C5「ダイナミック・マイクロプログラミング」シンポジウム報告

萩原　宏（京都大学）

1. シンポジウムの趣旨など

第14回プログラミング・シンポジウムの拡大運営委員会で，今年の夏のシンポジウムは，テーマを「ダイナミック・マイクロプログラミング」1本に絞って行なうことが決定され，早稲田大学の宇都宮公訓氏と私が幹事の大役をお引受けすることになった。

マイクロプログラミングに関しては，1971年にIEEEのTransaction on Computerで特集号が組まれ，その先年Hussonが「Microprogramming：Principles and Practices」を著述，我が国でも，今年情報処理6月号で特集号が出された。マイクロプログラミングに対する評価（評価は光の当て方によって変わるものであるが）は，システムの柔軟性（flexibility）が重視され，ソフトウェアのファームウェア化が要求されるようになって，ますます高くなって来つつある。さらに，マイクロプログラムを動的に変化させる技術が現実のものとなり，将来の新しい可能性を求めて，ダイナミック・マイクロプログラミングが本格的に研究されるようになった。

このような現状をバックに，次のようなお誘い状を関係各方面にお送りし，御協力をお願いした。

ダイナミック・マイクロプログラミング・シンポジウム出席のお誘い

情報処理学会プログラミング
シンポジウム委員会
夏のシンポシウム幹事　萩原　宏
宇都宮公訓

趣　　旨

バランスのとれた性能のよい計算機システムを目指して，ミニコンピュータから大型計算機に到るまで，マイクロプログラミングが広く採用されるようになり，いまや本格的なマイクロプログラミング時代に突入した感があります。マイクロプログラミングそのものは目新しくはありませんが，最近の集積回路技術の発達により，ソフトウェアで計算機の制御ロジックを動的に変更するダイナミック・マイクロプログラミングや，従来ソフトウェアで行なっていた作業のファームウェア化など革命的な応用が現実のものになってきたように思われます。

そこで，ダイナミックマイクロプログラミングの計算機システムへの導入や，ソフトウェア化に関する一般的もしくは具体的検討結果を持ち寄って討論し，この方面の発展に寄与したいと考え，本シンポジウムを開催することにしました。

実りあるシンポジウムにするため，上記のような趣旨に御賛同いただき，御参加いただければ幸甚であります。

なお，密度の高い討論を主体とする会合にするため，参加者総数は30名弱に制限したいと思っております。もし，出席を推薦していただける方がございましたら，幹事まで御連絡下さるようお願い致します。

シンポジウムは，7月17日（火）12：00から7月19日（木）13：00まで，軽井沢にある早稲田大学追分セミナー・ハウスで開催され，23名の参加者があった。参加者は必ず講演する義務を負った。熱心な講演と活発な討論のため，コーヒー・ブレークは有名無実となり，食事の時間まで圧迫された。例によって，夜の討論は連日0時過ぎまで続けられた。この方面の実践的な研究・開発に何らかの形で携っている方々が中心であったことが，このシンポジウムをより実りあるものにした。

本報告では，まずダイナミック・マイクロプログラミングの歴史的概観と展望を詳述し，次に各参加者の講演の題目と内容梗概を紹介する。なお，討論を含めた講演の詳細については，「ダイナミック・マイクロプログラミング」報告集を読んでいただきたい。

2. ダイナミック・マイクロプログラミングの発展

2－1　マイクロプログラムの変更について

microprgramming 方式はComputerのControl Unitをsystematicにdesignする方法として，M.Wilkesによって1951年に提案された。そして，その最初の論文においてmicroprogramを変更することについてつぎのように論じている。

fixed control storeがerasableなものにおきかえられ，必要なときに主記憶から情報を転進されるものとすると，固定した命令コードを全くもたない機械を得ることとなる。プログラマは実際彼自身の要求に合った命令コートを選択することができるであろうし，また，もし必要ならばプログラムの途中でそれを変更することも可能になるであろう。このような機械は多くの魅惑的な可能性を有している。しかし，多分実際に可変コードの計算機の必要性はないだろうといっている。

その後の論文においても，private order codeのsystemの融過性のために生じる混乱に対する心配が強調されている。

2－2　KT pilotの設計試作

1961年筆者らはmicroprogramming方式のComputer KT pilotを試作したが，実験型究用としてmicroprogramの変更可能なものとした。この時点では高速の書き換え可能な記憶装置がなかったので，photo transistor matrix，plug board等を用いてmicroprogramを変更できるようにした。変更できるmicro stepの数が余り多くないので，多倍長の算術演算，乱数の発生パタン処理のルーチンその他microprogramで作成した。

2－3　Stored logic computer

1961年Semarne，Porterにより stored logicの概念が発表され，writable control storeをもった具体的なComputer AN／UYK－1が発表された。更に続いてDatamatinの1964年Febに Stored logicについての論文と具体的な計算機としてTRW－133，PB－440，C－8401が発表されている。

これらはこの時代には特徴のある計算機であったが，WCSとして高速の適当なものが得られなかったので，余り発展をみなかった。

2－4　Grasselliの提案

1962年A.GrasselliはProgram-modifiable Control Unitとしてpath finder memory（PFM）を用いた方式を提案した。PFMのwordはmicro order addressのsequenceから成る。これを利用してControl memory（CM）の読み出しを行う。PFMの内容はプログラム制御によって，WM（working memory）から情報を転送することによって部分的に，或は完全に変更することができる。CMにstoreされているmicro orderは反

series化

復はない。

2－5　Emulation

ＩＢＭ　System／３６０が発表され，Emulationによって，旧機械のプログラムをそのままの形で効率よく処理するのにmicroprogramが用いられた。

2－6　Firmwareの概念

１９６７年Ａ．Oplerは第4世代のcomputerについて論じ，種々の目的のために制御記憶に入れられるmicro-programに対してFirmwareという新語を提案した。

2－7　Dynamic microprogramming

１９６７年Ｍ．Ｊ．Flynn 他はdynamically alterable storage（ comparable read & write performance ）to perform the combinatorial and sequential decoding function of machine controlを用いりmicroprog-rammingについて論じ，更にdynamically microprogrammed processorについて論じている。また，Ｃ．Ｖ．RamamoorthyとM.Tsuchiya はuser-microprogrammable computerについて論じている。

2－8　Microdiagnostics

故障診断技術としてＦＬＴが使われていたが，ＩＢＭ System／360 Model　85でＷＣＳが使用されるようになって，microprogramによってhardwareの部分をtestするmicrodiagnosticが使われるようになり，System／３７０になってから広い範囲に使われるようになった。

2－9　2 level microprogramming

microprogramming方式は制御記憶から順次取り出されるmicroinstructionを利用して制御信号を生成している。特に，直接制御方式の場合には，直ちに制御信号が生成される。vertical microprogrammingの場合にはdecodeその他の回路が必要である。そこでその部分に更に microprogramming の方式を適用することが考えられる。このとき，第２段目は第１段目のmicroprogramのmicroinstructionをdecodeして制御信号を作るだけのもの，と第１段目のmicroinstructionを何stepかに分解してその機能を実行する制御信号を生成するものがある。何れの場合でも第１段のmicroinstructionの情報が第２段目の制御記憶を介さずに，そのまま利用されるものもある。

microprogramはmachine instructionをmicrooperationのseguenceに分解して，これをmicroinstruetionで書き表わしたものであるが，その機能はmicrooperationの順序制御と，制御信号の生成の２つに分けて考えることができる。

この２ level microprogrammingではこの機能を分離している。すなわち，第１段のmicroprogramはmicro-operationの順序制御を行う部分であり，第２段はmicroinstructionをdecodeして制御信号を生成する役割をしている。したがって，第１段の制御記憶はmicroprogramのstep数だけの語数が必要になるが，第２段はmicro-instructionの種類だけの語数があればよい。一般にvertical microprogrammingの場合，microinstructionの語長は短いので第１段の制御記憶は語数が多いがそのbit数は少なくなる。第２段の制御記憶は語長は長いが語数は余り多くとる必要がない。また，第１段，第２段の制御記憶の読み出しはoverlapさせることができるので，２重構造にしたことによる速度の低下は考えなくてよい。

制御記憶は経済的になり，しかも速度は低下しないので，vertical microprogrammingの特長を発揮させるcomputerの一つの行き方であると考えられる。

2－10　高いレベル言語の処理

High level languageで書かれたprogramを直接実行する計算機については古くから論じられている。

一方，計算機の使用が進むと共に high level languageの処理についても種々の方式が考えられるようになった。大別すれば，

Compiler方式，　Interpreter 方式

Compiler－Interpreter方式（ high level languageとmachine codeの中間の表現に変換した programを interpretive に実行する）。

この interpreterをmachine languageのprogram で作成すれば時間的な効率が著しく悪くなるので，これをmicroprogram で作成して処理効率を上げることが考えられる。すなわち，中間表現の programをmachine codeの program によってではなく，microprogram の levelで解釈実行することにより，machine codeの objectprogram に翻訳することなく効率よく処理できる。

この際

中間表現の選び方，

中間表現への変換方式，（ どの levelの program によるか）

などが問題になるであろう。

バローズＢ１７００はこの考え方に沿ったもので，中間表現をＳ－ languageと呼び，各High level languageに対してそれぞれＳ－ languageが用意されて居り，それぞれのＳ－ languageに対するinterpreter がある。Ｓ－ language へ変換する compilerはＳＤＬ（ Software Development Language ）で書かれて居り，ＳＤＬ interpreterにより実行される。

2－11 Microprogram の記述

Dynamic microprogrammingの発展に伴いmicroprogram 作成の効率化が望まれる。microprogram 記述システムとしては，ＣＡＳ，ＭＰＬ，ＭＰＧＳ，ＭＤＳなどが発表されており，バローズＢ－１７００にはＭＩＬ（ microprogram implementation language ）が用意されている。

2－12 むすび

Dynamic microprogrammingについてはmicroprogramming 方式の最初の提案のときから論じられていたのであり，１９６０年代前半ですでに stored logic computerとして作られていた。しかしこの当時はＷＣＳとして適当なものがなかったことと応用面が余り開発されなかったため余り発展しなかった。

最近ＩＣmemoryの実用化に伴い，信頼性も高く速度も速く制御記憶として十分使用できるためＷＣＳが実際的なものとなり，またDynamic microprogrammingの応用面も種々開いてきたので，最近再び注目されるようになったと考えられる。

Dynamic microprogrammingの具体的な応用としてはすでに述べた emulation，microdiagnostics，hi－level languageの処理などの他に

実時間処理への応用，

特殊な処理の高速化

入出力処理，

ＯＳの効率向上

制御用その他特殊目的の計算機

通信制御

その他,

がある。

これらについては今後の研究開発にまつところが大きく，応用面の開発と同時にその目的に便利な機能をもったdynamic microprocessorを開発しなければならない。

3. 各講演について

(1) ダイナミック・マイクロプログラミングの発展　　萩原　宏（京都大学）2節参照

TRW-133

(2) ＭＥＬＣＯＭ－１５３０データプロセッシングシステム　　田中千代治・田淵謹也（三菱電機）

ダイナミック・マイクロプログラミングのはしりとも言うべきストアド・ロジック計算機の代表例として，ＭＥＬＣＯＭ－１５３０の論理設計とソフトウェア体系が紹介された。制御記憶としてコアを用いているため，現時点で見れば色々と批判もあるが，今なお学ぶべき点が多い。

(3) プログラミング言語処理におけるダイナミック・マイクロプログラミング　　杉本正勝（富士通）

プログラミング言語処理に中間言語を採用し，その中間言語に対してマイクロプログラム化されたインタプリタを採用する方式について，中間言語の効用とダイナミック・マイクロプログラミングの速度効率が議論された。

(4) 高級言語マシンの一実験について　　永井義裕（日本電気）

対象システム指向型計算機システムの対象としてＢＡＳＩＣインタプリタをとり上げ，ファームウェア化ＢＡＳＩＣインタプリタとその実験評価システムが述べられた。

(5) Microprogrammingにおけるトピックス　　飯塚　肇（電総研）

マイクロプログラミングに関する最近の話題が，(i)設計言語，(ii)高級言語マシン，(iii)故障診断とモニタリング，(iv)アーキテクチャ，(v)応用，(vi)ハーデストウェア，(vii)その他の面から紹介された。

(6) Virtual StorageとDynamic Address Translationについて　　長谷川昌昭（日本アイ・ビー・エム）

ＩＢＭシステム／３７０におけるvirtual storageの概念と，それをサポートするハードウェア，とくにDynamic Address Translationが具体的に解説された。

(7) ダイナミック・マイクロプログラミングに関するいくつかの考案　　葛山善基（大阪大学）

まず，ダイナミック・マイクロプログラミングの応用として，高級言語向き計算機をファームウェアで実現することに関して，Ｂ１７００の方法を例にあげながら，考察が行なわれた。続いて，富士通のＵシリーズ計算機中のダイナミック・マイクロプログラミング方式を採用する計算機（仮称ＵＸ）を用いて，簡単なバーチャルメモリ・システムを実現させるための検討結果が述べられた。

(8) オペレーティンダ・システムとマイクロプラミンダについて　　亀田寿夫（電気通信大学）

オペレーティング・システムの機能に関して，マイクロプログラムとソフトウェアはそれぞれのような役割を

分担し合うかが具体的に検討され，ダイナミック・マイクロプログラミング導入の長所・短所が議論された。

(9) Microprogrammed System Consoleとその応用の一考案　稲葉延武（日本アイ・ビー・エム）

従来使用されてきた統合タイプの配線論理コンソルに代り，マイクロプログラム可能なミニプロセッサがコンソル機能を受持つようになってきた。このミニプロセッサの詳細が紹介され，ユーザマイクロプログラミング的応用が述べられた。

(10) Dynamic Microprogrammingの応用 ― 診断など　三浦晴久（東芝）

dynamic microprogramingの応用として，ＴＯＳＢＡ Ｃ－５４００／１５０におけるマイクロ診断システムが紹介された。

(11) マイクロプログラム・ジェネレータ　馬場敬信（京都大学）

horizontal typeのマイクロ命令を構成する作業を機械化する方法が具体的に述べられた。このシステムは，まずマイクロプログラム制御方式の計算機を記述しておき，これに対してより高級な言語でアルゴリズムを書いて，これからできるだけコンパクトなマイクロコードを生成しようとするものである。このテーマは諸外国でも発表されておらず，貴重な研究としてその成果が期待される。

(12) 初等関数近似のファームウェア化　小柳　滋（京都大学）

Volderによって考案されたＣＯＲＤＩＣ（ＣＯordinate Rotation ＤＩgital Computer）法は加減算とシフトをデータ転送のみで行なえるので，ファームウェア化に適している。このＣＯＲＤＩＣのアルゴリズムとＨＩＴＡＣ８３５０のＲＣＭ（Relocatable Control Memory）を用いた実験結果が述べられ，実用性が議論された。

(13) マイクロプログラミングのデバギングへの応用について　宇都宮公訓（早稲田大学）

マイクロプログラムの長所を利用して，ソフトウェアデバッグを行なうことが提案され，その具体的手法が述べられた。１つは機械がデバッグモードで走りトレースを行なうことであり，他の１つは，デバッグのため，ハードウェアの制御の変更やスイッチの設定をユーザプログラムで可能にすることである。

(14) パイプライン計算機とマイクロプログラム　所真理雄（慶応大学）

パイプライン計算機における，マイクロプログラムの設計が具体的に議論された。

(15) MultiprogrammingとMicroprogramming　塚本克治（武蔵野通研）

最近のミニコン技術およびＬＳＩ技術の発達などと関連させて，マルチプログラミング方式へのマイクロプログラミング方式の導入が議論された。

(16) ダイナミック・マイクロプログラミングとマイクロ・プロセッサのプログラミング　石田晴久（東京大学）

ユーザマイクロプログラミングの本命と考えられるマイクロプロセッサについて，マイクロプログラムの設計とソフトウェアの開発が議論された。

(17) ダイナミック・マイクロプログラムの活用案　和田英一(東京大学)

ダイナミック・マイクロプログラム計算機でバックトラックとレコード・タイプの処理を行なうことが提案された。ここで言うバックトラックとは，計算機の命令がある時点まで進んだとき，1命令ずつバックすることである。この機能はハードウェア(マイクロプログラム)として組込むべきであり，それが実現すると面白い応用が多いことを指摘している。また，レコードタイプの処理とは，データストラクチャーが宣言されたとき，フィールドに見合ったマイクロプログラムが用意され，目的のフィールドを容易にセレクトしようというものである。

(18) ダイナミック・マイクロプログラミングの応用としての"ダイナミック・マイクロプログラミング・マシンによる情報処理システムにおける機密保護"と"ダイナミック・マイクロプログラミングによるマイクロプログラムの作成の機械化"　安井　裕(大阪大学)

computer systm，data communication system などを含むいわゆる情報システムにおける機密保護をより強力にするための一方式と，講演者等によるプログラムを計算機自身に作らせる実験をベースにしたマイクロプログラムの自動作成方法が紹介された。非常に示唆に富む話であった。

(19) マイクロプログラムによる効率向上と制御記憶のアドレシング　飯塚　肇(電総研)

マイクロプログラミングによる効率向上に関して，文献による例と講演者自身の実験例が紹介された。また，制御記憶のアドレシングについて，マイクロプログラムの入れ換え，ダイナミック・マイクロプログラミングの実施法の検討結果が述べられた。

(20) マイクロ・キャッシュに関する一考案　相磯秀夫(慶応大学)

マイクロ・キャッシュの容量とミスヒット率との関係，制御方式とミスヒット率との関係をミニコンピュータ・レベルで実測した数値と，その結果に対する検討結果が報告された。

(21) マイクロプログラムメモリについて　石井　治(電総研)

各種のマイクロプログラムメモリの長短と，メモリ・ハイアラーキに関する議論が行なわれた。

(22) 図形処理とダイナミック・マイクロプログラム　石田勝則(日本アビオトロニクス)

グラフィックディスプレイ装置による図形処理のうち，(i)拡大・縮少，(ii)平行移動，(iii) 回転，(iv) 投象について時間的な運動を表示する処理をファームウェア化する検討の結果が述べられた。

(23) ファームウェアによる計算機システムの特殊化について　大野直哉(日本電気)

多数のハードウェアあるいはファームウェアで用意されたオプションの中から，自分の応用に適したオプションを任意に選択し，各応用ごとに最適の形にハードウェアおよびソフトウェアを特殊化するための方法がいくつか述べられた。

本 PDF ファイルは 1965 年発行の「第 6 回プログラミング―シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの https://www.ipsj.or.jp/topics/Past_reports.html に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の捜索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者捜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日 〜 2021 年 3 月 19 日
掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程
https://www.ipsj.or.jp/copyright/ronbun/copyright.html