Vol. 27 No. 4 情報処理学会論文誌 Apr. 1986

ユニバーサル・ホスト計算機 QA-2 の低レベル並列処理方式†

北村俊明††* 中田登志之††** 柴山潔††
富田眞治†† 萩原宏††

算術論理演算装置(ALU)-レジスタ・レベルでの並列処理機能を有するマイクロプログラム制御計算機 QA-2 を開発した．QA-2 は，旧型機 QA-1 における各種の応用実験を通じて行った性能評価に基づいて，そのハードウェア構成方式を改良したシステムである．QA-2 は，そのマイクロプログラムを書き換えることによって，図形処理や信号処理などのリアルタイム処理，および高級言語処理に対する効率の良いシステムを提供できる．QA-2 では，256 ビット長の水平型マイクロ命令の相異なるフィールドによって，4 個の可変長 ALU 演算，4 個の主記憶アドレスへのアクセス，1 個の高機能順序制御を同時に指定できる低レベル並列処理方式が採用されており，マイクロプログラム制御方式の柔軟性を生かして，多様な応用に対処できるようになっている．また，処理速度や適応能力のほかにマイクロプログラムの生産性を上げることをも考慮して，マイクロ・アーキテクチャを一様に構成している．本論文では，QA-2 システムの設計思想を明らかにし，その最大の特徴である低レベル並列処理機構の実現方式について述べる．また，実際に QA-2 を応用した結果を示し，低レベル並列処理方式の有効性について評価を加える．

1. まえがき

LSI 技術の発展に伴い，計算機の利用形態は，従来の TSS による共有利用形態からホスト計算機をネットワークで結んだローカル計算機網による個人的利用形態に転換しつつある．この場合のローカル・ホスト計算機には，高級言語プログラムの高速実行および，図形，画像や信号のリアルタイム処理などが要求される．しかし，現在市販されているマイクロプロセッサを基本としたワーク・ステーションでは，これらの応用に対して十分な性能を得ることができない．

ユニバーサル・ホスト計算機[1]は，固定的なマシン命令セットを念頭に入れて設計された計算機ではなく，各種応用や言語に対して最適な中間言語を設定して，それをマイクロプログラムで解釈実行する計算機である．ユニバーサル・ホスト計算機として，Burroughs 社の B-1700/1800, Nanodata 社の QM-1, SCC 社の MLP-900, Stanford 大学の EMMY, 電子技術総合研究所の PULCE などがある[1]．また，LISP マシンを始めとする高機能ワーク・ステーションには，この方式を基本にして設計されているものが多い[2]．しかし，マイクロプログラム制御による逐次実行のみでは大量データのリアルタイム処理に対処することが困難であるので，並列処理方式の導入が必要とされる．そこで我々は，マイクロプログラム制御方式との親和性が良い低レベル並列処理方式（算術論理演算装置-レジスタ・レベルでの柔軟な並列処理方式）を考えて，これらを有機的に組み合わせることによって，ユーザの広範な要求を満たし得るユニバーサル・ホスト計算機 QA-2 を開発した．

QA-2 は，1976 年に完成した旧型機 QA-1[3] 上での種々の応用実験を通じての性能評価[4)-6)]に基づいて設計されており，QA-1 を基に種々のアーキテクチャ上の工夫を施した計算機となっている[7]．

本論文では，QA-2 の設計思想を明らかにし，特にその大きな特徴である低レベル並列処理機構の実現方式とその効果について述べる．順序制御方式の詳細，およびシステム管理用プロセッサ[19]については，各各別論文に譲る．

2. QA-2 の設計思想

2.1 ユニバーサル・ホスト計算機と低レベル並列処理方式

QA-2 の適用分野は，図形処理や信号処理などのリアルタイム処理，および高級言語マシンなどの各種（仮想）計算機のエミュレーションであり，QA-2 は研究室の研究環境の中心となるホスト計算機として位置付けられている．ユニバーサル・ホスト計算機[1]は，ソフトウェアによる処理をファームウェアで置き

† Low-Level Parallelism of a Universal Host Computer QA-2 by TOSHIAKI KITAMURA, TOSHIYUKI NAKATA, KIYOSHI SHIBAYAMA, SHINJI TOMITA and HIROSHI HAGIWARA (Department of Information Science, Faculty of Engineering, Kyoto University).

†† 京都大学工学部情報工学教室
* 現在 富士通(株)本体事業部
Main Frame Division of Computer Systems, Fujitsu, Ltd
** 現在 日本電気(株) C & C システム研究所
C & C Systems Research Laboratories, NEC Corporation.

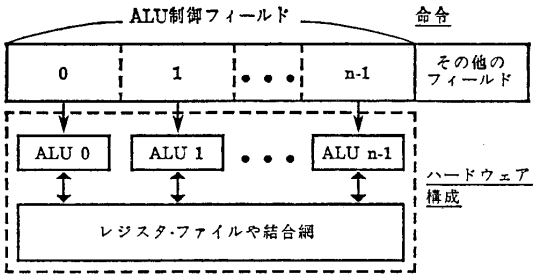

図 1 低レベル並列処理方式
Fig. 1 Low-level parallel processing.

換えることによって，高速化を達成することを基本原理としている．したがってユニバーサル・ホスト計算機では，多様な応用に適応できるように，柔軟なマイクロプログラム制御方式が採用されている[1]．しかし，従来のユニバーサル・ホスト計算機の実行過程は逐次処理によっており，高級言語処理などの逐次処理向き応用の高速化を達成できる反面，図形・画像・信号処理などの処理対象に明示的な並列性を含む応用に対しては，十分な性能を提供できないものであった．

低レベル並列処理は並列演算方式の一種であり，図1に示すように，比較的長い語長の命令の相異なるフィールドで多数の算術論理演算装置（ALU），メモリ・アクセス機構やバスなどをきめ細かく同時に制御する方式である．この方式は，演算機能レベルが低いので，マイクロプログラム制御方式との親和性が良い．我々は，QA-1 における経験から，処理対象に明示的な並列性のある応用[4]に対してはもちろんのこと，高級言語処理やリスト処理などの明示的に並列性を有しない分野[5),6)]においても，柔軟なマイクロプログラム制御方式による低レベル並列処理方式が有効に適用できることを示してきた．また，処理性能の向上率，ハードウェア規模やマイクロプログラミングの容易さなどの観点からの考察により，並列演算可能な ALU の個数は4が適当であると判断した．QA-2 の設計思想は，QA-1 のそれを継承して，「マイクロプログラム制御方式と低レベル並列処理方式との有機的結合」を基本にしている．

2.2 ユニバーサル・ホスト計算機のハードウェア構成方式

大規模マイクロプログラムの作成において，ユーザはプログラムの論理的構造（演算式や制御構造によって記述するアルゴリズムやデータ構造）をマイクロプログラムが実行される物理的構造（物理的なレジスタや主記憶空間におけるデータ表現，マイクロ命令の順序制御を行うハードウェア構成）に変換しなければならない．あるいは，高水準マイクロプログラム記述言語を用いる場合には，コンパイラが効率の良い目的マイクロプログラムに変換しなければならない．多様な応用に対して，この論理的構造と物理的構造のセマンティック・ギャップをできるかぎり小さくできるハードウェア構成方式が，ユニバーサル・ホスト計算機として必要である．ユーザに対するマイクロプログラミングの負担が増大したり，高水準マイクロプログラム記述言語のコンパイラが普及しないのは，その計算機ハードウェアの設計時にマイクロプログラムの生産性についての配慮がなされていないことに起因している場合が多い[8),9)]．我々は，QA-1 の開発や応用によって得られた経験から，ユニバーサル・ホスト計算機の評価の基準として，処理速度や適応能力と同様に，ファームウェアの生産性をハードウェア構成方式の設計時にも考慮することが重要であると判断した．したがって，QA-2 のマイクロ・アーキテクチャの実現においては，その並列演算機構や順序制御機構が一様な構造を保つように配慮した．

2.3 ALU の構成方式と低レベル並列処理

計算機アーキテクチャは，その命令起動方式の違いに応じて，コントロール駆動方式，データ駆動方式，要求駆動方式に大別される．このうちコントロール駆動方式では，制御装置は唯一であり，ALU の構成方式によって，単一命令・単一データ流（SISD）方式，スカラ並列演算方式，単一命令・複数データ流（SIMD）方式，演算パイプライン方式，およびこれらを複合化した複数命令・複数データ流（MIMD）方式の一形態であるマルチプロセッサ方式に分類される．このうちスカラ並列演算方式には，CDC 6600 に代表される方式と，QA-2 に代表される低レベル並列処理方式がある[10)]．

CDC 6600 で採用された方式では，命令パイプラインによって次々に生成される命令が複数の ALU で実行され，演算結果は命令パイプラインに入った順序とは異なった順序で出力され得る．CDC 6600 では，機能の相異なる 10 個の ALU がレジスタを共有しながら，スコアボードの制御の下で並列演算を実行する．スコアボードは ALU やレジスタの状態を保持するものであり，命令の実行に際して必要な ALU の割り付けや，必要なオペランド・データが既に他の ALU やメモリから供給されているかどうかなどの検査を行い，命令が実行可能である場合には所定の ALU の実行が開始される．したがって，この方式は，連続する命令の実行に際してデータ駆動方式で制御を行ってい

表 1　低レベル並列処理方式のマシン
Table 1　Machines with low-level parallelism.

名　称	発表年	開発機関	文献	応用分野	命令ビット長	命令実行サイクル(ナノ秒)	並列ALU演算数	ALU の種類と特徴
AMP	1973	Argonne National Laboratory	11)	リアルタイム処理(グラフィックスなど)	74	430	2	∘8 ビットおよび 16 ビット・整数演算器
QA-1	1976	京都大学	3)	図形・画像・信号処理,高級言語処理	160	350	4	∘16 ビット整数演算器（乗算器付き）
AP-120 B	1976	Floating Point Systems	12)	アレイ演算，信号処理(FFT 演算など)	64	167	3	∘38 ビット・浮動小数点加算器および乗算器（パイプライン制御） ∘16 ビット・整数演算器（逆 2 進変換器付き）
Vanguard (1100/60)	1979	Univac	13)	汎　用	283	116	2	∘36 ビット・汎用演算器(MC 10800 を使用)
MUNAP	1979	宇都宮大学	14)	非数値処理(記号処理など)	188	555	4	∘16 ビット・非数値処理用プロセッサユニット ∘2 レベル・マイクロプログラム制御
QA-2	1983	京都大学	—	図形・画像・信号処理,高級言語処理	256	600	4	∘8/16/24/32 ビット・整数演算器(乗算器付き)
ELI-512	1983	Yale Univ.	15)	科学技術計算	500以上	—	20～30	∘32 ビット・整数演算器 ∘64 ビット・浮動小数点演算器 ∘Trace Scheduling によるコード・コンパクション
ベクトルプロセッサ	1984	日立製作所	16)	(ミニコン内蔵型)ベクトル演算	96	167	3	∘32 ビット・浮動小数点乗算器(パイプライン制御) ∘64 ビット・浮動小数点加算器(パイプライン制御) ∘32 ビット・整数演算器
CYBER-PLUS	1984	CDC	17)	科学技術計算	240	20	8	∘8/16/32 ビット・整数加算器(2 台)，整数乗算器(1 台)，および論理演算器(2 台) ∘32/64 ビット・浮動小数点加算器，乗算器，および除算(平方根演算) 器（パイプライン制御，各 1 台）
MC	1985	松下電器産業・無線研究所	18)	グラフィックス	96	250	5	∘浮動小数点加算器および乗算器 ∘関数発生器（三角関数，平方根，逆数など） ∘浮動-固定小数点数変換器 ∘24 ビット・汎用演算器

ると言える[10].

低レベル並列処理方式は，図 1 に示したように，比較的長い命令を多数のフィールドに分割し，各々のフィールドで ALU などの機能装置を独立に制御する方式である．データ駆動方式や CDC 6600 が実行時に並列演算の可能性を判定するのに対して，この方式のマシンではそれをコンパイル時に行う．コンパイラは，ソース・プログラムから並列演算可能なものを抽出して，一つの命令に合成する．その制御装置は，データ駆動方式などと比較して単純化できる．また，ALU などの機能装置間の通信は命令内のフィールドで直接制御され，そのオーバヘッドは少ない．この方式では，ALU 個数に近い並列度が得られる時に高速処理を達成できる．しかし，並列度が低い時には ALU 制御フィールドに遊びができ，命令のビット使用効率が低下する．また，複数 ALU で多数のステータスが生成されるので，順序制御部を柔軟な構成にしておく必要がある．低レベル並列処理方式を採用したマシンとして，**表 1** のようなものがある．これらの大半は，単一の応用目的を意識して，非均質の ALU 構成（機能の異なる多数の ALU を実装）方式を採用している．しかし QA-2 では，ユニバーサル・ホスト計算機として，より広範な応用に適応できるように，均質な(同一機能の複数 ALU を実装) ALU 構成方式を採用している．

2.4 QA-2 システムの構成方式

ファームウェア工学[9] の観点に立脚して設計された QA-2 システムの構成方式の特徴は次のようにまとめ

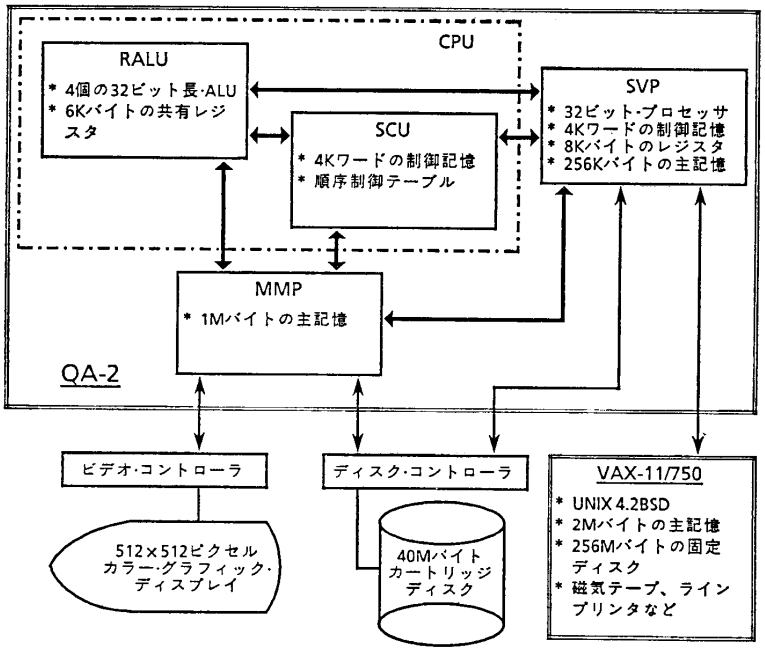

図 2 QA-2 のシステム構成
Fig. 2 System organization of QA-2.

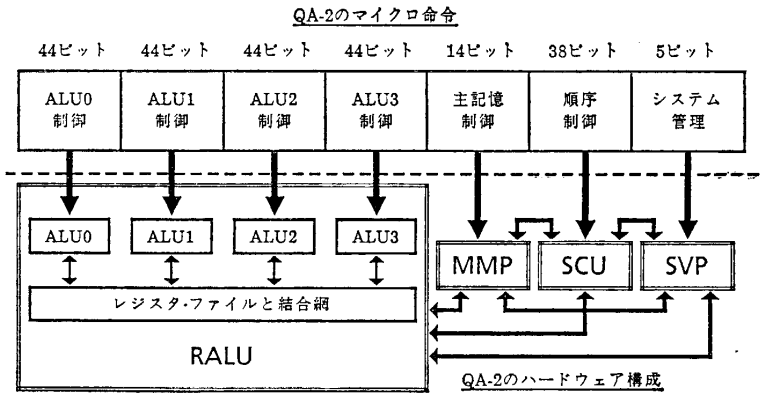

図 3 QA-2 の低レベル並列処理方式
Fig. 3 Low-level parallel processing mechanism of QA-2.

られる.

(1) 処理機能を分散化したシステム構成方式.

図 2 に示したように, QA-2 はレジスタ・ALU 部 (RALU) と順序制御部 (SCU) から成る CPU と, 主記憶管理プロセッサ (MMP), システム管理プロセッサ (SVP) の三つの機能部分を独立に構成し, これらを相異なるクロックと水平型マイクロ命令 (1ワード=256 ビット) の相異なるフィールドによって独立して制御する機能分散方式によって構成されている.

(2) 同一で高機能な4個の ALU による低レベル並列処理方式. QA-2 では, **図 3** に示すように, 4個の可変長 ALU が1ワード (256 ビット) の水平型マイクロ命令の相異なるフィールドで独立に制御され, それらが均一で大容量のレジスタ・ファイル (6 kバイト) を共有して動作する. 4個の ALU は互いに独立な四つのオペランド群に対して並列演算を実行できるだけでなく, 1個の演算結果を他の ALU 演算の入力オペランドとして使用する ALU 連鎖演算も可能である.

(3) マイクロプログラムの生産性の向上を指向した順序制御構造. 大規模なユーザ・マイクロプログラミングを実現するためには, ユーザが記述する論理レベルのマイクロプログラミング構造と物理的なハードウェア構造との対応関係を容易にとり得ることが要求される. 特に高度な制御構造は構造化マイクロプログラミングを実行する場合に必須であり, "IF-THEN-ELSE" 文や "CASE-OF" 文などの強力な条件判定分岐機能はハードウェアとして実装されていることが望ましい. QA-2 では, 低レベル並列処理機能を十分に生かせるように, 高度の順序制御構造が採用されている.

3. QA-2 の低レベル並列処理機構

図 4 に示したように, QA-2 の CPU は RALU と SCU との二つの機能部分に分けられる. さらに, RALU は1個の均一構成のレジスタ・ファイル (RF) と4個の同一機能の ALU とから成る.

3.1 均質・大容量共有レジスタ・ファイルの構成方式

低レベル並列処理方式では, ALU がレジスタを共有して動作する. したがって, 共有レジスタの構成方式がプログラミングの負担とシステム性能に大きな影響を及ぼす. 理想的には, 共有レジスタは大容量で, しかも任意の ALU の左右入力にデータを供給でき, 任意の ALU の出力結果を格納できることが望ましい.

QA-2 では, 以下のような方法で均質・大容量の共有レジスタを実現している. QA-2 の RALU に装備

されているレジスタ群としては，図4に示すように，定数レジスタ・ファイル (CRF : 1k バイト)，汎用レジスタ・ファイル (GRF : 1k バイト)，間接レジスタ・ファイル (IRF : 1k バイト×4 バンク)，および特殊レジスタ (SR) がある．このうち GRF，CRF，IRF は4個の ALU のいずれからも一様な共有レジスタ空間としてバイト単位でアクセスすることができるので，ユーザによる複雑なレジスタ割り付けは不要である．また，これらのレジスタ・ファイルは ECL・RAM で実装されており，実際のハードウェア構成と

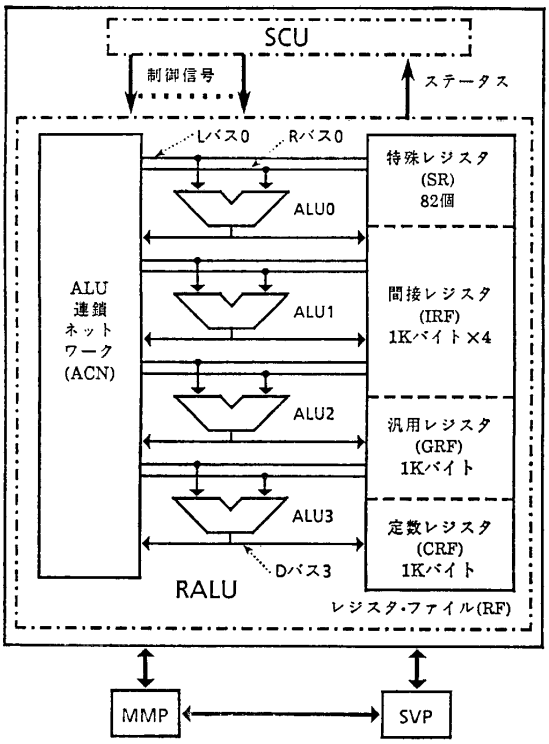

図 4　RALU のバス構成
Fig. 4　Bus organization of the RALU.

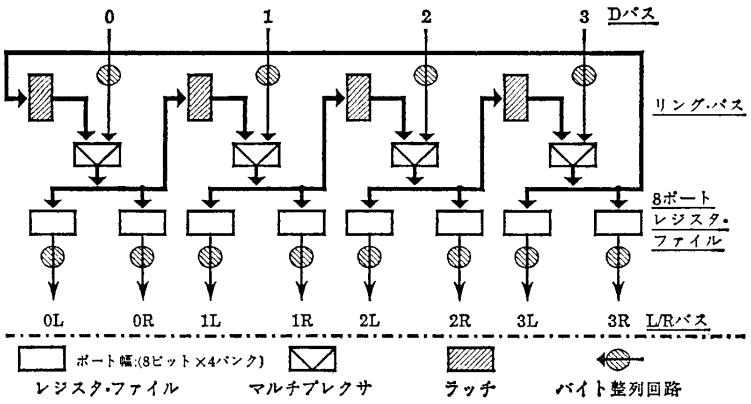

図 5　レジスタ・ファイルの構成方式
Fig. 5　Register file organization.

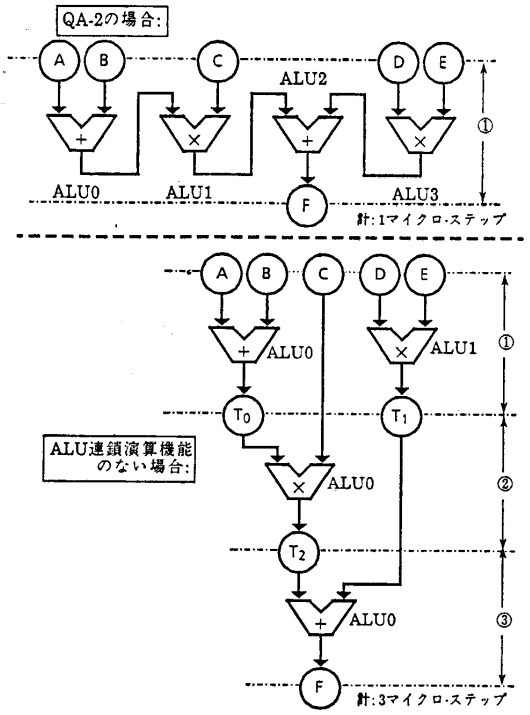

図 6　ALU 連鎖演算機能の効果
Fig. 6　An effect of the ALU chaining mechanism.

して，各 ALU の右入力，左入力ごとに各々同一コピーを持つレジスタ・ファイルが設けられている．したがって，4個の ALU 演算のソース・オペランドとして相異なるアドレスに存在する8個のデータを，同時に読み出すことが可能である．また，**図5**に示すように，ALU 演算終了時にはこれら合計8枚のレジスタ・ファイルすべてにリング状の ALU 出力バスを通して順々に4個の演算結果が格納され，これらのレジスタ・ファイルには最終的に同一データが保持される．レジスタ・ファイルの各ポートは4バンク構成になっており，複数バイト・アクセスの際には，バイト整列がハードウェアでなされる（図5のバイト整列回路）．

3.2　均質な ALU 構成による並列/連鎖演算機構

低レベル並列処理方式では，マイクロ命令の各フィールドが活用されて初めて，高速処理を達成できる．したがって，ユーザが論理レベルで考えた並列演算構造をそのまま物理レベルの構造に対応付けることが可能でなければならない．さらに，性能を上げるためには，最適化手法による並列操作の1マイクロ命令への合成（埋め込み）を行うことが必

要である.

QA-2の各ALUでの演算長は，バイト単位に最大4バイトまで可変であり，レジスタへのアクセスも演算データ長に合わせて自動的に制御される．さらにQA-2では，**図6**に示したように，1個のALUによる演算結果を別のALUの入力データとして指定するALU連鎖演算機能も備えているので，ユーザが4個までのALU演算を並列／逐次の区別なく1マイクロ命令で記述できる．したがって，1マイクロ命令で実行できるALU演算機能レベルが向上し，データの一時退避というオーバヘッドもなくなっている．1マイクロ命令で最大4組の演算の実行が保証されているので，ユーザ・マイクロプログラムの論理構造と物理的なALU構成との対応付けが容易であり，低レベル並列処理機能を活用した大規模ユーザ・マイクロプログラミングが可能である.

例えば図6の例では，変数 A〜F をすべて共有レジスタに割り付けて，ALU連鎖演算機能を用いると，この演算："$F=(A+B)\times C+D\times E$" を1マイクロ・ステップで行うことが可能となる．しかしALU連鎖演算機能を持たない場合には，同様の演算を行うのに3マイクロ・ステップ必要である.

3.3 バス構成とマイクロ命令形式

各ALU入力ラッチは，**図7**に示すマイクロ命令のLSRCおよびRSRCフィールドによって指定されたレジスタ・ファイル(RF)内のいずれかのレジスタとLバスおよびRバスによって結合される．ALU演算の結果であるALU出力データはDバスを経由して，DSTフィールドによって指定されたレジスタに格納される．また，各Dバスは ALU 連鎖ネットワーク回路(ACN)を経由して任意の他のALU入力ラッチとも結合することができる(ALU連鎖演算機能).

各ALUオペランドのデータ長はOPLフィールドで指定され可変(1, 2, 3または4バイト)である．したがって，ユーザは1マイクロ命令で4個の相異なる可変長データに対する並列 ALU 演算を指定できる．ALUへの入力データはバイト単位でアクセス可能な RF から，右端(最下位ビット)調整されて読み出される.

3.4 ALU 演算機能とその制御方式

マイクロ命令の OP フィールドが ALU による演算を指定する．複数 ALU 演算を連鎖演算機能によって組み合わせることができるので，高機能 ALU 演算(例えば，データ・フィールド操作，ビット入れ換え操作あるいは浮動小数点数の正規化など)も1マイクロ命令で実行することが可能である．OP フィールドで指定できる特徴的な ALU 演算には次のようなものがある.

(1) 16 ビット・ECL バレルシフタ (MC 10808) による高速並列シフト(最大32ビット)演算．シフト桁数は，RSRC フィールドで指定する.

(2) 8×8ビット・乗算器 (Am 25S558) による最大 32 ビット長整数の乗算．符号付きと絶対値表示の2種類の整数乗算が可能である.

(3) 浮動小数点演算を支援するためのプライオリティ・エンコード(データの最上位ビットから連続する "0" や "1" のビット列長の計数)演算.

(4) マスク機能付きのバイトあるいはビット演算．鏡像(逆2進)演算.

(5) 専用レジスタの内容による間接演算指定.

QA-2の各ALU機能モジュールは32ビット長演算を基本に設計されている．またALU部には，例えば浮動小数点プロセッサ・チップなどの機能モジュールの付加を簡単に行うことができる構成になっている．そして，可変長データ演算や実行時間長の異なる演算に対処するために，機能モジュールの各部分は，演算タイミングなども含めて，ナノ記憶(256ワード)から読み出された1ワード(64ビット)のナノ命令によって制御する方式を採用している．ナノ記憶はECL・RAMチップで構成されている．ナノ命令は，マイクロ命令のOPとOPLフィールドに従ってフェッチされ，各ALUの機能モジュールをきめ細かく制御する.

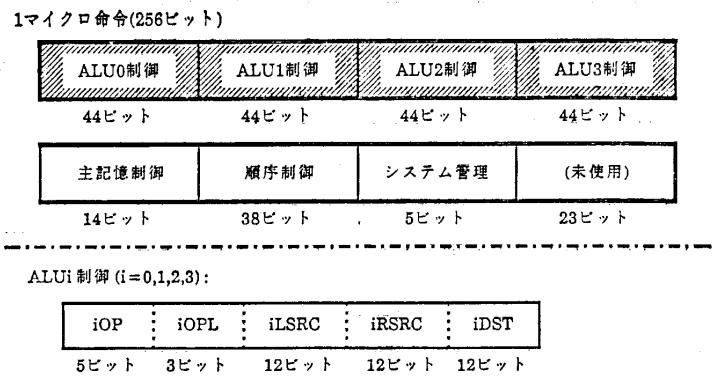

図7 マイクロ命令形式
Fig. 7 Micro-instruction format.

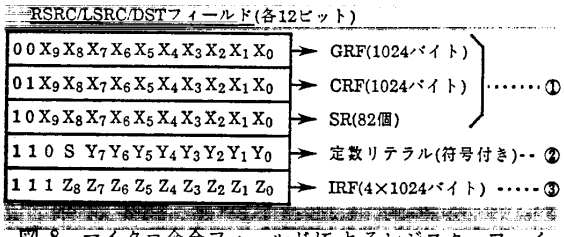

図 8 マイクロ命令フィールドによるレジスタ・ファイルの指定

Fig. 8 Register file access method specified by a micro-instruction.

3.5 レジスタ・ファイルへのアクセス方式（図8参照）

レジスタ・ファイル (RF) に対するアクセス方式はマイクロ命令の LSRC, RSRC および DST フィールドによって指定される. GRF, CRF と SR への直接アクセス（図8の①）は，各フィールドの下位 10 ビット (X_9, …, X_0) によってアドレス指定が行われる. CRF にはマイクロ・アセンブラによって必要と判定された定数が，マイクロプログラムのロード時に SVP によって格納される. 9ビット以上の定数は LSRC または RSRC フィールドによって指定されたアドレスの CRF から読み出されるが，8ビット以下の定数は直接マイクロ命令のフィールドを利用して生成され（図8の②），CRF には格納されない.

IRF へのアクセス（図8の③）は12個ある IRF 用アドレス・レジスタ (IRFAR) のうち1個を LSRC, RSRC あるいは DST フィールドの下位4ビット (Z_3, …, Z_0) によって選択することによって行う. また，(Z_5, Z_4) によって IRF アクセス時の IRFAR の自動増減機能を制御する. IRFAR はアクセスされたバイト長だけ自動的に増減されるので，ユーザは IRF を論理的なスタックあるいはキュー構造のレジスタ空間とみなして利用することができる. IRF は4組装備されており，そのうちの1組を (Z_7, Z_6) で選択する（ただし，Z_8=0 の場合）. なお，4個の IRF を連続した 4 kバイトの大容量レジスタ空間としたアクセス・モードにすることも可能である（Z_8=1 の場合）.

SR は QA-2 内部のハードウェア・ファシリティを直接制御するために設けられており，各々が独立した MSI チップによって構成されている. SR には，例えば主記憶管理用にメモリ・アドレス・レジスタやメモリ・データ・レジスタ，順序制御用にカウンタ，さらに RF アクセス制御用に前述した IRFAR などが装備されている.

また，RF アクセスに対する間接アドレス指定も可能となっている. この場合には，RF 内の1ワードがまず読み出され，その内容（図8と同一形式）に基づいて，RF へのアクセスが再度実行される.

3.6 レジスタ・ALU 部の制御方式

RALU では，異なる実行時間を有する演算のタイミング制御，LSRC/RSRC/DST フィールドによる直接アクセスや IRF アクセスの制御，間接アドレス指定の制御，ALU 連鎖演算の制御などが必要である. RALU の各 ALU に対して，LSRC, RSRC, EXEC DST 制御オートマトンを配置し，これら合計16個の制御オートマトンが相互に同期をとりながら状態遷移し，制御信号を生成するようになっている.

例えば，ALU0 の OP フィールドで乗算，LSRC で IRF アクセス，RSRC で ALU1 の演算結果，DST で間接アドレスをそれぞれ指定した場合における，ALU やレジスタの制御信号について説明する. この例では，マイクロ命令がフェッチされると，ALU0 の LSRC および RSRC 制御オートマトンがまず起動される. LSRC 制御オートマトンは，IRFAR の内容を読み出し，IRF にアクセスし，指定されたデータをラッチする. RSRC 制御オートマトンは，ALU1 の EXEC 制御オートマトンから演算終了を通知されるまで待ち状態となり，ALU1 の演算が終了するとその結果をラッチに取り込む. ALU0 の演算に必要な両 SRC データがそろうと，ALU0 の EXEC, DST 制御オートマトンが起動される. DST 制御オートマトンは，間接アドレスの計算を行って，全 ALU の EXEC 制御オートマトンからの終了通知を待つ. EXEC の終了信号がそろうと，全 ALU の DST 制御オートマトンは一斉に，RF に対する書き込み（図5参照）ステージに入る. ALU0 の EXEC 制御オートマトンは，ナノ命令の情報を使って乗算の演算時間長だけ待ち，全 ALU に終了通知を発する.

4. QA-2 の低レベル並列処理方式の評価

QA-2 の低レベル並列処理方式の有効性を確認するために，QA-2 を実際の問題に応用し，動的な性能評価を行った. 例題としては，処理対象に明示的な並列性のある応用として，（1）スキャンライン・アルゴリズムによる3次元グラフィックス，また明示的に並列性を有しない応用として，（2）逐次型 Prolog のインタプリタ，（3）Lisp のインタプリタ，を選んだ. いずれの応用においても，QA-2 のマイクロ・アーキテクチャについて様々な観点から定量的な評価

表 2 1マイクロ命令における平均使用 ATU 個数
Table 2 Mean number of ALUs used in one micro-instruction.

応用問題	(個/4個)
(1) 3次元グラフィックス	3.04
(2) Prolog インタプリタ	2.93
(3) Lisp インタプリタ	2.93

表 3 ALU 構成を変えた場合の実行マイクロ命令ステップ比
Table 3 Relative proportion of the number of micro-instruction steps executed with modified ALU organization.

応用問題	ALU 個数				連鎖演算機能なし
	1	2	4 (現状)	8	
(1) 3次元グラフィックス	3.04	1.66	1	0.87	1.20
(2) Prolog インタプリタ	2.94	1.65	1	0.79	1.38
(3) Lisp インタプリタ	2.51	1.65	1	0.97	1.49

を加えたが，本章では QA-2 の低レベル並列処理機能の効果の概要を述べるにとどめ，詳細については別稿[20] に譲る．

各応用における QA-2 の代表的な特性を抽出するために，(1) では 30 個の立方体をランダムに表示する場合の可視部分の決定ルーチン，(2) では 30 個の要素から成るリストを反転するプログラム，(3) では「tarai-4」のベンチマーク・プログラムを，それぞれ具体的な問題として選んだ．

まず，**表 2** に示すように，1マイクロ命令で使用されている平均 ALU 個数は，いずれの問題においても約3個であり，問題における明示的な並列性の有無にかかわらず，QA-2 の低レベル並列処理機能を十分に活用できることが分かった．

また，QA-2 から ALU 連鎖演算機能を削除すると，各問題において 20〜50% のオーバヘッドが生じる（**表 3** 参照）．特に，明示的な並列性を持たない Prolog マシンや Lisp マシンのエミュレーションにおいては，この機能の効果が大きいことが分かる．

さらに表3に示すように，QA-2 において4個の ALU を並列動作させる効果は，1個しか ALU を持たない場合の 2.5〜3 倍もあり，逆に ALU を8個（2倍）にしても，数%からせいぜい 20% の性能向上しか得られない．ALU を8個にした場合，ハードウェア規模は確実に2倍以上になることを考慮すると，並列演算可能な ALU 個数としては4が妥当であり，広範な問題適応性を持つと判断できる．

図 9 QA-2 の全体写真
Fig. 9 Photograph of QA-2.

5. むすび

QA-2 は，旧型機 QA-1 による応用で得られた種種の定性的・定量的評価データを基にして新しく設計されたものであり，マイクロ・アーキテクチャはユニバーサル・ホスト計算機として高性能かつ柔軟性に富んだ斬新なものとなっている．特に，その低レベル並列処理機能の有効性は，各種の応用によって確認されている．

しかし，**図 9** の全体写真で示したように，実装規模は予想外に大規模化し（QA-2 全体での IC チップ数は約 22,000 個），最小マイクロ命令実行サイクルは約 600 ナノ秒（設計時の性能予測では，200 ナノ秒）となっている．QA-2 のハードウェアは，メモリ（MOS チップ）を除いて ECL とショットキ TTL チップで構成されている．乗算器やシフタなどの ALU 関連あるいは ECL メモリで構成されている汎用レジスタ・ファイルなどでは LSI が使われているが，ビットスライス・マイクロプロセッサは使用していない．また，システムの設計開始からハードウェア本体の開発の終了まで4年半を要した．現在では，AMD 社の Am 293XX シリーズや Weitek 社の浮動小数点演算器など，新しいビルディング・ブロックが出そろってきたので，これらを利用して，よりコンパクトでしかも高性能な低レベル並列処理計算機を実現できるものと考えている．

謝辞 我々と共に，QA-2 の設計・開発を行った山下博之（日本電信電話（株）），栗山和則（(株) 日立製作所），中島浩（三菱電機（株）），藤井誠（(株) 東芝），河村武司（松下電器産業(株)），釜田栄樹（(株)日立製作所），村上和彰（富士通（株））の各氏，および QA-2 システムの開発に際しご協力いただいた萩原研究室の皆様に感謝いたします．

参考文献

1) 飯塚 肇：ユニバーサル・ホスト・プロセッサ，ダイナミック・アーキテクチャ，相磯，飯塚，坂村（編），*bit*・臨時増刊，Vol. 12, No. 10, pp. 1329-1350 (1980).
2) 安井 裕：LISP マシン，情報処理，Vol. 23, No. 8, pp. 757-772 (1982).
3) Hagiwara, H., Tomita, S., Oyanagi, S. and Shibayama, K.: A Dynamically Microprogrammable Computer with Low-Level Parallelism, *IEEE Trans. Comput.*, Vol. C-29, No. 7, pp. 577-590 (1980).
4) 和泉，川本，柴山，富田，萩原：実時間色彩動画システムの開発，電子通信学会論文誌，Vol. 63-D, No. 2, pp. 161-168 (1980).
5) 北村，柴山，富田，萩原：マイクロプログラム制御計算機 QA-1 による直接実行型高級言語計算機の構成とその問題適応化方式，電子通信学会論文誌，Vol. J65-D, No. 7, pp. 882-889 (1982).
6) 柴山，中田，北村，富田，萩原：ユニバーサル・ホスト計算機 QA-1 による各種高級言語プロセッサのエミュレーション，電子通信学会論文誌，Vol. J65-D, No. 11, pp. 1374-1381 (1982).
7) Tomita, S., Shibayama, K., Kitamura, T., Nakata, T. and Hagiwara, H.: A User-Microprogrammable Local Host Computer with Low-Level Parallelism, *IEEE Conf. Proc. of the 10th Annual Int. Symp. on Comput. Architecture*, pp. 151-157 (1983).
8) Jones, L. H.: Instruction Sequencing in Microprogrammed Computers, *AFIPS Conf. Proc. of NCC*, Vol. 44, pp. 91-98 (1975).
9) Davidson, S. and Shriver, B. D.: An Overview of Firmware Engineering, *Computer*, Vol. 11, No. 5, pp. 21-34 (1978).
10) 富田眞治：処理装置の構成，VLSI コンピュータ I，元岡 達（編），岩波書店，pp. 12-172 (1984).
11) Barr, R. G., Becker, J. A., Lidinsky, W. P. and Tantillo, V. V.: A Research-Oriented Dynamic Microprocessor, *IEEE Trans. Comput.*, Vol. C-22, No. 11, pp. 976-985 (1973).
12) 伯東株式会社(訳)：*AP-120B Processor Handbook 7259-02*, Floating Point Sytems Inc., p. 139 (1976).
13) 中西直之：最近のコンピュータ技術とバンガード・システム，インターフェース，Vol. 6, No. 3, pp. 166-173 (1980).
14) 馬場，石川，奥田：2 レベルマイクロプログラム制御計算機 MUNAP のアーキテクチャ，電子通信学会論文誌，Vol. J64-D, No. 6, pp. 518-525 (1981).
15) Fisher, J. A.: Very Long Instruction Word Architectures and the ELI-512, *IEEE Conf. Proc. of the 10th Annual Int. Symp. on Comput. Architecture*, pp. 140-150 (1983).
16) 阿部，高藤，坂東，平沢，加藤：スーパミニコン内蔵型ベクトルプロセッサの演算制御方式，情報処理学会論文誌，Vol. 25, No. 4, pp. 614-621 (1984).
17) Bongiorno, V.: The CYBERPLUS Multiparallel Processor System, Donaldson, R. and Kreisler, M. N. (eds.) *Proc. of the Symp. on Recent Developments in Computing, Processor and Software Research for High-Energy Physics*, pp. 321-331 (1984).
18) 中瀬，日高，西村，宮崎，野口，鷲島：高速浮動小数点演算機能を持つユニット・コンピュータ・MC のアーキテクチャ，情報処理学会・研究会資料，Vol. CA-85, No. 15, pp. 1-8 (1985).
19) 中田，北村，柴山，富田，萩原：マイクロプログラム制御計算機 QA-2 のシステム管理プロセッサ，情報処理学会論文誌，Vol. 27, No. 3, pp. 356-364 (1986).
20) Tomita, S., Shibayama, K., Nakata, T., Yuasa, S. and Hagiwara, H.: A Computer with Low-Level Parallelism—Its Applications to 3-D Graphics and Prolog/Lisp Machines—, *IEEE Conf. Proc. of the 13th Annual Int. Symp. on Comput. Architecture* (1986, to appear).

(昭和 60 年 11 月 18 日受付)
(昭和 61 年 1 月 17 日採録)

北村 俊明（第 27 巻第 3 号参照）

中田登志之（第 27 巻第 3 号参照）

柴山 潔（第 27 巻第 3 号参照）

富田 眞治（第 27 巻第 3 号参照）

萩原 宏（第 27 巻第 3 号参照）