マイクロコンピュータ　60－4
とワークステーション
（１９９０．５．８）

ＭＰＵテスト支援機能の一考察

石田　光*　　平尾　繁晴**
*株式会社 東芝　府中工場，**株式会社 東芝　システム・ソフトウェア技術研究所

多くの高級ＭＰＵではテスト用の命令や機能を持たせることが一般化してきている。シングルステップなどはその一例である。　ＭＰＵは一般に２つの実行モード（１.ノーマルラン・モード，２.シングルステップ・モード）を持っている。

本論文では、さらに２つの実行モード（３.ブランチストップ・モード，４.スタック操作ストップ・モード）の追加を提案し、これらの実行モードの効果について検討した。

A STUDY OF TESTING FUNCTION IN MICROPROCESSORS

Mitsuru Ishita*　Shigeharu hirao**

*TOSHIBA Corporation　Fuchu Works
**TOSHIBA Corporation　SYSTEM & SOFTWARE Engineering Laboratory

*1,Toshibacho,Fuchu,Tokyo,183 Japan
**70,Yanagicho,Saiwai-Ku,Kanagawa,210 Japan

A lot of high performance MPUs have various debug support function. Single step execution is one of them.

MPU has generally two execution modes. (1.Normal run mode, 2.Single step mode)

In this paper,We propose to add two execution modes further. (3.Branch stop mode, 4.Stack operation stop mode)

And we examined efficiency of these modes.

１．はじめに

一般にＭＰＵの内部状態を観測するには、その外部状態（端子に入出力される信号など）から推察するしかない。　このテストの際の「チップ外部からの観測性の悪さ」を解決するために、多くの高級ＭＰＵではテスト用の命令や機能を持たせることが一般化してきている。

シングル・ステップなどはその典型といえ、１ステップ毎にＭＰＵの内部状態を観測できるメリットを持つが、ソフトウェアテストにおいてはそのリアルタイム性の悪さが致命的となることもある。また、ステップ実行させずに指定した停止条件が成立するまで、通常の実行速度で動作させる機能を持つものもあるが、この場合は逆にリアルタイム性は保証されるが、観測性は悪い。

今回は、このリアルタイム性とＭＰＵ内部状態の観測性のトレードオフの問題を解決するのに有効と思われるＭＰＵのテスト用機能を考案して、同時にその機能を持つＭＰＵを用いたテスト装置について検討したので報告する。

２．従来の問題点

ＭＰＵの内部状態は直接観測することはできず、ＭＰＵの各端子の信号を観測するなどしてその動きを推察することになる。　その「内部状態の観測性」の悪さを解消すべく多くの高性能ＭＰＵでは、

１）シングル・ステップ
　１命令毎にプログラムの実行にブレークがかけられる機能。
　ＭＰＵの内部状態の観測性が良い。

２）ブレーク
　プログラムを設定した停止条件が成立するまで、実行させられる機能。
　停止条件としてはブレークしたいアドレスだけでなく、メモリへのアクセスなどを持つものもある。

３）トレース
　分岐命令などでプログラムの実行順序が変化した場合、その変化（分岐したアドレスなど）を記録する機能。
　プログラムの実行順序を確認するのに有効。

などのテスト用機能を持っている（参考文献[1]～[3]）。　しかし、これらの機能にも問題がある。

まずシングル・ステップは、１命令実行毎にプログラムの実行を停止するのであるから、当然リアルタイム性は大幅に低下する。

次にブレークは、リアルタイム性は良いが実行途中のＭＰＵの内部状態の観測性は悪い。　また、内部条件だけでなくプログラムの実行順序の観測性も悪い。

最後にトレースは、ＭＰＵ内部にはトレース情報を記録する大きなトレース・メモリを置くことはまだ現実的でなく、数箇所の変化点の情報を記録するのみである。

３．テストに有効なＣＰＵの機能の検討

２章で述べた従来の機能によるプログラムの実行形態を、ここでもう一度整理する。

従来の実行形態は、大きく以下の二つのモードに分けられる。

１）ノーマルラン（通常実行）モード

２）シングルステップモード

この両者は、リアルタイム性とＭＰＵ内部状態の観測性のトレードオフの問題を持っている。

そこで、これらに以下の２種類のモードの追加を考えた。

３）ブランチストップモード

４）スタック操作ストップモード

まず、ブランチストップモードはプログラムの実行順序に変化があった場合に停止する機能である。

ここでいう実行順序の変化とは、分岐命令やサブルーチンのコール命令，割り込みの発生，サブルーチンや割り込みからのリターン命令などの実行によるものである。

この機能により停止した時点の変化点の情報（分岐点のアドレスなど）から、プログラムの実行順序を完全にトレースできる。　この機能は数種のＭＰＵで似た機能や、外付けのＭＭＵで実現可能なものがあるが、このモードでは変化前のアドレスを保持するバッファをＭＰＵ内部に持たせることを考えている。　キャッシュの内蔵などを考慮すると最終的にはＭＰＵの内部にすべて持つべきと考えている。　また、実行順序の変化が起こらない間は、通常実行されているので、リアルタイム性はシングルステップモードに比べて当然良い。　この機能は、リアルタイム性を必要以上低下させることなく、プログラムのテスト網羅率を測定したい場合などに有効である。

次に、スタック操作ストップモードはスタックポインタの内容が変化した場合に停止する機能である。　つまり、サブルーチンのコール命令，割り込みの発生，サブルーチンや割り込みからのリターン命令などの実行によって停止する。

この機能により停止した時点の変化点の情報（分岐点のアドレスなど）からサブルーチンの呼び出し，割り込みの発生は完全にトレースできる。

この機能は、ブランチストップモードよりさらにリアルタイム性の低下が少なく、また、各時点での時間を記録できるようにしておけば、ルーチンごとの実行時間を測定することができる。

４．テスト装置の検討

４．１　従来のテスト装置

第１図は従来のソフトウェアテスト装置のブロック図である。

ユーザは端末から被テストプログラムの実行を制御できる。　被テストプログラムの実行モードには、以下の２種類がある。

１）ノーマルランモード

制御装置は、端末からノーマルランの指令が与えられると、ＭＰＵの実行モードをノーマルランモードに設定し、スタート信号を発生させてＭＰＵに実行を開始させる。　次に端末から停止の指令が与えられると、ストップ信号を発生させてＭＰＵの実行を停止させる。

ＭＰＵ停止後、制御装置は端末からの指令に応じて、ＭＰＵの内部状態を端末に表示したり、制御装置内のタイマにより実行時間を求めて端末に表示する。

2）シングルステップモード

制御装置は、端末からシングルステップの指令が与えられると、ＭＰＵの実行モードをシングルステップモードに設定し、スタート信号を発生させてＭＰＵに実行を開始させる。 シングルステップモードの場合は、ＭＰＵは１命令実行後、自動的に停止するので、停止したＭＰＵの内部状態を端末に表示したり、内部状態バッファに格納する。

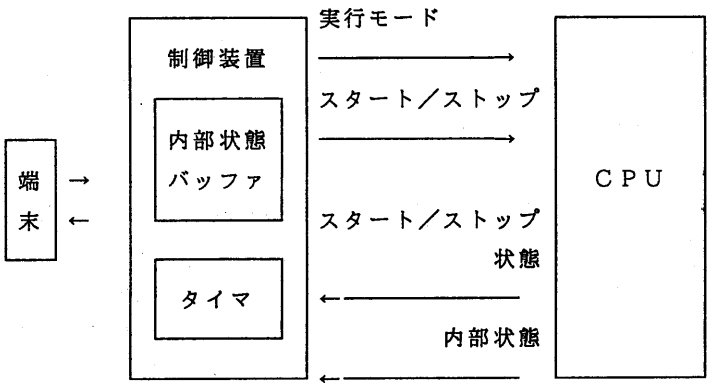
さらにシングルステップモードには、以下の２種類がある。

２－１）１命令実行モード

１命令実行毎にＭＰＵの内部状況を端末に表示するモード。

２－２）ｎ命令実行モード

１命令実行毎にＭＰＵの内部状況を内部状態バッファに格納しておき、被テストプログラムの実行終了後に格納しておいたＭＰＵの内部状況を端末に表示するモード。



第１図　従来のテスト装置

４．２　新実行モードを追加したテスト装置

第３章で述べた以下の新実行モードを追加したソフトウェアテスト装置は、従来のテスト装置とそのブロック構成において基本的に変更はない。 ただ、実行モード選択信号が、従来の２種類と新実行モード２種類、つまり４種類から選択可能になっている。

１）ノーマルランモード　　（従来の実行モード）

２）シングルステップモード　　（　　〃　　）

３）ブランチストップモード　　（新実行モード）

４）スタック操作ストップモード　（　　〃　　）

第２図は今回のソフトウェアテスト装置に使用されるＭＰＵの概要（実行モード追加に関連する部分のみ）を示している。 このＭＰＵには、新実行モード２種類を実現するために、以下の回路の追加が必要である。

ブランチストップモード関連

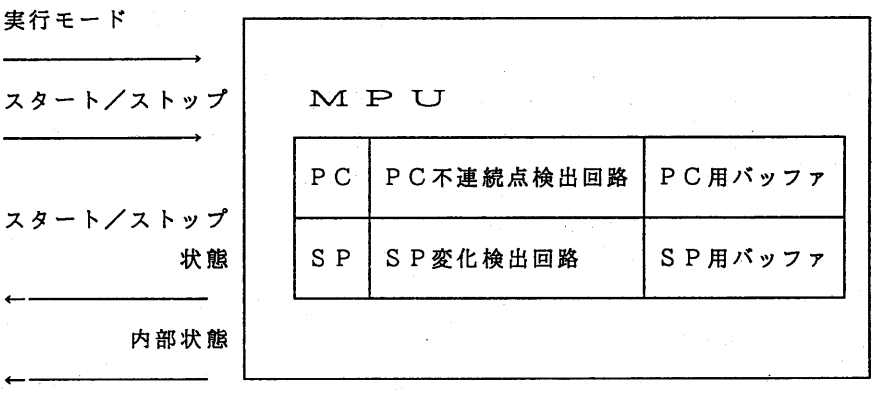
・ＰＣ（プログラムカウンタ）用バッファ

・ＰＣ不連続点検出回路

スタック操作ストップモード関連

・ＳＰ（スタックポインタ）用バッファ

・ＳＰ変化検出回路



第２図　新実行モードを持つＭＰＵ

このＭＰＵの新実行モード時の動作について説明する。

まず、ブランチストップモードは、ＰＣ不連続点検出回路によってＰＣが不連続に変化した場合に停止する。　この際、変化する直前のＰＣの内容がＰＣ用バッファに格納される。　ＰＣ用バッファと（変化直後の）ＰＣを調べることにより、プログラムの実行を完全にトレースできる。

ここでの「不連続な変化」とは、分岐命令などによってＰＣの値が強制的に書き換えられた場合を指す。　現在、いくつかのＭＰＵが持つ「トレース」機能では、このようなプログラム実行の不連続点のＰＣの値を、最新の二カ所ほど記録するのが主である。　（プログラム実行は停止されない。）

次に、スタック操作ストップモードは、ＳＰ変化検出回路によってＳＰが（サブルーチンコール，割り込み発生等によって）変化した場合に停止する。　この際、変化する直前のＳＰの内容がＳＰ用バッファに格納される。

ＳＰ用バッファとＳＰ（変化直後のＳＰ）を調べることにより、サブルーチンコールや割り込み発生を完全にトレースできる。　例えば、スタックが増加方向へ変化したか、減少方向へ変化したかが判れば、サブルーチンをコールしたのか、逆にリターンしたのかが判断できる。

５．実行モードの比較

４種類の実行モードを、以下の項目について比較してみた。　第１表がその結果である。

１）リアルタイム性

説明：　ノーマルランにどれだけ近いか。　これが悪いと微妙なタイミングが必要とされるようなソフトウェアのテストには用いることができない。

結果：　どのモードも実行中にブレークをかけているため、ノーマルランに比べてリアルタイム性は悪い。　ブレークがかかる頻度で考えた場合に、表１の結果となった。

２）ＭＰＵ内部観測性

説明：　ＭＰＵの内部情報（レジスタの内容など）の観測性。

結果：　ＭＰＵの停止した状態でその内部情報を見ているため、当然リアルタイム性とは逆の結果となる。　しかし、ブランチストップもスタック操作ストップもシングルステップと比較しては、大幅に観測性は悪い。

３）実行トレース

説明：　被テストプログラムの実行を完全にトレースすることが可能か。
これはテスト網羅率の測定などが行えることを意味する。

結果：　シングルステップは勿論、ブランチストップも可能。　それ以外のモードでは不可。

４）ルーチン単位のトレース

説明：　ルーチンの呼び出し関係や割り込みの発生などをトレースすることが可能か。

結果：　ノーマルラン以外は可能である。

	リアルタイム性	ＭＰＵ内部観測性	実行トレース	ルーチン単位のトレース
ノーマルランモード	◎	×	×	×
スタック操作ストップモード	○	×	×	◎
ブランチストップモード	△	×	◎	◎
シングルステップモード	×	◎	◎	◎

表１　実行モードの特徴比較

4種類の実行モードを、リアルタイム性と観測性について、並べてみると以下の様な結果になる。

リアルタイム性		観測性
（良）	ノーマルランモード	（悪）
↑	スタック操作ストップモード	↑
↓	ブランチストップモード	↓
（悪）	シングルステップモード	（良）

この結果より、新しく考案した2種類の実行モードは、従来のノーマルランモードとシングルステップモードとの中間的特徴を持つことが判る。

6．考察

各実行モードの特徴比較の結果より、各モードの具体的な使用例を考えてみた。

1）ノーマルランモード

他のモードではテストできないようなリアルタイム性重視のテスト向き。

また、他のモードでのテスト後の最終テスト。

2）スタック操作ストップモード

ルーチン単位の実行順序や実行時間の計測。

被テストプログラムの効率化のボトルネックとなっているルーチンの発見などに必要な情報を得ることができる。　また、テスト時間も比較的少なくて済む。

3）ブランチストップモード

実行順序の確認。テスト網羅率の測定。

被テストプログラムの実行順序を完全にトレースできるため、不具合箇所の発見に（シングルステップほどではないが）有効である。　特に高速にテスト網羅率の測定を行いたい場合に有効である。

4）シングルステップモード

被テストプログラムの不具合箇所の発見。

ＭＰＵの内部状態を1命令実行ずつ完全にトレースできる。　しかし、テスト時間が増大するので、他のモードで不具合箇所の存在する範囲を絞ってから使用すると有効である。

特に、被テストプログラムの実行解析にこれらの実行モードを利用することを検討した場合、新実行モードでは

ルーチンの使用回数

ルーチン内の消費時間

テスト網羅率

ｅｔｃ．．．

などの実行解析に必要な情報を、最低限のリアルタイム性の低下で得ることができ、大変有効である。

ここで「リアルタイム性の低下」について、以下のような仮定を設けて検討した。

・各モードで停止した時点で、実行の解析に必要な処理を行なうのに、２０命令必要だとする。

（つまり、1回の停止で２０命令余分にかかるものとする。）

この仮定で、各実行モードでの実行命令数をノーマルランモードの場合と比較すると次のようになる。

1）ノーマルランモード

「リアルタイム性の低下」はない。

2）スタック操作ストップモード

サブルーチンコールなどの命令の出現頻度によるが、DHRYSTONEベンチマークテストのプログラムでは２．６％の出現頻度であるので、実行命令数は１．５２倍となる。

3）ブランチストップモード

2）同様に分岐命令の出現頻度によるが、DHRYSTONEベンチマークテストのプログラムで７．８％の出現頻度であるので、実行命令数は２．５６倍となる。

4）シングルステップモード

１命令ごとに実行を停止するため、実行命令数が２１倍になり、一番「リアルタイム性の低下」が大きい。

（注）DHRYSTONEのプログラムはＣ言語で書いたものをＴＸ１Ｃコンパイラでコンパイルしたものを使用。

結論としては、新しく考案した２種類の実行モードは、従来のノーマルランモードとシングルステップモードとの中間的特徴を持ち、これらを加えることでソフトウェアテストの際により最適な実行モードの選択が可能となり、テスト効率の向上が期待できる。

７．おわりに

高機能ＭＰＵのテスト用機能を検討して、従来の実行モードであった「ノーマルランモード」と「シングルステップモード」に、新しい概念である「ブランチストップモード」と「スタック操作ストップモード」を追加することを考え、その有効性を検討した。

その結果、今回考案した２種類の実行モードは、いままでのリアルタイム性と観測性のトレードオフについて、その選択の幅を広げることにより、テストの効率化に有効であると考えられる。

ＭＰＵのテスト用機能には、今回検討した実行モードの選択だけでなく、実行のトレースを支援するために、分岐したアドレスや分岐前にシーケンシャルに実行された命令数のカウンタを持つものもある。　今後もテスト項目とそれに必要な情報を与えるＭＰＵの機能について検討していく。

［参考文献］

［１］南 宗宏：32ビットマイクロプロセッサ入門，ＣＱ出版社（1986）

［２］Misao Miyata,Hidechika Kishigami,Kosei Okamoto
"The TX1 32-Bit Microprocessor: Performance Analysis, and Debugging Support",IEEE MICRO, Vol.8,No2,pp37-46 (1988)

［３］W.B.Surjanto：80386の使い方，オーム社（1987）