

多重モジュール構成連想プロセッサ DREAM-II のハードウェア

THE HARDWARE OF DREAM-II, MULTI-MODULE ASSOCIATIVE PROCESSOR

後藤厚宏 大和 理 上森 明 元岡 達
Atsuhiko GOTO Osamu YAMATO Akira UEMORI Tohru MOTO-OKA

東京大学 工学部

Faculty of Engineering, University of TOKYO

[1] はじめに

同一の処理モジュールを数多く並べる(マルチモジュール構成)ことによって、大規模連想プロセッサを実現する方式を研究中である。^{[1][2]}

マルチモジュール構成方式には次の利点がある。

- (i) 構成単位が小規模であるため設計が容易である。
- (ii) LSI化が可能であるため安価である。
- (iii) 応用によってシステムの規模を自由に選択できる。

一方、マルチモジュール化においては次のような解決すべき問題がある。^[3]

- a) モジュールの処理機能の選択。
- b) 多数のモジュールの制御方法。
- c) モジュールに対するデータの供給方法。

本研究では、マルチモジュール構成の連想プ

ロセッサ DREAM-II とホスト計算機(汎用ミニコン)から構成されるシステムの開発を行っている。[Fig.1]

すでに処理モジュールとして、二次元アクセス記憶を有する連想モジュール(Associative Module)を開発し、モジュール試験システム(DREAM-I)によって性能評価を行った。連想モジュールは、二次元アクセス記憶、マイクロプロセッサ、および連想処理に有用な外部ロジックによって構成されている。^{[4][5]}

連想プロセッサ DREAM-II は、前述 b), c) の問題を解決することを目指し、現在製作中のマルチモジュールシステムである。

DREAM-II では、モジュール群の制御、およびデータ転送において

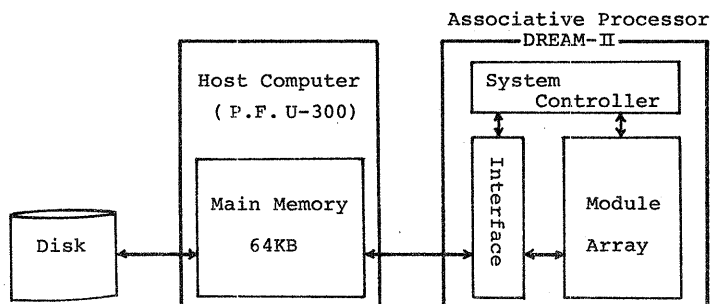
「連想」

の概念を導入することを試みた。

本報告では、DREAM-II のハードウェア/ファームウェアの実際について示す。

[Fig.1]

An Overview of Associative Processing System.



[2] DREAM-IIアーキテクチャと「連想」

連想処理は、

「複数の中から、ある特定の性質を示すことによってデータを取り出し、そのデータの持つ性質に対して処理を施す」

ことである。

連想処理は、基本的に並列性を持っており、SIMD型プロセッサによって連想プロセッサを構成することができる。

DREAM-IIでは「連想」の概念を

- ・ モジュール群の制御
- ・ シーケンス制御
- ・ データ転送

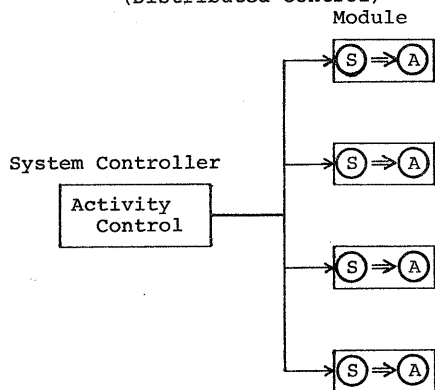
に導入している。

(2-1) 連想制御^[6]

DREAM-IIはSIMD型であり、各モジュールは基本的に同じ動作をする。そのため、モジュール群の個別制御は、モジュールの動作/非動作を指定することによって行われる。

DREAM-IIでは、モジュールの数が変わってもモジュール群を制御するコントローラの規模がほとんど変化しない制御方式(分散制御)を導入した。

[Fig.2] Associative Control
(Distributed Control)



- (S) : Module Status
- (A) : Module Activity

これは、コントローラがモジュールに対して「状態」を指定することによって、各モジュールが指定された状態と自身の状態を比較し、各々の動作/非動作を決定する方式である。この制御方式は連想処理の概念にあっているため「連想制御」と呼ぶ。[Fig.2]

連想制御では、各モジュールに

- ・ 動作/非動作を示すフラグ
- ・ モジュール固有の値(モジュール番号)を示す擬似レジスタ
- ・ 状態レジスタ

を置き、通常のレジスタと同様のALU演算を可能にしておく。これにより、全てのモジュールに同一のレジスタ間演算命令を与えることによって、特定のモジュールの動作を変更することができる。

コントローラがモジュールの状態を収集する手段としては、各モジュールに通信用のフラグ(モジュール情報フラグ)を置く。これは各モジュールの状態を代表し、コントローラは各モジュールのフラグの値をプライオリティエンコードした値のみを知る。これにより、コントローラとモジュール群の間の制御線の数を少くすることができる。

(2-2) 多重ループの高速化^[6]

連想処理の応用では、

一度に処理できる データ量 << 処理すべき データ量

であり、処理のループ演算が数多くある。

DREAM-IIでは、ループ演算の高速化、および制御の簡素化を検討した。

一重ループの実現には、ループ回数を数えるカウンタと戻り先番地を示すレジスタがあればよい。DREAM-IIでは、多重ループ制御用にループカウンタのスタック(カウンタスタック)と戻り先番地のスタック(アドレススタック)を設け、通常の演算命令中にループエンドを示すマーカ(EOL: End of Loop Marker)をつけた。これにより、ループ制御は次の二つの操作のみで実行できる。

(i) カウンタスタックへのループ回数のロード

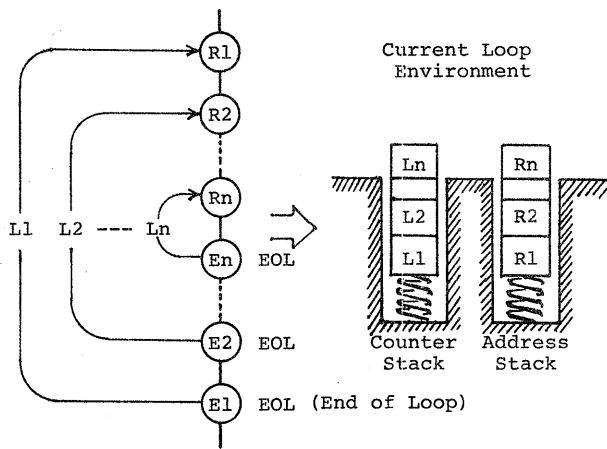
カウンタスタックに、それ以後開始するループ演算の回数をプッシュし、同時に次命令のアドレスをアドレススタックにプッシュする。

(ii) エンドオブ・ループ

EOLマーカの付いた命令を実行する際、現在実行中のループカウンタ（カウンタスタックのトップ）をテストする。カウンタが1でなければ、カウンタの値を1減じアドレススタックのトップに飛ぶ。カウンタの値が1であれば、カウンタ、アドレス両スタックをポップしてループを終了する。

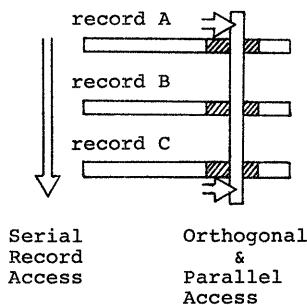
これにより、[Fig. 3] 左の様な多重ループの制御が簡素化され、高速化される。特に、ループ内に処理以外の命令（ループ制御用の命令）がないため、プログラミングが容易になることが重要である。

[Fig. 3] Loop Nesting

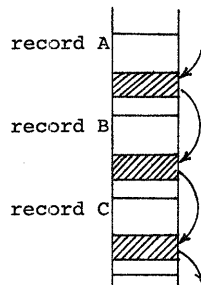


[Fig. 4]

(a) Table Format (2-Dimensional Data Space)



(b) Linear Data Space



(2-3) 連想プロセッサとホスト間のデータ転送 [2]

連想処理では、データ量が多く、ホスト主記憶とのデータ転送がシステムのスループットに大きな影響を与える。

単一データソース（ホスト主記憶）とマルチモジュールシステムを結合する場合、

- スキャンモード
- フロードキャストモード

によってデータ転送を実行する必要がある。DMAによって転送を行う場合、制御の手続きが複雑であるため、DREAM-IIでは専用ハードウェアによって転送を制御することにした。

これにより、複雑なデータ転送をひとつの転送命令によって実行でき、かつ、処理と転送のオーバーラップが可能となる。

また、ホスト側インタフェースには、ホスト主記憶に対するワード単位の二次元アクセス機能を設けた。

ホスト主記憶は一次元のデータ空間であるため、[Fig. 4] (a)の様な二次元のテーブルは(b)の様に配置される。処理すべきデータが図の斜線部である場合、連想モジュールにおける二次元アクセス記憶と同様に、斜線部のみを縦にアクセスすることが有効である。[2][4] これは主記憶上で(b)の様に斜線部のみをスキップしながらデータをアクセスすることに相当する。

一方、モジュールから処理結果を出力する際に、データ転送の無駄を省く手段として、必要な処理結果のみを出力する機能を設けた。これは、出力結果の有無を示すフラグをモジュールに置き、転送の制御部がこのフラグに従ってデータを収集し、ホスト主記憶に転送することによって行われる。

[3] DREAM-IIのハードウェア構成

本システムのホスト計算機は、主記憶64KBのミニコン(P.F.U-300)である。

ホスト計算機のバスには、連想プロセッサDREAM-IIと二次記憶としての磁気ディスク、および基本入出力が接続される。

DREAM-II内には主記憶に相当するものはなく、データの供給源はホスト主記憶である。DREAM-IIは、ホスト主記憶から見た場合、ファンクショナルモジュールの様に見える。

そこでDREAM-IIには次のことが要求される。

- i) ホストから、複雑な処理を一度に指示できる高レベルのコマンドが与えられるだけでDREAM-IIが自走してデータの処理を進める。
- ii) ホスト主記憶から、ホスト側バスの最高転

送速度で送られてくるデータに処理を施し、最高転送速度で主記憶に送り返す。

連想プロセッサDREAM-IIは次の3部分から構成される。

<システム制御部>

マイクロプログラムによってシステムの制御を行う。マイクロプログラムはホストから書き換え可能であり、問題向きのシステム構成が可能である。

<連想処理部>

連想モジュールの一次元アレイ。

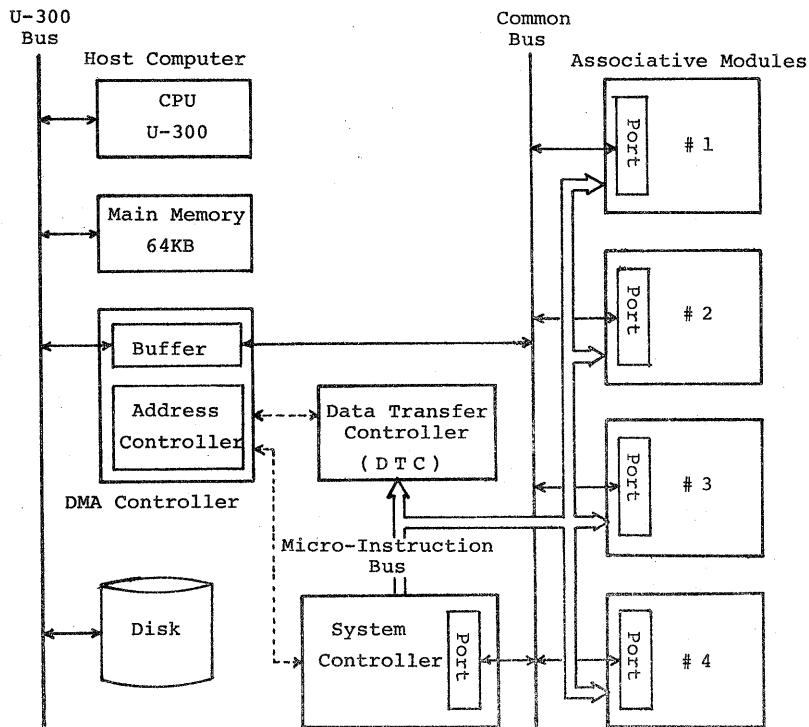
<インタフェース部>

ホスト計算機と連想プロセッサ間のデータ転送を行う。

DREAM-IIは基本的KSI M Φ マシンであり、DREAM-II内の全ての処理はシステム制御部からのマイクロ命令によって実行される。

[Fig.5]

[Fig. 5] Block Diagram of Associative Processing System.
--- Host Computer & DREAM-II ---



(3-1) 連想モジュール

連想モジュールには、主に以下の6つの処理機能がある。[4]

- f1. 二次元アクセス
- f2. バレルシフト
- f3. ビットグループ交換
- f4. バブルロジック
- f5. プライオリティ エンコーディング
- f6. ALU演算

具体的な連想モジュールの構成は[Fig.6]の様になっている。処理単位は全て16ビットであり、ALUにはAm2901を用いている。

処理機能の中で、f2, f3 は二次元アクセス記憶に用いているSE (Shuffle Exchange Network) 回路によって実現している。

Am2901内には、16ワードのレジスタファイル、およびQレジスタがある。ALUは、演算結果をレジスタファイルに格納する際に、全部で24通りのシフトが実行できる。シフトモードは、通常の算術シフト、論理シフトの他に、Qレジスタを含めた倍長シフト、モジュール間連結シフトが可能である。

モジュール間シフトは、画像データの様にデータのつながりが重要であるデータ処理に有効である。

制御レジスタには次のものがある。

MEF :モジュールアクティビティを示すフラグ
MIF :モジュールからシステム制御部への通信用フラグ

MIDR:モジュール番号を示す擬似レジスタ

OF :出力データの有無を示すフラグ

MSR :モジュールのALUステータスを示すレジスタ

これらは全てALUの外部レジスタであり、レジスタ間演算によって制御データを与えることができる。

各モジュールは、I/Oポートを通して、モジュール間共通バス(C-Bus: Common Bus)に接続される。[Fig.5]

(3-2) システム制御部

システム制御部は、演算部とマイクロ命令制御部に分けることができる。[Fig.7]

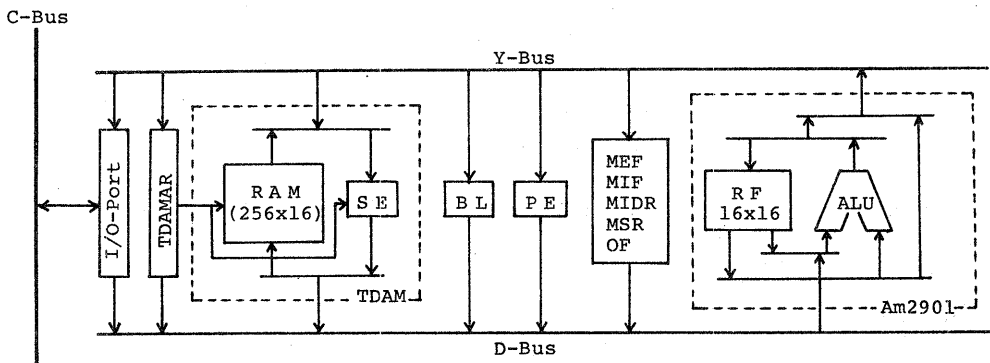
▷マイクロ命令制御部

マイクロ命令制御部は

- シーケンサ
- コントロールメモリ(CM)
- マイクロ命令レジスタ(MIR)

から構成される。MIRはパイプラインレジスタであり、現在実行中の命令が格納され、CM出力には次命令が用意される。

[Fig.6] The Detail of Associative Module



TDAM: Two-Dimensional Access Memory.

SE : Shuffle Exchange Network.

PE : Priority Encoder.

MEF : Module Enable Flag.

MIDR: Module Identify Register.

OF : Output Flag.

TDAMAR: TDAM Address Register.

BL : Bubble Logic.

RF : Register File.

MIF : Module Information Flag.

MSR : Module Status Register.

シーケンサでは、

[Fig.8]

- ① プログラムカウンタ (PC)
- ② CMアドレスレジスタ (CMAR)
- ③ アドレススタック (ADSK)
- ④ マイクロ命令のアドレスフィールド

から、次命令のCMアドレスが選択される。

多重ループ制御用カウンタスタック (CSK) は、減算カウンタとレジスタファイルによって構成される。カウンタスタック、アドレススタック (ADSK) は深さが16段であり、最高16レベルのサブルーチン、多重ループが可能である。

CSKのトップ (ループカウンタ) および両スタックのポインタ (CSK-P, ADSK-P) はALUの外部レジスタであり、ALU演算によってシーケンサ制御が行われる。

▷ 演算部

ALUは連想モジュールと同じAm2901であり、外部レジスタとして次のものがある。

MIFR: 各モジュールのMIFのPE出力を格納しているレジスタ

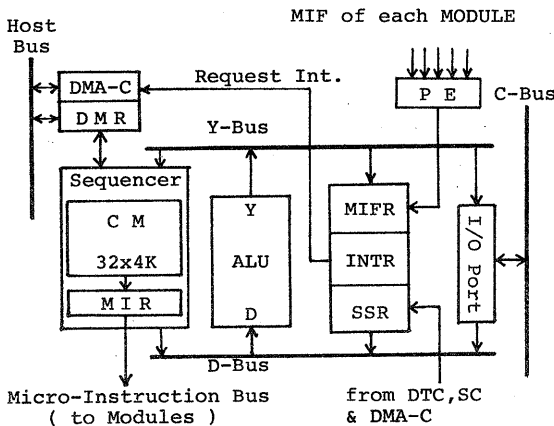
INTR: ホストへの割込みをかけるレジスタ

SSR: システムステータスレジスタ

▷ ホストからのモニタ

ホストは、DSR (Device Status Register) によってDREAM-IIの動作モード (MANUAL / RUN) を指定する。MANUALモード時は、DMRを通してCM, MIR, CMARをモニタすることができる。DSR, DMRはホスト主記憶上にアドレス付けされたレジスタであり、転送制御レジスタと共にホスト側インタフェース内にある。

[Fig.7] Block Diagram of System Controller



(3-3) 転送制御部

転送制御部 (DTC: Data Transfer Controller) はシステム制御部からのマイクロ命令によって起動され、DMA-C (ホスト側)、I/Oポート (各モジュール) を制御することによって、後述する各モードのデータ転送を実行する。

DMA-Cはホスト主記憶に対するワード単位の二次元アクセスを行うアドレス制御回路を持ち、サイクルスチールモードでDMAを実行する。

▷ 転送モード

DTCは転送命令 (CMV) によって指定された以下のモードのデータ転送を実行する。

a. SELECTIVE READ, SELECTIVE WRITE

特定のモジュールについて、1ワードのデータのリード/ライトを行う。

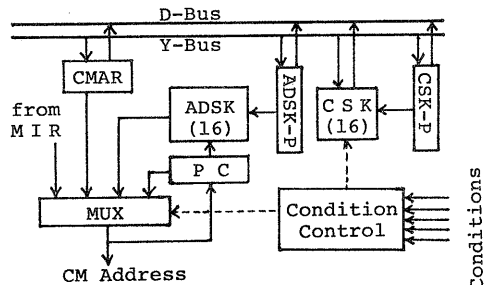
b. SCAN READ

ホスト主記憶から全てのモジュールに対して1ワードずつ別々のデータを転送する。

c. SCAN WRITE

全てのモジュールから、出力ポートに格納さ

[Fig.8] Sequencer



- PC: Program Counter
- CMAR: CM Address Register
- CSK: Counter Stack
- ADSK: Address Stack
- CSK-P: CSK Pointer
- ADSK-P: ADSK Pointer

- DMR: Device Monitor Register
- DMA-C: DMA Controller
- CM: Control Memory
- MIR: Micro-Instruction Register
- PE: Priority Encoder
- MIFR: MIF Register
- INTR: Interrupt Register
- SSR: System Status Register

れているデータを順にホスト主記憶に転送する。

d. BROADCAST

ホスト主記憶から全てのモジュールに同一のデータを転送する。

e. SCAN READ/write

上記 b, c を連続して行う。

f. GATHERING-WRITE

SCAN WRITE の拡張モードである。各モジュールは出力すべきデータの有無を事前に判断してフラグを立て、DTC は各モジュールのフラグに従ってデータを収集、転送する。この機能によって情報検索などにおけるデータ転送の冗長性を省くことができる。

▷ 処理と転送のパイプライン化 [Fig.9]

DTC は、転送命令を受けると自走して転送制御を行い、転送の終了をフラグによってシステム制御部に示す。システム制御部は、データ転送中にモジュールの処理を進めることができる。このため、データ入出力と処理のパイプライン化が可能であり、システムのスループットが向上する。

▷ ホスト主記憶への二次元アクセス機能

DMA-C 内のアドレス制御回路は、4個のパラメータを格納するレジスタ、ループカウンタ、アタチによって構成され、ホスト主記憶に対してワード単位の二次元アクセスができる。

この機能は、画像データの様に二次元的性質を持つデータを各モジュールに SCAN READ モードで分配する場合、また情報検索において処理したいフィールドのみを取り出したい場合等に適する。

[4] マイクロ命令の構成

DREAM-II のマイクロ命令は、32ビット長であり、以下の3種に大別される。[Fig.10]

- 演算命令 (1)~(8)
- 分岐命令 (9)
- 転送命令 (10)

(4-1) 演算命令

演算命令は、システム制御部、連想モジュールの ALU 演算を実行する命令である。

システム制御部、連想モジュール内の制御レジスタは各 ALU の外部レジスタであり、ALU の入力、出力として指定可能であるため、DREAM-II の制御は演算命令で実行される。

演算命令の先頭4ビットは、

E : End of Loop Marker.

R : Return from Subroutine Marker.

TG : Target of Micro-Instruction.

である。

E ビットはループ制御用ビットであり、ループの終了を示す。E ビットが 1 である演算命令が実行されると、システム制御部のループカウンタ (CSK のトップ) がテストされ、

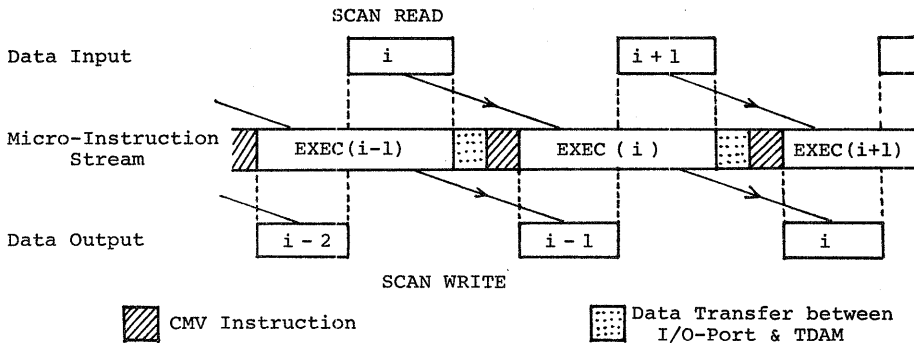
カウンタ ≠ 1 → カウンタをデクリメントしてループの先頭へジャンプ

カウンタ = 1 → ループ終了。CSK と ADBSK をポップして次の命令へ。

となる。

R ビットはサブルーチンの終了を示す。R ビットが 1 である演算命令が実行されると、アドレススタックのトップ (戻りアドレス) が次の

[Fig.9] Data Processing & Data Transfer



命令アドレスとして参照され、アドレススタックはポップされる。

TGフィールドは演算命令の対象を示す。

- TG = 00 : System Controller
- TG = 01 : Modules whose MEF=1
- TG = 10 : Modules whose MEF=0
- TG = 11 : All Modules

▷ ALU-0 ~ ALU-2 命令

この3命令は、システム制御部および各連想モジュール内での内部/外部レジスタ間の演算を実行する。

内部/外部レジスタは、

RA, RB, RAB : Internal Register of ALU
 RSX, RDX, RSDX : External Register of ALU
 の各フィールドで指定される。(A, BはALUソースセレクトにおけるポート名を示し、S, Dは外部レジスタのソース、デスティネーション指定を示す。)

EFN, SC, EDCの各フィールドは以下の意味を持つ。

- EFN : キャリーコントロールを含めたALU演算指定フィールド
- SC : ALUの2つの入カソース指定フィールド

[Fig.10] MICRO-INSTRUCTION OF DREAM-II

	31	28	24	20	16	12	8	4	0	
(1) ALU-0	E	R	TG	0	EFN	RB	RA	SC	EDC	RSX
(2) ALU-1	E	R	TG	1	EFN	RDX	RAB	SC	EDC	RSX
(3) ALU-2	E	R	TG	2	EFN	RB	RA	SC	EDC	RSDX
(4) EXLG	E	R	TG*	3	EFN	RB	RA	SC*	(SE-P)	EXLG
(5) MIX	E	R	TG	0 1 0	D	EFN	RDX	(DATA)		
(6) CXMX	E	R	TG*	0 1 1	D	EFN	RDX			RSX*
(7) MII	E	R	TG	1 0	SD	EFN	RAB	(DATA)		
(8) CXMI	E	R	TG*	1 1	SD	EFN	RAB			RSX*
(9) BRA	1	1	0				BRM	PC	BRC	(ADDRESS)
(10) CMV	1	1	1				TMD	MODULE NUMBER		

- E : End of Loop Marker.
- TG : Target of Micro-Instr.
- SC : Source Control of ALU.
- RB : Internal Registers of ALU (B-Port).
- RA : (A-Port).
- RAB : (A&B-Port).
- RSX : External Registers of ALU Source.
- RDX : Destination.
- RSDX : Source & Destination.
- BRM : Branch Mode.
- BRC : Branch Condition.
- R : Return from Subroutine Marker.
- EFN : Extended ALU Functions.
- EDC : ALU Dest. and Shift Control.
- PC : Polarity Control.
- TMD : Transfer Mode.

EDC: 内部レジスタにおけるALU出力の格納先およびシフトモード指定フィールド

▷ EXLG命令 (EXTERNAL LOGIC)

EXLG命令は、ALU演算に加え、連想モジュールの外部ロジック(SE, BL, PE)を並行して動作させる命令である。

外部ロジックはEXLGフィールドにおいて、以下の4種のいずれかを指定する。

SEM: Shuffle Exchange by Micro-Inst.
SER: Shuffle Exchange by TDAMAR
PE: Priority Encording
BL: Bubble Logic

命令のフローは、

RA → (EXLG) → ALU演算 → RB

である。

SEM, SERは、SE回路の制御データ(ex. バレルシフトのシフト量)をマイクロ命令のフィールド(SE-P)で与えるが、またはTDAMARから与えるかの違いである。

▷ MIX命令, MII命令

(MOVE IMMEDIATE EXTERNAL REGISTER)
(MOVE IMMEDIATE INTERNAL REGISTER)

この2命令は、DATAフィールドに示される16ビットデータをALUソースとする命令である。ALUソース、デスティネーションは、D, SDフィールドで部分的に指定可能であり、ALU出力は

MIX命令: RDXで指定される外部レジスタ

MI命令: RABで指定される内部レジスタに格納される。

▷ CXMX命令, CXMI命令

(MOVE SC's EXTERNAL REGISTER
to MODULE's EXTERNAL REGISTER)

(MOVE SC's EXTERNAL REGISTER
to MODULE's INTERNAL REGISTER)

この2命令は、MII, MIX命令における16ビットのデータフィールドを、システム制御部の外部レジスタ(RSX*)の内容に置き換えた命令である。この命令によって、システム制御部から連想モジュールへ制御データ(ex. ステータス, モジュール番号)を転送できる。

(4-2) 分岐命令 (BRANCH)

分岐命令では次の4つのモードの分岐が実行できる。

- (i) JUMP SUBROUTINE
- (ii) JUMP
- (iii) LOOP EXIT
- (iv) RETURN FROM SUBROUTINE

(i) ~ (iii)については

DIRECT MODE: 命令中のアドレスを参照

INDIRECT MODE: システム制御部内のCMARを参照

があり、全部で7通りの分岐モードがBRMフィールドで指定可能である。

これらの分岐は、

PC: Polarity Control

BRC: Branch Condition

に指定された分岐条件によって実行される。

BRCはシステム制御部内のSSRの各ビット(条件分岐)および1(無条件分岐)を指定する。

(4-3) 転送命令 (COMMON-BUS MOVE)

転送命令には、転送モード(TMD: Transfer Mode)フィールド、およびセレクトモード時に参照されるモジュール番号フィールドがあり、転送制御部(DTC)に起動をかけることによってTMDに指定されたデータ転送を実行する。

[5] まとめ

DREAM-IIは現在製作中であるため、実際のマシン上でマイクロプログラムを走らせることによって性能評価を行うことはできない。ここでは、DREAM-I(モジュール試験システム)と同じ応用プログラムを書いた場合について考える。

2値画像の細線化処理^[5]について検討すると、DREAM-IIでカウンタスタックを採用したことにより、実行速度で20~30%高速化される。これは最も内側のループの実行時間が短縮されるためであり、プログラムの作成も容易になる。また、DREAM-IIにおけるモジュール間連結シフトは、二次元アクセス記憶の各アレン境界処理の負担を軽減した。

連想制御方式、データ転送方式については、性能や効果を定量的に評価するまでには現在到っていない。

しかし、システムのスループットが、ある最適モジュール数を越えるとホスト計算機からの最高転送速度で抑えられることはわかる。

これは、[Fig.9] の1サイクルの処理 (SCAN READ → 処理 → SCAN WRITE) において、

データ転送時間 \propto モジュール数
であるためであり、応用 (処理のステップ数) によって最適モジュール数が決定されることがわかる。

現在、ハードウェアの製作と並行して、応用プログラムの開発を行っている。近日中に、プログラムを走らせることによって DREAM-II の定量的評価を行う予定である。

終りに、DREAM-II の設計において協力された、大学院生の久門耕一君に感謝する。

[参考文献]

- (1) 元岡、田中、上森 「パターン処理用プロセッサのアーキテクチャ」 情処全大第17回
- (2) 元岡、田中、上森、鈴木 「二次元記憶を用いた連想処理システム」
信学技報 EC 76-80
- (3) 上森、田中、元岡 「連想処理システムのモジュール化」 情処全大第18回
- (4) 元岡、田中、上森、河村 「連想プロセッサのハードウェア構成と処理機能」
信学技術報告 EC 77-77
- (5) 上森、田中、元岡 「連想プロセッサ DREAM-I の2値画像前処理への応用と評価」 信学技報 EC 78-19
- (6) 久門、後藤、上森、元岡 「マルチモジュール構成による連想プロセッサ DREAM-II の制御方式」 情処全大第20回
- (7) 後藤、久門、上森、元岡 「連想プロセッサ DREAM-II のデータ転送方式」
情処全大第20回