

制御用計算機とマイクロプログラミング

曾 我 正 和†

1. はしがき

最近、制御用計算機においては、マイクロプログラミングを省いて考えることはできなくなってきたり、超小型機にも一般中型機にもマイクロプログラムが使われはじめている。ただし、超小型機と中型機とでは、マイクロプログラムの形態が、対照的な相違をみせている点が面白い。

2. パーチカルマイクロプログラムと 水平マイクロプログラム

超小型制御用計算機において使用されているマイクロプログラムは、いわゆるパーチカルマイクロプログラムである。これは、一般に多い水平マイクロプログラムに対し、マイクロプロセッサのハードウェア構成をできうるかぎり簡単にし、その量を少なくするという立場にたって、マイクロインストラクションの横幅を極力縮めた型である。そのため

1. マイクロコマンドは若干エンコードされたフォーマットをとる、
2. シーケンスコントロールは簡単な方式をとる、こととなり、総合的には
3. マシンインストラクションの実行時間は長くなるが、ハードウェアは最小になる、という点に特長がある。

パーチカルマイクロプログラム式の超小型制御用計算機では、ユーザープログラムそれ自身がマイクロプログラムにおとされ、実行されることもある。その場合この種の計算機は、自らはコンパイルはもちろん、アセンブルする能力をもっていないのが普通なので、ユーザープログラムは、別の上位計算機によってマイクロプログラムレベルまでおとしてもらうことになるか、もしくはユーザーがじかにマイクロプログラミングを行なう。このようにユーザープログラム自身がマイクロプログラムになるケースは、プログラムの書き換えをほとんど必要としない固定プログラムシステムに適している。制御用システムとしては、シーケンスコントローラの種類がこれにあたる。

最近の素子技術の進歩はめざましいものがあり、パーチカルマイクロプログラム方式でも総合処理速度はどんどん向上してきている。従って、単に制御用に限らず、各種の小型専用装置としての応用分野が拡大しつつある。

一方、数十ビットから場合によって百ビットを越す水平方向に横長なフォーマットをもつマイクロインストラクション方式は水平マイクロプログラムと呼ばれる。この方式では、各種のデータバスや演算ゲートに対して直接制御信号を発するマイクロコマンドフィールドが並列に用意されており、マシンインストラクションをできるだけ迅速に実行することに重点がおかれている。

制御用計算機においても、中型機はこの方式をとっている。前述のパーチカルマイクロプログラムが、固定プログラム、専用小型機、に適しているのに対し、水平マイクロプログラムは、融通性と高速性を要求される中型機大型機に適している。

3. マイクロプログラムの制御用計算機への 応用

制御用計算機に対する要件を満たすために、次のようにマイクロプログラムが応用されている。

3.1 高信頼性

高信頼性とは、単的にいって、システムの無故障稼働時間がより長いことであり、MTBFがそのメジャーである。MTBFは、ハードウェアレベルでは、その計算機を構成している素子の一品一品の故障率、それらの使用状態（周囲温度、ピーク電圧等）およびそれらの部品の総数量とによって決まってくる。

ところで、マイクロプログラムは、CPUの部品総数量に非常に深い関連をもっている。何故なら、従来の全ハードワイアドロジック型の計算機に比して、マイクロプログラム計算機では、そのCPUハードウェア構成が

- ハードワイアドロジック部、
- マイクロプログラム部（制御メモリ部）

の二者に分化する。前者は、全ハードワイアドロジック式の場合に比べて、複雑なシーケンス回路が全く不

† 三菱電機鎌倉製作所計算機工場

要で、レジスタとデータバスがほとんどを占める単純明瞭なものとなる。この結果、そのほとんどを MSI にて構成しうる。一方、後者は、ROM または RAM に実装される。従って、この部分には LSI を採用しうることになり、ハードワイヤドコントロールの場合のこの部分に相当するシーケンス制御回路に比し部品点数は激減する。

現状においては、中規模以上の高速計算機の演算ロジック部を LSI 化することには、種々の困難があり、ROM とか RAM とかのレギュラーな構造のもののみが LSI 化されている。この事情がつづく限り、マイクロプログラム技術を更に拡大推進することが計算機の LIS 化への最も重要なステップでありつづける。制御用計算機の立場からみると、この点がマイクロプログラム方式をとるべき最大の理由である。

3.2 安全性

制御システムでは、信頼性の次に、或いは信頼性と同等に重視されるのが安全性（修理または点検のしやすさ）である。システムによっては、そのダウンが長びくことは、単に経済的損失を招くだけでなく、人命に影響を与えたり社会不安を招くものすらある。

従来統計では、障害の発生率は、CPU よりも、圧倒的に入出力機器に多い。しかしながら、一旦障害が発生した場合、その修理時間が長びいて問題を起すのは、CPU とか I/O チャンネルに多い。

そこで、スタート・スモール方式といわれる確保前進型的手法を採用できるマイクロ診断が積極的に進められている。

3.3 瞬時故障対策

信頼性、安全性を云々するときに忘れてはならないのが瞬時故障対策である。瞬時故障とは、ごく短時間の障害を起したあと、自然復旧してしまう現象である。

このとき、診断プログラムはほとんど役に立たない。何故なら、相手は瞬時故障を引き起したあと姿をくらましており、システムは自然復旧しているため、プログラムは正常にパスしてしまうからである。

これに対抗するため考えられている手段に次の二つがある。

- 自動再試行、
- 故障状態の自動ログアウトおよびアナリシス。

いずれに対しても、まず必要なのは、時間的および場所的分解能の高い、きめ細かなエラー検知回路である。これによって異常を即刻キャッチすれば、そのあとは、ただちに CPU の制御動作の流れを切替え、そ

の時点での各種ステータスのログアウト、および適当なポイントまでバックしてデータを復元して再試行を行うことになる。マイクロプログラム式 CPU では、このように制御動作を切替え、ログアウトしたり再試行したりすることは、単にプログラムステップの増加によって達成できる部分が多く、ハードワイヤド CPU に比べて大いに実現しやすい。（もちろんセーブレジスタ等のハードウェア増加も不可欠である。）

3.4 プログラム生産性

制御用計算機分野でも、アプリケーションプログラムはユーザー自身が作成するのが主流になってきており、どんどん高位言語が使われている。この場合、制御用アプリケーションとしては、オブジェクトプログラムの速度が問題であり、マイクロプログラムによって、アプリケーションに合ったマクロ命令を与える点は大きなメリットである。

3.5 機種連続性

電子計算機一般に、技術革新が非常に急テンポで進行しているが、実用面では非常に保守的な一面がある。すなわち、過去に蓄積したソフトウェアの量が膨大になり、これが巨大な慣性をもっているため、急激な変化をなすえない。制御用計算機の場合は更にこの傾向が強い。それは、ソフトウェアの問題以外にハードウェアにおいて、プラントとの接続仕様の問題および、保守の重要性にからんで保守員教育の問題があり、いずれも変化を嫌う性質をもっている。従って、かりに新機種を導入する場合でも、常に過去の既設システムとの互換性をもっていることが必須条件となる。こうなると、新機種の持ち込みようがないかに思われるが、マイクロプログラム技術は、少くとも古い機種とのソフトウェア互換性を失わずに、最新の素子や、新しい性能を導入することを容易にしてくれる。CPU のみならず、チャンネルや I/O 制御装置にもマイクロプログラムを適用することにより融通性が增大することももちろんである。

3.6 機能単位の分割

マイクロプログラム CPU では、バスとかアダーのように全機能に共通に使用される部分がハードワイヤドロジックとして存在する一方、各命令、各動作、個個に対応する制御信号はすべてマイクロプログラムとしてコントロールメモリ中に格納されている。しかも各機能別に各ルーチンとして分離されているため、制御動作の把握がきわめて容易である。このことは、開発の容易さ、保守の容易さ、ログアウトアナリシスの

容易さ等につながる。

4. マイクロプログラム方式における注意点

以上では、マイクロプログラムが制御用計算機にもたらすメリットを見てきたが、実際にこれを採用する場合には、以下に述べるように若干の注意を要する。

(1) デバッグの問題

マイクロプログラムを格納する素子として ROM を選ぶ場合、その内容変更に関係が手間がかかるので、開発初期のマイクロプログラムデバッグに際しては注意を要する。対策としては、

- 別計算機の上で事前にシュミレーションする、
 - デバッグ期間中は ROM でなく RAM を使う、
- 等が考えられる。

(2) スペアパーツの問題

制御用計算機では MTTR に関連して、スペアパーツの重要性が高い。ここでもマイクロプログラム素子として ROM を選ぶ場合は、ROM を搭載する何枚かのプリント基板パッケージが、おのおの異った記憶

パターンをもつため、それぞれ個別にスペアパーツを要することになり、スペアパーツコストが上昇する。

(3) 制御メモリ素子の選択

マイクロプログラム格納メモリとして ROM を選んだ場合は、上記 (1)、(2) のような注意を要する。一方、RAM を選び、WCS 方式をとった場合は、上記 (1)、(2) の問題はないが、そのかわりに制御メモリ自身へのロードパスというかなり大きなハードワイヤドロッジックを必要とする。

制御用計算機として忘れてはならない瞬時停電の対策も、WCS に対しては十分考えておかねばならない。しかし、WCS にはマイクロプログラムの入替えが可能になるという大きな利点がある。両者それぞれに特長があり単純に比較はできない。一部に ROM を使い、その部分のマイクロプログラムルーチンにて残りの RAM へのロードとか、IPL とかをやらせる方式もあり、これは両者の特長をともに生かそうという一つの工夫である。

(昭和 48 年 3 月 12 日受付)