現在, 機能の拡充, 処理速度の高速化を目標に, ①A-listの配列化, ②即時回収を含むバジコレクタ, ③ハッシュ領域の設定 (ハッシングによるアトム格納, 連想計算及び配列の確保のため) などを中心とした改訂版をデバッグしている。

メモリ構成を図4に示す。バルクメモリはRAM上の固定された16通りのバッファを通してアクセスされる。(LISPインタプリタではSET・AAM命令はインタフェースに対して一度ずつしか出されていない)。図5にコンパクト化されたアトムP-listを示す。

また, インタプリタアルゴリズムの改良とあわせて, 8ビットμコンでは改良に限度があるので, セルアクセス機構・変数束縛機構・CPUアーキテクチャなどを根本的に変えた新しいハードウェアの検討も進めている。



図4 メモリ構成

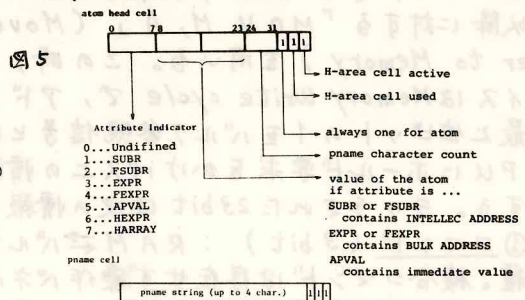


図5