

マイクロプロセッサのFunctional Test に関する一考察

On the Functional Test of Microprocessors

徳竹芳男⁺古屋清⁺⁺当麻喜弘⁺⁺

Yoshio TOKUTAKE

Kiyoshi HURUYA

Yoshihiro TOHMA

⁺富士通 (株)⁺⁺ 東工大 情報工学科

Fujitsu Co Ltd.

Tokyo Institute of Technology

あらまし 本論文は近年注目を浴びているマイクロプロセッサのファンクショナルテストに関するものである。命令セットを用いたファンクショナルテストの実現手法に関して、従来の方法の限界を指摘しプロセッサ内にScan方式を利用した若干の診断回路を組み込むことによりテストセット、生成アルゴリズムが小さく簡単化されることを示す。

1. まえがき

半導体技術の進歩によりプロセッサの高集積化、高速化は目覚ましい勢いで進んでいる。そのため、従来のゲートレベルテストはそのテスト生成に多くの時間を要する上に、技術的にも困難になりつつある。その解決策としてスキャンパス、LSSD等の回路構成法が提案され、現実に採用されている。しかし、これらの回路構成法はゲートレベルテストの生成を容易にはするが、テスト生成問題本来の解決策とはなっていない。

又、現在の主要なデバイスであるCMOS, NMOS等はタイミングセンシティブ、パターンセンシティブな故障メカニズムを持つことが知られている。(1)(2)

このような状況の中でプロセッサ本来の命令セットを用いるファンクショナルテストが多くの魅力あるテストとして注目されている。

ファンクショナルテストには次の様な利点がある。

(1) ダイナミックなテストが行なえるため、

タイミングセンシティブ、パターンセンシティブな故障を含む幅広い故障を検出の対象とすることができる。

(2) テスト生成に必要な情報量が少なく、より経済的に作成できる。

(3) テスト生成を論理設計と同時にこなえ、より低いレベルの設計変更には影響を受けない。

又、一般論理回路においては素子の機能ブロック化が進んでおり、ゲートレベル迄の追跡が容易となってきている。そのため、ゲートレベル情報を用いることによるテスト生成のデメリットも増えている。

そこで本論文では、まずプロセッサのファンクショナルテスト生成における従来方法⁽³⁾の問題点を指摘し、その解決策としてプロセッサの中に若干の診断回路を組み込むことによりテストセット、生成アルゴリズムが小さく簡単化されることを示す。さらにゲートレベルテストにおけるスキャンパス回路構成法と本論文で提案している診断回路の構成法とを比較評価する。

2. 従来のテスト方法と問題点

従来の小規模回路に対しては、D-Algorithm等を用いたゲートレベルテスト生成法が充分その目的をはたしてきた。しかし、回路が大規模化、高速化し、さらにLSI化されるに従い従来方法ではそのテスト生成に多くの時間が必要となり技術的にも困難になってきている。この問題の解決のため、現在もゲートレベルテ

ストに関するテスト容易化のための回路構成法、テスト生成アルゴリズム等が研究されている。ファンクショナルテストに関しては次の様な多くのむづかしい問題があるため、発表されている文献等も少ない。

- (1) 回路構成法が設計者により異なるため、そのテスト生成も個別のケーススタディとなってしまう。
- (2) テスト生成をアルゴリズム化しにくい。
- (3) 検出率等の評価手法がない。

プロセッサのファンクショナルテストを行なう場合、内部状態の制御性、観測性が大きな問題となる。

2.1 従来のテスト方法

プロセッサのファンクショナルテスト法に関しては、これまでに次の様な研究が報告されている。S. M. THATTEは各レジスタに対して入力端子からの距離を「制御困難さ」、出力端子までの距離を「観測困難さ」と定義する。ここで距離とは、その間をデータ転送するのに要する最小命令数をいう。 $|L(R_i)|$ は、レジスタ R_i を出力端子に読み出すのに必要な命令数をいう。更に命令の「制御困難さ」及び「観測困難さ」をそれぞれソースレジスタの「制御困難さ」、及びデスティネーションレジスタの「観測困難さ」と定義する。そして観測性の最も高い命令から順にその正常性を保証してゆく形のテスト生成法を発表している。⁽³⁾

ここで次の様なケースを例にこの方法を考えてみる。

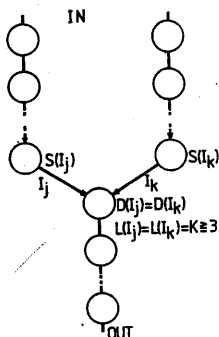


図1. レジスタ-命令グラフ

図1は命令 I_j のソースレジスタ $S(I_j)$ 及び命

令 I_k のソースレジスタ $S(I_k)$ から同一のデスティネーションレジスタ $D(I_j) = D(I_k)$ に転送される様子を示している。ここで○印(NODE)はレジスタを示し、有向枝(EDGE)は命令実行に伴うデータの転送経路を示している。

ここで故障 $f(I_j/I_k)$ を考える。故障 $f(I_j/I_k)$ とは命令 I_j を実行したにもかかわらず、命令 I_k が代りに実行されてしまう故障現象をいう。

このような故障 $f(I_j/I_k)$ は次の手順で検出される。

手順 $f(I_j/I_k)$ $|L(I_j)| = |L(I_k)| = K \geq 3$, $D(I_j) = D(I_k)$, $L(S(I_k)) = K$

ステップ1. $S(I_j)$ にオペランド1を、 $S(I_k)$ にオペランド2を格納する。
(オペランド1 ≠ オペランド2)

ステップ2. i を1から K までの間、次のことを実行する。

I_j を実行する。
 $Read(D(I_j))$ を実行し、 $D(I_j)$ を読み出す。

(期待データ = オペランド1)

ステップ3. I_k を実行する。

ステップ4. $Read(D(I_k))$ を実行し、 $D(I_k)$ を読み出す。

(期待データ = オペランド2)

ここで $Read(D(I_k))$ とは、命令 I_k のデスティネーションレジスタを読み出すために用いる命令シーケンスをいう。

この手順中で、 $S(I_k)$ にデータを書き込むために用いる命令群に故障が存在すると、オペランド1が $S(I_k)$ に格納される可能性がある。 $D(I_j)$ を読み出すためには K ステップかかる。この K ステップ中の全ての命令が $S(I_k)$ に対してオペランド1を格納する」という故障の影響を1から K ステップの間どこでも及ぼし得る。そのため、ステップ2を K 回繰り返す必要があり、その繰り返しの実行により故障の影響を排除している。

2.2 問題点

2.1で示したものはS. M. THATTEの示したテスト生成アルゴリズム中の一手順であるが、これら生成法は次の点から現在の高性能なプロセッサに対しては実用的でない。

- (1) テスト生成アルゴリズムが複雑である。

(2) テスト生成法中でなされている次の様な仮定は現実のプロセッサにあてはまらない。

- ・ 命令IJのデスティネーションレジスタ数は1である。
- ・ 命令のパーシャル実行を引き起こす様な故障、データの流のみ変化させる様な故障は起こらない。

(3) テストセットが大きくなる。

命令の観測性をパラメタとしてファンクショナルテストを組み立てようとする、生成アルゴリズムが複雑となりプロセッサごとのケーススタディとなってしまう実用的でない。

ここでプロセッサの構成を示す。プロセッサの構成は一般に図2に示す様に、コントロール回路部分とオペラティブ回路部分に分けることができる。プロセッサにおいては、制御信号経路、データ経路といったものが命令という形の中に定義され決まっている。そこでファンクショナルテストはオペラティブパートの入力端にテストデータを供給し、命令セットを実行することによりデータを伝播し、オペラティブパートの出力端を正常値と比較する方式のテストであるといえる。このことからファンクショナルテストの生成アルゴリズムを複雑にし、テストセットを大きくする原因はプロセッサ内のデータラッチ（オペラティブパート内のデータレジスタ、命令レジスタ等をいう。）の制御性、観測性の低さにあるといえる。そしてファンクショナルテストにおけるデータラッチの制御性、観測性の低さは次のことに起因する。

- (1) 制御性が低いとは、各データラッチに任意のデータをセットすることが困難であること。即ち、特定命令の実行によってのみしかデータの初期値セットができず、その正常性保証が困難なことをいう。
- (2) 観測性が低いとは、各データラッチに記憶されている内容の検証が困難であること。これは命令実行により各データラッチに制御信号（例えば、Set Clock, Count Up, Write Enable等の信号）の出したこと、又は誤まって出なかったことの検証がその内容判定によってのみしか行なえないことによる。

このことからファンクショナルテストにおけ

るテスト生成を容易に、かつ小さなテストセットにするためには、各データラッチを制御する制御信号の制御性、観測性の改善が最も必要であることがわかる。

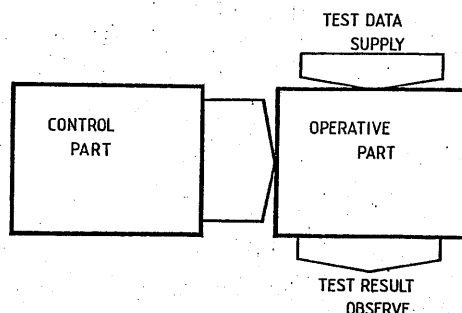


図2. プロセッサの構成

3. 診断回路

そこで本研究ではファンクショナルテストのテスト生成を容易にするため、プロセッサに若干の診断回路を導入することを考えた。

図3は本研究の診断回路の考え方を示したものであり、診断回路により斜線部分の機能の実現をはかる。つまり診断回路により初期値セット、結果の読み出しに必要な制御信号が供給でき、命令実行によるプロセッサ本来の制御信号の発生を観測できる様にする。

図4はプリセット端子付D型F.F.の構成例を示している。図5は図4のF.F.の一構成要素として組み立てた各段プリセット端子付シフトレジスタを示す。これが本研究で用いる診断回路である。

図6, 7は本診断回路の適用例を示している。本診断回路は、データラッチの初期値セット、その内容読み出しに必要な最低限の制御信号を供給する機能（データは本来のデータバスを経由する。）、そして命令実行時に出される各データラッチへのプロセッサ本来の制御信号をダイナミックにサンプリングする機能とを有している。ここでバス（MPX等）の切替えは命令レジスタIRの制御性を高めることにより、セットされた命令本来のデコード信号を利用することで簡単に行なえると考えている。

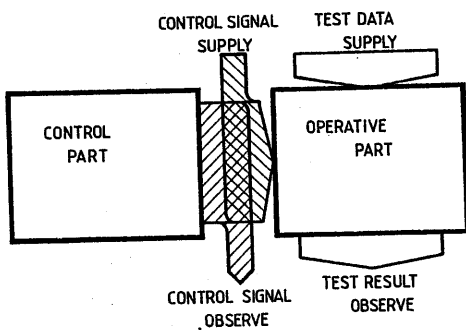


図3. 診断機能の概念図

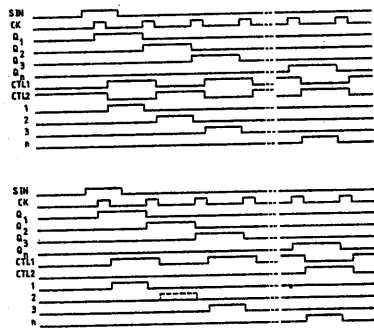
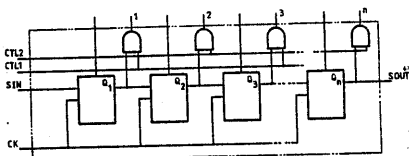


図5. 診断回路 構成例

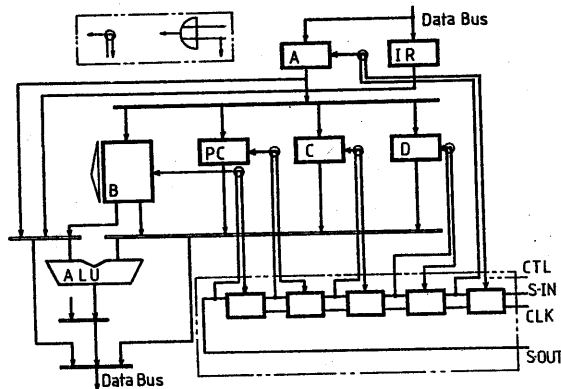


図7. 診断回路 適用例2

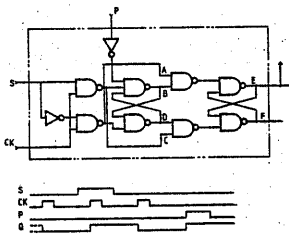
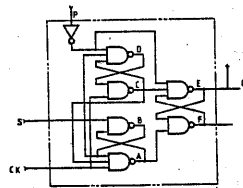


図4. P端子付D型F.F.

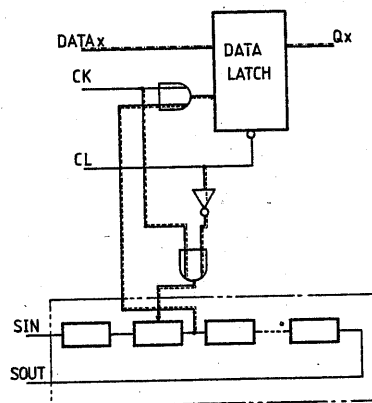


図6. 診断回路 適用例1

ここで図6, 7を用いて診断回路の機能について説明する。図6に示したデータラッチとは、図7の様なブロック図上で表現される1つのレジスタを表わしている。

本診断回路の制御信号供給機能について述べる。本診断回路の各段F.Fの出力はCTL信号によりゲートされ、プロセッサ本来の制御信号にORされる形で接続されている。制御信号はCTL信号により任意のデータラッチに供給可能である。ここで本診断回路はプロセッサ本来のデータバスDATAx等には何の影響も与えていない。この診断回路からの制御信

号供給機能を用いてプロセッサ内全データラッチへの初期値セット、結果の読み出しに必要な最低限のデータ転送経路の正常性をテストする。データバス IN からレジスタ A へ、レジスタ A からレジスタ C、D へと診断回路を用いてデータを伝播し、データバス OUT にて正常値と比較する。

次に本診断回路の制御信号観測機能について述べる。本診断回路ではデータラッチの内容を変化させるプロセッサ本来の制御信号が各段 F.F. のプリセット端子に接続されている。命令実行中に与えられる各データラッチへの制御信号は本診断回路にサンプリングされる。CTL 信号='0' 状態でスキャンアウトすると命令セット実行中に各データラッチに出された制御信号はシリアルデータとして読み出すことができる。プリセット端子は波形のエッジを取らえるため、全ての波形変化をサンプリング可能である。

命令のテストは次の様に行なう。各データラッチに初期値をセットし、命令レジスタ IR にテストする命令をセットする。(このテスト実行時点では、各データラッチにセットされている初期値が正常であること、そしてその内容を正しく読み出せることは検証済である。)そしてテストする命令を実行する。この命令実行により各データラッチに出された制御信号は本診断回路にダイナミックにサンプリングされる。データトランスファー命令 J (レジスタ C → レジスタ D) を実行したとすると、命令実行後診断回路をスキャンアウトした時 '0 0 0 1 0' が読み出される。それ以外であれば故障が存在することになる。その正常性判定の後、レジスタ D を読み出し期待値であることを確かめる。ここでデータラッチの内容判定が必要となるのはテスト命令のデスティネーションレジスタのみとなり、それ以外のレジスタについては診断回路を用いて行なわれる制御信号の判定のみでよいことになる。このため本診断回路はテストセットを小さくする効果がある。

本診断回路を用いたテストは図 8 の様な構成となる。ここで、初期値セット、結果判定はダイナミックに行なう必要はない。又、実スピードで行なわれる命令実行中に発生するグリッチ、波形抜け等のタイミングエラーは本診断回路で確実にサンプリングされる。そのため、

本診断回路を用いることによりファンクショナルテストの持つダイナミックな特性が失なわれることはない。

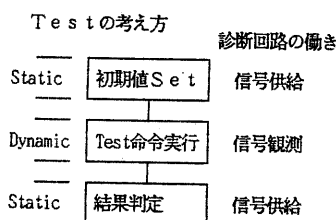


図 8. テストの構成

4. スキャンバスとの比較

本研究の診断回路の考え方、回路構成法も現在注目を集めている LSSD, Scan Path⁽⁴⁾⁽⁵⁾の考え方、回路構成法と比較してみる。

ゲートレベルテストは論理回路の入力ピンよりテストデータを供給し、出力ピンの応答を観測し正常値と比較する方式のテストである。ゲートレベルテストのテスト生成を容易にするためには、信号の伝播を妨げるプロセッサ内全 F.F. の制御性、観測性を高める必要がある。これはゲートレベルテスト生成においては各信号線の持つ性質、つまり制御信号である、又はデータラインであるという概念が全く入り込まないため、プロセッサ内の全記憶素子を Path 活性化問題の対象とする必要があるからである。LSSD, Scan Path 等もこれらをぬらった回路構成法といえる。

本研究の診断回路はファンクショナルテストにおいて障害となるデータラッチの制御性、観測性を高めるため、データラッチを制御する制御信号の制御性、観測性を高めることをスキャンバスと同様に、シフトレジスタのテスト容易性、付加端子数が少ない等の特徴を利用して実現したものである。

次に従来のファンクショナルテスト、スキャンバス方式、本研究の方式の 3 通りについてテストの考え方を概略的に比較する。

(1) ファンクショナルテスト

入力データバスよりテストデータを供給し、命令セットを実行することによりデータを伝播し出力データバスにおいて正常値と比較する方式のテストである。

(2) スキャンパス方式

ゲートレベルテストにおいてPath活性化の障害となる論理回路内の記憶素子の制御性、観測性を高めるため、記憶素子に付加回路をつけることにより記憶素子を全てシリアルに接続し、擬似入出力端子として用いる。

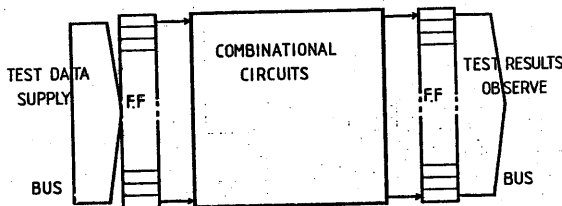
(3) 本研究の方式

ファンクショナルテストにおいてそのテスト生成の障害となるデータラッチの制御性、観測性を高めるため、診断回路を付加することにより制御信号を任意に供給、そしてダイナミックに観測することを可能にしたものである。

本診断回路には次の様な利点がある。

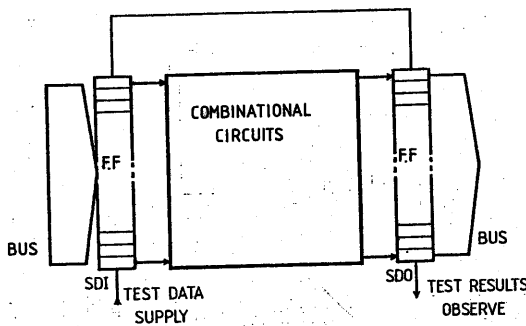
- (1) 診断回路自身のテストが容易である。
- (2) 必要な追加ハード量が少ない。
- (3) スキャンパス方式と異なり本来の論理回路には診断回路が入り込まないので、論理設計、構造設計への影響が少ない。
- (4) プログラムデバッグ機能等への応用が容易である。
- (5) ダイナミックな診断を可能とする。
- (6) 本来必要であるデータラッチの内容判定が制御信号のみの判定で済むため、テストセットを小さくできる。

FUNCTIONAL TEST

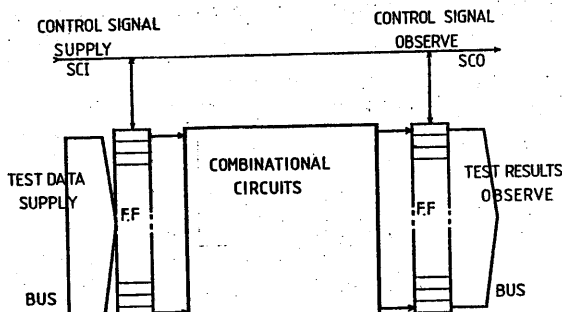


ファンクショナルテスト

SCAN PATH



スキャンパス方式



本研究の方式

図9. ファンクショナルテスト、スキャンパス、及び本方式のブロック図

5. 機能分類と故障モデル

本研究ではプロセッサの機能を表1の5つに分類し、そのそれぞれに対し故障の仮定を行っている。

機能分類	故障モデル
Instruction Decoding & Control Function	$f(I_j \neq I_j)$
Register Decoding Function	Mapping fDR from R to R U { ϕ } Mapping fDW from R to R U { ϕ }
Data Storage Function	S-a-0,1 at any number of cells of any number of Register
Data Transfer Function	1-Line S-a-0,1 Two Line Coupled Fault
Data Manipulation Function	—————

表1. 機能分類と故障モデル

この分類を次の例を用いて概説する。

「配列構造を持つR1, R2レジスタ集合のうち、オペフィールドのR1, R2パートで指定されるレジスタをFunc指定に従って演算し、その結果をR2パートで指定されるレジスタに格納する。そして演算結果によるステータス情報をR3レジスタにセットする。」

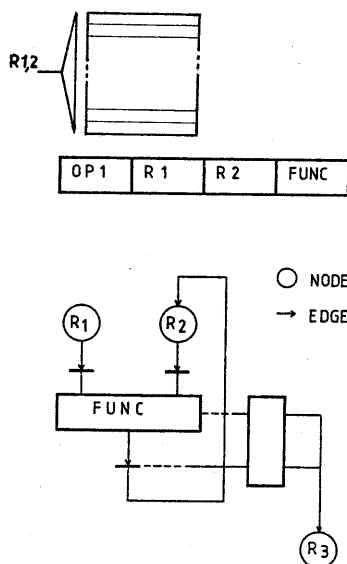


図10. 命令のグラフ表現例

本研究では、Node ○印をデータ記憶機能、有向枝→印をデータトランスファー機能、○印で示される配列レジスタ群から指定レジスタを読み出し又は書き込みを行なう機能をレジスタデコーディング機能、Funcパート等で指定される機能をData Manipulation機能としてボックス□で示している。そしてオペコードを解釈し、その命令をイニシエイトする機能をインストラクションデコーディング&コントロール機能と定義し、オペコードとそのグラフ形状の違うものを全て別個の命令として扱う。(実際にはもう少し細かな定義を必要とするが、これも命令の性格、レジスタの属性等から容易に判断でき、ハードウェアの詳細情報は必要としない。) 以上の分類でテストセットの大きさに最も影響力を持つのは、インストラクションデコーディング&コントロール機能である。

又、以上の様な定義による命令分類、機能分類はレジスタ等に属性をつけることにより、計算機による自動判別も可能であり、ファンクショナルテストにおけるテスト生成を自動化する一段階であると考えられる。

5.1 $f(I_j \neq I_j)$ のテストについて

本研究では、テストする命令のソースレジスタに正しい初期値がセットできること、デスティネーションレジスタが正しく読み出せることは診断回路を用いてオーにテストされる。その正常性を保証した後、インストラクションデコーディング&コントロール機能のテスト等を行なう。故障 $f(I_j \neq I_j)$ の存在しないことの検証のため次のテストを行なう。

- (1) テスト命令を実行後、その命令のデスティネーションレジスタのみに制御信号が出され、それ以外のレジスタには出なかったことを判定する。
- (2) (1)の正常性を確認した後、テスト命令のデスティネーションレジスタを読み出し正常値であることを判定する。

ここではデータラッチが他から何らの制御信号も受けずにその内容が変化してしまう様な疑時的な故障現象は考慮する必要がない。この様な故障は、データストレージ機能のテストで検出可能である。

5.2 レジスタデコーディング機能の

故障モデル

本研究におけるレジスタデコーディング機能とは、プロセッサ内配列構造レジスタ、つまりRAM構造を持つレジスタのデコーディング機能とみることができる。そして故障現象は次の様に考える。

レジスタ集合Rから指定レジスタを読み出す場合、次の現象として現われる。

- (1) $f(R1/\phi)$ R1を読み出そうとしたが、誤まって何も読まない。(ハードにディペンドした情報を読み出す。)
- (2) $f(R1/R2)$ 誤まって異なるアドレスの内容を読む。
- (3) $f(R1/R1+R2)$ 異なるアドレス内容とOR又はANDされた内容を読み出す。

レジスタ集合R中の指定レジスタに書き込もうとする場合、次の現象として現われる。

- (1) $f(R1/\phi)$ 誤まってどのレジスタにも書き込まない。
- (2) $f(R1/R2)$ 誤まって異なるアドレスに書き込む。
- (3) $f(R1/R1+R2)$ 指定レジスタに加え、異なるアドレスにも書き込む。

5.3 Data Manipulation機能について

本研究における当機能に対しては、全て組合せ回路に対するテスト生成問題となる。これは既知の最大テストセットからいかにテストセットを小さくするかという問題となり、ここでは取り扱わない。

6. テスト方式

次に本診断回路を用いたテスト生成法で、テストセットの大きさに最も支配的な影響力を持つインストラクションデコーディング&コントロール機能について述べる。図1を例に再度考える。この例でわかるように、テストアルゴリズムを複雑に、そしてテストセットを大きくする原因は次の点にある。

- (1) 初期値セット、結果判定に用いる命令の正常実行の保証が困難である。
- (2) テスト命令の正常性の判定がレジスタの

内容比較によってのみが行なえない。

この点を解決するため、診断回路から各データラッチへの書き込み信号を任意に供給することを可能とし、テスト命令の正常性判定を容易にするため各データラッチへのプロセッサ本来の制御信号をダイナミックにサンプリングすることも可能とした。

診断回路を用いた基本的なテストの考え方は次のようになる。

- (1) 診断回路を用いてテストする命令のソースレジスタにオペランド1を、それ以外のレジスタにオペランド2を格納する。
(オペランド1≠オペランド2)
- (2) テストする命令を実行する。
- (3) 診断回路の内容をスキャンアウトし、制御信号の発生が正しいか判定する。
- (4) テストした命令のデスティネーションレジスタを診断回路を用いて読み出し、その内容が正常値であるか判定する。

本方式のテストでは、従来ファンクショナルテスト生成の大きな障害であったデータラッチの制御性、観測性の低さの問題が、診断回路の機能により解決されている。

これにより次の効果がある。

- (1) テスト生成アルゴリズムが単純化され、機械的となる。
- (2) テストセット(プログラム、テストデータ等)が小さくなる。
- (3) 故障の指摘がより細かな機能単位に行なえる。
- (4) 検出できる故障範囲が拡張される。

7. テストセットの評価

テストセットの大きさをはっきりした数値で表わすことは大変むづかしい。そこで単純な場合に分けてその概略値を示す。

表2はインストラクションデコーディング&コントロール機能のテストに必要なテストセットの大きさをケース1, 2に分けて示したものである。ここでは次の仮定のもとにたっている。

- (1) 命令としてはデータトランスファー命令のみを考える。

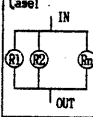
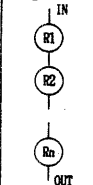
・Case 1. レジスタ-レジスタ間の転送命令

のみを考える。

- Case 2. IN-レジスタ, レジスタ-OUT 命令も考える。

(2) INからレジスタ, レジスタからOUTへは1ステップの命令実行で転送される。

表3はレジスタデコーディング機能のテストに必要なテストセットの大きさをケース1, 2に分けて示したものである。ここでは1Cellへの書き込み, 読み出しは1ステップの命令実行で行なえるとしている。(CellとはMEMにおけるアドレスでアクセスされるワードをいう。) テストセットの大きさは、 nI : 命令数, nR : レジスタ数, nR : Cellの数をパラメータとして示す。

Instruction Decoding & Control Function nI : 命令数 nR : レジスタ数		
Case 1	故障モデル	テストセットの大きさ
	$f(Ij / \neq Ij)$	$nI (nR + C2)$
	同上	$nI (1 + C2) +$ $C1 (4nR - 1)$ $(nI = nR + 1)$

C1: 各Registerの初期設定, 読み出しをするために診断回路を用いて行う処理
C2: 命令実行後サンプリングされた制御信号を正常判定する処理

表2. テストセットの大きさ1

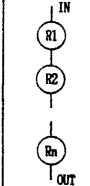
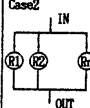
Register Decoding Function nR : Cell数		
Case 1	故障モデル	テストセットの大きさ
	Mapping fDR from R to R U {φ} Mapping fDW from R to R U {φ} 各CellのS-a-0,1	$nR (nR + 4)$
	同上	$10 (nR - 1)$

表3. テストセットの大きさ2

8. むすび

本論文ではファンクショナルテストのテスト生成における従来の方法の限界を指摘し、プロセッサ内に若干の診断回路を組み込むことによるテスト生成における効果等を考察した。

本研究の趣旨は、図11に示す様にゲートレベルテストにおいて採用されているスキャン方式のテスト容易性をファンクショナルテストの中に取り入れたことにある。

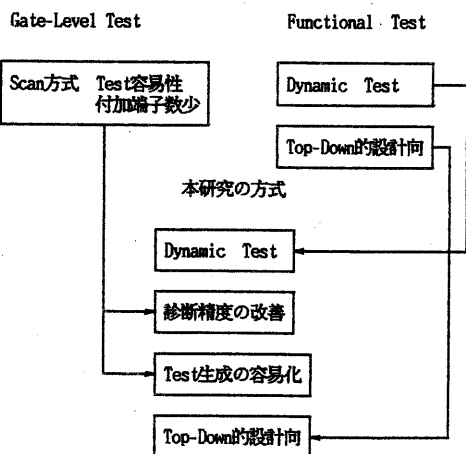


図11. 本研究の趣旨

本研究で採用した診断回路はスキャン方式と同様、シフトレジスタのテスト容易性等を利用したものであり、その回路構成法の持つ特徴は失なわれていない。例えば、複数プロセッサシステムにおいてはスキャン方式と同様、次の様な応用が可能である。

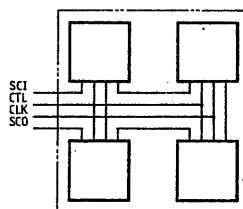


図12. 診断回路 応用例

今後、プロセッサの高集積化、高速化はますます進むであろう。この様な状況の中でゲートレベルテストに代るテスト生成法としてファンクショナルテスト生成の必要性がますます高まってくる。ファンクショナルテストをより一般的にするためには、ゲートレベルテストと同様に、機能仕様レベルでのテスト生成を手手によらず自動作成させる必要がある。そのためにはテスト生成アルゴリズムをより簡単化し、機械的に行なえる様にする必要がある。これらの実現のためには回路構成法へのフィードバックは不可避のものである。そして今後、ファンクショナルテストの検出率等の評価手法の確立も重要である。

謝辞

本研究を進めるにあたり終始適切な助言をいただきました東工大 情報工学科 当麻研究室の各位に心から感謝いたします。

又、本研究を援助していただきました富士通の関係者各位に心から感謝いたします。

文献

- (1) Case, G.R : "Analysis of Actual Fault Mechanism in CMOS Logic Gates", Proc. 13th Design Auto. Conf., PP.265-270 1976.
- (2) Wadsack, R.L : "Fault Modeling and Simulation of CMOS and MOS Integrated Circuits", Bell System Tech Journal vol.57, no.5, PP.1449-1474 May-June 1978.
- (3) S.M. Thatte : "Test Generation for Microprocessors", ILLINOIS UNIVERSITY May 1979.
- (4) E.B. Eichelberger, T.W. Williams : "An Enhancement to LSSD and Some Applications of LSSD in Reliability, Availability and Serviceability" FTCS 11 1981.
- (5) E.B. Eichelberger, T.W. Williams : "A Logic Design Structure for LSI Testability", Proc 14th Design Auto. Conf., PP.462-468 June 1977.
- (6) H. Ando : "Testing VLSI with Random Access Scan" Compcon 80.