Hardware Fault Tolerance through Artificial Immune System

Vadim Kataev
vkataev@upb.de

University of Paderborn

June 11, 2008

Abstract

A modern approach to hardware fault tolerance is presented. This approach is based on the principles similar in many respects to the human immune system. Immune system protects an organism from infection by pathogens. Immune system is composed of a complex constellation of cells. Analogous principles are used in the design of hardware fault tolerance. This paper introduces some novel methods for providing fault tolerance in the design of state machine based hardware systems.

1 Introduction

Exponential growth of the complexity in areas as varied as electronics of all kinds, communications, and software, is observed since many years. Human-made systems demonstrate high effectiveness and they provide us with required functionality which is often not directly accessible from the nature. In comparison with some natural systems, human-made systems are not very reliable. Even if a small piece of a regular human-made system accidentally stops to work, full system does not operate properly.

Biological natural systems is observed to have an amazing stability. They have decentralized architecture, providing at the same time a complex behaviour. The cost to have such properties is huge - a reliable system usually has redundant information about itself. Even essential faults cannot stop operability of the full system, because

information about structure of the whole system and algorithms to self-repair are kept in decentralized manner.

Natural immune system makes life of a complex organism more stable to possible damages generated by a complex environment. Having the capability of extracting information from the infectious agents and making it available for future use in cases of re-infection by the same or a similar agent, the immune system demonstrates properties similar to the intelligent behaviour. Information is extracted, processed and is enabled for future decisions.

Todays typical integrated circuit has up to several millions transistors. Normal operations require special narrow environmental conditions in order to operate without faults and damages. Fault avoidance is not a practical realization. Faults better to be tolerated and their effects to be minimized. This can be done effectively by processes similar to the biological immune system.

General immune-inspired design principles, negative selection algorithm, and feature mapping are discussed in Section 2. Section 3 introduces existed methods and approaches to hardware fault tolerance. Section 4 demonstrates modern self-repair approaches. Results are presented in Section 5. The paper concludes in Section 6.

2 Design

Typical electronic hardware is a system which includes many state machines. Each state machine is to operate only in a valid way. Any faulty state is to be avoided. Fault tolerant systems need to be capable to recognize and to process a fault state of the system.

Unlike in the natural immune system, where B cell highlights an invader and signals to T cell to kill an invader, an electronic solution consists of forcing the state machine back from a state recognized as 'faulty' to a known safe state.

Mapping biological immune system to electronic hardware helps to identify correctly and effectively a possible invalid state of any state machine. This approach can help to exclude a faulty state machine from the normal operations in the whole hardware system. It has to return a faulty state machine to one of the states known as a valid.

Many functional systems are complex heterogeneous structures. Direct validation of all states and conditions would require enormous amount of the spacetime. Significance of nonself and negative selection methods increases with the increase of system's complexity.

Immune models for hardware fault tolerance based on the negative selection algorithm have being widely accepted.

2.1 Negative selection algorithm

The system has to distinguish between self, i.e. elements that belong to the system, and non-self, i.e. elements that does not belong to the system. Valid states are considered to be self data. They are commonly defined as the bit strings or self-

vectors. Negative Selection algorithm is instructed to identify any changes in self data (also known as protected data).

The algorithm has two phases:

1. Set of detectors must be created. Each detector is a vector different to each of the self-vectors. Special threshold is used to determine an affinity factor. Hamming or Euclidean distance can be used in order to define the similarity between two vectors.
2. Monitor phase includes constant comparison of detectors with self-vectors. Matching condition indicates that non-self state is presented.[1]

Following simple python code demonstrates one cycle of the monitor phase.

```
selfVectors=[ [1,0,1,1], [1,1,1,0] ]
detectors=[ [1,0,0,0], [0,0,1,0] ]
for vector in selfVectors:
  if vector in detectors:
    nonselfDetected()
```

2.2 Feature mapping

Many features of the immune system can be mapped to the hardware. Following tables summarise feature mapping for the hardware fault tolerance.

Immune System	**Hardware Fault Tolerance**
Self	Acceptable state/state transition
Non-self (antigen)	Invalid state/state transition
Antibody	Error tolerance conditions
Genes	Variables forming tolerance conditions
Paratope	Invalid state/transition verification conditions
Epitope	Valid state/transition verification conditions
T-Helper	Recovery procedure activator
Memory cell	Sets of tolerance condition

Table 1: Entity feature mapping[1]

Immune System	Hardware Fault Tolerance
Recognition of self	Recognition of valid state/state transition
Recognition of non-self	Recognition of invalid state/state transition
Ontogenetic learning	Learning correct states and transitions
Humoral immunity	Error detection and recovery
Clonal deletion	Isolation of self-recognized tolerance conditions
Inactivation of antigen	Return to normal operation
Life of an organism	Operation lifetime of a hardware

Table 2: Process feature mapping[1]

3 Immunotronics and Embryonics

Immunotronics means immunological electronics. This is design of electronic systems inspired by the natural immune system. Embryonics is embryo-electronics. Hardware is represented by multicellular structure, which has many important properties of decentralized development.

3.1 Centralized approach

In centralized case, a separate Immunotronic monitor is used to control a separated state machine. It has to be trained to differentiate between self(valid) and non-self(invalid) transitions of the state machine.

In order to create an array of self-vectors, immune system should undergo a test phase. During this phase, immune system learns what is self, how full system works, its structure, and its states. This information helps to define a list of antibodies patterns that identify non-self operations.

- Special Software/Hardware testbench is used to define self-data from the state machine. A number of loops with altering information on the inputs of the state machine is generated. Systems outputs are inspected and self-data is collected.
- Tolerance conditions are generated and analysed. Self vectors are inputed into a greedy detector generator. Tolerance conditions are produced for all potential match lengths.
- Tolerance conditions are downloaded into the hardware immune system, which monitors the state machine.
- Current state of the state machine is extracted in the form of a bit vector and is analysed by hardware immune system. It uses a partial matching content addressable memory (CAM) in order to validate the extracted vector. No match indicates valid state: in this case OK signal needs to be sent. [1] If a match is

[1]For instance, nothing must be sent if the signals are clocked appropriately

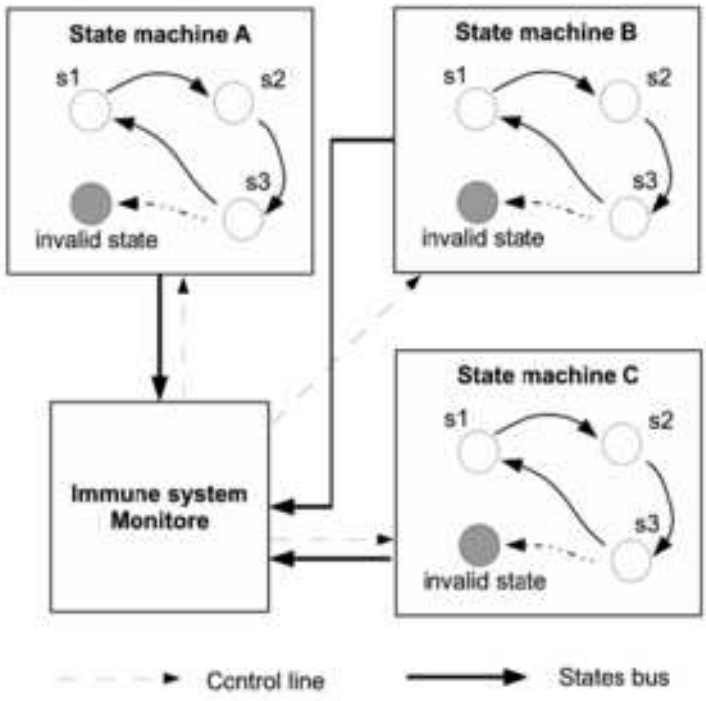

Figure 1: Hardware-based immune system for the fault tolerance

found then the immune system sends KILL signal that has to be processed by controlled system independently (i.e. controlled system is immunised).[2]

3.2 Distributed approach

More effective decentralized implementation is based on the multi-cellular architecture.

The embryonic array is distributed structure that has no central control unit. Embrionic array is self-capable to spot possible fault states. Thanks to its multi-cellular decentralized architecture, wrong operating cells could be excluded relatively painless from the whole operating array. Like in a biological multi-cellular organism, corrupted cells die and other cells replace their functionality.[3]

Special lymphatic network of immune cells integrated into the embrionic array helps to increase reliability of the full system. Having immune cells inserted in a regular way between embryonic cells, more than only one immune cell controls any embryonic cell (Fig. 3). Immune cells monitor and evaluate states of each embryonic cell. Each immune cell controls more than one embryonic cell. High redundancy of the control on an embryonic cell from surrounding immune cells requires a mechanism of subordination. Majority vote dictates the behaviour of an embryonic cell.[4]

Three important communication channels are used:

- Embryonic channel connects embryonic cells only.

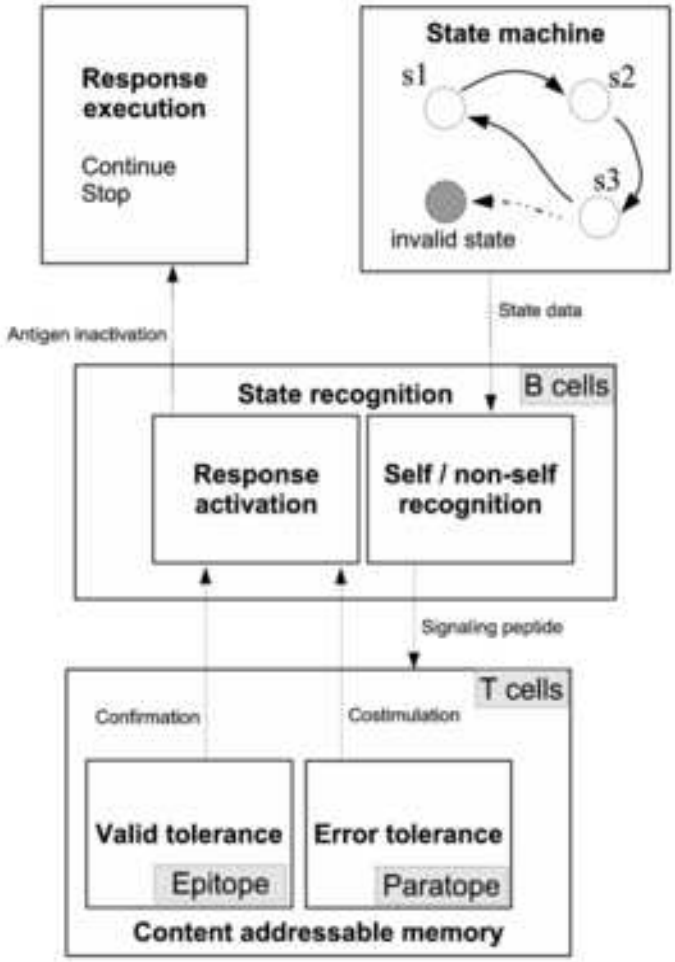

Figure 2: Fault recognition stage[2]

- Lymphatic network serves to transfer information between immune cells.[5]
- Trans-layer communications channels enable immune cells to control embryonic cells.

Depending on implementation, either a row or a column of the embrionic cells can be switched off from the normal work. The reason is technological complexity of the accurate components interconnection.

3.3 The Cell

Main blocks of a typical immune cell are usual hardware components:

- Serial Counter
- Compare Unit

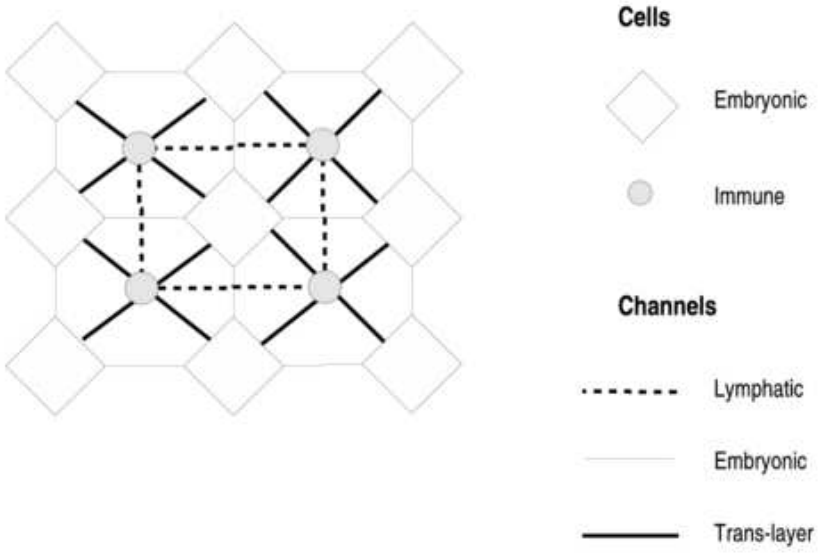

Figure 3: Immune cell controls four surrounding neighbours

- Memory Unit
- Multiplexers
- Demultiplexer

Considering that each immune cell has in its neighborhood at least four embryonic neighbours, it stores information about correct configuration of the four neighbouring cells (Fig.4). Thus, each of the immune cells stores self-tolerance conditions. This vectors are constantly compared with current states of the neighbouring embryonic cells. [2] If a vector matches, OK signal is generated, a mismatch creates a KILL signal. Such signals are to be processed further by an embryonic cell which is to be connected through the trans-layer communication channel to a corresponding output of the immune cell.

Embryonic cells in addition to some functional mission, include parts responsible for trans-layer communication and for decision making, based on majority vote.[3]

Assuming that initially an embryonic array works correctly, an initial configuration is read directly from the surrounding embryonic cells during the initialisation

[2]Potentially it could extremely slow down a work of the full system. To avoid this effect, comparison mechanism can work on its own independent clock.

[3]In the model with 4 immune neighbours surrounding one embryonic cell is to avoid a situation where two cells have opposite views with other two cells. The simplest solution is to have only three connected immune cells.

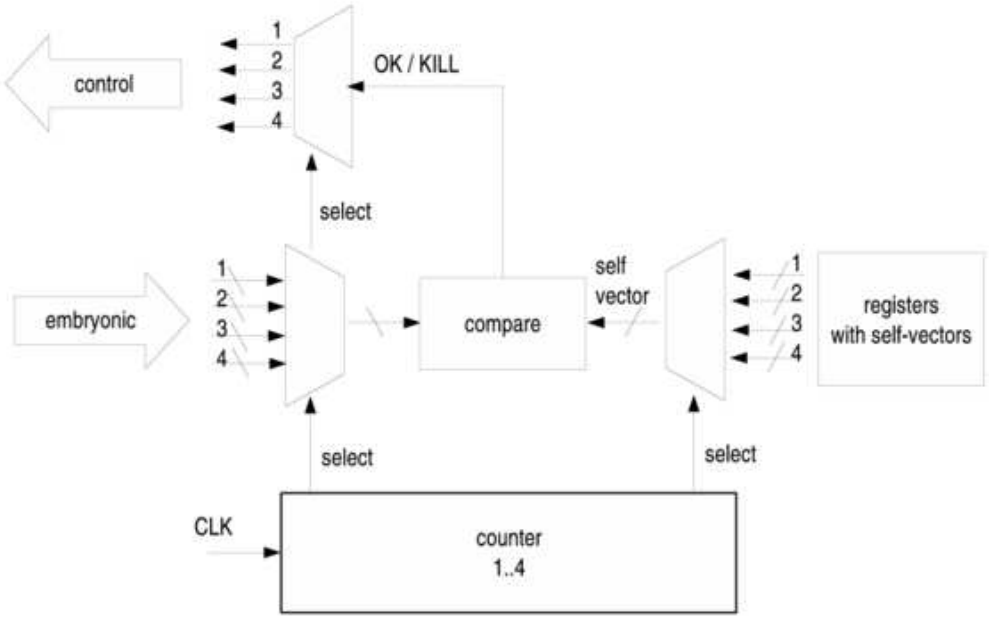

Figure 4: Immune cell structure

phase.

4 Self repairing cellular architecture

The need to recover faulty hardware and return it to full functionality quickly and efficiently is great. Mechanism for self-repair is based on faulty cells elimination.

Like biological cells, each artificial cell contains information about whole system. Depending on its position in the system, each cell have access only to corresponding part of this information, providing cellular differentiation. Thus, such superabundance allows each cell to change its behaviour in the emergency cases, such like damages or operational faults.

Duplication of critical system components is based on the addition of redundant cells to the system. Functional behaviour of an incorrect running cell can be replaced by its neighbouring redundant cell.

The simplest of possible methods is following. Providing there are spare cells available, self-repair at the cellular level is accomplished by deactivating a column containing a faulty cell. All cells to the right of a excluded [4] column will renew their behaviour, changing their functions (Fig.5).

[4]Column is disabled by switching off corresponding lines

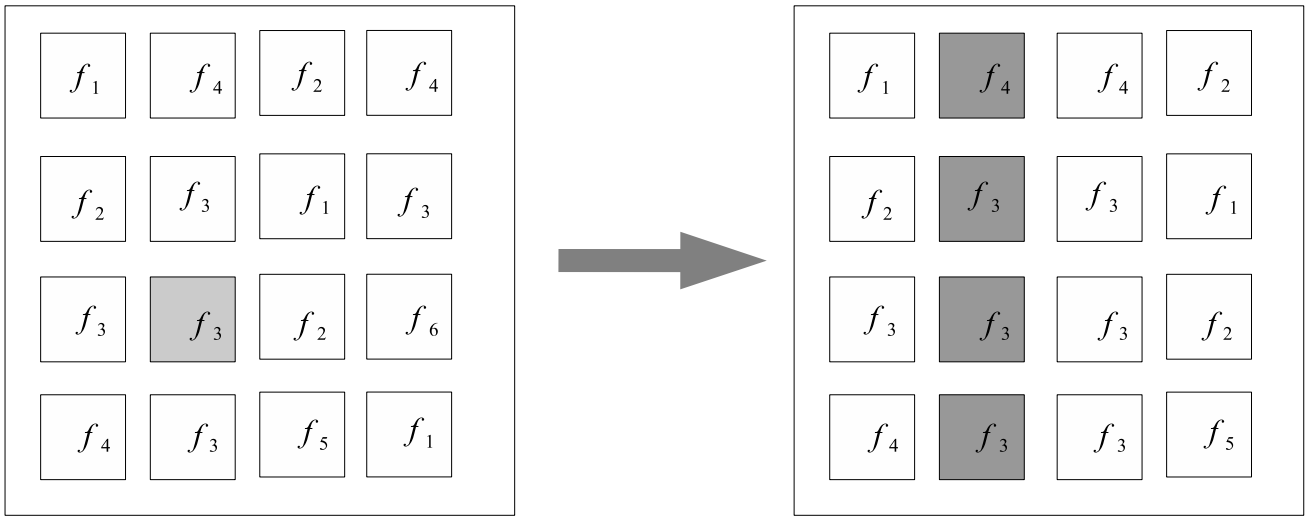

Figure 5: Self-repair of the cellular level through column shift

5 Results

New design improvements and existed problems are currently discussed in the research community.

Some researchers propose to use one simple innate immune cell for every embryonic cell of embryonic array.[5] They give an example of the case of concurrent monitoring, when computation loading of the immune cell which controls four its neighbours may be extremely large.

All of existed implementations of hardware fault tolerance inspired by the immune system have limitations. An important one is imperfect detection of invalid conditions.[1]

6 Conclusion

Current implementations of the hardware fault tolerant systems for the most part are FPGA-based. They use negative selection algorithm and finite state machines.

This paper has introduced immunology based approach in the field of hardware based techniques for improving fault tolerance. Inspired basically by the biological immune systems, hardware solutions can solve many problems which are considered as limitations for the further growth of reliability of the hardware electronic systems.

References

[1] D.W.Bradley, A.M.Tyrrell *Immunotronics—Novel Finite-State-Machine Architectures with built-in self-test using self-nonself differentiation.* IEEE Transactions on evolutionary computation, vol.6 no.3 June 2002

[2] D.W.Bradley, A.M.Tyrrell *Immunotronics: Hardware fault tolerance inspired by the immune system.* ICES 2000, LNCS 1801, pp. 11-20 2000

[3] R.Canham, A.M.Tyrrell *A hardware artificial immune system and embryonic array for fault tolerant systems.* Genetic Programming and evolvable machines, 4, 359-382 2003

[4] R.Canham, A.M.Tyrrell *A learning, multi-layered, hardware artificial immune system implemented upon an embryonic array.* ICES 2003, LNCS 2606, pp. 174-185 2003

[5] X.Zhang, G.Dragffy, A.G.Pipe, Q.M.Zhu *Artificial innate immune system: an instant defence layer of embryonics.* ICARIS 2004, LNCS 3239, pp. 302-315, 2004 2004