

大型計算機の装置診断容易化支援システム

6V-6

(2) 故障辞書作成方式

†志賀 博 ††西田 隆夫 ††西根 裕久 †森 隆 †林 義明
††(株)日立製作所 ††日立電子サービス(株)

1. はじめに

装置診断技術における設計効率の向上、設計品質の向上を目的として、装置診断容易化支援システム CONDORを開発した[1]。本稿では、CONDORの主要機能の一つである故障辞書作成に関して、その処理方式を中心に報告する。

2. 障害検出回路とチェック領域

計算機システム内に組み込まれた障害検出回路をチェックラッチと呼ぶ。このチェックラッチの障害検出可能な領域をチェック領域と称する(図1)。計算機システム内で障害が検出されたとき、どのチェックラッチで検出されたかを知ることにより、そのチェック領域を故障被疑範囲として指摘することができる。故障辞書には、チェック領域内の部品(パッケージ等)が交換候補として記述されている。従って、故障辞書の精度を向上させる為には、チェック領域の確実な抽出が不可欠である。

次章では、チェック領域抽出のための基本となるトレース処理方式について説明し、さらに第4章では、大規模装置に対して大幅な性能向上を可能とする階層分割圧縮方式について説明する。

3. トレース処理方式

3.1 抽出開始点と停止点の決定方法

チェック領域を正確に抽出する為には、その抽出開始点と停止点とを正確にプログラムで認識する必要がある。

チェック領域抽出の開始点はチェックラッチであり、停止点は他のチェックラッチのチェック対象となるレジスタ(例えば、パリティチェック対象レジスタ:PCR)、自分自身のパリティを生成する対象となるレジスタ(PGR)、入力端子、伝播不能信号線などである。

論理記述した論理ファイルの中から抽出開始点、停止点を、プログラムで自動認識することは困難を極める。そこで、チェック領域開始点、停止点抽出のキーとなる信号を付加情報として与えてもらい、それをもとに以下の方法でプログラムで自動抽出する。

(1) チェックリセット信号に連なっているラッチを、チェックラッチとして抽出し、トレースの開始点とする。

(2) チェックラッチを起点として、ソース側へバックトレースして到達したラッチ群をパリティチェック対象レジスタとし、トレースの停止点とする。

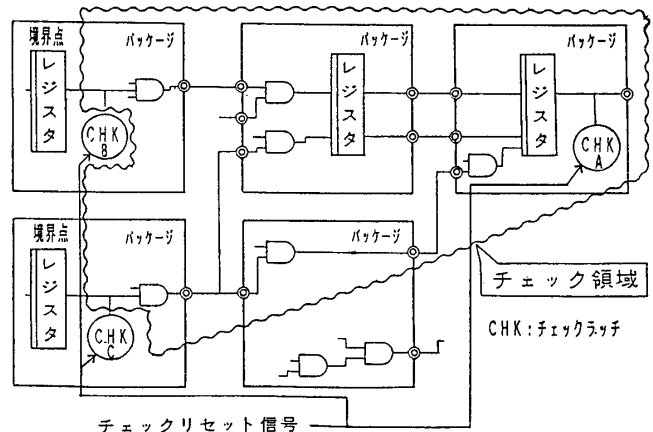


図1 チェック領域の範囲

(3) パリティジェネレータを認識し、ソース側へバックトレースして到達したラッチ群をパリティジェネレート対象レジスタとし、トレースの停止点とする。

(4) 装置の通常動作時に固定値をとる信号のために、故障の伝播が不能となる信号群を抽出し、そこを停止点とする。

3.2 チェック領域抽出方法

チェック領域は、チェックラッチを抽出開始点として、上述した停止点までバックトレースする事によって得られる領域である。

上述した開始点と停止点の他に、チェック領域抽出精度向上のために、以下に示す機能をサポートしている。

(1) コントロール信号線はPCRといったチェック領域の境界を持たず停止点が明確でないため、チェック領域が急激に拡散し、不必要な部品指摘を招く可能性がある。これに対処するため、トレース中にコントロール信号を検出した場合、チェックラッチからの通過トレース情報をもとに、トレースを停止する。

(2) タイミング信号や、スキャン信号など故障率の低い特殊信号線を検出した場合、トレースを停止する。

Diagnosis Systems for Large Computers

(2) Research on Fault Dictionary Generator

†Hiroshi SHIGA, ††Takao NISHIDA, ††Hirohisa NISHINE, †Takashi MORI, †Yoshiaki HAYASHI

††Hitachi, Ltd. †Hitachi Electronics Services Co., Ltd.

4. 階層分割圧縮方式

大規模計算機システムを一括して処理するためには、過大な主メモリ量、処理時間が必要であり、設計支援システムとしてのTAT(Turn Around Time)の保証が困難となる。これに対処するために、階層分割方式を採用した[2]。本方式は、対象装置の結線情報を、LSI内部とLSI外部との2階層に分割し(図2(a))、LSI内部の結線情報の中からチェック領域抽出に必要な論理要素のみを抽出し(図2(b))、圧縮する方法である[3]。本方式により主メモリ量と処理時間を、圧縮しない場合に比べて、1/10以下に削減することが出来た。

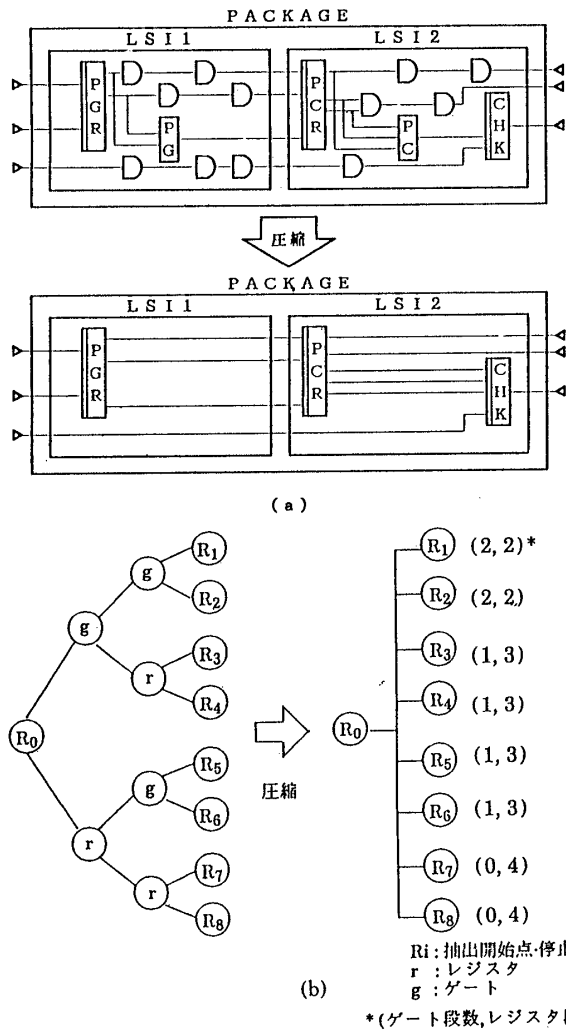


図2 階層分割圧縮方式

5. CONDORの処理フロー

図3にCONDORの処理フローを示す[4]。図に示すようにCONDORは、入力された論理ファイル内の結線情報をチェック領域抽出に必要な情報だけに圧縮し(階層分割圧縮方式)、中間ファイルを作成する。中間ファイル内に在るチェック領域抽出開始点からバックトレースをスタートし、抽出停止点までトレースすることによりチェック領域を抽出する。抽出したチェック領域内に在る部品(パッケージ等)を交換部品として故障辞書に出力すると共

に、故障辞書作成情報をリスト出力する。又、チェック領域抽出の結果得られたチェックラッチ性能、チェック領域に含まれない非チェック領域情報もリスト出力する[5]。

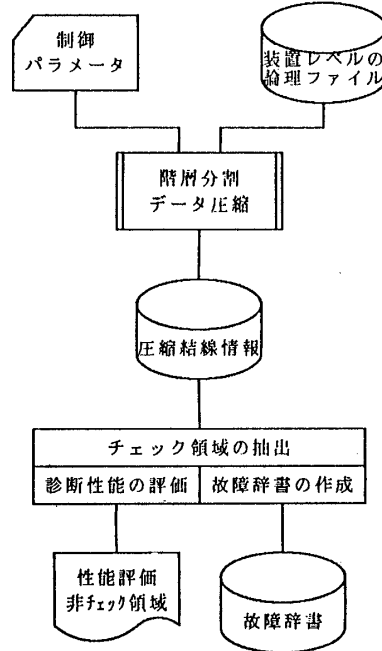


図3 CONDORの処理フロー

以上説明した処理方式により、CONDORは大規模な計算機の故障辞書を短時間で効率良く作成する事が出来る。

6. おわりに

CONDORを開発し、大規模計算機システムに適用した。その結果故障辞書作成に要する人工工数を1/10以下に削減することが出来た。また、階層分割圧縮方式を採用したことにより、主メモリ量、処理時間共圧縮しない場合と比較して1/10以下に削減することが出来た。今後は、チェックラッチ設計、検証の効率化、不良部品指摘分解能の向上等が課題である。

[参考文献]

- [1]西田 他;大型計算機装置診断支援方式について;設計自動化研究会1989.2
- [2]津布久 他;大型計算機装置の診断容易化支援システム;第20回FTC研究会1989.1
- [3]Y.Tsubuku,T.Nishida,H.Shiga;'Main Frame Diagnosis Support system',Proc.ITC,Aug.1989.
- [4]津布久 他;大型計算機の装置診断容易化支援システム(1)-支援システムCONDOR概要~;情報処理第39年全国大会,1989.10
- [5]西田 他;大型計算機の装置診断容易化支援システム(3)-性能評価指標の計算方式~;情報処理第39年全国大会,1989.10
- [6]H.Tanaka,M.Kawai,I.Sugasaki,T.Hakuba;'System Level Fault Dictionary Generation',Proc.Int. Test Conf.,pp884-887,1988.