

マイクロコードのテストカバレッジ測定装置

4G-6

前沢安則、高橋典子、中村孝

(日本アイ・ビー・エム株式会社 大和研究所)

1. はじめに

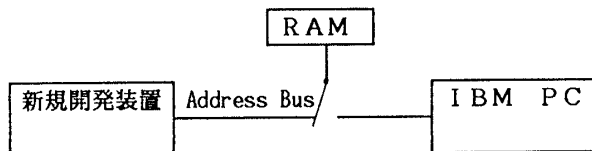
従来MicrocodeのTestは、評価基準が無いため闇雲に行われることが多く、幾つかの点で問題が生じていた。そこで本論文では、二種類のTestの評価基準を採用した。更に、これらの評価基準を用いたTestの実施例を示し、従来存在した幾つかの問題点を解決出来ることを示した。

2. Testの評価法

Testを、実装置を用いて、定量的に評価する一手法として、次の二種類の方法を採用した。

2.1 Test Coverage

Testにより走行したMicrocodeのAddressをTraceしたものを、Test Coverageとする。このTest Coverageを測定するためのAdapterの原理図を第1図に示す。新規開発装置のAddress Bus上の走行Addressは、RAMにTraceされる。RAMは予めPCに接続され全領域すべて0値に設定されており、次に新規開発装置のAddress Busに接続され、走行AddressはそのAddressに対応した所に1値としてStoreされる。Test終了後、再度RAMはPCに接続され、そのRAMの内容を読み取る事によりTest Coverageが得られる。



第1図 Test Coverageの測定原理

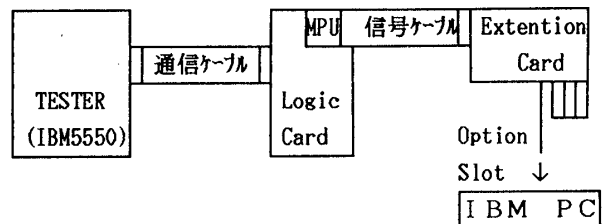
このAdapterを用いたTestの構成図を第2図に示す。新規開発装置は、Tester (例えばIBM5550)と通信ケーブルを介して接続されており、ディスク上のTestcaseに従ってTestされる。Test Coverage測定用Adapterは、PC用Extention Cardに組み込まれ、新規開発装置と信号ケーブルを介して接続される。この信号ケーブルは、新規開発装置のLogic Card上のMPUソケットに接続される。

このTestを行ない、Test Coverageを求め、その結果を第3図に示す。左端の16進4桁の数字は、その右に示されているTest結果の最初のAddressを示す。Test結果は2進の数字で示され、1桁がそのAddressに対応したMicrocodeの1Byteに対応している。0値の連続している部分は、それと対応したAddressの

F200	0000000000000111	1111111111111111	1111110000000000	1111111111111111
F240	1111111111111111	1111111111111111	1111100000000011	1111100111111111
F280	1111110110000000	0000001000000000	0000000000000000	0000000000000011
F2C0	1100101111111000	1111101100000000	0000000001111111	1111100000000000

第3図 Test Coverageの測定例

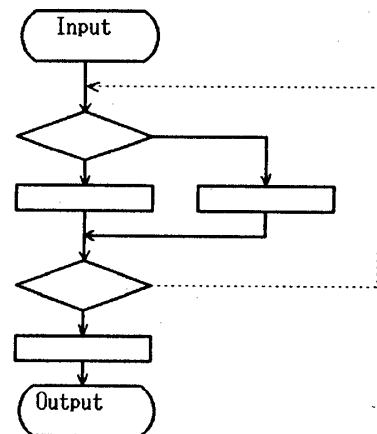
MicrocodeがまだTestされていない事を意味している。



第2図 Test 構成図

2.2 Test Depth Coverage

Testにより走行したMicrocodeの条件付Jumpの中で、条件JumpしなかったAddressをTraceしたものを、Test Depth Coverageとする。第4図は、Program Flowの一例を示している。このFlowにおいて、第2.1節で説明したTest Coverageの測定により、Testされた事を確認できる部分は実線で描かれている部分である。点線で描かれている部分は、Jump先が独立した処理を持っていないため、Testされたかどうかを確認することができない。そこでTest Depth Coverageを測定する事により、このようなJump先が独立した処理を持っていない条件付JumpについてもTestされたかどうかを確認することができ、より一層複雑な論理をもつMicrocodeに対してもTestの確認ができる。



第4図 Program Flowの一例

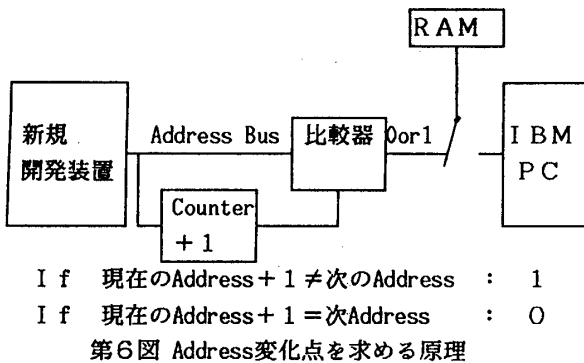
このTest Depth Coverageを求める方法はいくつか考えられるが、採用した方法は前記AdapterでTest Depth Coverageを直接求めるのではなく、先ずAddress変化点を求める。Address変化点とは、第5図に示すごとく、MPUが

読むDataのaddress Bus上の現在のAddressに1を加えた値が次に読むDataのAddressと異なった場合、現在のAddressでAddressが変化したとするもので、そのことを1で示し、それらが同じ場合を0で表わす。第5図は条件付Jump命令で条件Jumpし、Address Bus上のAddressが8004番地から9000番地に変化した場合で、8004番地に変化点が生じる。条件付Jump命令の内でのこのAddress変化点を求めれば、Test Depth Coverageを求めた事になる。しかし、単純Jump等においてもAddress変化点が生じるので、後で条件付Jump以外のAddress変化点をPC上で取り除いている。

Address 変化点	Address Bus	Data Bus	Instruction
0	8000	FE	CPI X'01'
0	8001	01	
0	8002	CA	JZ X'9000'
0	8003	00	
1	8004	90	
0	9000	78	MOV A,B

第5図 Address変化点

第6図にAddress変化点を求める原理図を示す。新規開発装置のAddress Bus上のAddressはCounterで1加えられ、次に有効となるaddressと比較され、異なれば1、同じならば0としてRAMにTraceされる。RAMは予めPCに接続され全領域すべて0値に設定されている。Test終了後、再度RAMはPCに接続され、そのRAMの内容を読み取る事によりAddress変化点を得られる。Testの構成図としては、前記第2図と同じである。



そこで、Test Depth Coverageを求める方法について次に述べる。第2.1節で解説したTest Coverageを用いる事により、少なくとも一度は実際に走行したAddressを知ることができる。条件付Jump命令のAddressを求める手法は色々考えられるが、Object Codeを逆Assembleすることにより求める。これら二種類のAddressの論理積を求めれば、Testにより少なくとも一度は実際に走行した条件付Jump命令のAddressを知ることとなる。このAddressの中でAddress変化点のなかつたAddressを求めれば、Test Depth Coverageを求められる。

2.3 実施例

第2.1節及び第2.2節で述べた二種類のTestの評価基準である、Test Coverage及びTest Depth Coverageを使用したTestの実施例を第7図に示す。

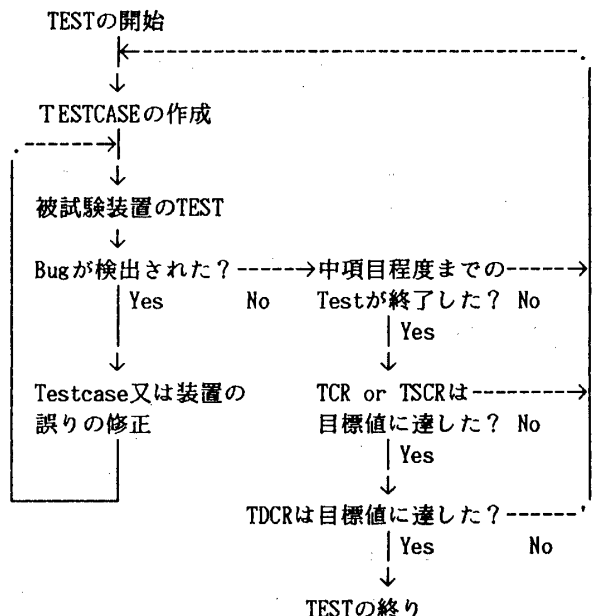
Microcode設計仕様書に基づき、先ず大まかに中項目程度のTestを終了させる。この中項目程度のTestまでは、Test case作成者の個人差によるTestのパラツキによる重複Testは比較的微小である。そこでTest Coverageを求め、まだTestされていないMicrocodeの内容から効率良く小項目のTestcaseを作成でき、Test漏れを防ぎかつ重複の無いTestが可能となる。更に、Test Depth Coverageを測定することにより、まだTestされていない条件付Jump命令を知ることができるので、それが条件JumpするようにTestcaseを効率良く作成できる。したがて、Test期間を短縮しつつより完全なTestが可能となる。

$$TCR = \frac{\text{Testにより走行したByte数}}{\text{Object Codeの総Byte数}} \quad (1)$$

$$TSCR = \frac{\text{Testにより走行したStatement数}}{\text{Object Codeの総Statement数}} \quad (2)$$

$$TDCR = \frac{\text{Testにより条件Jumpした条件付Jump数}}{\text{Testにより走行した総条件付Jump数}} \quad (3)$$

TCR : Test Coverage Ratio
 TSCR : Test Statement Coverage Ratio
 TDCR : Test Depth Coverage Ratio



第7図 評価基準を用いたTestの実施例

3. おわりに

実装置を用いて、Testを定量的に評価する方法を提案した。Test Coverageを求めるために、PC Extention Card及び算出Programを作成し、その効果を確認した。Testの深さを確認するためのTest Depth Coverageを求めるPC Extention Card及び算出Programを作成し、その効果を確認する予定である。本論文で提案したTestの評価法を用いて、現在次期開発Display装置のMicrocodeのTestを行っており、その結果については次の機会に発表する予定である。