

C_018

自己組織化ハードウェアによる耐故障性向上手法の CPU への適用について An Example of Fault Tolerant CPU based on Self Organization Hardware

実方 友裕† 新井 浩志†
Tomohiro Jitsukata Hiroshi Arai

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るようになってきている。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^[1]。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自己修復機能を実現するための研究が行われている^[2]。書き換え可能デバイスは任意の機能を代替できるため、耐故障回数に対して用意すべき予備要素の規模を少なく抑えることが可能である。

我々は、自己組織化ハードウェアを用いた耐故障性の向上について研究している^[3]。自己組織化ハードウェアとは進化型ハードウェアの一種であり、複数のセルから構成され、各セルが自分自身の機能を自律的に決定する機能を持つハードウェアである。

従来我々は、多段のパイプライン状の処理に本手法を適用してきた。本報告では、自己組織化ハードウェアを CPU に適用する際の問題点と解決策を提案する。

2. 自己組織化ハードウェア

従来の我々の研究ではパイプライン状の処理を対象としてきた。パイプライン状の処理は、いくつかの機能ブロックを経る逐次的な処理であり、一方方向にデータを処理するという限定的なものである。

自己組織化ハードウェアで F1~F4 の 4 つの機能ブロックから構成されるシステムを構築した場合の例を図 1 に示す。この図では左から A~F 列の 6 個のセルを用いて構成されている(図 1(a))。Fn は n 番目の機能を持つセル、SP は入力信号をそのまま出力する予備のセルである。各セルは外部に故障検出ユニットを持つものと仮定する。F2 のセルが故障した場合の自己修復の流れを以下に示す。

Step1. 故障したセルの機能停止

故障を検知した B 列のセルは、自身の機能を停止し入力側から送られてくるデータを処理せず直接右隣のセルに出力する(図 1 (b))。

Step2. セルの再構成

B 列のセルの機能を補うために C~E 列のセルは自身の機能を F2~F4 へと再構成する(図 1(c))。

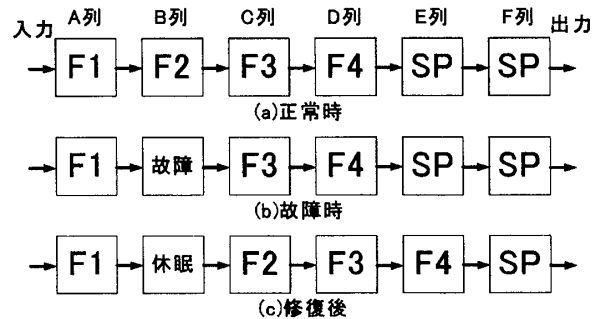


図 1: 自己組織化ハードウェア

3. CPU への適用

本報告では自己組織化ハードウェアを CPU に適用することを考える。CPU を自己組織化ハードウェアで構成するにあたって、以下の 2 つの問題を解決する必要がある。

まず第一に従来の自己組織化ハードウェアでは隣接したセル同士のみ通信しか許されていないという点である。これに対して CPU は内部の複数の機能ブロック間で様々な信号伝達を行う必要がある。そこで各セルはバスを経由して信号の伝達をおこなうこととする(図 2)。

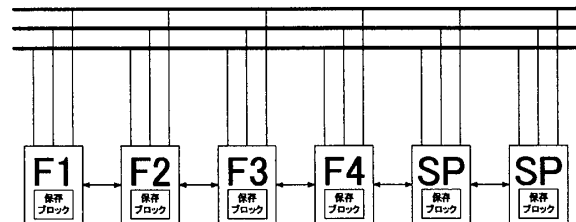


図 2: バス型自己組織化ハードウェア

第二に従来はストリーミング形式の処理を想定していたため、故障時におけるデータの破損、取りこぼしを許容できたという点である。CPU では故障から修復する際に一部のデータが消失することは致命的である。

よって図 2 のようにセル内部に左隣のセルにある処理中のデータを逐次保存するための保存ブロックを設ける。そしてセルを再構成した後は保存ブロックに保存されているデータを用いて処理を再開する。これにより故障前の状態から処理を続行することが可能となる。

現在教育用 8 ビット CPU である KUE-CHIP2^[4]を対象として本手法を用いた自己組織化 CPU の設計を進めている。

†千葉工業大学 工学研究科 電気電子情報工学専攻
Department of Electrical, Electronics Engineering and
Computer Science, Graduate School of Engineering, Chiba
Institute of Technology.

KUE-CHIP2を解析した結果、68ビットのバスと各セルに44ビット分の保存ブロックが必要になることが分かった。これは実装するうえで実現可能な許容範囲内であると考えている。

KUE-CHIP2を自己組織化ハードウェアとして分割設計した結果の各機能ブロックの内容を表1に示す。

表1: KUE-CHIP2の機能ブロックへの分割結果

機能ブロック	F1	F2	F3	F4
機能	Rgu	Clk_gen Sync	Cont Ir_idc	Pc_mar

4. 有効性の検討

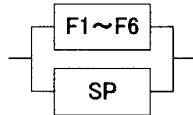
本節では、従来の単純な2重化システムに対して、本手法の有効性を検討する。以下では、システム全体を6個の機能ブロックに分割するものとする。様々なシステム構成のバリエーションを図3に示す。

図3(a)は単純にシステム全体を2重化した場合である。一般にシステムをn重化すると、n-1回の故障に耐えることができる。

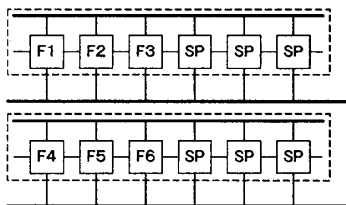
これに対して図3(b)はシステム全体を6個のセルに分割し、さらに6個の予備のセルを用意した自己組織化ハードウェアである。この場合、(a)と同等のロジック規模で6回の故障に耐えることができる。

しかしより大規模なCPUを自己組織化ハードウェアで実現しようとするときバスの数が増大し実現が困難になる。そこでセル同士の接続が密な機能を集めてブロックを構成し、ブロック単位内で自己修復する方法が考えられる。この例を図3(c)に示す。この構成は(b)の構成を2個のブロックに分けて自己組織化した例である。この例では、最低3回、最大6回の故障に耐えることができる。この場合(b)よりも耐故障性が低下するが、より大規模で複雑なシステムに本手法を適用可能になる。

さらに図3(d)では(c)の構成に予備のブロックを導入している。これによりあるブロックの予備セルを全て消費したとしても自己修復が可能となる。この場合、最低3回、最大5回の故障に耐えることができる。



(a)単純な2重化



(c)ブロック分割

以上、各方式のシステム全体の稼働率Pをまとめたものを表2に示す。この表においてpは1セルの稼働率、nは多重化した数、sはブロック内の予備のセル数、bはブロックの数、cは予備のブロック数を示す。

表2: 稼働率

構成(図3参照)	稼働率P
(a)	$1 - (1 - p)^n$
(b)	$1 - (1 - p)^{s+1}$
(c)	$\{1 - (1 - p)^{s+1}\}^b$
(d)	$1 - (1 - p)^{s+c+1}$

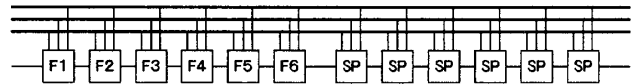
表2の全ての場合についてロジック規模を同等にしたと仮定すると、(b)が最も稼働率が高くなる。しかし大規模なシステムでは多くのバスを必要とするため現実的ではない。そこで(c)のように稼働率と実現度のバランスを取る必要がある。また(d)のように予備ブロックを導入することによりさらに稼働率を向上させることができる。

5. おわりに

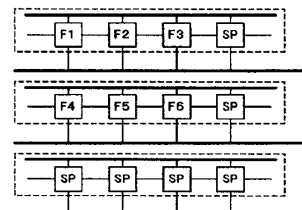
今後は本システムを実装し、その有効性を確認した上で、BIST等による故障検出システムについて検討を重ねる予定である。また本手法を実用規模のCPUへ適用することも検討中である。

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”, 榎書店, 1991.
- [2]樋口哲也, “進化型ハードウェア”, 情報処理, Vol.40, No.8, pp.795-800, 1999.
- [3]椿龍也, 新井浩志, “自己組織化ハードウェアによるフォールトトレラントシステムのための自立制御機構についての研究”, 情報処理学会第67回全国大会, 2ZB-2, 1-143, 2005.
- [4]神原弘之 他, “KUE-CHIP2 設計ドキュメント” ASTEM RI/Kyoto, 1993.



(b)自己組織化



(d)予備ブロックの導入

図3: 様々な耐故障のためのセル構成