

進化型ハードウェアを用いた自己修復機能の実現手法について

5 J - 4

平野 健司 新井 浩志  
 千葉工業大学 大学院工学研究科 電子工学専攻

1. はじめに

デジタルシステムの利用分野の拡大に伴い、ハードウェアコンポーネントにも高い信頼性が求められている。従来より、一部の構成要素に故障があった場合にその故障を検出し、それを予備要素に置き換え、故障発生前と同等な機能を果たすフォールトトレラントシステムが提案されている<sup>[1]</sup>。これらは比較的粗い粒度での自己修復機能を実現する手法である。これに対して近年は、書き換え可能なデバイスを用いた進化型ハードウェアによって細かい粒度での自己修復機能を実現するための研究がおこなわれている<sup>[2]</sup>。

ハードウェアの自己修復、自己複製を目標としたものとして Bio Watch というハードウェアが提案されている<sup>[3]</sup>。しかし、故障検出の方法としては BIST(Built-in self test)を想定しており、システムが稼動中に故障を検出することは困難である。また自己修復作業中は機能を停止せざるを得ないという問題がある。本報告では故障検出を自動でおこない、自己修復中も稼動させることが可能な自己修復機能について提案する。

2. システム構成

本手法では、いくつか機能ブロックが逐次的に処理するようなハードウェアを想定する。今回はこのようなハードウェアの例として、カウンタとデコーダからなる単純な回路を用いる。図 1 に対象システムの構成を示す。この図において、左から A, B, C, D 列, 上から  $\alpha$ ,  $\beta$ ,  $\gamma$  行とし、各機能ブロックをセルと呼ぶことにする。

各セルは内部にカウンタとデコーダと予備セルとしての機能を持っており、縦横ともに隣のセルの機能を把握することによって、その機能を切り替える。予備セルは入力信号をそのまま出力する。

本システムの初期状態では、A 列の 3 つのセルが

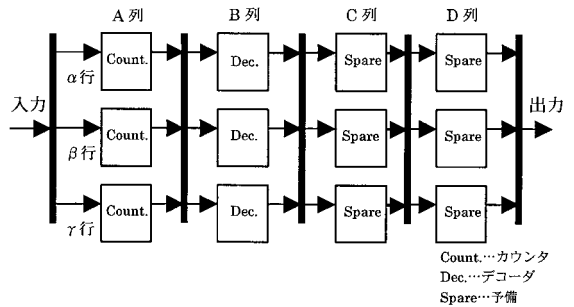


図 1 : システム構成

カウンタ、B 列の 3 つのセルがデコーダとなり、その他のセルはすべて予備のセルである。本手法では、各列の 3 つのセルで同機能を同時に動作させる。また、各セルには機能モジュールの他に他セルからのデータと自セルの出力データを比較する、比較モジュールを設ける (図 2)。各セルは自セルの出力と他の 2 つのセルの出力とを比較して、その出力の多数決を取ったうえで、多数派の最上行のセルが信号をバスに出力する。

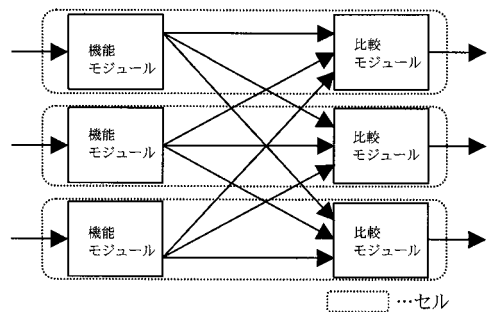


図 2 : 故障検出

3. 故障検出

比較モジュールは、自セルのデータが他の 2 つのセルの出力データと一致していれば、正常に動作していると判断する。自セルのデータが他の 2 つと異なった場合、自セルが故障状態であると判断する。故障状態になったセルはデータを透過させる。

この時に、その列の出力データは、故障状態でない他の2つのセルのデータとする。3つのセルのデータが全て異なっていた場合には、本システムは動作不能となる。

#### 4. システム再構成

データ比較の結果、故障が検出された場合、故障したセルが含まれる列以降のすべてのセルで再構成作業をおこなう。各セルは縦、横の隣接するセルの機能を監視しているため、ある行が再構成をして機能モジュールの配置が変わった後でも、自分が何の機能になればよいか判断することができる。これは、同じ列のセルは同機能になることと、本システムがA列の方からカウンタ、デコーダ、予備の順番で並ぶことによる。本システムでは以下の2段階の再構成をおこなうことによって、再構成作業中もシステム全体としては常に正しい値を出力する機能を実現する。

##### Step1. 故障セルを含む行の再構成

故障が起きた場合はまず、故障セルがある行の再構成をおこなう。故障セルよりも右にあるセルは左隣のセルの機能を参照しながら自分の機能を再構成していく。予備セルまで順次切り替わることによって、故障セルのある行の再構成は終了となる。

故障セルを含む行の再構成終了後、それまでのように、多数決で出力をとると、故障セルがある列は故障していない2つのセルの出力が多数派とみなされ、この信号が再構成が終了したセルにも流れてしまう。これを避けるために、故障セルがある行の再構成が終了した時点で故障セルがある行の出力をバスを介さずに直接隣のセルに流すように切り替え、システム全体としては再構成が終了した行の出力を最終的な出力とすることとする。

##### Step2. 故障セルを含まない行の再構成

故障セルを含まない他の2行のセルは、Step1の再構成が終わった行のセルを見て、機能を変える必要性を認識する。しかし、いつの時点でStep1が完了したのかを認識できない。そこでこれらの行のセルは、自分の行の右隣のセルの機能が切り替わっていない間は自分はまだ機能を変えられないと判断する。よって最初に予備セルが再構成される。予備セルは上下のセルをみて、機能を変えなければならないと判断する。その後、予備のセルが切り替わったのを見て、その左側のセルが機能を変えるというこ

とを順次繰り返していく。そして故障したセルと同じ列のセルは、故障セルをみて、自分は機能を持たずに休眠状態（故障でも予備でもなく信号をそのまま透過させる）へ変化する。各セルが機能を変更した後、すぐに各列で多数決によって出力を決定することはできない。そのため、左側からの出力をうけて多数決を採れるようになった時点でバス出力に切り替えていく。

例として、図1のB列 $\beta$ 行のセルが故障した場合を図3に示す。以下では $m$ 列 $n$ 行のセルを $(m, n)$ と表すことにする。 $(B, \beta)$ が故障と判断されると、予備であった $(C, \beta)$ がデコーダとなり、 $\beta$ 行の再構成が終了する。

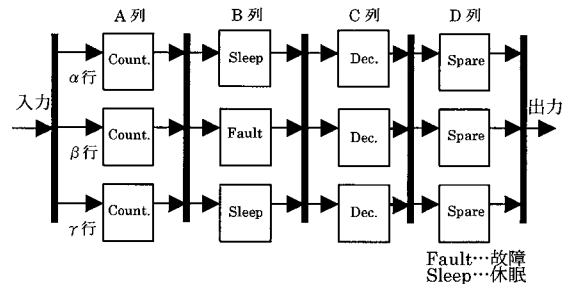


図3：(B,  $\beta$ )が故障した時の再構成後のシステム

$(C, \alpha)$ ,  $(C, \gamma)$ は $(C, \beta)$ がデコーダになったのを見て、自分がデコーダになると判断する。 $(B, \alpha)$ ,  $(B, \gamma)$ は、 $(C, \alpha)$ ,  $(C, \gamma)$ がデコーダになったことと $(B, \beta)$ が故障中であることを認識したのちに休眠状態となる。

#### 5. おわりに

本稿ではカウンタやデコーダなどの回路の出力を比較することによる故障検出や自己修復機能の実現手法について述べた。しかし、順序回路の内部状態の複製までは至っていない。そのため、今後の課題としては順序回路の内部状態の複製を含んだ自己修復機能の実現があげられる。

#### 参考文献

- [1]当麻, 南谷, 藤原, “フォールトトレラントシステムの構成と設計”, 横書店 (1991).
- [2]樋口, “進化型ハードウェア”, 情報処理 Vol.40, No.8, pp.795-800, Aug. (1999).
- [3]Sipper, Mange, Sanchez, “Quo Vadis Evolvable Hardware?”, communications of the ACM, Vol.42, No.4, pp.50-56, (1999).