

システム370モデル155のマイクロ診断機能

松岡 亜樹 (日本アイ・ビー・エム株式会社)
木村 博 (同上)

1. 序

最近の3.5世代と呼ばれる計算機は、ハードウェアの飛躍的な発展に伴い、大規模となり、オンラインあるいはマルチプログラミング等、複雑かつ多様なアプリケーションに使用されている。

計算機の応用の高級化に伴い、ハードウェアの誤動作が与える影響は大きく、有効な故障診断プログラムは、計算機の設計において考慮されるべき重要な部分となっている。

マイクロプログラム制御の計算機の診断プログラムは、マイクロ診断プログラムとマシン・インストラクションで構成されたものに大別される。マイクロ診断プログラムの利点は、ハードウェアの部分的な診断が可能であるため、基礎となるハードウェアから順次診断がすすめられ、プログラムによる故障ユニットの指摘が容易となり、後者の診断プログラムに比べ故障回復時間が極めて少なくなり、また効率よいプログラミングが可能となることである。

本稿では、システム370モデル155における、マイクロ診断プログラムの診断メカニズムを述べる。

2. マイクロ診断のハードウェア

図1に示されるシステム構成において、マイクロ診断のため特に必要となるハードウェアは、EXTERNAL DEVICE、LOG BUFFER および SERVICE ADAPTER と呼ばれるユニットから構成され、それぞれ下記の機能を有する。

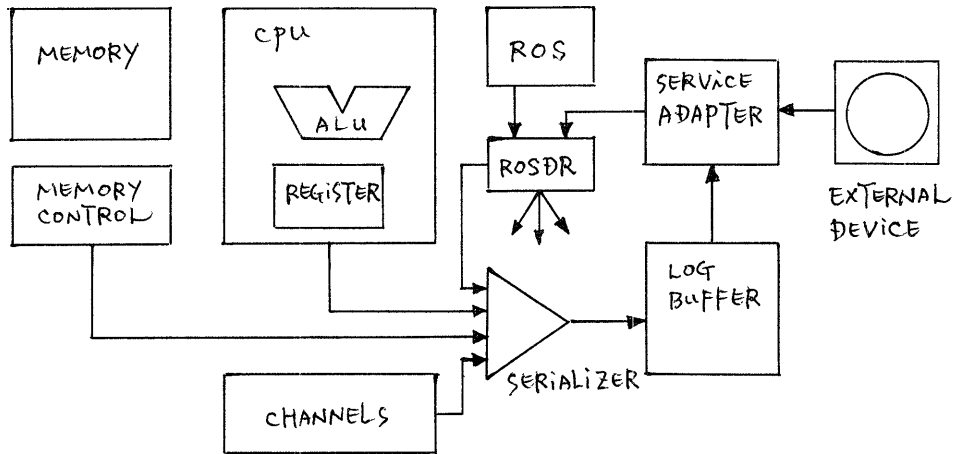


図1. システム構成

- (a) EXTERNAL DEVICE: マイクロ診断プログラムの入力装置となり、ソリッド型磁気ディスク(約60キロバイト/ディスク)より、診断プログラムを構成する制御ワードおよびデータを読み込む。
当ユニットは、診断プログラムのみならず、WCS (Writable Control Storage) を持つシステムに対しては、マイクロプログラム入力装置としても利用される。(例 システム 370 モデル 135)
- (b) LOG BUFFER: ハードウェアの状態即ちシステムの論理回路のオン、オフ状態が貯えられるメモリーであり、この内容が診断プログラムのデータと比較される。システム稼働時において、このメモリーは、故障が起きた場合のハードウェアの状態が記憶され、プログラマーによる故障回復プログラムやマシン・インストラクションレベルの診断プログラムのデータとして使用される。
- (c) SERVICE ADAPTER: EXTERNAL DEVICEからの入力がバイト単位に編成され、まず制御ワードとデータに識別される。最も重要な機能は、制御ワードに従いデータを ROSDR に転送し診断用マイクロインストラクションを構成することおよび LOG BUFFER の内容とデータ (Expected Data) を比較することである。更に、マシン・インストラクションレベルの診断プログラムを EXTERNAL DEVICEより読み込み、主記憶装置にロードすることも可能である。

以上3つのユニットのロジック・ダイヤグラムが図2に示される。

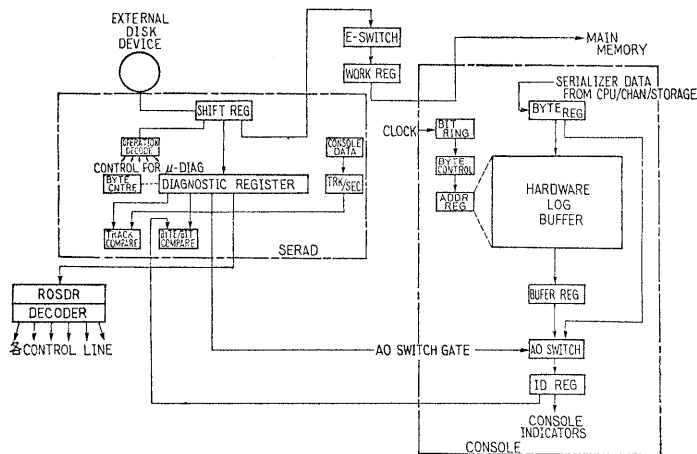


図2. マイクロ診断のハードウェア

EXTERNAL DEVICE からシフト・レジスタに入りバイト単位に編成された情報は、ビット7が1であれば制御ワードと識別され、図3に示されるハードウェアの制御に使用されるが、ビット7が0であれば、データとみなされ診断レジスタに転送される。

診断レジスタに貯えられたデータは、制御ワードに従って、ROSDRに送られマイクロ・インストラクションを構成するデータあるいは、LOG BUFFERよりIDレジスタに移された内容と比較されるデータとして使用される。

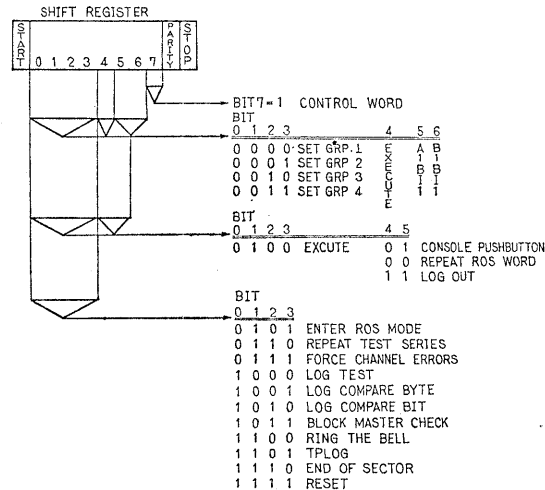


図3 制御ワード

3. 制御ワード

図3に示されるシフト・レジスタのビット7が1であれば制御ワードと識別され残りのビット0~6がデコードされハードウェアのコントロールを司る。このうち主要な機能について述べる。

- (a) SET GRP 1~4: 診断レジスタの内容をROSDRにセットする。マイクロ・インストラクションは72ビットより構成されるため、診断レジスタ(21ビット)の内容は4回に分けてROSDRに送られる。
- (b) EXECUTE: ROSDR内のマイクロ・インストラクションが実行される。
- (c) LOG OUT: ハードウェアの状態をLOG BUFFERにストアさせる。
- (d) LOG COMPARE BYTE/BIT: LOG BUFFER(512バイト)内の任意の1バイトあるいは1ビットが診断レジスタの内容と比較され等しくなければ診断プログラムは停止する。
- (e) LOG TEST: 診断レジスタのビット0~9でアドレスされたLOG BUFFERの1バイトがIDレジスタに送られる。

(f) RESET: ハードウェアの状態をリセットする。

(g) ENTER ROS MODE: 診断プログラムのコントロールから ROS コントロールに切り換えられる。この制御ワードは通常 EXTERNAL DEVICE からのデータを主記憶装置にロードする場合に用いられる。

ここで簡単な例として、ROSDRを構成する 72 ビットのレジスタの診断方法を述べる。

診断パターンの例

FEFEFE 01112131 3C1EF8 8191
グループ1 グループ2 グループ3 グループ4

グループ1: 診断レジスタの 21 ビットがすべて 1 にセットされる。

グループ2: 診断レジスタの内容が ROSDR に入れられ 72 ビットすべてが 1 にセットされる。

グループ3: 診断レジスタに、比較のためのデータが次のようにセットされる。

ビット $\begin{array}{cccccccccccccccccccc} 0 & & 5 & & 10 & & 15 & & 20 \\ \hline 0 & 0 & / & / & / & / & 0 & 0 & 0 & / & / & / & / & / & / & / & 0 & 0 \end{array}$
LOG BUFFER のアドレスを示す 比較されるパターン

グループ4: 制御ワード "81" は診断レジスタのビット 0~4 と 6~9 でアドレスされた LOG BUFFER の 1 バイトを ID レジスタにセットする。この例では、ROSDR ビット 0~7 の状態が ID レジスタにロードされることになる。

制御ワード "91" は、診断レジスタのビット 10~18 で示されるパターンと ID レジスタの内容を比較する。即ち ROSDR ビット 0~7 とパリティビットがすべて 1 であることがチェックされている。

比較の結果、等しくなければ ROSDR を構成しているレジスタ (ビット 0~7) が正しく動作していないことを意味し、診断プログラムは停止する。

このような診断メカニズムによるマイクロ診断プログラムの実例を次節で述べる。

4. マイクロ診断プログラムの実例

マイクロ診断では、前述したように ROSDR にデータを入れマイクロ・インストラクションを構成して実行させるので、診断辞書に書かれるマイクロ・ステートメントは、通常のマイクロプログラム・ステートメントと同じ形式で記述され、診断プロ

グラムのユーザーにとって解説が容易となっている。

図4の実例中、BLK MASTER CHECK、REPEAT、LCBYなどのマイクロ診断用制御ワードを除き、他は通常のマイクロテストメントの記述法となる。

この例は、ALU(演算ユニット)周辺の4バイト・データパスの診断をする部分で、ALUに直結する1つのレジスタ(4バイト)からALUを経由して他のレジスタにデータを送り、結果が正しいかどうかを診断されている。

“LCBY”と表示されているのが、LOG COMPARE BYTEの制御ワードを意味し、その右に示される2つのパターン、例えば“10F 000”はLOG BUFFERのアドレス“000”の内容が“10F”となることを期待している。“10F”は、パリティビットが“1”で、データが“0F”ということの意味する。

次に示される四角で囲まれた部分が、比較の結果一致しない場合の診断辞書部にあたり、最も可能性の高い故障ユニット(FRU-Field Replaceable Unit)のロケーションと、そのロジック・ダイヤグラムのページが示される。

通常、カスタマー・エンジニアは、この部分を調べ、相当するユニットを交換することになる。

5. 結び

以上システム370モデル155のマイクロ診断プログラムを紹介したが、現在使用しているマイクロ診断プログラムは、90%の故障発見率を目標としており、それに近い実績を示している。

モデル155では、ROSが使用されているため本稿に述べた方法が取られているが、WCSを持つ計算機では、診断プログラム自体をWCSにロードし、マシンサイクルで実行、診断することが可能で、更に診断精度を上げることができる。この方法は、システム370モデル135、145、158等で使われている。