設計自動化 46－14
（1989. 2. 21）

大型計算機装置診断支援方式について

西田隆夫、　西根裕久、　志賀博
（日立）　（日立）　（日立電子サービス）

計算機装置の大規模化に伴い、装置の信頼性の向上に対する要求はますます高まっており装置診断技術の向上が不可欠である。計算機が稼動中に発生した障害を見逃すことなく検出し、かつ正確迅速に不良部品の交換を可能とするためには、障害検出回路の設計等のハード面と故障辞書の作成等のソフト面の両面に対して設計品質の維持向上が必要である。

今後、一段と計算機の大型化に拍車がかかることを想定するならば、従来技術の延長線上での対処には限界を感ぜざるをえない。本文ではこのような背景に鑑み、装置診断技術のより一層の飛躍のためにハード、ソフト両面の設計効率の向上、設計品質の向上を可能とする設計支援方式について検討した結果を報告する。

Research on Diagnosis Systems for Large Scale Computers

Takao NISHIDA*, Hirohisa NISHINE*, Hiroshi SHIGA**
* Hitachi,Ltd.　**Hitachi Electronics Services Co.,Ltd
Kokubunji,Tokyo 185,JAPAN

With computer complexity increase, it becomes more and more difficult to detect and locate failures in a terget system. To overcome this problem, diagnosis system for concurrent error detection is seriously needed. Careful hardware and software design is necesarry for successful diagnosis. In hardware design, proper insertion of error detection circuits is required. In software design, accurate fault dictionary must be generated for fault analysis.

These hardware and software designs have been conventionally performed by skilled desiners with much labours. In this paper, we report research on the diagnosis system for large scale computers to increase design efficiency and to enhance design quality.

1．はじめに

近年の計算機装置の大規模化、高機能化、高速化に伴い、装置の信頼性の向上、保守性の向上に対する要求はますます高まっている。これに対処するためにはＬＳＩ等の部品レベルでの信頼性向上はもちろんであるが、装置全体としての検査、診断技術の向上が必須である。

計算機が正しく動作しているかのチェック（検査）と不当な動作をした場合にその原因の究明（診断）とは、計算機のライフサイクルー組立、稼動、保守ーを通して厳密に実施されなければならない。特に稼動中に発生した障害を見逃すことは計算機利用者に多大な損害を及ぼすことになる。また、障害を検出したとしても、その原因である故障部品を速やかに同定し修復できなければ、無意味である。このような背景から、計算機の稼動時に同時に検査が可能であり、また再現性の無い間欠的な故障に対しても診断が可能な ｺﾝｶﾚﾝﾄｴﾗｰﾁｪｯｸ診断方式の重要性が認識される。本診断方式では計算機内部に予め障害検出回路を組み込み、障害検出時に故障部品を指摘するための故障辞書を準備することが不可欠である。一方、障害検出回路の設計（ハード面）と故障辞書の作成（ソフト面）は特殊な設計知識が必要であり、熟練者の人手作業に依存する度合いが強かった。しかしながら、今後の大型計算機の飛躍的な大規模化に対しては、現状のままでは設計工数の急増、設計品質の低下が懸念される。

障害検出能力（検出率）は障害検出回路の配置方式に直接依存する。故障位置の指摘範囲（分解能）は障害検出回路の配置方式と故障辞書の精度とに依存する。故障部品の指摘精度（適中率）は故障辞書の精度そのものに依存する。すなわち、正確に効率良く装置診断を実施するためには、上述したハード面、ソフト面の両面にわたる設計支援が不可欠である。以下ではこれらのハード、ソフト両面の設計効率の向上、設計品質の向上を目的に設計支援方式について検討した結果を報告する。

2．装置診断の課題

計算機を装置のレベルで診断する目的として、（１）出荷前の動作確認、（２）ユーザサイトで稼動中に発生した障害に対する修復、（３）ユーザサイトでの予防保守等があげられる。ここでは特に（２）に重点を置き、その課題を以下に列挙する。

（ａ）ハードウェア設計
　(ｲ)障害検出回路設計工数の削減
　(ﾛ)障害検出回路の適正配置
（ｂ）ソフトウェア設計
　(ｲ)故障辞書作成工数の削減
　(ﾛ)故障辞書の精度向上
（ｃ）診断性能、機能
　(ｲ)検出率、分解能、適中率の向上
　(ﾛ)間欠故障に対する検出能力の向上

近年、特に装置の大規模化が進み、障害検出回路の適切な設計が困難となってきた。また、部品レベルでのテスト性能の向上と共に、装置レベルでは、間欠故障の比率が増大しており、間欠故障の検出能力が重視されている。従来、装置診断方式として、ＦＬＴ(Fault Locatin g Test)，ＭＤ(Micro Diagnostics)，ＬＯＡ(Log-Out Analysis)等が用いられてきたが、間欠故障に対してはＦＬＴ，ＭＤ方式は対処が困難であり、ＬＯＡ方式の重要性が高まっている。

このような背景から、以下では ｺﾝｶﾚﾝﾄｴﾗｰﾁｪｯｸ方式の一つであるＬＯＡ方式を中心に話を進める。第３章でＬＯＡ方式の概要を、第４章で上述した装置診断の課題に対処するために必要な装置診断支援システムのあり方について検討する。

3．ＬＯＡ方式概要

図１にＬＯＡ方式の開発工程と運用工程を概念的に示した。

（１）ハード設計

パリティチェッカ、チェックラッチ等の障害検出回路を組み込むと同時にその

検出率、分解能を評価し配置を改善する。

（２）ソフト設計

チェックラッチごとの障害検出領域（ドメイン）を抽出し、それをもとに故障辞書を作成する。また、故障辞書とログアウトデータ（障害検出時に退避されたラッチの状態）とから故障部品を指摘するＬＯＡ実行プログラムを開発する。

（３）障害検出、不良解析、修復

装置内部で障害が発生し、これが障害検出回路で検出されると、システムはログアウト機構を介してラッチの内容を退避した後、再試行を試みる。再試行に成功した場合には処理を続行するが、失敗した場合にはシステムを停止し、ＬＯＡ実行プログラムにより故障部品を見つけ出す。指摘された部品を交換することにより、処理を再開できる。

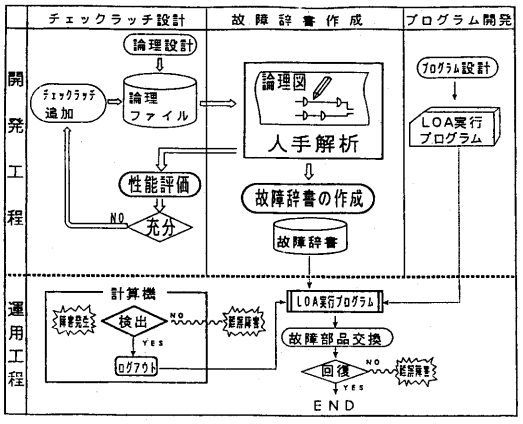

図１　ＬＯＡ開発のしくみ

故障検出もれ、交換部品の指摘もれは、難解障害として長時間のシステムダウンを引き起こしかねない。これを回避するためには前述した検出率、分解能、適中率の向上を目指して、ハード設計、ソフト設計の品質の向上が不可欠である。次章では装置診断の設計品質、設計効率の向上のために必要となる設計支援システムについて検討する。

４．装置診断支援システムに対する要請

本章ではＬＯＡ方式の性能を左右する障害検出回路の設計と故障辞書の作成に関して、その支援システムの必要機能、必要性能について検討する。

４．１　障害検出回路の設計支援

第２章で述べたごとく、今後の診断対象装置の大型化を考慮するならば、障害検出回路の設計品質向上が重要な課題となる。

第一に、設計した障害検出回路が期待通りに動作するか否か厳密に検証する必要がある。障害検出回路の正しさを保証するためには、論理シミュレータ等を用いた効率的な動作確認が必要である。

動作確認と並行して、障害検出回路の検出能力、分解能を定量的に評価し[1]、設計にフィードバックし、性能向上を図る必要がある。設計結果を基に、自動的に短時間で精度良く検出率、分解能等の評価指標を計算すると同時に、性能向上のための論理変更を補助するための情報を表示できる設計支援システムが望まれる。

４．２　故障辞書の作成支援

検出率、分解能の高い障害検出回路が正しく設計されたとしても、その設計結果を反映した故障辞書が不正確ならば、ＬＯＡ適用時に多大な問題を引き起こすため、精度の良い故障辞書の作成が重要な課題となる。

障害検出回路により故障の発生を検出可能な対象論理をチェック領域、あるいはチェックドメインと称する（図２参照）。装置で障害が検出されたとき、どの障害検出回路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みなすことができるので、その範囲に含まれるＰＫ（パッケージ）等の部品を交換の候補として絞り込むことができる。故障辞書とはこのように、エラーチェッカと交換部品候補の対応を記述したものである。従って、故障辞書の精度を向上させる為には、チェック領域の正確な抽

出が不可欠である。

従来、故障辞書の多くは人手により作成されていた。このため、ミスによる精度の低下が問題であった。精度良く、短期間で故障辞書の作成を可能とする支援システムが望まれる[2]。

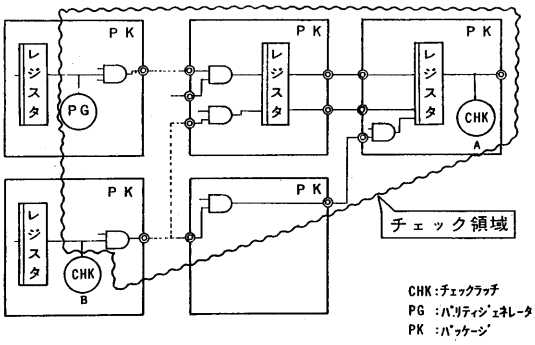

図2　チェック領域の範囲

4．3　支援システム構築上の留意点

前述したように、設計品質の向上のためには、障害検出回路の設計結果を基に、下記の処理を自動化した設計支援システムが不可欠である。

（a）検出率、分解能の算定
（b）性能改善に必要な補助情報の表示
（c）故障辞書の作成

以下ではこのような支援システムを構築する上で留意すべき点について記述する。

（1）入力情報に対する留意点

本支援システムの入力情報を準備するために派生する余計な作業に対して考慮すべきである。チェック領域を正確に抽出するためには、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。この開始点と停止点は障害検出回路を設計した時点では明白であるが、障害検出回路を論理記述した論理ファイルの中からプログラムで自動的に認識することは困難を極める。そこで、開始点や停止点等の設計情報を何らかの形で、プログラム側に伝達する必要がある。その手段としては、

・信号名の統一等の設計制約を設ける
・論理ファイルに記述を追加する
・新たに専用の設計ファイルを設ける

等が考えられるが、これらの作成作業を最小限に抑えることを留意すべきである。

（2）処理方式に対する留意点

検出率、分解能等の評価指標の計算が不正確であると、性能が不十分なまま設計が終了する恐れがある。あるいは逆に、必要以上に障害検出回路を含んだ高価な装置を設計する可能性もある。故障辞書の作成が不正確であると、障害検出時に不良部品の交換もれ、あるいは逆に不要な交換により、修復時間の増加をもたらす危険性がある。

すなわち、支援システムの精度が不十分であると装置の設計コスト、製造コスト、保守コストの増加を招くので、人手以上の精度を確保できるような処理方式を考案する必要がある。

（3）運用に対する留意点

本支援システムは設計の途中で繰り返して使用される。また、論理変更が発生した際にはそれに応じて故障辞書を作り直す必要がある。従って、このような運用に対処できるようにＴＡＴ(Turn Around Time)を保証する必要がある。

以上３つの側面から装置診断支援システム構築上の留意点を検討してきた。最後に特に注意すべきことはこれら３つの留意点は互いに相反する要素を含んでいる事である。従って、トータルなシステムとして、バランス良く、これらのトレードオフを解決することが最も重要であると考える。

次章では、以上の観点をもとに、我々が開発した装置診断支援システムであるＣＯＮＤＯＲ(Concurrent error checker Diagnosability Analyzer)[3]について、その概要を説明し、第６章で今後の課題について述べる。

5．ＣＯＮＤＯＲ概要

本章では装置診断支援システムＣＯＮ

ＤＯＲの用途、機能、処理方式について説明する。

５．１　ＣＯＮＤＯＲの用途

ＣＯＮＤＯＲは障害検出回路の設計の支援と故障辞書作成の支援を目的としたシステムである。障害検出回路の論理設計段階では検出率、分解能を評価し、性能向上のための指針を与える。論理設計完了時にはその結果をもとに故障辞書を自動的に生成する（図３参照）。

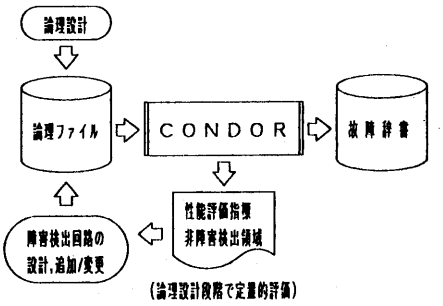

図3　CONDORの使用方法

５．２　ＣＯＮＤＯＲの機能

（１）　機能

ＣＯＮＤＯＲの主なる機能を以下に列挙する。

（ａ）検出率、分解能の算定
（ｂ）いずれのチェックラッチのチェック領域にも属さない領域（非チェックドメインあるいは非障害検出領域と称する）の表示
（ｃ）チェック領域の重なり状態の表示
（ｄ）各チェックラッチごとのチェック領域の表示
（ｅ）故障辞書ファイルの作成

論理設計時に（ａ）－（ｄ）の情報を参照することにより、障害検出回路の適正な配置を効率良く実現できる。

（２）　入出力情報

ＣＯＮＤＯＲの出力情報は（１）で示した（ａ）－（ｄ）のリスト類と（ｅ）の故障辞書ファイルとである。

入力情報に関しては、４．３節で述べた方針に基づき、論理ファイルと付加情報の２種とした。即ち、論理ファイルは論理設計結果を格納したマスタファイルであり、開発フェーズを通して一元管理保守される。ＣＯＮＤＯＲ運用時の可用性、信頼性、保守性を考えるならば、設計マスタファイルそのものを入力することが最も望ましい。ＣＯＮＤＯＲ用に新たに設計ファイルを設けることは、入力作業、記述ミスのチェック、論理変更に伴う保守等のための派生作業を必要とするため、我々はＣＯＮＤＯＲの主入力情報を既存の論理ファイルとした。一方、チェック領域抽出のための抽出開始点、停止点を論理ファイル中から自動的に誤り無く認識することは困難である。そこで、次節で説明するように自動抽出と人手抽出の折衷方式を採用することとし、人手で指定すべき項目を付加情報としてＣＯＮＤＯＲに入力する。付加情報の内容については次節で説明する。

５．３　ＣＯＮＤＯＲの処理方式

図４にＣＯＮＤＯＲの処理フローを示した。以下に主なる処理内容を説明する。

（１）階層分割データ圧縮

大規模な装置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の保証が困難となる。これに対処するために、階層分割データ圧縮方式を採用した。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、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、圧縮する方法である。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る。

（２）チェック領域の抽出

障害検出回路の構成上の特徴を利用して、チェック領域の抽出開始点であるチェックラッチと抽出停止点であるパリテ

ィジェネレート対象レジスタを抽出する。この抽出はあくまでも障害検出回路の標準的な回路構成を前提としており、それに準じない場合には、人手で付加情報としてその位置を指定する必要がある。

抽出開始点よりバックトレースすることにより、チェック領域を抽出することが出来る。しかし、上述した停止点だけでは範囲限定が十分でなく、チェック領域が拡散してしまう場合があるため、種々の拡散防止手段を講ずる必要がある。チェックラッチからの通過ゲート段数、通過ラッチ段数、通過ＬＳＩ個数等を用いて、トレースの打切りが可能である。また、人手で任意の信号線を停止点として指定することもできる。これらの、打切り段数や停止点を付加情報としてＣＯＮＤＯＲシステムに与える。

チェック領域抽出の結果を用いて、故障検出率と分解能を計算する。また、各チェックラッチのチェック領域内に含まれる部品情報を故障辞書として登録する。チェック領域内のゲート数をチェックラッチからの距離で重み付けした値に基づき、部品の交換優先順位を決定する。

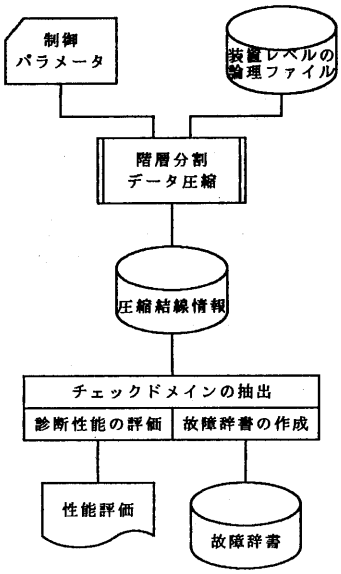

図4　CONDORの処理フロー

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できるだけでなく、障害検出回路の設計効率の向上、設計品質の向上に寄与できる。

６．今後の課題

ＣＯＮＤＯＲを用いることにより、障害検出回路の設計効率、設計品質の向上と故障辞書作成工数の大幅削減が可能となるが、装置診断全体を対象に、より一層の体系化、自動化を図るために、今後以下の課題に対して検討を進める必要がある。

（１）障害検出回路の設計、検証の効率化
（２）不良部品指摘分解能の向上
（３）ハード面、ソフト面を含めたＬＯＡ方式全体の検査の効率化

特に（２）に関しては、障害発生時の交換単位が大規模化し、搭載部品数が増えるにつれ、修理コスト、修理時間が増大するため、分解能の向上が重要である。

７．おわりに

装置の稼動時に障害の検出が可能であり、また間欠故障に対しても診断が可能な ｺﾝｶﾚﾝﾄｴﾗｰﾁｪｯｸ 診断方式を実現するためには、障害検出回路の設計と故障辞書の作成が必要である。これらの設計効率向上、設計品質向上のためには支援システムが不可欠であり、我々はその一端としてＣＯＮＤＯＲシステムを開発し、その有効性を確認することができた。今後は分解能の向上等が課題である。

［参考文献］

[1] P.M.Almy et al.;Using Error Latch trace to obtain Diagnostic Information;DAC,1981,355-359pp

[2] T.Hidetoshi et al.;System Level Fault Dictionary Generation;ITC,1988,884-887pp

[3] 西田他；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