

マイクロコンピュータ用会話形テスト支援システム “HITS”

Interactive Cross Testing System for Microcomputer Software “HITS”

マイクロコンピュータの普及拡大は目覚ましいものがあり、最近ではシステム製品への適用が急増しつつある。これに伴い、ソフトウェアの大規模化、複雑化が急速に進み、従来のマイクロコンピュータ用ソフトウェア開発の在り方を根本的に改革する必要が生じた。日立製作所では、マイクロコンピュータ、特に16ビット系の大規模ソフトウェア開発環境の構築を目的とし、数多くのソフトウェア開発支援システムを実用化してきた。

ここで述べるシステムは、ソフトウェア開発で多くの期間と労力を要するテスト作業の効率向上をねらったもので、HITAC Mシリーズホストコンピュータ上で実用に供している。本システムは、多くの製品ソフトウェアの開発に適用されており、品質、開発期間の短縮などに大いに寄与している。

黒崎 徹* Tôru Kurosaki
本田明徳* Akinori Honda
増井光幸* Mitsuyuki Masui
中所武司** Takeshi Chûsho

1 緒 言

ソフトウェアの品質及び生産性を飛躍的に向上させるためには、ソフトウェアの設計から保守までのライフサイクルを通して、一貫性のあるソフトウェア開発環境を実現させることが重要である。

この考え方に基づいて実用化を行ってきたのが、16ビットマイクロコンピュータ“HMCS68000”(Hitachi Microcomputer System以下、単に68000と略す。)用統合化ソフトウェア開発支援システム^{1), 4), 7)}である。

このシステムの中の一つとして位置付けられているのが、会話形テスト支援システム“HITS”(Highly Interactive Testing and Debugging System)^{2), 5)}である。HITSは、プログラムモジュールの単体テストからシステムテストまでを系統のかつ効率的に行なえる豊富な機能をもつシステムである。

本システムの実用化マシンとしては、大規模ソフトウェアの開発に対応できること、高いスループットが得られること、

豊富な入出力機器が容易に使えることなどから判断して、HITAC Mシリーズをクロスホストコンピュータとして採用した。

本報では、HITSの開発のねらいと特徴、機能構成及び適用成果について述べる。

2 HITSの実用化方針

2.1 HITSの概要

HITSは、HITAC Mシリーズホストコンピュータ上で“68000”用ソフトウェアのテスト及びデバッグを行なうことができる。これにより、図1に示すようにプログラム編集からテスト十分性評価に至るまでの一連のソフトウェア開発作業をホストコンピュータのTSS(Time Sharing System)環境下で効率的に行なうことができる。

HITSの機能としては、テスト機能、デバッグ機能、プロ

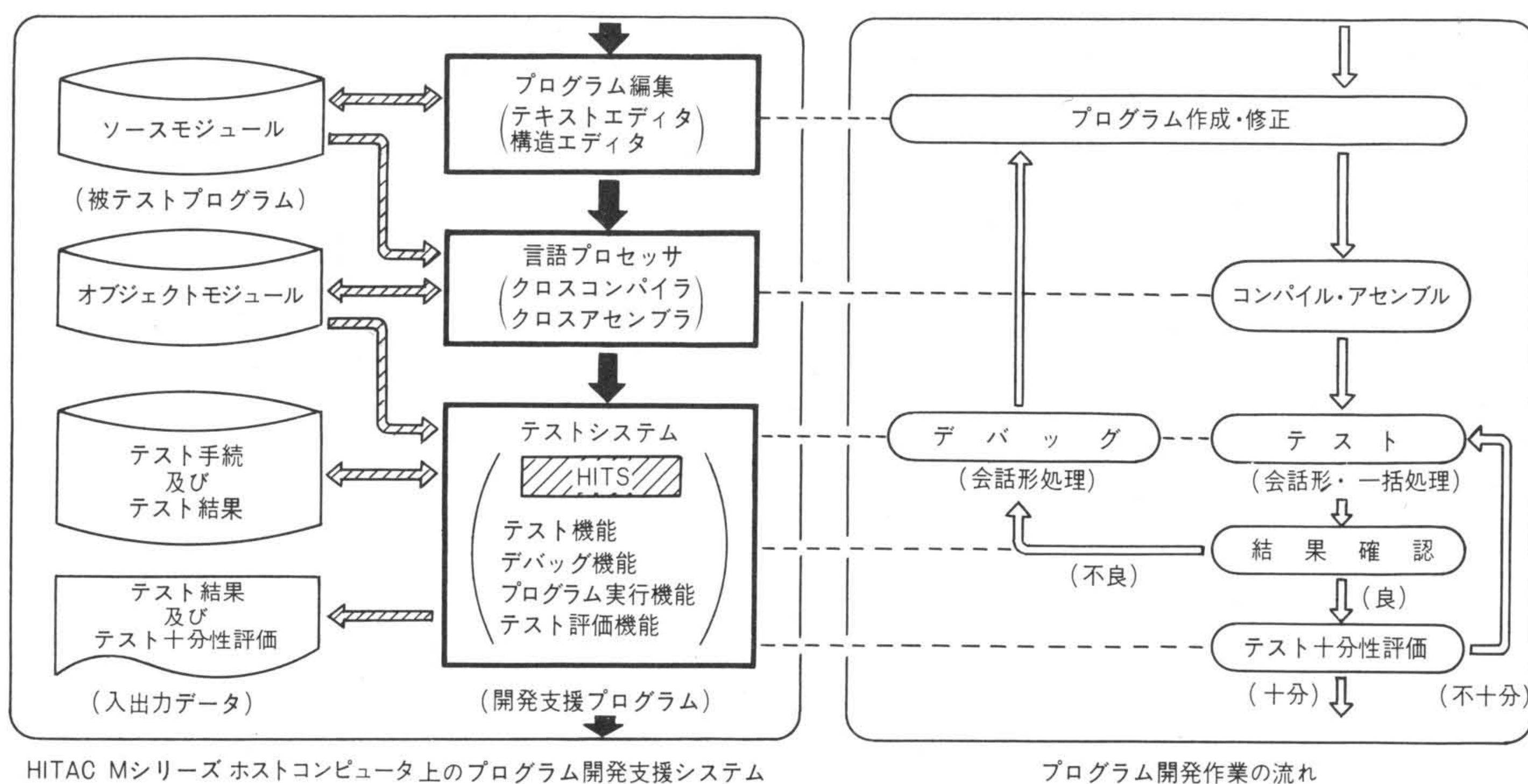


図1 会話形プログラム開発支援システムとHITSの機能構成

16ビットマイクロコンピュータ“68000”用プログラム開発作業者は、プログラム開発の各作業に対応し、かつマンマシンインタフェース性に優れたプログラム開発支援プログラムを会話形式で簡単に利用することができ、プログラム品質、生産性の向上に役立っている。

* 日立製作所戸塚工場 ** 日立製作所システム開発研究所

グラム実行機能及びテスト評価機能の四つに大別でき、これらの機能を会話形で効果的に利用することによって、テスト作業の効率化を図ることができる。

2.2 HITS開発のねらいと特徴

HITSは、ホストコンピュータの高速性と、大容量かつ高速の二次記憶装置といった特長を生かし、次に示すような開発のねらいと特徴²⁾を実現した。

(1) テスト手順の保存による再テストの容易化

通常、プログラムのテストには、テストデータの設定、被テストプログラムの起動、テスト結果の出力などを行なうテストプログラムがあり、一方ではテストを実行するためのジョブ制御文など多くのものが必要である。これらを再利用できるように保存管理するのは案外難しく、使い捨ての場合が多い。HITSでは、上述のテストデータやテストプログラムの内容など、テストに必要な操作手順、すなわちテスト手順を一括してコマンドにより記述でき、コマンドを保存管理することにより再テストが容易にできる。

(2) 非破壊テスト及びデバッグ

HITSでは、被テストプログラムを変更せずにプログラム中の変数の値の変更や表示、それに中断点の設定などの操作ができ、テスト及びデバッグ時のプログラム変更による誤りの発生を防ぐことができる。

(3) シンボリック操作

高級言語S-PL/H^{*)}で書かれたプログラム中の変数名、

表1 HITSのコマンド機能一覧 HITSのコマンドを組み合わせることによって、被テストプログラムのテスト環境の設定、シミュレータによるプログラムの実行、テスト結果の確認などの作業を効率良く行なうことができる。

分類	コマンド	機能
テスト手順の枠組みの定義	PROC	テスト手順の開始
	CASE	テストケースの開始
	DO	コマンドのグループの開始
	END	テスト手順やテストケースなどの終了
プログラムや領域の割付	LINK	プログラムの割付
	INCLUDE	オブジェクトモジュールの取込
	GET	領域の確保
	EQUATE	名前の定義
データの操作	QUALIFY	変数名、文番号などの属するモジュールや手順の宣言
	LIST	変数などの値の表示
	SET	変数などへの値の設定
	DATA	定数列の定義
シミュレーション制御	CALL	高級言語プログラムの起動
	GO	アセンブラプログラムの起動
	RESUME	中断点からのシミュレーション再開
テスト向きコマンド	STUB	スタブの定義
	CHECK	テスト結果の判定
	IF	条件判定
デバッグ向きコマンド	AT	中断点の設定
	OFF	中断点の解除
	TRACE	トレースの表示
	LISTMAP	変数やセクションの割付状況などの表示
マクロ機能	CLIST	マクロコマンドの定義
	マクロコマンド名	マクロコマンドの実行
その他	STOP	実行の中止
	OPTION	各種オプションの指定
	EXEC	ライブラリ中のテスト手順の実行

ラベル及び文番号並びにアセンブリ言語で書かれたプログラム中のラベルなどを用いて、テストやデバッグの操作が記述できる。特に、高級言語の場合、機械命令やレジスタなどのハードウェア構成を知らない人でも、容易にHITSを使いこなすことができ、テスト及びデバッグ作業の効率が飛躍的に向上する。

(4) 効率の良い会話形処理

HITSは、一括処理だけでなく会話形でも使用でき、特に会話形処理時の操作性を良くするよう配慮してある。

(5) 大規模テスト

プログラムモジュール単体テストだけでなく、プログラム規模の大きなテストを可能とするため、複数プログラムの結合機能を含むほか、高性能の機械命令模擬(シミュレーション)を実現している。

3 HITSの機能

HITSは、表1に示す27種類のコマンドにより、その機能を実行することができる。図1に示した四つの機能について以下に述べる。

3.1 テスト支援機能

テスト手順の作成からテスト結果の確認までを支援する代表的な機能について説明する。

(1) テスト手順の入力

テスト用コマンドは、TSS端末から会話形で入力してもよいし、図2に示すように一連のコマンドをまとめてテスト手順として、あらかじめファイルに作成しておいたものと呼び出してもよい。更にテスト手順は、幾つかのテストケースに分割することができ、種々の異なる条件のもとでのテスト手順の記述を、一つの手続内にまとめて保存管理することができる。

(2) データの操作

プログラムで用いている変数へのデータの設定、値の表示、

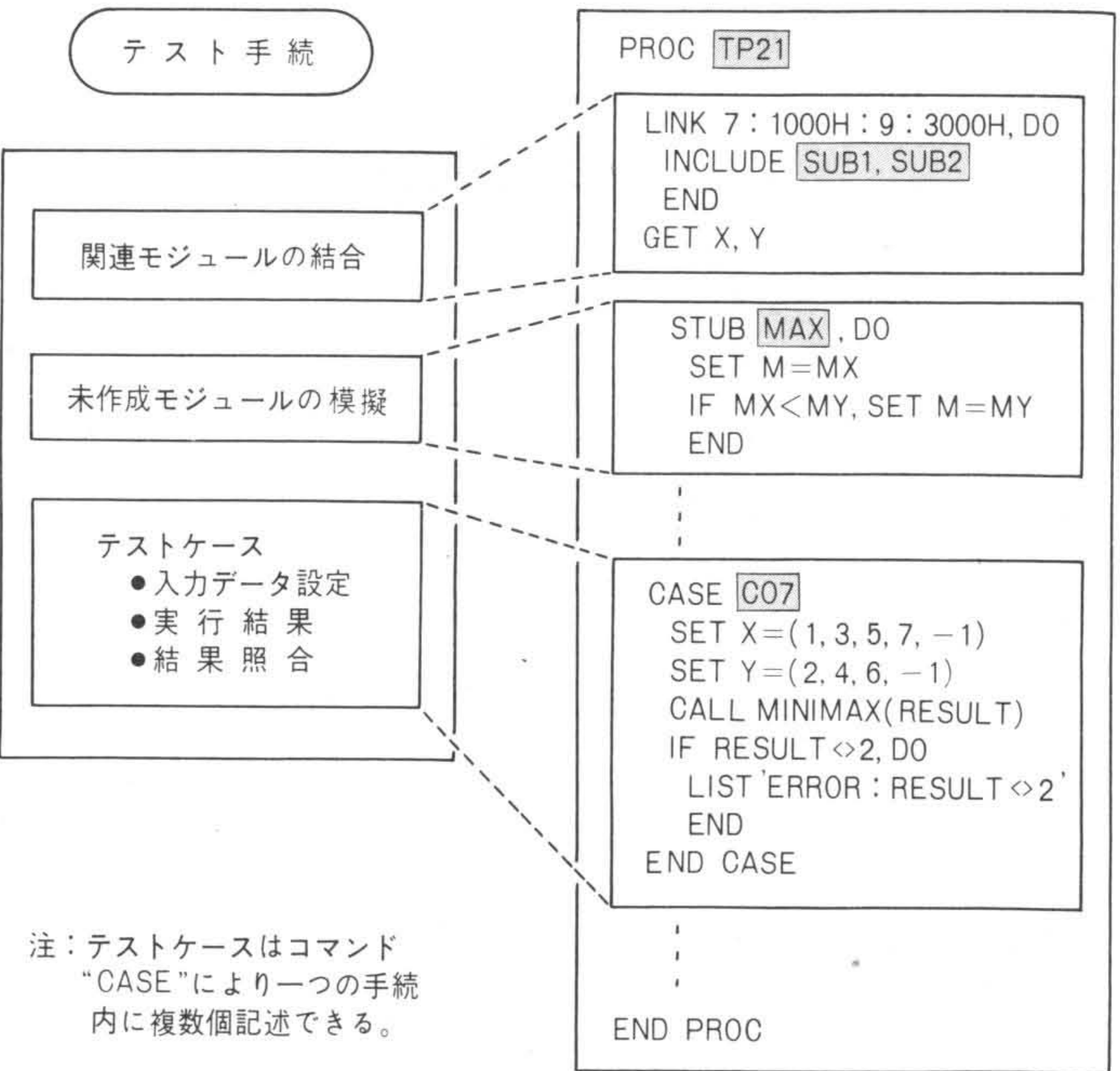


図2 テスト手順の例 ここで示すHITSコマンドの使用例(テスト手順)は、テスト手順“TP21”によって被テストプログラム“SUB1”、“SUB2”の結合、プログラム“MAX”の模擬などのテスト環境の設定を行ない、テスト仕様“C07”を使って被テストプログラムをテストする場合の例である。

※) S-PL/H:日立製作所で開発した“68000”用システム記述言語である。

比較などのデータ操作は、テスト作業の中で最も頻繁に行なわれ、労力を要するものである。

表2の例に示すように、プログラム中の変数が利用できるだけでなく、構造体変数、配列変数の全体や部分範囲を指定してデータ操作ができるなど、機能の充実を図っている。

(3) プログラムの模擬

被テストプログラムから呼び出されるプログラムが未完成であったり、完成していても多大の処理時間を要するなど、実際に実行すると不都合な場合に、そのプログラムの動作を模擬するような簡単なプログラム、すなわちスタブを作成してテストを行なうのが一般的な手法である。

HITSでは、図3に示すように、コマンドによりプログラムの処理を模擬することができ、スタブ作成労力を大幅に減らすことができる。

(4) テスト結果の自動判定

プログラム中の変数が指定された関係式を満たしているかどうかの条件をコマンドで指定することにより、テストの合

表2 データ操作の例 被テストプログラム用テストデータの設定や、表示にプログラム作成者が宣言して使っている変数名が使えただけでなく、変数の型や構造に応じて種々の指定ができる。

例	コマンド及び表示の例	説 明
1	(コマンド) SET A=(0,1,2,3,4)	配列Aの各要素に順に0から4までの値を設定する。
2	(コマンド) LIST B(3:4) (表 示) MOD1#B(3).B1=3 B(3).B2=1 B(4).B1=4 B(4).B2=1	モジュールMOD1内で宣言された構造体の配列Bの要素のうち、添字が3から4までの範囲のものの表示。
3	(コマンド) LIST B(*).B1 (表 示) MOD1#B(0).B1=0 B(1).B1=1 B(2).B1=2 B(3).B1=3 B(4).B1=4	例2と同じ変数Bのすべての添字の値について要素B1を表示する。

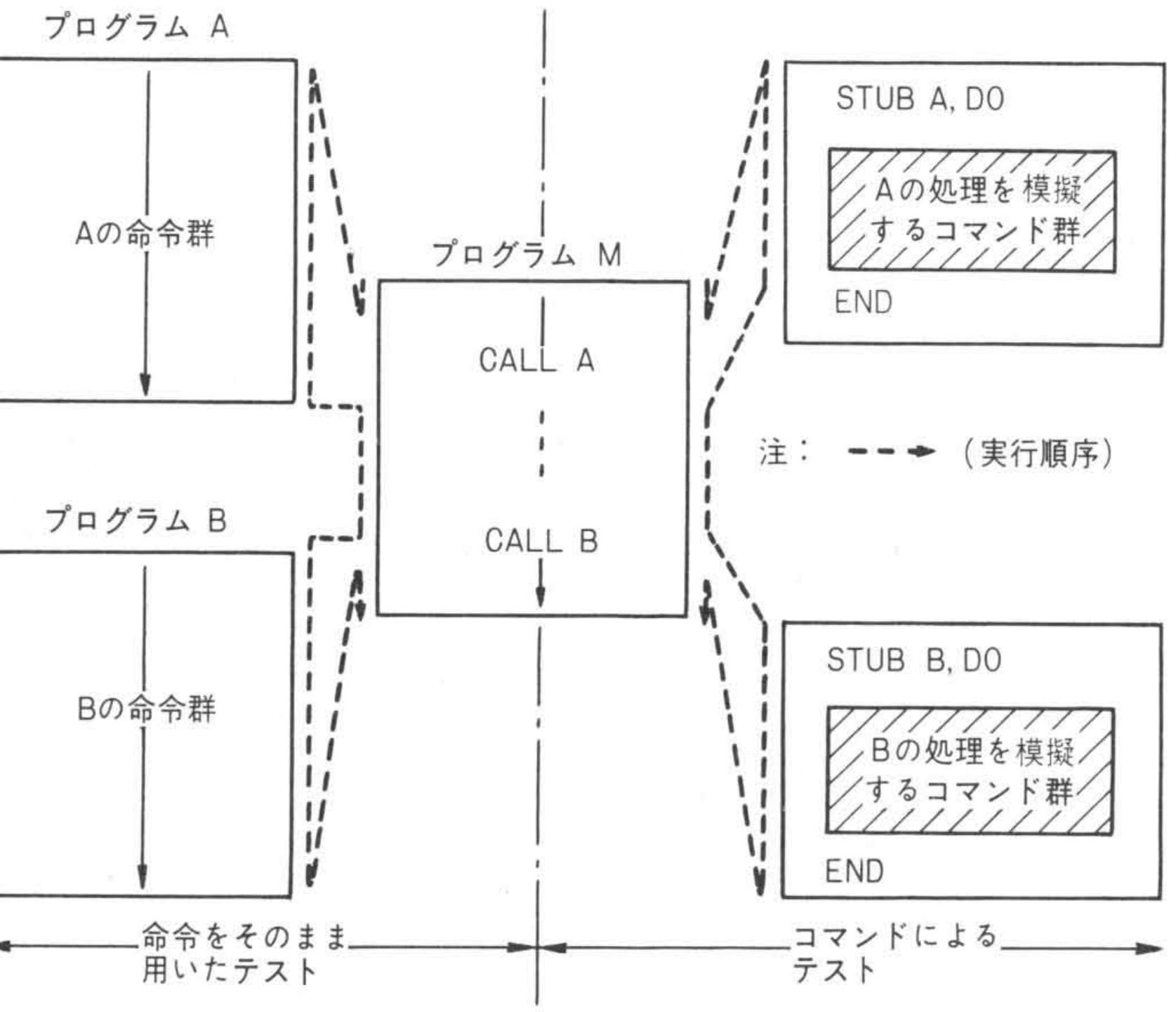


図3 プログラムの模擬機能 被テストプログラムMのテストで、未完成のプログラムA、Bを必要とする場合に、HITSのコマンド“STUB”を用いてA、Bを簡単に模擬することができ、しかもA、Bの実体が存在するときと全く同じ環境で動作させることができる。

表3 プログラム模擬機能の諸元 HITAC Mシリーズホストコンピュータ上で、マイクロコンピュータ“68000”の大規模プログラムの実行を、高速に模擬することができる。

項 目	特 性
模 擬 対 象 ア ド レ ス 空 間	“68000”の全アドレス空間 (16Mバイト)
模 擬 可 能 プ ロ グ ラ ム サ イ ズ	3 Mバイト以上
ハードウェア機能 の 模 擬	ユーザー・特権両モードを模擬する。 外部割込を除く例外処理を模擬する。
命 令 模 擬 速 度	機械命令当たり平均約250 μ s (3 MIPSのHITAC Mシリーズ計算機を) 使用した場合のCPU時間の実測値)

否判定を自動化することができる。

このように、テストの可否を判定するコマンドをテスト手続中に記述しておくことにより、再テストを行なった場合のテスト結果のチェック作業を省力化できる。

3.2 プログラム実行機能

HITSでは、“68000”の機械命令を模擬することにより、被テストプログラムをHITAC Mシリーズホストコンピュータ上で実行することができる。

表3に示すように、模擬できるアドレス空間、命令模擬速度などからみて、システムプログラムや大規模プログラムのテストも実用的に行なうことができ、また、プログラム実行中の誤り検出機能としては、

- (1) 存在しないメモリ番地の参照
- (2) 初期設定されていない変数の参照
- (3) コマンドで指定した実行命令数の上限に達した場合などがある。(2)の誤り検出例を図4に示してあるが、これはプログラム誤りを発見するのに効果大きい。

3.3 デバッグ機能

プログラムの実行中に誤りが検出された場合には、プログラムの実行を中断し、速やかに誤り原因の究明とその削除を行なう必要がある。この作業は、一般的にデバッグ作業と言われ、この作業を支援する、すなわちデバッグ機能としては、プログラム実行の中断点設定と解除、それに表4に示すトレース機能などがある。

中断点の指定には、プログラムのラベル、文番号及び手続

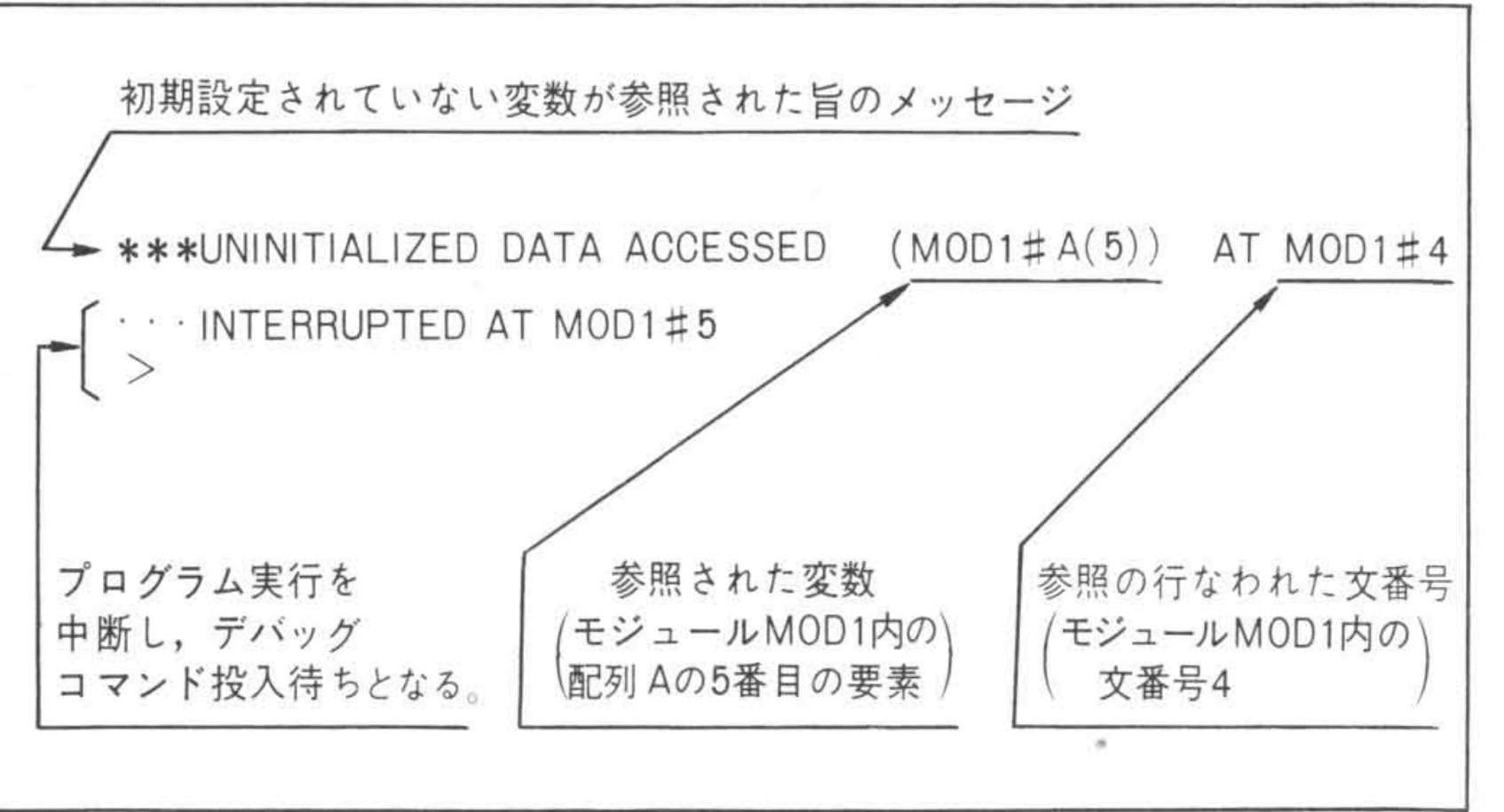


図4 プログラム実行中の誤り検出例 プログラム実行中に初期設定されていない変数を参照した場合の変数名及び文番号を表示した例であり、HITSは、プログラム誤り自動検出機能も併せもっている。

表4 HITSのプログラムトレース機能 HITSがもっている5種類のトレース機能は、会話形式で任意のものが選択でき、プログラムのデバッグ作業を行なう上で効果的である。

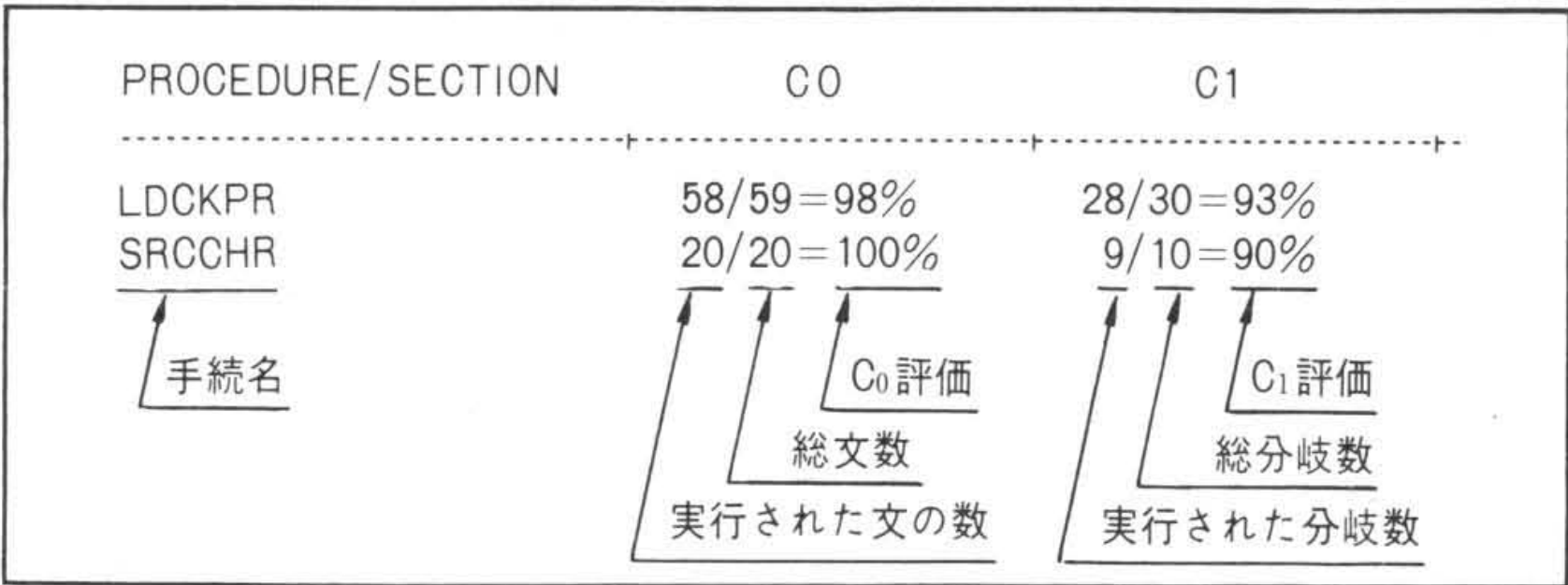
項番	トレース機能	説明
1	分岐トレース	分岐の実行ごとに飛び元と飛び先の文番号を表示する。
2	手続トレース	手続の呼び出しと戻りごとに手続名と呼び出し元又は戻り先文番号を表示する。
3	データトレース	変数の内容が更新されるごとに変数名、更新前後の値及び文番号を表示する。
4	逆分岐トレース	分岐トレースを、過去にさかのぼって表示する。
5	手続ネストトレース	現在実行中の手続から順に呼び出し元の手続名と文番号をたどり、表示する。

名を用いることができ、また、会話形処理の場合は、TSS端末の割込キーを押すことによって、プログラムの実行を中断することができる。

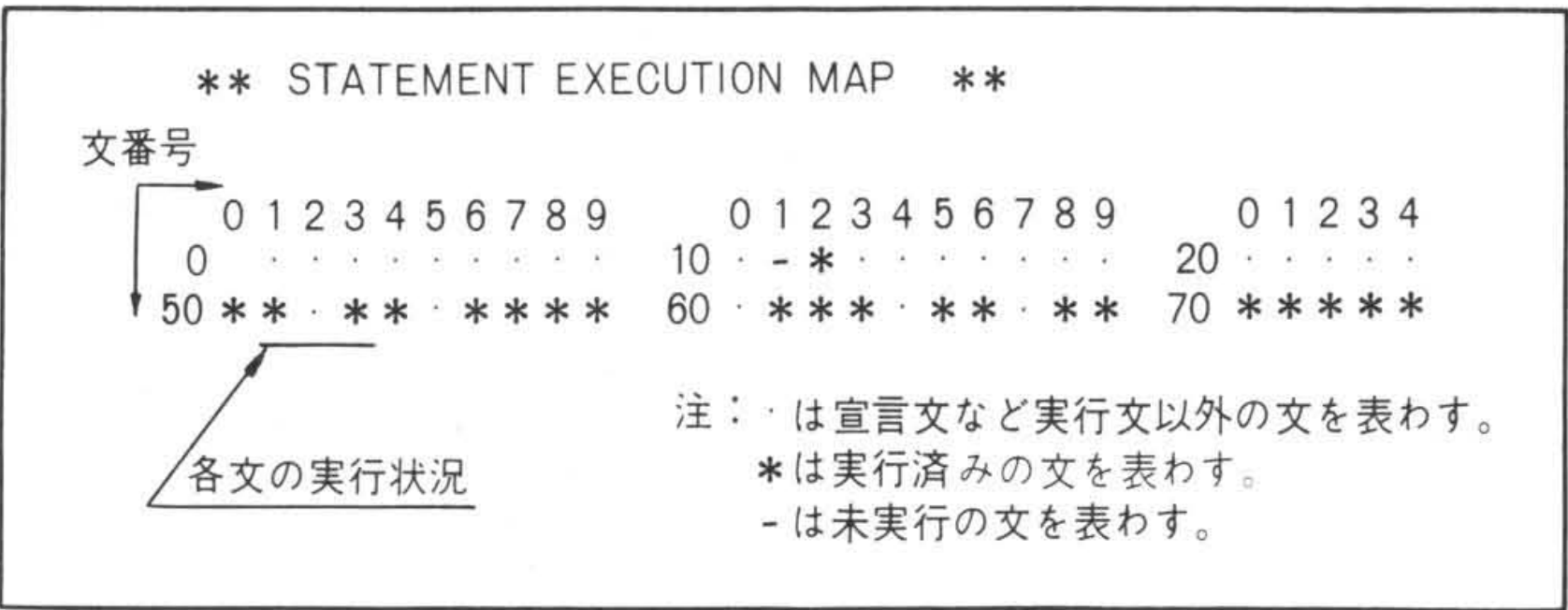
3.4 テスト評価機能

プログラムのテストがどの程度十分に行なわれたかの目安として、HITSではC0とC1の2種類の評価尺度を表示できる。C0とはプログラム中の実行された文の割合であり、C0が100%に達しない場合は、全くテストされなかった文が存在することを意味する。C1とはプログラム中の実行された分岐の割合であり、C1が100%に達しないことは、テストされなかった実行経路が存在することを意味する。これら評価尺度の表示例を図5に示す。

また、HITSは評価尺度のほかに、実行されなかった文や分岐も表示するので、どのようなテストを追加する必要があるかを知る手掛りとなる。このように、C0、C1の2種の尺度を用いてテストの十分性を評価し、不十分であれば必要なテストケースを追加し、十分高い評価が得られるまでテストを繰り返すことにより、プログラムの品質を高めることができる。



(a) C0、C1 評価尺度の表示例



(b) 文の実行状況の表示例

図5 テスト十分性評価の表示例 HITSのテスト評価機能による表示例で、テストの十分性及び命令の実行・未実行の箇所などの可視化により、テスト作業の進捗を容易に把握することができる。

4 適用成果

本システムは、既に情報通信及び産業分野の多くのシステム製品のソフトウェア開発に適用されており、その成果を次に述べる。

- (1) 情報通信システムをはじめとする数多くのシステムに適用されており、テストしたプログラム規模は、数百キロステップに達した。
- (2) HITAC Mシリーズホストコンピュータによるシミュレーション速度は、68000を用いた実システム、すなわちターゲットマシンの200～300倍程度(クロスホストコンピュータ性能：3MIPSの場合)と高速化を図ることができた。
- (3) 上記(2)の実現により、会話形機能のかなめ一つである応答時間で好結果を得ることができた。
- (4) 効果としては、次の点を挙げることができる。
 - (a) テスト工数、特にプログラムモジュールの単体テスト工数を半減することができる。
 - (b) 系統的テストが可能になったことにより、不良原因究明の範囲を極少化でき、不良対策を効率良く行なえる。
 - (c) ハードウェア開発と並行にソフトウェア開発作業が可能になり、システム製品開発期間の大幅短縮ができる。

5 結 言

16ビットマイクロコンピュータ“68000”用ソフトウェアのテスト作業の効率化をねらいとして開発した“HITS”について、その開発のねらい、機能構成、成果などについて述べた。

本システムは、HITAC MシリーズコンピュータのTSS環境下で実用化したもので、操作性、機能及び性能面で優れたシステムと言える。

前章で述べたように、その適用実績からみて、当初のねらいとした効果³⁾を十分に達成している。

今後は、16ビットマイクロコンピュータの普及拡大とともにHITSの適用量が増えるのは確実であり、運用経験の積重ねと課題のフィードバックを行ない、より良いシステムとしてゆく考えである。

最後に、本システムの開発に御協力いただいた関係各位に対し、感謝の意を表わす次第である。

参考文献

- 1) 黒崎，外：通信用ソフトウェア生産技術の高度化，日立評論，64，3，171～174(昭57-3)
- 2) 中所，外：16ビットマイコン“68000”用クロス形テストデバッグ支援システムHITS，情報処理学会，マイクロコンピュータ研究会資料，20-3(昭57-3)
- 3) 岡本，外：マイコン用クロス形テストデバッグ支援システムHITSの適用評価，情報処理学会第27回全国大会，575～576(昭58-10)
- 4) T.Kurosaki, et al.: Software Development Support System for Communication Systems, Hitachi Review, Vol.31, No.5(1982)
- 5) T.Chusho, et al.: HITS: A Symbolic Testing and Debugging System for Multilingual Microcomputer Software, AFIPS-Conference Proceedings, Vol. 52, 73～80(1983-3)
- 6) 神野，外：16ビットマイクロコンピュータ用高級言語S-PL/Hの開発，情報処理学会，マイクロコンピュータ研究会資料，17-3(昭56-7)
- 7) 増井，外：通信用ソフトウェア開発支援システムの一構成法，電気通信学会交換研究会(昭58-12)