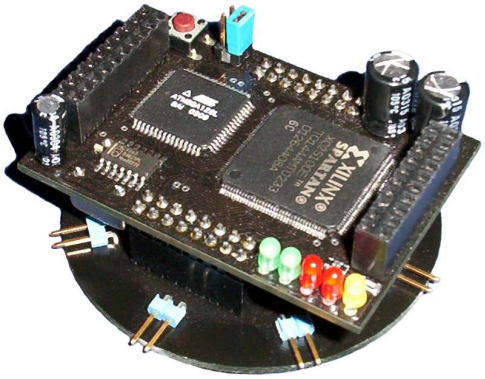
A Reliability Engineered Multicellular Architecture Inspired by Endocrinology:

The BioNode System

Andrew John Greensted

Ph.D. Thesis



THE UNIVERSITY of York

Bio-Inspired Architectures Laboratory,
Intelligent Systems Group,
The Department of Electronics,
Heslington, York, YO10 5DD
http://www.bioinspired.com

Thesis committee:
Prof. Andy Tyrrell, University of York
Dr. Steve Smith, University of York
Dr. Geof Carpenter, Aston University

Submitted: 30th September 2004

To Dad,

Every time I put up a shelf, hold open a door, or take pride in a job well done I will remember you. I have a great deal to thank you for.

With much love and admiration,
Andy

Abstract

There exists within us all a complex control system that continually manages the healthy state of our body. Mediated by a variety of specialised cell-to-cell communication techniques, the endocrine system maintains our physiological state whilst adjusting to a diversity of dynamic environments. This, like so many biological systems, exhibit amazing robustness to internal failures due to the highly redundant nature of the underlying cellular structure.

This thesis introduces a novel multicellular electronic architecture that draws inspiration from the endocrine system. The architecture's cellular structure models the cell based construction of biological organisms. The fault tolerant properties this provides is fully taken advantage of by an overlaying artificial endocrine system that coordinates the behaviour of cell groups, creating a system that performs arbitrary computation on a data stream.

The design and application of software simulations were used to acquire an initial level of confidence in the correct and fault tolerant operation of the developed system. As a result, a custom built hardware platform, called the BioNode System, was created, building on the initial success and allowing the proposed system to be more rigorously tested.

Initial results have provided positive feedback to the systems operation, demonstrating the system's capability of correct computational operation and resilience to cell failure. It is concluded that the endocrine system and the cellular architecture of biological organisms provide a suitable source of inspiration for a reliability engineered electronic system.

Contents

List of Figures

List of Tables

List of Circuits

Acknowledgements

Four years of research leaves you with many kind people to thank. I hope I have remembered everyone…

My thanks goes to the department's technicians, without whom a lot of hardware would never have been built. In particular, Pete Durkin deserves a special mention for his willingness to impart his boundless expertise in the area metalwork. A considerable amount of thanks also goes to Mark Hough, whose advice, continual encouragement to keep 'chipping away' and positive outlook made making well over 100 PCBs possible.

My lab partners, new and old, have been a pleasure to work, drink and argue with. My thanks goes to Carl, Crispin, Dave, Joel, Natasha, Rod and Will. I'd especially like to thank Paul for always stopping to chat in the corridors, Alex, for being a fellow hardware type, and providing answers to my endless hardware questions, and Mic, for providing a reminder of the lighter side of life, continued distractions and solid beer advice. You're all good friends.

Thank you also to Tim for always showing an interest in my work and continually filling me with enthusiasm. The last of my colleagues to thank is my boss, Andy, who has always been available for advice and guidance, both personal and work related.

Closer to home, my five dear friends Andy, Chris, Chris, Lee and Mark have always been willing to come to York to relieve the tension and remind me there is a life outside the university. Also to my Mum, Dad and sister Jennie who have supported and encouraged me all my life.

My final thanks is saved for my loving girlfriend Liz. Without you I would never have got this far. Ever since I met you, you've provided me with the love and support to overcome all of life's hurdles. Thank you, you mean everything to me.

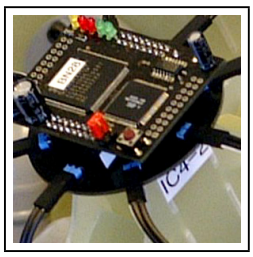
Declaration

The work presented within this thesis is entirely the author's own work, except where explicitly referenced. The following items represent work that has been previously published in the course of this research.

- Andrew GREENSTED and Andy TYRRELL. *Fault tolerance via endocrinologic based communication*. In *Proceedings of ICES 2003, 5th International Conference on Evolvable Hardware*, number 2606 in LNCS, pages 24–34. Springer Verlag, March 2003.

- Andrew GREENSTED and Andy TYRRELL. *An endocrinologic-inspired hardware implementation of a multicellular system*. In *Proceedings of EH 2004, 6th NASA/DoD Conference on Evolvable Hardware*, pages 245–252. IEEE Computer Society, June 2004.

Andrew John Greensted



Chapter 1

Introduction

The introduction of electronics as a mainstream engineering technique has revolutionised the way people live. Electronic systems have provided the tools enabling the rapid processing and interpretation of information, which in turn has made a dramatic impact on our ability to communicate, control physical systems and understand the world around us.

However, as the utilisation of electronic solutions becomes more widespread, increasing demands are being placed upon the core electronic components. Whether the pressure is for miniaturisation, greater speed, better efficiency or improved robustness, the response almost always results in an increase in device complexity, especially that of integrated circuitry. Unfortunately, greater complexity and increased levels of circuit integration normally come at a cost. As device transistor counts rise, and feature sizes get smaller, guaranteeing component reliability becomes an increasingly difficult problem.

As IC technology has developed, improvements in feature reliability have been required. This has enabled the production of devices with greater levels of feature integration to maintain realistic usability. The difficulty facing engineers now is the sheer magnitude of device integration. Even if the latest generation of IC design and fabrication techniques deliver highly reliable features, the huge quantities involved promote failure. Consequently, the utilisation of fault tolerant techniques will become a necessity. The difficulty that remains is finding fault tolerant architectures that are suitable for highly complex devices. The solution will require more diverse, and potentially less traditional approaches to be researched.

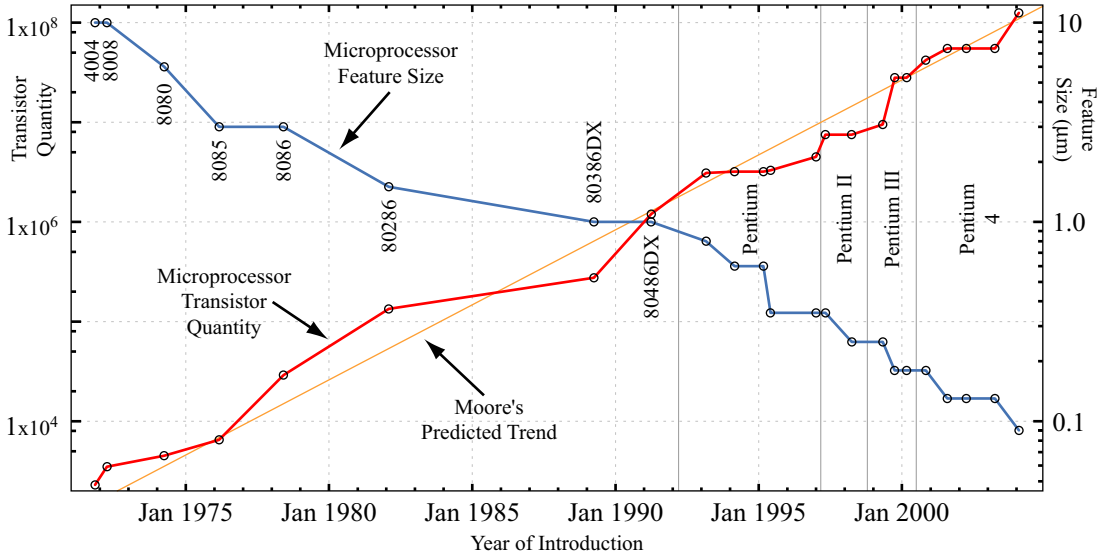


Figure 1.1: *The change in transistor quantity and feature size of Intel® microprocessors provides a good demonstration of Moore's Law.*

Biology and nature are no strangers to complexity. Even a moment's consideration of the many millions of cells that form the human body makes clear the ability of biology to construct complex systems. Furthermore, biological organisms are amazingly resilient to the effects of injury, demonstrating a high degree of fault tolerance. It is no wonder that engineers are asking themselves if there is something that might be learnt from biology which can be applied to the construction of complex, but reliable, electronic systems.

1.1 The Growth of Integrated Circuit Complexity

In 1965 Gordon Moore predicted that the number of transistors that could be integrated into a single device would double every two years [Moo65; Inta]. Surprisingly, to this day his prediction, famously known as *Moore's Law*, has held true.

An interesting trend that supports Moore's Law is how the number of transistors used in successive generations of Intel's® 'state-of-the-art' microprocessors has increased with time. Figure 1.1 shows this growth superimposed upon a plot representing a true doubling of transistors every two years. The real data can be seen to follow Moore's prediction remarkably closely.

Such a growth in integration generally requires a corresponding miniaturisation of transistors. Also included in Figure 1.1 is a plot of the decreasing microprocessor feature size, which also shows a exponential trend. Feature size reduction incurs a number of

reliability problems, an issue introduced later in Chapter 2. The source data for these plots, acquired from [Intb], is shown in tabular form in Appendix A.5.

The development of these microprocessors supports the fact that ICs are becoming increasingly complex. If that trend continues, the failure of an internal component during device operation will become a practical certainty rather than just a highly probable event.

1.2 Biologically Inspired Electronic Systems

Biology has been a source of inspiration to engineers for centuries, the classic early example being the bird-like ornithopters designed by Leonardo Da Vinci [Cut56; Mon87]. However, Da Vinci's visual and anatomical observations represent only a small portion of the wealth of biological inspiration on offer.

Electronic engineers and computer scientists have made use of a broad range of ideas from biology in the creation of so called *bio-inspired* systems and techniques. The following sections provides an introduction to a number of these. The selection is made to highlight those techniques that have provided novel solutions to the problem of handling electronics faults.

1.2.1 Evolutionary Algorithms

The *evolutionary algorithm* (EA) is a problem solving tool inspired by Darwin's theory of natural selection [Dar59]. There exist a number of different forms of EA, however the operation of each is based upon the same biological principle: 'survival of the fittest' in the context of evolution [Rid04].

The process operates upon an artificial population of individuals, each representing a potential solution to the problem being solved. The initial population is formed from randomised individuals. Each is then evaluated by the application of a *fitness function* that determines how 'fit', or how close to a satisfactory solution an individual is. A selection and variation process that is biased towards the fitter individuals is then used to create a second generation of potentially fitter individuals. Subsequent stages of evaluation and selection are undertaken until a suitably fit individual is found [Fog94]. A diagrammatic explanation of this process is shown in Figure 1.2.

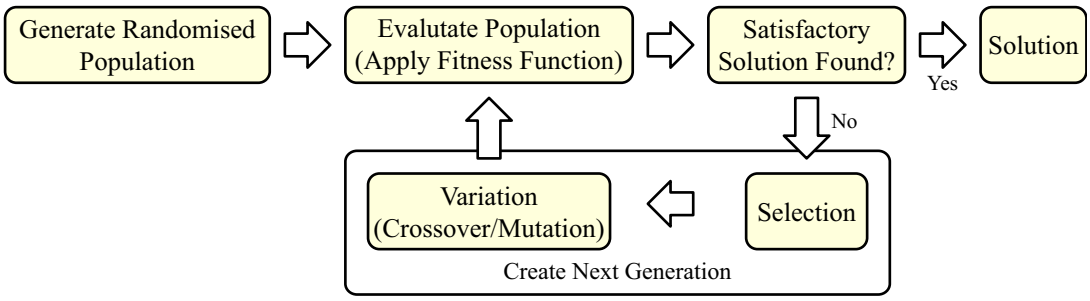


Figure 1.2: *The Evolutionary Algorithm is a reiterative process that mimics the evolution of a population of problem solutions in successive generations. The goal is to generate increasingly fitter individuals until a sufficiently good solution is found.*

The various forms of EA are distinguished by the representation used for individual solutions, the method used to select and generate subsequent populations and the size of the population involved. A full categorisation of approaches is unnecessary for this discussion, however, the *genetic algorithm* (GA) deserves a specific mention due its popular use in electronic hardware design, an area of EA application commonly referred to as *evolvable hardware* [Zeb02].

The GA representation is based upon the genetic code of biological organisms [Mit96]. In circuit design this takes the form of a list of component parameters, such as value, position and connection state. The variation process used to create successive generations is based upon the crossover and mutation of genetic information that occurs in natural reproduction. The effect is that successive generations tend to favour the more successful application of components.

One of the attractive features of GAs, and EAs in general, is their property of exploring solutions that traditional human-driven design techniques would not have considered. This has led to the creation of a number of circuits with comparable performance to human designed alternatives [Koz04; Kea03]. Furthermore, the potential of using GAs for finding novel circuits has been greatly extended by allowing the search to exploit the underlying physical properties of the hardware being evolved [Tho97; Har04].

GAs have also been applied to the creation of fault tolerant circuits. By including a measure of an individual's ability to tolerate faults into the algorithm's fitness function, circuits that exhibit a increased overall robustness to faults may be evolved [Can02]. An alternate approach involves the reapplication of evolution once a fault has been detected. In this case, evolution is used to find a circuit that ignores the fault, or even utilising the defective components as if they were functioning parts of the circuit [Key00].

1.2.2 Artificial Immune Systems

The immune system of vertebrates is a highly complex defence mechanism that provides biological organisms with protection against the dangers of pathogenic material. A fundamental principle of the immune system is that of the distinction between self and non-self: matter that belongs in the body, and that which does not. Specialised cells are able to recognise the intrusion of materials foreign to the host, and subsequently invoke specific reactions to eliminate them [Jan99].

The main interest held by electronic engineers in the immune systems is in producing artificial versions of the detection and recovery mechanisms. However, the complexities associated with implementing artificial immune systems (AIS) has biased their development towards the software domain rather than hardware. An early AIS, created by Forrest, was used to provide detection of abnormal behaviour within the UNIX operating system [For96]. Patterns of system calls were matched against a database of confirmed normal call patterns. Sequences that did not match were flagged as abnormal, and also as a potential indication of computer attack.

Of particular interest to engineers is mimicking the immune system's ability to learn to detect new pathogens. This trait is a feature of the highest tier of immunity, the adaptive immune system, which allows an organism to 'remember' past pathogenic attacks. Such memory means subsequent infections may be dealt with more swiftly, this being the principle behind vaccine inoculation. An adaptive AIS by Hofmeyr exhibiting such a characteristic was used for intrusion detection on a computer network [Hof99]. The system remembered the signature of past attacks so that they would be more readily recognised in the future.

Bradley introduced the notion of *immunotronics* to describe the fusion of immunology and electronics hardware [Bra01]. The implementation of an AIS utilised the self/non-self principle to provide runtime detection of erroneous states and transitions within a finite state machine [Bra02]. The progress of the state machine was monitored via the use of content addressable memory that held detectors of erroneous behaviour.

Although AISs is an area of research that is just emerging from infancy, engineers and computer scientists alike are now starting to really unfold the knowledge gleaned by immunologists into new bio-inspired techniques.

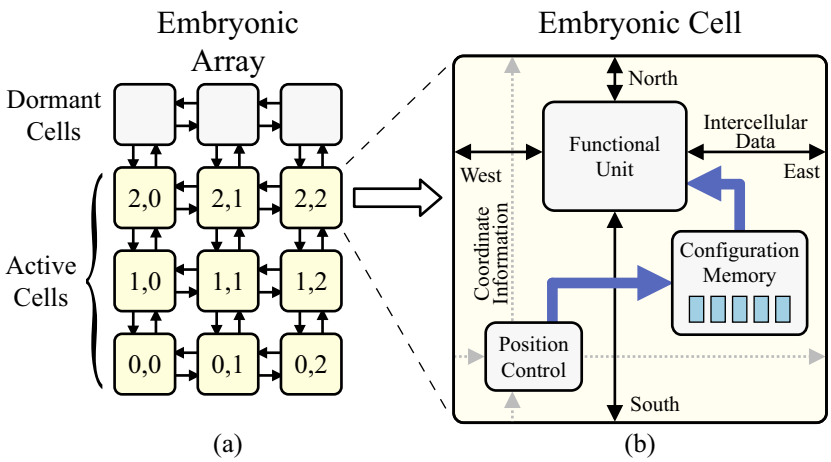


Figure 1.3: *The embryonics architecture is constructed from a structurally homogeneous array of cells whose function is determined by their position within the array.*

1.2.3 Embryonics

The *embryonics* architecture is a generic computing array that exhibits fault tolerant properties [Man00; OS00]. The architecture involves a structurally homogeneous array of cells that is representative of a biological tissue. As is the case in nature, each individual cell holds all the necessary information required to specialise as any cell type found within that organism or, in the embryonics case, system. The name embryonics is derived from the similarities the architecture shares with developing embryos [OS00].

Figure 1.3a depicts an embryonic array and Figure 1.3b, the basic construction of an individual cell. The functionality of the implemented system is distributed across the embryonic array, with separate system sections being implemented in individual cells. The particular function of a cell is determined by its position within the array. Cartesian coordinate information propagates from a single origin cell outwards across the array via dedicated communication links. The given coordinate is then used to calculate which part of the complete set of cell instructions should be used to determine the cell's function.

Fault tolerance in the embryonic array is achieved by the reconfiguration of cells to make use of spare rows and/or columns. To describe the reconfiguration process only spare rows shall be considered. Starting with a fault free array, Figure 1.4a, when a cell is detected as failed, Figure 1.4b, the complete row of which the cell is a member is made transparent to the array's positioning system, Figure 1.4c. The row directly above the failed cell now effectively receives its y coordinate information from the row below the

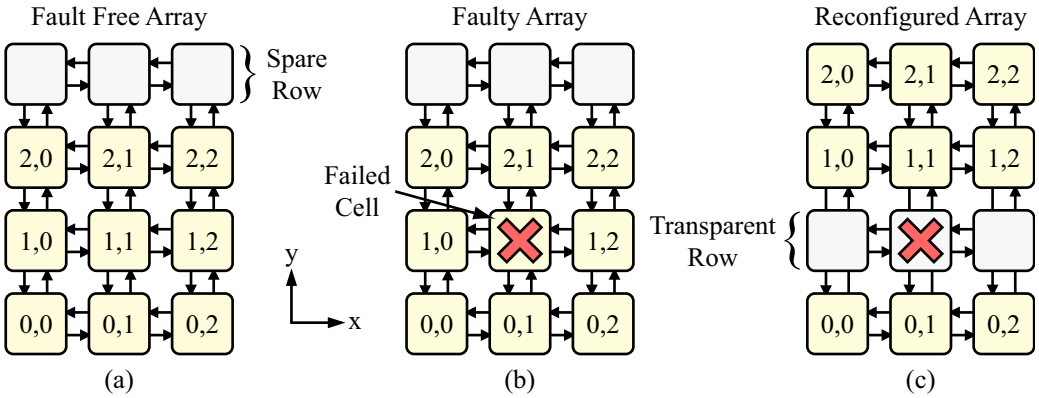


Figure 1.4: *Cell failure cause cells to receive new coordinate information which triggers a functional reconfiguration of the array.*

failed cell, resulting in a change in each cells y coordinate. A similar y coordinate change ripples through the remaining rows effectively causing a shift in circuitry as each cell reconfigures to its new function. The net result is that the circuitry from the transparent row onwards will have shifted upwards by one place, restoring the complete system and excluding the failed cell. The utilisation of spare columns occurs in a similar manner apart from the direction of reconfiguration follows the other axis. Using both spare rows and columns provides further flexibility in the choice of reconfiguration direction.

The embryonic architecture can be applied to both coarse and fine functional granularities. The finer grain approach utilises cells of limited functionally, providing just a few logic gates and sequential elements [OS98]. The configuration information held in the cell determines how the circuitry is interconnected and presented to neighbouring cells. The courser grained array uses individual microprocessing elements as cells [Man96]. In this case the cell location determines which sections of software are used to define the cells function.

1.2.4 Bio-Inspired System Summary

The preceding section introduced a number of different bio-inspired systems, summarised in Table 1.1, each of which is applicable as a technique in reliability engineered electronic circuit design. However, the huge amount of inspiration on offer from biology demands that further research is undertaken to ascertain what else may be learnt and applied to electronic design. This research represents a further step in that direction, to find further useful novelty from biology.

Bio-Inspired System	Biological Sources of Inspiration
Evolutionary Algorithms	Darwins's theory of natural selection, the concept that survival of the fittest leads to the improvement of individual characteristics with successive generations.
Artificial Immune Systems	The pathogen detection and recovery systems of biological immune systems, mainly those of vertebrates.
Embryonics	All cells contain the same genetic information that underpins their initial ability to specialise as any cell type within that organism.

Table 1.1: *A summary of bio-inspired approaches to reliability engineering techniques.*

1.3 Research Motivation and Hypothesis

The motivation for undertaking the research presented in this thesis was the search for a new source of biological inspiration that may be applied to the creation of a computationally capable, reliability engineered electronic architecture. In order to guide the direction of this search and find a research hypothesis, a number of biological observations were made through background and preparatory research, the details of which are presented in subsequent chapters. The observations are presented now, followed by the resulting hypothesis.

The majority of bio-inspired architectures that aim to exhibit reliability characteristics are motivated by a desire to reproduce biological fault handling mechanisms, these being fault tolerance, fault detection and fault repair. When looking for reliability related inspiration it is sensible to consider at what structural level these biological processes manifest themselves. Such a question leads to the cellular level where all three fault handling mechanisms truly operate.

Observation 1: The biological cellular structure, that contains fault handling mechanisms, is inspiration for a fault tolerant architecture.

Biology's cellular architecture provides a robust platform that underlies many of the biological systems that keep organisms alive. These systems gain robustness themselves by operation upon the cellular structure. This provides a source of biological systems that could inspire the creation of an artificial system that exhibits fault handling characteris-

tics. In selecting a system that could more directly inspire a computational architecture, the endocrine system presents itself as a suitable candidate.

Observation 2: The endocrine system is inspiration for a computational system that achieves fault tolerance by operating upon the robust cellular structure.

These two observations provide a direction for research and support the following hypothesis, the aim of this research being to prove.

Hypothesis: The operation and underlying architecture of the endocrine system can provide inspiration for the structure and operation of a reliability engineered electronic system.

1.4 Contributions

The undertaking of this research has led to the following main contributions to the field of electronic engineering:

- The design of a novel fault tolerant architecture believed to be the first electronic system inspired directly by endocrinology [Gre03].
- The creation of a multicellular hardware platform designed for the implementation and testing of cell based bio-inspired techniques [Gre04].

The following secondary contributions have also been made:

- A multicellular architecture that suitably maintains the functional capabilities of biological cells in the conversion to electronic nodes.
- The design and realisation of an expandable and networkable node, with both hardware and software reconfigurability, appropriate for implementing biological cell models.

1.5 Thesis Structure

The organisation of this thesis is as follows; the first of three parts, Chapters 2 and 3, collectively provide background information for the presented work.

- **Chapter 2**: *Reliability Engineering* introduces the concept of IC dependability. Included are explanations of a variety of the actual mechanism that cause IC failure, and an introduction to classical fault handling techniques.

- **Chapter 3**: *Biological Inspiration* serves as an introduction to the biological processes that inspire the research presented in this thesis, mainly cell communication, endocrinology and DNA.

In the second part, Chapters 4 to 6, collectively present the stages of development for the proposed endocrinology inspired fault tolerance system from analysis, through modelling and realisation to testing.

- **Chapter 4**: *Analysis and Development* feeds from the biological knowledge presented in Chapter 3 to explain the mapping from biology to electronic architecture.

- **Chapter 5**: *The BioNode System* contains details of the custom built hardware platform that was created to implement the proposed fault tolerant architecture.

- **Chapter 6**: *Experiments and Results* includes details of preliminary experiments to test the architecture's fault tolerance properties.

The final part, Chapter 7, provides a summary of and concluding remarks about the research undertaken.

- **Chapter 7**: *Conclusions* contains a summary of the presented work and conclusions drawn from it. There are also suggestions for directions of further work and a discussion of the potential future of bio-inspired electronics.

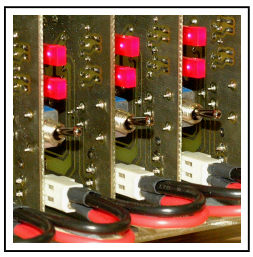
Supporting information is provided in the following appendices:

- **Appendix A**: *Derivations and Data* includes full derivations of quoted formulae and data used for graphs.

- **Appendix C**: *BioNode System Datasheets* provides detailed information for users of the BioNode system.

- **Appendix B**: *Circuit Diagrams* includes PCB layouts and circuit diagrams of the BioNode system.

- **Appendix D**: *Photographs* contains a variety of pictures of the BioNode System.



Chapter 2

Reliability Engineering

Electronics plays an ever increasing role in our everyday lives. Within the developed world it is unusual for a day to pass without having been affected by electronics. Whether this contact be direct; using a personal computer, or indirect; payments via a credit card, electronic systems have had a part to play.

However, many of these electronic systems are hidden from sight and consequently, it is understandable that we only become aware of their presence when they fail or underperform. Generally, such failures only lead to inconvenience; losing a days work on a document, or missing a favourite television programme. But delving deeper into the electronic infrastructures that support our lives reveals a far greater dependence on these systems, the failure of which may have far worse consequences than just inconvenience.

Many jobs, originally only trusted to human operators, have now been given to electronic systems. The benefits are clear; tireless operation, repetitive precision and in some cases, cheaper running costs. But these characteristics all depend on the correct operation of the systems involved. Medical equipment, industrial plant controllers and transport systems all benefit from electronics but still have the potential to put peoples lives at risk.

It is this demand for electronics to provide continued, correct and uninterrupted operation that drives the field of *Reliability Engineering*. Without the techniques generated by this field, many of the services we take for granted would be either impossible or too dangerous to operate.

This chapter introduces the subject of reliability engineering, starting with a discussion in Section 2.1 on how dependable electronics can be defined, what can be done to achieve

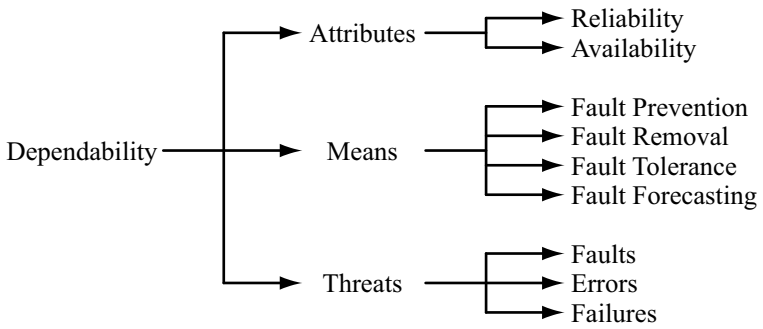


Figure 2.1: *The Dependability Tree. This graphical representation shows the key elements that hinder, help achieve and quantify dependability.*

it and how it may be measured. Section 2.2 provides a detailed look at the real drive for reliability engineering: faults and the mechanisms that create them. Finally, Section 2.3 introduces the actions and techniques that can be employed to avoid electronic system failures and the consequences they may have.

2.1 Dependability

An engineer designing a circuit, or a end user selecting a new system will want some form of assurance that their chosen components or products will provide sustained and correct operation. In short, they need to know the level of *dependability*. A system that cannot be depended upon to perform its required task in the required time is of limited use.

Dependability can be more formally defined as a system property that allows a justifiable reliance to be placed on the services that system offers [Lee90]. Reliability engineering is about providing the *means* to achieve dependability, by considering the *threats* that prevent it, and potentially of most importance to the user, *measures* of the *attributes* that quantify that dependability. These relationships are graphically expressed in the *dependability tree* [Avi00], as shown in Figure 2.1.

2.1.1 Threats

The reason that necessitates reliability engineering is the prevention of failure. A system is said to have *failed* if its operation deviates from the stated system specification [Lal85]. Systems with complex specifications are open to failure in a number of ways, each potentially requiring a different form of response. It is therefore usual to categorise modes of failure into severity ranges, so that the correct action is taken.

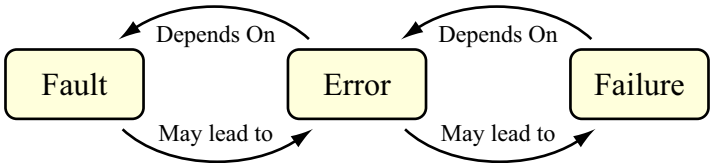


Figure 2.2: *Failure is caused by errors, which are in turn are caused by hardware faults. However, the relationships are not bidirectional, faults can exist without error, and errors without failure.*

The cause of failures are *errors*. An error is defined as a deviation from a correct system state, which may be an externally visible state, or an internal subsystem or component state [Lee90]. It is important to note that an error does not directly infer a failure; a system can harbour errors without ever deviating from the specified operation. For example, a microprocessor can perform many operations, placing the results into memory. If a stored value is erroneous, whether by incorrect calculation, or corruption once placed in memory, and the value is never required again, it will not contribute to a failure.

The underlying reason for errors are *faults*. Hardware faults are the physical defects and abnormalities in the system circuitry. There are many sources and types of faults, some due to human mistake, others by natural events [Lap89]. Section 2.2 introduces some of these. Faults do not always produce an error. In a similar manner to errors not always leading to failure, the effect of a fault depends on the operation of the affected system or component. For example, an incorrectly designed register that is never used to store a value, will never produce erroneous data. The inter-dependency of failure, errors and faults is depicted in Figure 2.2.

2.1.2 Attributes and Measures

The dependability of a system is determined by the system property, or properties, upon which reliance is placed. Consequently, there exists a variety of different system attributes that may be used to define dependability. The two attributes that most commonly gauge dependability are *reliability* and *availability*. Reliability is the attribute of continued and correct service, availability is the readiness to correctly perform a service [Ben99].

However, to be of real use, these attributes require quantification. A user of a system where availability defines dependability will want to know exactly what percentage of

time they can expect the system to start, or perhaps not to start. For dependability to have real meaning it requires a form of measure.

A common measure of reliability is *mean time between failure* (MTBF), a probabilistic guide to a system's expected failure-free operation time. However, 'guide' is an important caveat of this measure. A system with a 1000 hour MTBF may fail a long time before this operational limit has been reached, likewise, operation may well continue for a long time after. Appendices A.1 and A.2 contain formal definitions of reliability and MTBF respectively.

It is generally the case that dependability attributes only have probabilistic measures due to the random nature of faults, and therefore failure. Reliability engineering is very much a subject of improving dependability such that the chance of failure is as small as possible, or sufficiently small as to be acceptable.

2.1.3 Means

The final branch of the dependability tree is means. This constitutes the practical steps that may be taken to increase the dependability of a system. The techniques involved are *fault prevention*, *fault tolerance* and *fault forecasting*.

Fault prevention is the combination of two techniques, *fault avoidance* and *fault removal*, that are conducted before a system is released into normal service. Fault avoidance is the application of quality control to minimise the initial introduction of faults. This is carried out in both design and manufacture stages and can take the form of correct component selection, utilisation of thorough design rule checks and maintenance of proper manufacturing environments. The fault removal stage is required to identify and correct faults that have occurred despite the avoidance techniques. Removal takes the form of testing followed by redesign of the system or the manufacturing process.

It is possible that fault removal may spread into the 'in-service' stage of a system's lifetime. The ability to reconfigure particular types of hardware allows designers to correct design faults found after system release. In fact the manufacturers of reconfigurable hardware have marketed this approach as a method of reducing the time to market for new systems [Bra99].

Good use of fault prevention will help to reduce the quantity of faults that may exist up to system release. However, what of those that remain, and those that develop whilst the

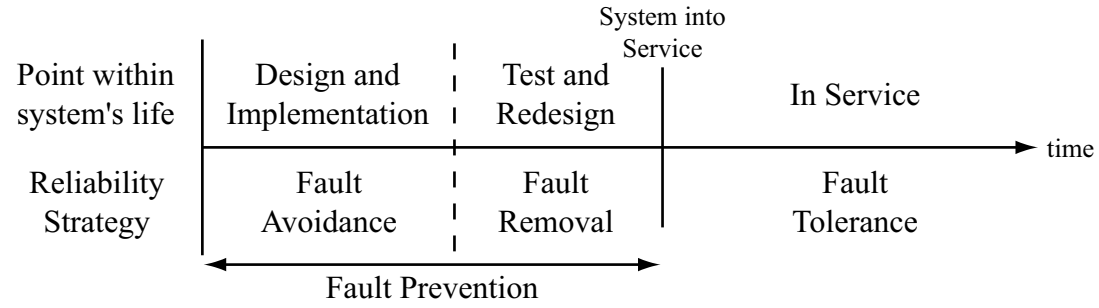


Figure 2.3: *Dependability improving techniques are employed at different stages of a systems lifetime [Lee90].*

system is in service? Fault tolerance techniques are employed so a system may maintain continued service despite the presence of faults. Further explanation of these techniques can be found in Section 2.3. The distribution of fault prevention and tolerance techniques over a system's lifetime is shown in Figure 2.3.

Finally, fault forecasting is the procedure of estimating the number of faults at a particular time, and the likely consequences of those faults. Such forecasting techniques can be used to generate measures of dependability.

2.2 Faults and Failure Mechanisms

Faults can be classified in a number of ways. Depending on the approach taken, classification can be useful in determining which reliability engineering techniques will be most suited to preventing or effecting tolerance of a particular type of fault. For example, intrinsic failure is caused by faults that are remnants of the design and manufacturing process. Extrinsic failure is caused by all other faults, such as those occurring due to device misuse, or appearing since the system's release into service [Jen95]. The faults that cause intrinsic failure may be most effectively dealt with by fault prevention techniques, whereas those that lead to extrinsic failure can only be handled by fault tolerance techniques.

An alternative classification approach is via the physical location of a fault. This highlights which area of component design may need further work to reduce the likelihood of those faults occurring. The following list identifies one particular grouping of locations [Sha97, Page 249] and examples of faults that may occur within them.

1. **Bulk** - Faults that develop in the actual semiconductor area of the component. Examples include, soft errors and CMOS latch-up

2. **Interface** - Faults that occur where semiconductor meets dielectric. Examples include hot carriers.

3. **Dielectric** - Faults occurring in the insulating layers of device active circuitry. Examples include electrical overstress such as electrostatic discharge.

4. **Metallisation** - Faults occurring in the metal interconnects between the active circuitry components. Examples include electromigration and corrosion.

5. **Packaging** - Faults related to device casings. Examples include bonding wire failure and sealing failure.

A process that generates a fault which then goes on to cause system failure is called a *failure mechanism*. The fault, or perhaps faults, being the physical manifestations of that mechanism. The cause of failure mechanisms can also be used as a criteria for classification. For example, Laprie classified faults into categories depending on the failure mechanism being induced by human mistake [Lap89].

Sections 2.2.1 to 2.2.4 provide an insight into the wide variation in integrated circuit faults by describing some of the failure mechanisms that lead to them.

2.2.1 Radiation Effects

Nuclear radiation can produce dramatic effects on the internal state of semiconductors. High energy particles hitting the right, or depending on your viewpoint, wrong area of circuitry can cause electrical charges to change, leading to flips of logic levels. Particles can also introduce undesirable electron flow that can trigger the build up of large currents capable of destroying circuit elements. This section describes some of the sources of nuclear radiation and the problematic effects it can have.

Two common sources of the radiation have been identified. The first is the decaying radioactive atoms that are naturally found in many materials. The α particles emitted by these atoms have a very limited range; only a few centimetres in open air. When the particles radiate through a solid material their range is far less, and therefore for an IC to be affected the source atoms are generally found in the device's packaging.

The second source of radiation is space. Cosmic rays continually stream towards Earth, enter our atmosphere and shower the ground with high energy particles. On entry into the atmosphere, a so-called *primary cosmic ray* will collide with air atoms such as oxygen and nitrogen, smashing them into different particles. Momentum is passed onto these new particles which continue towards the earth and, in turn, hit further atoms. The resulting chain reaction of smashed atoms produces a cascade of particles, known as an *air shower*.

Many of the air shower particles will not have enough energy to reach ground level, being absorbed into the atmosphere. The shower density reaches a maximum when the particles no longer have enough energy to smash further atoms. From this point on, as altitude decreases the density of particles decreases as more are absorbed. However, a single cosmic ray can still produce an air shower that covers an area of many square miles. Particle flux at ground level is in the order of one particle per square centimetre per second, at 15km above sea level, this increases to 100 particles per centimetre per second.

2.2.1.1 Total Ionizing Dose Effect

The accumulation of ionizing radiation will slowly degrade the performance of a device over its lifetime until it fails completely. This is known as the *total ionizing dose effect*. High energy particles entering a device transistor form electron hole pairs. The mobile electrons are easily swept away leaving the holes, or more accurately, a positive charge. A build up of holes affects the parameters of the struck transistor, such as the switching threshold [Sha97, Page 325].

2.2.1.2 Single Event Effects

As the name implies, *single event effects* (SEEs) result from penetration by a single energetic particle. As such a particle moves through the device it leaves an ionizing trail behind it, and changes the net charge of that area. Two common forms of single event effects are *Single Event Latch-Up* (SEL) and *Single Event Upset* (SEU). SEL is a special case of CMOS latch-up that is caused by the current flow induced by the particle ionizing trail. This phenomenon is described more fully in Section 2.2.3.2 when latch-up is properly introduced.

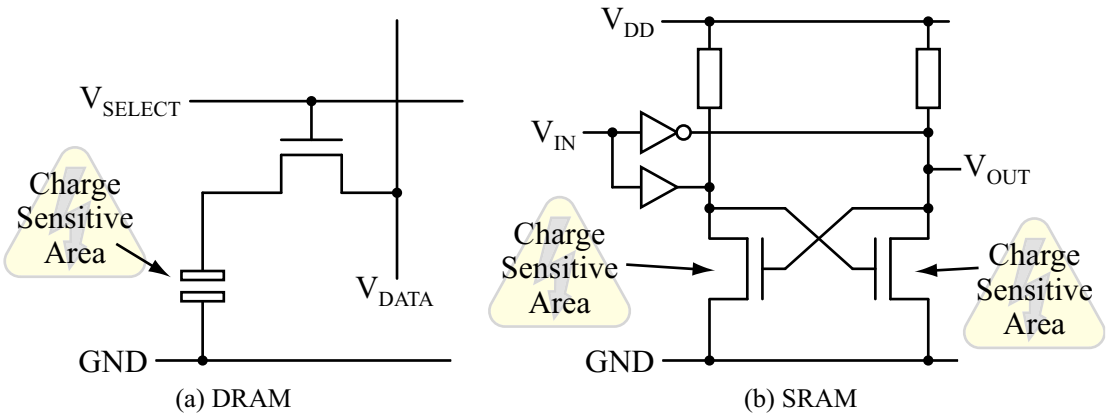


Figure 2.4: *Different memory types are sensitive to different levels of charge change. DRAM (a) uses a relatively large charge store and is less susceptible to radiation than SRAM (b) which relies on gates to hold logic levels.*

Single event upsets are caused by a change in electrical charge. Components that rely on charge to hold a signal level are susceptible to corruption if that charge is altered unintentionally. Memory cells are an example of such a component as they use electrical charge to store digital bit values, and are consequently particularly prone to radiation [Zie79]. A SEU does not cause permanent physical damage to the struck device, it only alters logic values. For these reasons this type of error is also called a *soft error* or *soft fail*.

The discovery of soft errors caused by α particles occurred in 1978 due to an investigation into operational errors of Intel®'s 2107 memory devices. An increase in demand for the memory chips meant a new production facility was built on the Green River, Colorado, to produce the ceramic device packages. Water from the river was used to supply the factory, but a uranium mine upstream meant the water was radioactively contaminated. The radiation levels turned out to be sufficiently high to contaminate the ceramics produced at the factory, and consequently be a source of alpha particles in the memory chips [Zie96].

The sensitivity of a device to single event upset depends on how it utilises charge. Figure 2.4 shows the basic construction of dynamic RAM (DRAM) and synchronous RAM (SRAM) cells. The DRAM cell uses a capacitor as a relatively large charge store. A nuclear particle would need to produce or neutralise a significant amount of charge to affect the logical value held by the cell. SRAM cells are, however, far more susceptible. The bit value is stored by the state of the two central MOS transistors, Figure 2.4b. A small change in charge within one of these transistors can alter its biasing, resulting in a flip of both transistor states, and of the overall stored value.

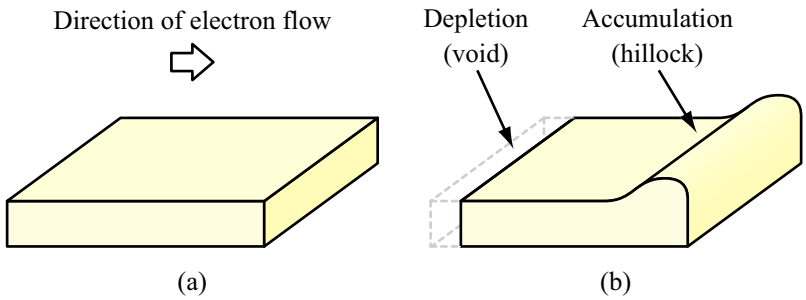


Figure 2.5: *At very high current densities, excessive electron-atom collisions cause interconnect metal to shift in a process called electromigration. A flat interconnect (a) can become deformed (b).*

The obvious solution for reducing the risk of soft errors in memory devices is increasing the level of charge used to store data. However, the decrease in IC feature size and increase in transistor density has meant that smaller charge levels are being used, making soft errors a worsening problem.

Memory cells are not limited to holding system data. Many reconfigurable devices utilise SRAM to store their configuration. In these cases the actual circuitry implemented on the device can be affected by radiation. The term *firm error* is sometimes used to describe such a failure by a single event [Act02].

2.2.2 Electromigration

High current density in a metal conductor can cause a shift within its atomic structure. This process, called *electromigration*, is a major cause of failure in the metal interconnects of integrated circuits [Bla69]. An illustration of electromigration is shown in Figure 2.5.

2.2.2.1 Integrated Circuit Current Densities

A perfect metal conductor would have all stationary atoms, arranged in a uniform lattice. Electrons would be able to flow freely through the metal without being impeded. However, the normal structure of metal is not ideal. There are normally irregularities in the lattice, described later in Section 2.2.2.3, and heat causes the metal atoms to vibrate, increasing the chance of electron collision.

When an electric potential is applied to a metal, the resulting electron flow is resisted due to collisions with the metal atoms. Each collision generates heat, a process called *Joule heating* named after the English physicist James Prescott Joule. For a fixed metal cross

section, an increase in current results in an increase in current density, this means more electron-atom collisions occur and more heat is generated.

The current density of standard wiring is limited to the region of 10^4 A cm^{-2}. Above this, the generated heat will cause the metal to melt. However, the metal interconnects used in integrated circuits are attached to a layer of silicon. This acts as a heat sink, efficient enough to allow current densities up to 10^{10} A cm^{-2} before melting occurs [Wan96]. The consequence is that the number of electron-atom collisions in metal interconnects can be considerably greater than in a normal air-exposed wire.

2.2.2.2 Interconnect Metal Migration

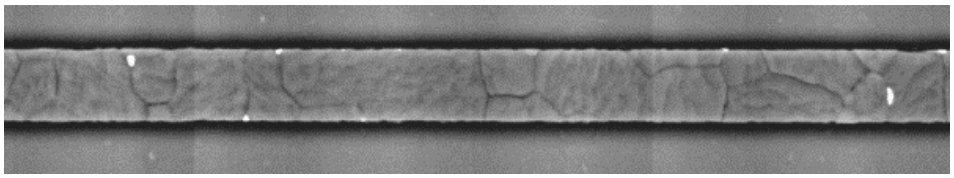
When an electron hits an atom, some of its momentum is passed on to that atom. The amount transfered can only be very small due to the huge mass difference between the colliding objects. The atom may vibrate further, but will otherwise be unaffected. However, if the current density is sufficiently high, enough momentum can be imparted such that the atom's position is shifted.

The migration of metal atoms along the direction of electron flow will eventually cause a depletion of atoms at the negative end, and an accumulation at the positive. It is possible that metal depletion can be so great that cracks occur, leaving the interconnect open circuit, see Figure 2.6a. The accumulation of atoms creates an area of stress that can cause splitting of IC layers, or if the space allows, deposits that spread outward into the device as shown in Figure 2.6b. If the spread of material occurs in the wrong direction, short circuits to other device features can occur, a result exacerbated by reducing IC feature size.

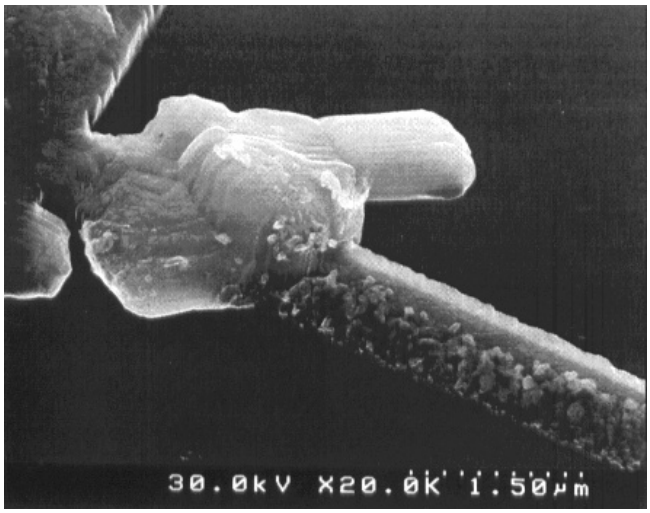
2.2.2.3 The Effect of IC Feature Size

The reduction of IC feature size with negligible change in current usage means interconnect current densities will increase. Unexpectedly this situation does not always lead to a worsening of electromigration. The reason is due to what lies in the path of the moving metal atoms.

When liquid metal cools, separate solid lattices called *grains* start to form [Sma99, Page 42]. As the metal cools further, the remaining free atoms attach themselves to the lattices until the point where all the growing grains have met. However, when this happens the



(a)



(b)

Figure 2.6: *Electromigration can result in the creation of voids (a) as atoms are removed or hillocks (b) as they are deposited. Figure a is reproduced with the permission of the Jan D'Haen, Institute for Material Research, Limburgs Universitair Centrum (http://www.imo.luc.ac.be). Figure b, with the permission of Gary H. Bernstein, Xun Pang, and Richard Frankovic, University of Notre Dame.*

grain lattices are not aligned, resulting in irregularities called *grain boundaries*. The same is true of the applied metal layers used on integrated circuits. After etching, interconnects may be left containing many grain boundaries. Figure 2.7a shows the disorientated nature of metal grains, and the boundaries that form between them.

As has been already established, these lattice irregularities are one of the reasons that electrons collide with atoms, but they also affect how the atoms may move. A network of connected grain boundaries provides a less restrictive path for migration to occur, as shown in Figure 2.7b, with the migration path shown in bold. However, if the interconnect width is comparable with the grain size, the boundaries end up being perpendicular to the direction of migration, as shown in 2.7c. When this happens, the boundary network is lost, and atoms have a far more restrictive grain lattice in their path.

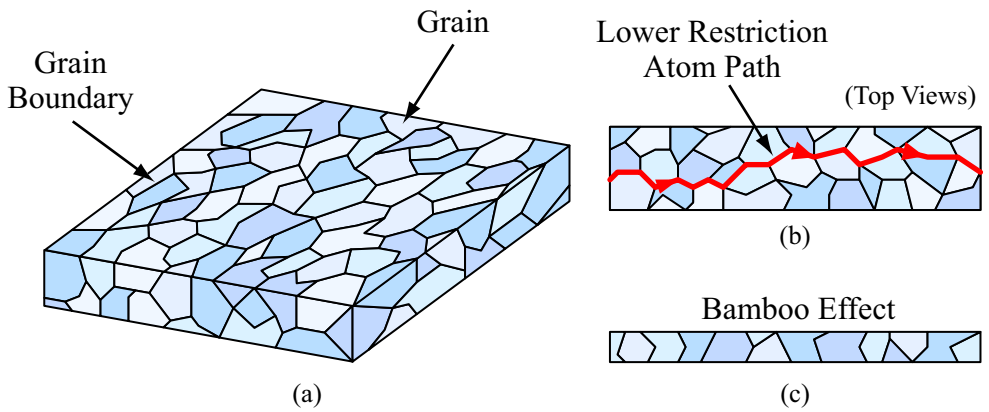


Figure 2.7: *Cooled metal forms separate disorientated grains of uniform lattices (a), high interconnect width to grain size ratios provide low restriction grain boundary paths for migrating atoms (b), lower ratios produces the bamboo effect (c), limiting atom movement.*

This particular ratio of grain size to interconnect width produces a formation called the *bamboo effect* [Llo97]. It creates a barrier to diffusion and reduces electromigration.

2.2.3 Electrical Overstress

All electronic components are limited to the level of electrical load they may endure before becoming damaged. *Electrical overstress* (EOS) refers to the condition when too great a voltage or current is applied to a device and damage occurs.

Electrical overstress generally occurs due to the improper handling or use of a device. For example, simple misalignment of a component can leave sensitive signal lines connected to power supply tracks, providing a potential source of large and damagingly currents. An example of the effect EOS can have is shown in Figure 2.8. In this case, too high a current has destroyed a pin to die bonding wire. More information on bonding wires can be found in Section 2.2.4.

2.2.3.1 Electrostatic Discharge

When a material has an imbalance of electrons and protons it will have a net electrical charge in the form of *static electricity*. The transfer of this charge to another material is called *electrostatic discharge* (ESD) and can be a source of electrical overstress. Humans are able to carry a considerable amount of charge and, on contact with a material held at a different electrical potential, discharge will occur, potentially with high current levels. If the contacted material is a sensitive electronic device, then the discharge may be enough to damage the component.

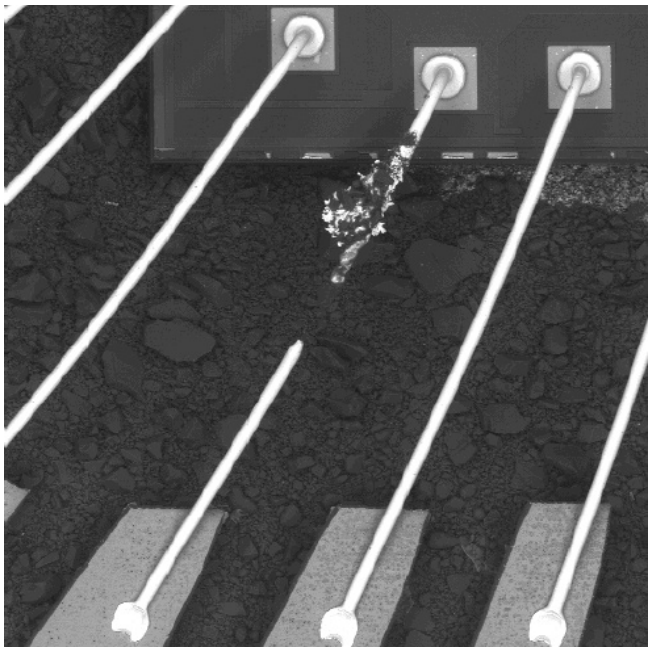


Figure 2.8: *A bonding wire within this device has been destroyed due to electrical overstress. This image is reproduced with the permission of Ed Hare, SEM Lab (http://www.semlab.com).*

Electrostatic charge is most commonly created during the contact and separation of materials. This process is called triboelectric charging, and is depicted in Figure 2.9. The atoms of materials without a net static charge have the same number of positively charged protons in their nuclei, as they do negatively charged electrons in orbit around them, Figure 2.9a. When the right combination of materials come into contact, electrons from the surface atoms of one material jump to the surface atoms of the other. The result is the two previously neutrally charged materials now individually have a net charge. The material losing electrons becomes positively charged and the other negatively charged, Figure 2.9b. The amount of charge transfer is dependant on factors which include the contact area of the opposing materials, the speed of material separation and humidity.

Discharge occurs as a single rapid event, in the order of nanoseconds. This normally requires physical contact but, if the involved static electric field is large enough, discharge may only require close proximity between materials. The effect discharge has on a ESD sensitive device depends on its ability to dissipate the sudden influx of energy. This ability, and thus how prone a device is to ESD is termed *ESD sensitivity* or *ESD susceptibility*.

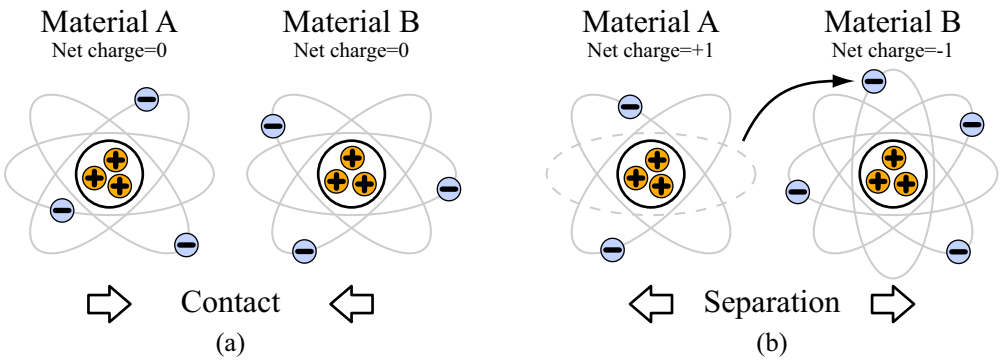


Figure 2.9: *Triboelectric charging is a common occurrence that can produce large charges that may lead to ESD. The bringing together and separation of materials allows electrons to transfer leaving an imbalance in each material.*

Devices can have different sensitivities to ESD depending on the way in which the charge is transferred. Therefore, to provide a guide to a component sensitivity, three different discharge models are used [ESD99], they are:

1. **Human Body Model** - The oldest and most commonly used model, it represents the transfer of charge from a standing human's fingertip to the device.

2. **Machine Model** - Similar to the human body model. However, in this case, the source of charge is a conductive object. This may be a hand tool such as a screwdriver, or automated assembly line equipment. The machine model, in fact, represents a worst-case human body model since contact resistances are much lower.

3. **Charged Device Model** - The final model represents the transfer of charge from the ESD sensitive device to another object. Such a situation may occur in automated PCB manufacture. As components are transferred towards a board, they may acquire charge that will be released on contact.

2.2.3.2 CMOS Latch-Up

Latch-up is a troublesome mode of failure that is particular to CMOS devices [Str95]. It can be considered a form of EOS, as a device suffering a latch-up event may experience currents far in excess of the specified maximum limit. Serious damage, or even destruction, of individual device transistors can occur.

Figure 2.10a shows the cross sectional structure of a basic CMOS inverter (Figure 2.10b shows the schematic). Close inspection of the structure reveals two parasitic bipolar

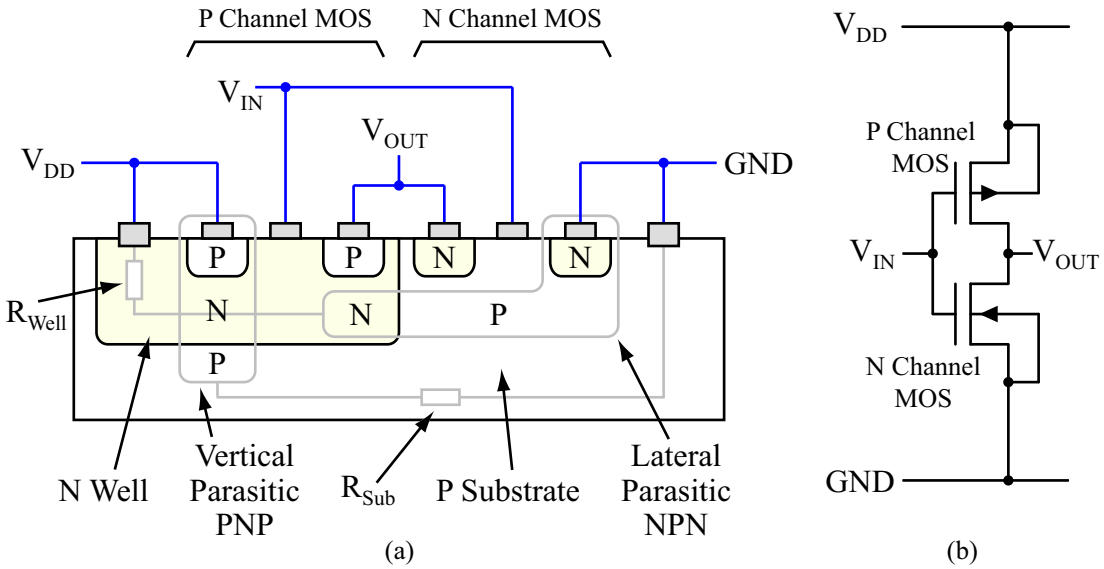


Figure 2.10: *The structure of a basic CMOS inverter contains two parasitic bipolar transistors connected in a highly unstable form. Under the correct conditions the bipolar transistors allow a destructive current to flow through the inverter.*

transistor forms, a vertical PNP, and a lateral NPN. The connectivity of these transistors can be redrawn as in Figure 2.11. The resistors are due to the larger N well and the P substrate. This alternative view of the inverter clearly shows where the problem lies. Unless both bipolar transistors are firmly 'off', a shorting current is able to flow from V_{DD} to GND.

Latch-up occurs as follows; if a current flows through R_{Well}, the base of the PNP transistor will be pulled down. This in turn causes an increased current flow in R_{Sub}, that

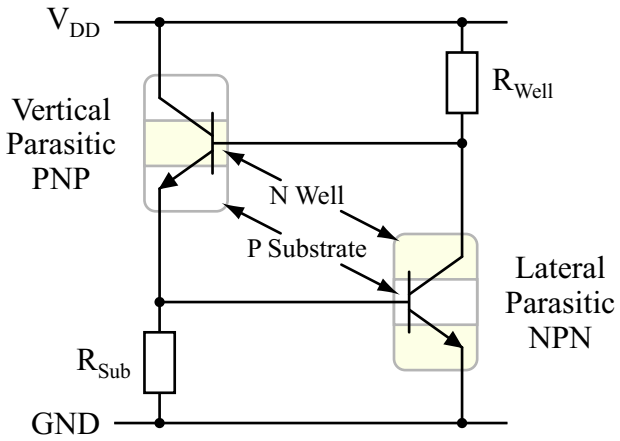


Figure 2.11: *The CMOS inverter parasitic transistors can be redrawn to show why they can be so destructive. A large shorting current can flow from V_{DD} to GND if both transistors are saturated.*

raises the NPN transistor base and increases the initial current through R_{Well}. Positive feedback means the transistors soon become saturated and the shorting current may freely flow. However, an initial current is required for this 'runaway' to occur. The most common cause is due to mishandling of the device. If the inverter output is pulled to a potential sufficiently lower than ground, a current will flow through the N Well to V_{DD}. This is the same as a current through R_{Well}, so the whole latch-up process starts. Latch-up can also be caused by single event radiation. High energy particles can ionize the N-Well atoms producing electrons that are pulled towards V_{DD}, this constitutes a current flow that can initiate latch-up.

The ensuing latch-up current flows directly through the heart of the CMOS inverter and, if allowed to rise, will destroy the transistors. The only way of stopping the process once underway is to remove power to the device. Designers use a number of techniques to help remove the potential for latch-up. These include:

1. Further separating the N channel MOS from the N well. This lengthens the NPN transistor and consequently reduces its current gain. The NPN transistor now requires more base current to drive R_{Well}. However, this technique is expensive in terms device area.

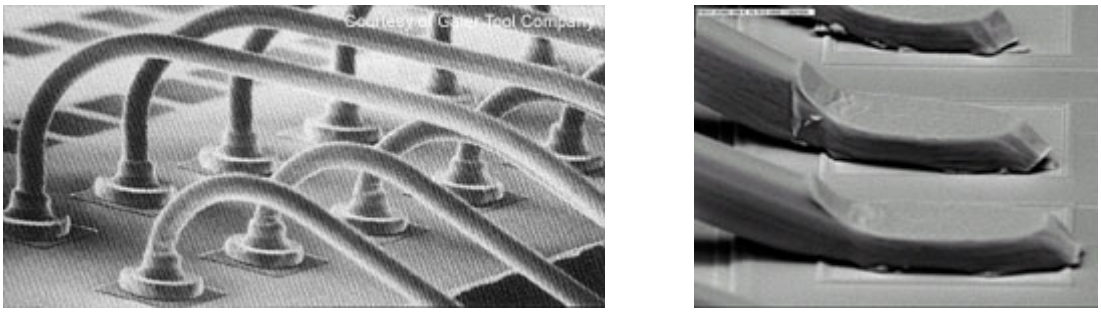
2. The addition of extra V_{DD} taps into the well and GND taps into the substrate reduce the effective resistances of R_{Well} and R_{Sub} respectively. This helps keep the parasitic transistors switched off as a greater current is required to create a biasing voltage.

3. Laying out the two MOS transistors in isolated trenches. This approach removes the parasitic transistor formations, but increases fabrication complexity.

2.2.4 Wire Bonding Failures

The manufacturing process for most IC components includes a connecting phase to provide electrical connections between the die and the package pins. Only where overall system size is critical are component dies directly connected to the circuit board. The two most widely used approaches to achieve pad to pin connectivity are wire bonding, or flip-chip bonding.

Flip-chip technology was first used by IBM in 1964 [Tsc01], and involves placing the die up side down in the package; hence the name. This realignment allows the die pads to make more direct contact with the package pins. Flip-chip packages are ideally suited



(a) (b)

Figure 2.12: *Bonding wire is used to connect die pads to package pins. Two methods are used to terminate the wire, a ball end (a) or a wedge (b). These images are reproduced with the permission of the Gaiser Tool Company (http://www.gaisertool.com).*

to components with very high pin counts as they may all be connected in a single step. However, the technology has taken a long time to be accepted in industry due to the extra steps required to ready the die for connection.

The alternative, and far more widely used approach, is *wire bonding* which uses individual wires, normally of gold, aluminium or copper, to connect die pads to package pins [Ser91]. Two wire termination forms may be used, either bonding balls or wedges. A complete bonding wire can be seen in Figure 2.8, bonding balls and wedges can be seen in Figures 2.12a and 2.12b respectively. The mechanical process of wire bonding has a number of failure mechanisms associated with it. A collection of these are introduced within this section.

2.2.4.1 Wedge and Ball Lifting

The detachment of a wire termination from its pad can leave a device pin unconnected, or with a high serial resistance. The two most common causes of lifting are a bad wire weld, or contamination at the joint [Ahm86]. A good weld relies on the proper set-up of the wire bonder machine and a firm placement of the device. Any misalignment and the ultrasonic welding process will fail.

The most likely sources of joint contamination are residues left from previous fabrication stages, such as etch resists, or sawdust from wafer cutting. Contamination can also promote the formation of inter-metallic compounds. When two dissimilar metals meet, diffusion across the join leads to the creation of a new material. In the case of a gold to aluminium join, five such compounds may form, one of which is purple in colour, and is often referred to as *purple plague* [Foo87; Sel69]. On their own, these materials are not

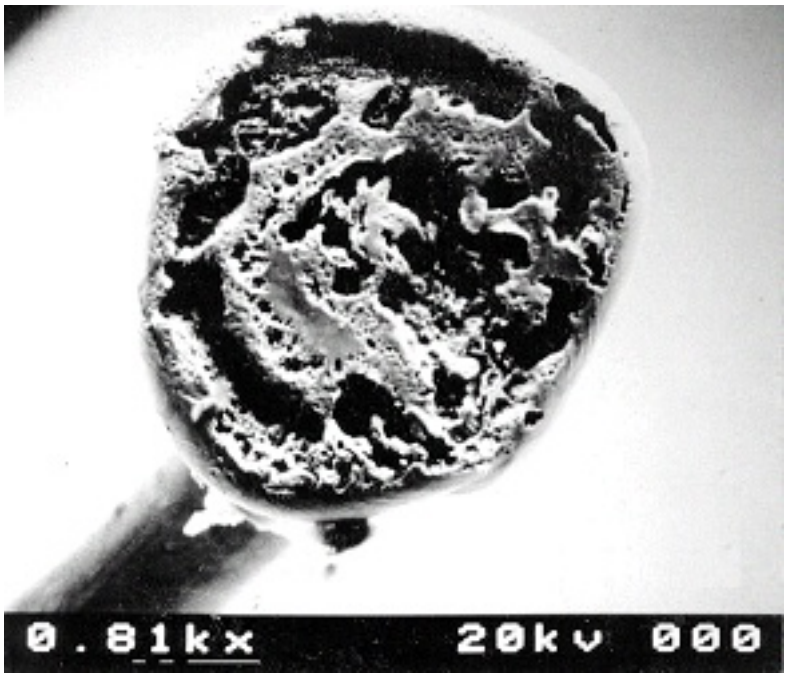


Figure 2.13: *A difference in inter-material atom diffusion rates can be a source of voids in bonding wire joints. This example shows how such an effect may weaken a ball bond weld.*

a problem. They are strong and have good electrical conducting properties. However, if the diffusion of material from one metal to the other is not equal and the temperature is sufficient, voids can form, as shown in Figure 2.13. These voids are called *Kirkendall voids* [Oku81], and, if enough form and then combine, they can seriously weaken the wire bond.

2.2.4.2 Neck and Heel Breaks

The bonding wire neck is the point at which the wire meets the ball bond. An incorrectly formed ball can leave a overly weak neck that is susceptible to breaking. Such a break can occur if the wire loop from die pad to package pin is formed with too much tension; the wire snaps at the weak point, leaving an open circuit pin.

The bonding wire heel is the point at which the wire tapers off to the pad or pin. In a similar manner to ball bonding, bad alignment or welding can leave the wire heel as a weak point. The fabrication stages that follow bonding can lead to the failure of both weak necks or heels. These stages include the application of the *die overcoat* that provides moisture and corrosion protection, and the moulding stage that forms the final device package. Both stages may produce movement in the bonding wires, and lead to breakages.

2.2.4.3 Shorting

There is a lot of scope for shorting in the wire bonding process. Incorrect placement of a wire ball or wedge may form a short circuit between the destination pad and a neighbouring die feature. Similarly, too great a welding pressure or undersized pads can again leave wire metal overlapping its allowed footprint.

If the wires themselves are badly formed they may short with other wires, or parts of the device. As can be seen in Figure 2.12a, wires may sometimes be stacked above each other. In this situation too much slack can cause wires to touch. The same wire movements that result in neck and heal breaks, can also cause such wire shorts. This is especially true if free wire ends have occurred due to breaks.

2.2.5 Failure Mechanism Summary

This section has introduced a number of the failure mechanisms that can lead to faults in integrated circuits, they are summarised in Table 2.1. Some, such as the bonding wire breaks and shorts, occur during device manufacture, and can consequently be dealt with by fault preventative methods: improved manufacturing. However, the others represent faults that can occur once the device is in service. Dealing with these faults require fault tolerance techniques, the subject of the next section.

2.3 Fault Tolerance Techniques

The use of proper design and manufacturing procedures can help reduce the number of faults present in a device when it enters service. However, after this point, even exhaustive fault prevention techniques cannot stop the appearance of new faults. Although certain design measures can be employed to limit the likelihood of their occurrence, guaranteeing continued freedom of faults for the lifetime of a device is near to impossible.

Dealing with this situation requires a set of design techniques that allow a system to resist failure even with the presence of faults. In effect, they enable the system to tolerate faults, and are therefore known as *fault tolerance techniques*. Early fault tolerance techniques are attributed to Von Neumann who studied the creation of reliable systems from the exceptionally unreliable components of the day [vN56]. However, the introduction of semiconductor devices saw a dramatic improvement in component reliability and, consequently, a downturn in the use of fault tolerance techniques. It was the start of the

Category	Location	Failure Mechanism	Comments
Radiation Effects	Device Silicon	Single Event Effects	The effects of single high energy particles such as Single Event Upsets, and Single Event Latch-Up. These occur due to radiation induced changes of electrical charge.
		Total Ionizing Dose	The build up of electron-hole pairs as high energy particles continue to radiate a chip.
Electro-migration	Metal Inter-connects		The movement of interconnect metal due to high current densities, can lead to the formation of short or open circuits.
Electrical Overstress	Whole Device	Electrostatic Discharge	An unintentional current caused by potential voltage differences between ICs and other objects. Can cause fusing.
		Latch-Up	Parasitic formations in CMOS transistors can be triggered to cause large and destructive currents within a device.
Wire Bond Failure	Die to pin wiring	Ball lifting	The lifting of wiring bonds from their pads due to bad welds or contaminants, can lead to open circuits
		Breaks	Where the die to pad wire breaks, normally due to wire movement during device packaging
		Shorting	Free ends of broken wires can short to other IC features. Inaccurate bonding can short pads to neighbouring features.

Table 2.1: *A summary of different failure mechanisms that can occur within Integrated Circuits.*

space program in the 1960's that revitalised their use when reliability became so closely connected to the support of human life.

The key to fault tolerance is redundancy: extra resources added into the system requiring protection. Furthermore, the organisation and utilisation of that redundancy is critical to the effectiveness of the tolerance that its inclusion is meant to provide. The method in which redundancy is used within a system may be categorised into either *static redundancy* or *dynamic redundancy* [Avi76], although the combination of both methods

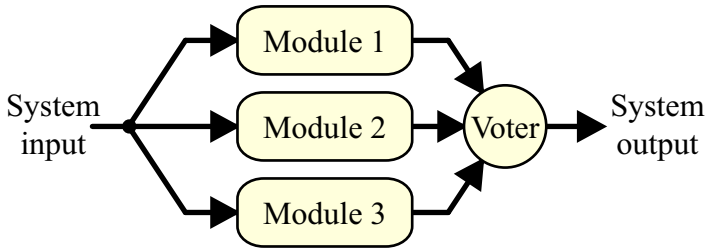


Figure 2.14: *Triple Modular Redundancy (TMR) is a form of static redundancy. A vote taken on the output of three functionally identical systems provides masking of errors.*

produces a third, *hybrid redundancy*. Each method is now considered in the following sections.

2.3.1 Static Redundancy

The term *static redundancy* refers to an architecture that utilises redundancy simultaneously, in a fixed and unchanging manner. It provides fault tolerance by masking the effects of a fault, so is also known as *masking redundancy* [Lal01].

2.3.1.1 Triple Modular Redundancy

The triple modular redundancy (TMR) architecture is one form of static redundancy. It is a popular method of implementing fault tolerance due to the relative ease with which it may be retro-fitted to a system [Avi76]. The architecture utilises three functionally identical copies of a system, all receiving the same input. Each system output is then fed into a voter which is used to generate an overall output. The key element in this architecture is the voter, which provides error masking. A diagram of TMR is shown in Figure 2.14.

The standard implementation of the TMR voter outputs a value representing the majority of the inputs; the truth table is shown in Table 2.2a. This being so, the number of failed systems TMR may tolerate is just one, otherwise a majority vote with a valid result cannot be made. However, there is a situation when this is not so. If two modules fail such that they always have complimentary outputs, the modules in effect cancel each other out, and the voter output follows the correct module. Such a situation is shown in Table 2.2b. The M_1' and M_2' columns show the complimentary outputs of the failed modules, but the voter output can be seen to still follow the non-failed M_3 module.

An attractive feature of TMR, and static redundancy in general, is that error masking is both automatic and immediate. When a TMR system with a full complement of operating

Module Outputs			Voter
M_1	M_2	M_3	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(a)

Module Outputs			Voter
M_1'	M_2'	M_3	Output
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1

(b)

Table 2.2: *The TMR voter implementation is very simple (a), it has the interesting property that two inputs with complimentary values, the result of double failure, still lead to the correct output (b).*

modules suffers a single module failure, there is no glitch in the voter output. In fact, to an observer of the TMR output, no indication of this failure event will be seen at all.

The transparent nature of TMR makes it ideally suited to implementation at any point within a system structural hierarchy. Depending on the tolerance coverage required, it may be necessary to triplicate a whole system, possibly requiring a considerable amount of hardware. Or it may be more appropriate and efficient to implement TMR further down the hierarchy, at the sub-system level, perhaps to cover a unit with greater susceptibility to faults, or a more critical need for fault tolerance.

However, the implementation of TMR does not automatically guarantee the increased dependability of a system. Figure 2.15 shows how the probability of survival, or reliability, decreases with time for a system both in simplex and TMR form. It can be seen that the TMR system has a greater probability of survival only up to a certain point in time, after which the simplex system is more likely to survive. The point at which this cut-off occurs is dependant on the failure rate of the simplex system, or TMR module. The lower the number of expected system failures per hour, the longer TMR architecture provides improved dependability.

If the system specification has a required operational time that fits within the TMR/simplex cutoff, then TMR is a viable fault tolerance technique to improve dependability. The time of sustained reliability may be lengthened by increasing the number of redundant modules used to produce a voted output. A more general form of TMR is called NMR

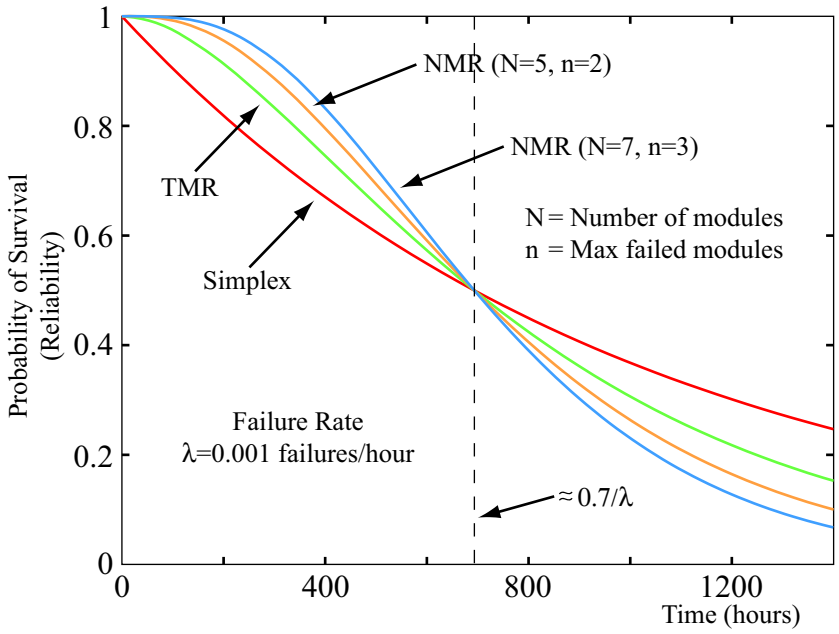


Figure 2.15: *The TMR architecture only provides improved dependability over its simplex alternative up to a specific point in the systems lifetime. This point is determined by the failure rate of the simplex system.*

where N denotes the number of modules. N is usually an odd quantity so that a majority vote can be established. As N increases, the number of tolerable failed modules, n, whilst still being able to form majority, also rises

The probability of survival for an NMR architecture with 5 and 7 modules is also shown in Figure 2.15. The formula used to generate these plots may be found in Appendix A.3

Whether TMR is used, or a greater number of redundant modules, the weak point of the architecture is the voter. The data portrayed in Figure 2.15 is based on a voter that is permanently fault-free. In certain circumstances this is almost a realisable aim. If the voter only operates on simple digital signals the required circuitry is small and, with the employment of greater fault prevention effort to this area, an improved resilience to faults can be achieved. However, where voting involves more complex signals the required voter represents a greater single point weakness.

Although strictly a fault prevention technique, it is worth noting that, when a set of functionally identical modules are required for a fault tolerance technique, efforts should be made to make their design and manufacture processes as dissimilar as possible. This measure ensures that a module with a weakness in one of these areas will not be present in the others. However, applying this procedure to a TMR or NMR scheme is not always

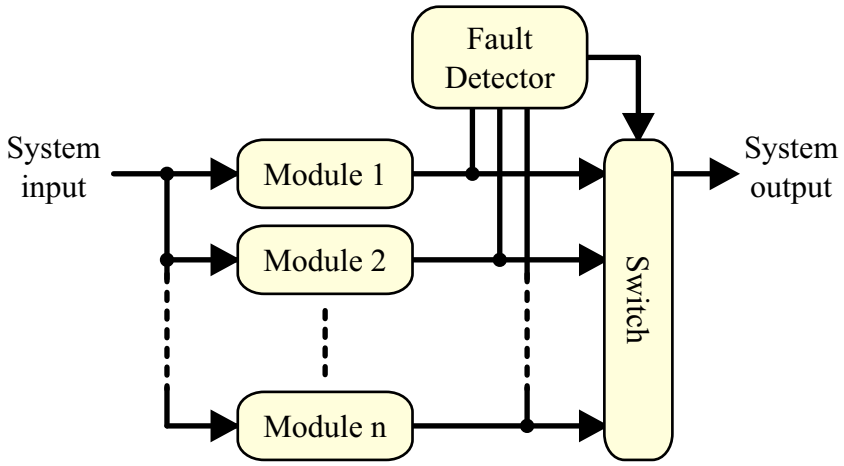


Figure 2.16: *The Dynamic Redundancy architecture uses a single operational module, which on failure is replaced by one of a bank of spares.*

straightforward. Any timing difference between the transitions in modules outputs will appear as glitches after the voter.

2.3.2 Dynamic Redundancy

The architecture of a *dynamic redundancy* scheme, in contrast to a static scheme, reconfigures to achieve tolerance of faults [Lal85]. System services are performed by a single module. If that module is detected as failed, it is replaced by one from a bank of spares. Such a system is shown in Figure 2.16.

Spare modules may remain dormant until they are required, a cold-standby system, or may be running along with the current in-service module, a hot-standby. The advantage of hot-standby spares is that they will have received the same data, and will hold the same state information as the in-service module, thus easing the switch over between faulty and spare modules. However, operational hot-standbys are susceptible to a wider range of failure mechanisms than cold-standby modules, and may well develop faults whilst waiting to be used.

For fault tolerance to be achieved, a fault detection circuit is required to decide when the current in-service module needs to be replaced. This process can represent a major disadvantage of the dynamic redundancy scheme. During the time it takes for the failure of the in-service module to be detected, and the new module to be switched in, the *reconfiguration latency*, the current system output will have been erroneous. Furthermore, depending on the format and value of the output data, the event of switching between operating modules may introduce glitches into the signal.

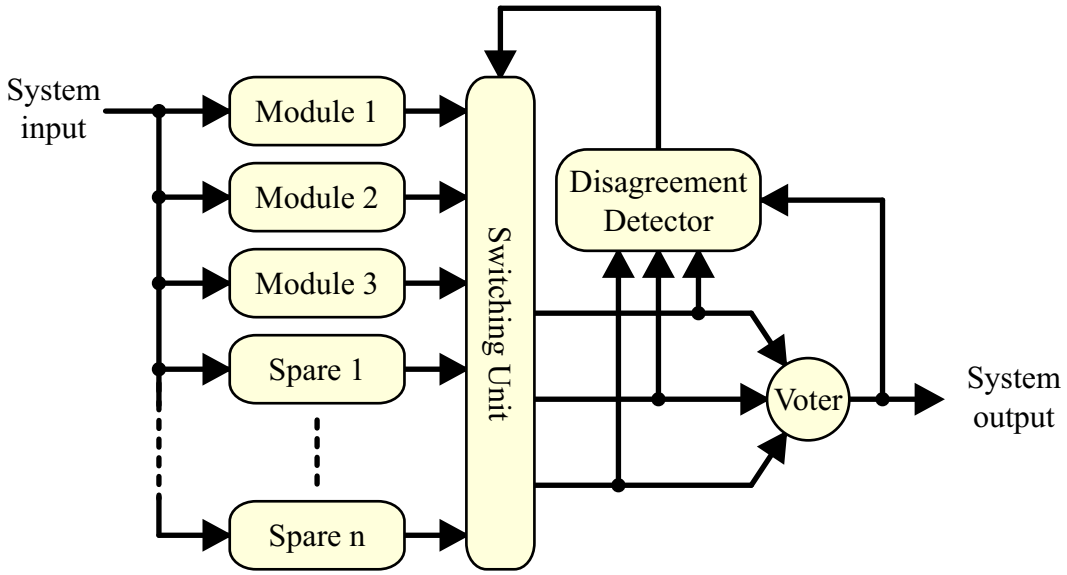


Figure 2.17: *This hybrid redundancy architecture utilises TMR for fault masking and a dynamic redundancy scheme to replace failed TMR modules.*

The design of the fault detection system must be carefully considered. If the system falsely detects failures, the use of the current in-service module will be wasted. Worse still, if incorrect detection occurs regularly, all the redundant modules will be used and removed far more quickly than is necessary. Conversely, a fault detector that allows failure to pass will result in erroneous data at the output.

The fault detection system and module output switch are two weak points of the dynamic redundancy architecture. Although the switch can be very simply implemented, and consequently can be dealt with in the same manner as the TMR voter, the fault detection system is likely to be more complex, and therefore a more likely candidate for failure.

2.3.3 Hybrid Redundancy

A *hybrid redundancy* architecture brings together features of both static and dynamic redundancy. The system, depicted in Figure 2.17, utilises TMR for fault masking, and dynamic redundancy to replace failed modules within the TMR structure [Lal85].

The switching unit is used to connect separate modules to each of the three voter inputs. On the event of a module failure, the disagreement detector, a simple comparator, identifies which unit is providing erroneous data. The switching unit is then able to remove the failed module, and replace it with a spare. The correct operation of this hybrid redundancy architecture is dependant on there only being a single failed module within the TMR structure at any time. If more than one modules fails, fault masking is

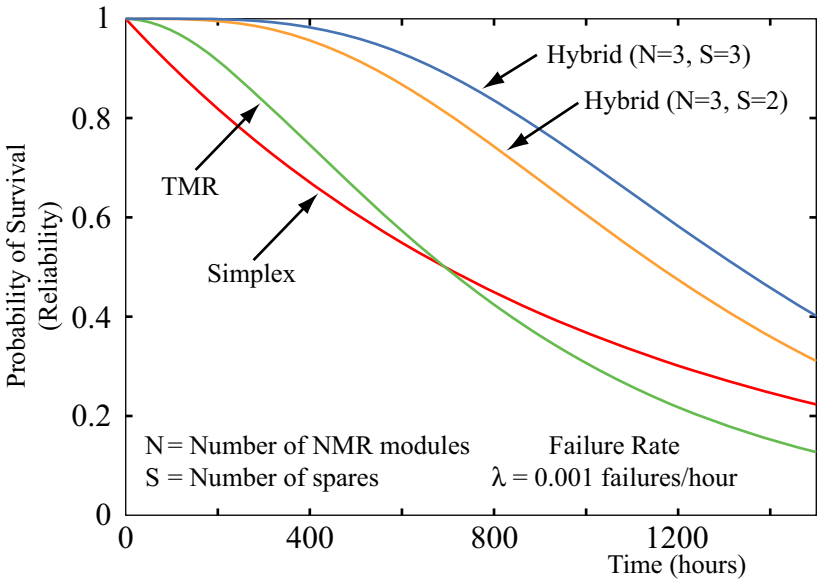


Figure 2.18: *This graph show the change in reliability with time of two hybrid redundancy systems compared to that of simplex and TMR systems. One hybrid system has two spares, $S = 2$, the other three, $S = 3$, both use TMR for masking. The voter, switch and disagreement detector are assumed to never fail.*

lost, and the disagreement detector is likely to switch out the wrong module. This limit may be overcome by using an NMR scheme with more than three modules.

Figure 2.18 shows the change in reliability with time for the hybrid redundancy architecture with varying numbers of spares. The reliability of the simplex system and TMR architecture is also shown for comparison. The formula used to produce these plots can be found in Appendix A.4.

2.4 Summary

Electronic systems are vulnerable to a large variety of failure mechanisms. Some are instigated by human mistake and consequently have scope for removal. However, a great deal of others occur due to the simple ageing process of components or uncontrollable natural phenomena. It has been shown that the risk these latter threats place on system dependability can be reduced by the application of assorted fault tolerance techniques.

As electronic systems find themselves put to ever wider use, more situations arise where reliable operation is a necessity. Consequently, the application of reliability engineering techniques is becoming more widespread. The combination of increasing

circuit complexity and the wider fault tolerance technique application advocates the need for further research into novel fault tolerant architectures.

Biology has evolved a number of impressive methods to ensure the dependability of animal systems. Biological organisms are highly complex, but still present fault tolerance characteristics. Furthermore, such tolerance is achieved in architectures that support both fault detection and recovery. There are a number of valuable and inspiring lessons to be learnt from biology that, on transfer to the electronics domain, may produce novel, but more importantly, more effective reliability engineering approaches. The following chapter provides an insight into the particular features of biological systems used in this research to inspire the creation of a new fault tolerant architecture.

Chapter 3

Biological Inspiration

This chapter provides background information on the two main features of animal biology that have served as sources of inspiration for this research. These being cell development and the endocrine system.

Section 3.1 provides an insight into the nature of cells, specifically the mechanisms that allow single cells to develop into a multitude of different cell types all using the same biological instructions of DNA.

Having established the core facts that underpin the power of cells, Sections 3.2 and 3.3 introduce the subject of cell communication. Advanced biological organisms make use of a variety of communication methods each tailored to specific situations. Using these techniques, cells are able to influence each other's behaviour and invoke an assortment of reactions.

The endocrine system, described in Section 3.4, provides animals with a control system to regulate their internal environment. By utilising a mixture of cell communication techniques, it enables organisms to maintain a stable physiological state.

3.1 Cell Development

Multicellular organisms such as mammals and other advanced vertebrates are constructed from a vast number of different cells and cell types. However, the entire cellular structure that forms these *higher animals* originates from just a single cell. To achieve this amazing feat of construction, that originating cell, called a *zygote*, requires a detailed plan

of how to develop. Furthermore, as subsequent cells are produced to allow the organism to grows, they too must contain a copy of this plan. [Wol91; Wol02]

3.1.1 DNA

The development of an organism depends on a cell's ability to store and interpret information. Likewise, the operation of a cell requires numerous sets of instructions that must be available for retrieval when needed. Biology has evolved such a database that is capable of storing all the development knowledge required for an organism to develop fully, and able to hold all the instructions needed for each cell type to operate. What is more, this database can be stored within a small proportion of every single cell that makes up the organism [Lod00].

The key to cellular function and development are proteins. They act as structural building blocks and form enzymes that initiate all cellular chemical reactions. Proteins also enable cells to move, and communicate. In fact, the proteins that a cell can produce almost entirely dictates the cells function and properties. Therefore, proteins can be considered an information common denominator for cellular life.

In the 1940s *deoxyribonucleic acid* (DNA) was identified as a potential information storage mechanism. However, it was James Watson and Francis Crick in 1953 who first provided clues that DNA may encode the instructions necessary to create proteins. The double helix structure of DNA is now well known and universally excepted as the blueprint of all living things.

3.1.1.1 DNA Structure

DNA is constructed from two long chains, each called a DNA strand. Each chain 'link' consists of a base and a sugar-phosphate which together are called a *nucleotide*. All the sugar-phosphates in one chain are connected together, forming a backbone which holds the strand together. The bases are connected via hydrogen bonds to those of the opposing strand, thus forming the double strand. This arrangement can be seen in Figure 3.1.

The DNA strands exhibit chemical polarity; that is one end may be readily discerned from the other. The nucleotide sugar-phosphate sections are arranged along the axis of the chain such that the backbone always follows an alternating sequence, sugar-phosphate-sugar-phosphate. This property ensures that each nucleotide is orientated the same way.

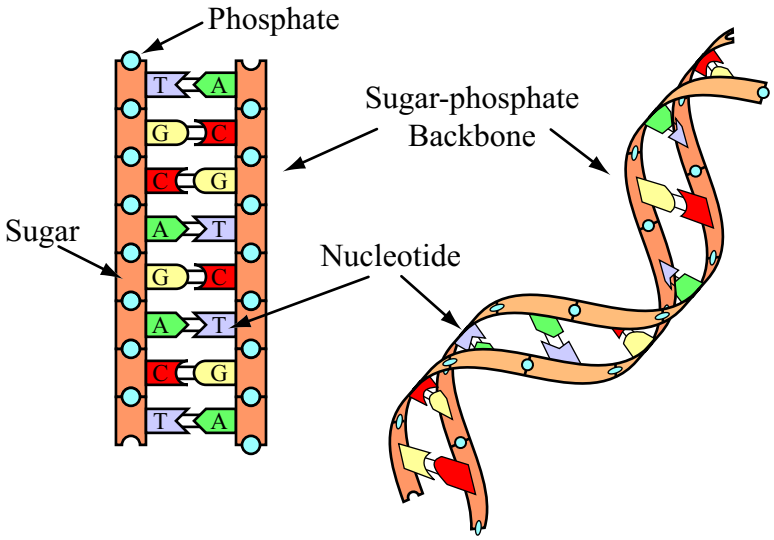


Figure 3.1: *DNA is the blueprint of life. A long chain of paired nucleotides encodes the information necessary to create an organism, in a format that may be copied and passed from cell to daughter cell.*

The polarity of the two DNA strands are opposite, with each strand's 'spare' phosphate being at opposite ends of the helix.

The controlled orientation of the sugar-phosphate backbone points the nucleotide bases towards the centre of the helix allowing them to bond. However, the pairings are not random. DNA consists of four base types, adenine, cytosine, guanine and thymine, given the symbols A, C, G and T respectively. Adenine is always paired with thymine, and cytosine with guanine. These pairings ensure that the opposing sugar-phosphates will be of equal widths, thus maintaining nucleotide alignment along the DNA molecule.

3.1.1.2 Genes, Chromosomes and Genomes

The nucleotide sequence of a DNA molecule contains subsequences called *genes*. It is these genes that provide a functional unit for the production of proteins. The complete set of DNA molecules that describe an organism is called the *genome*. The human genome contains approximately 3200 million nucleotide pairs. *Chromosome* is the name given to a single length of DNA molecule that consists of many genes. The human genome is divided into 24 different chromosomes, of which 2 are sex chromosomes, and 22 are non-sex types or autosomes.

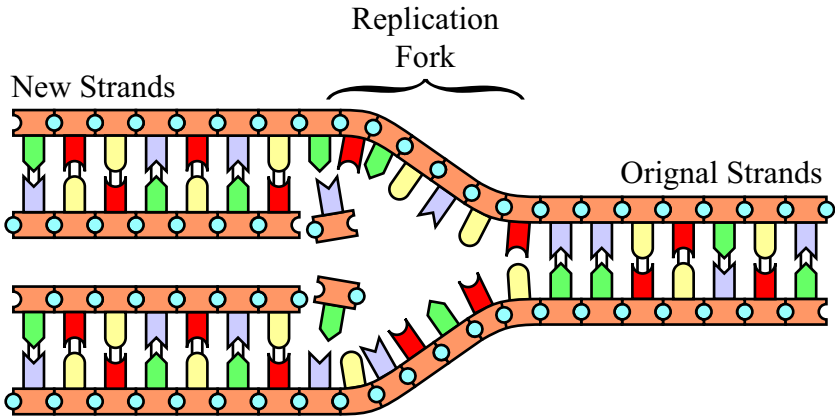


Figure 3.2: *The two complementary strands that make a DNA molecule allow it to be accurately replicated.*

3.1.1.3 DNA Replication

Cells must be able to replicate their DNA if they are to pass on the vital information it contains to daughter cells. The consistent and exclusive pairing of nucleotide base types is fundamental to this process. The base sequence of one DNA strand is always exactly complementary to its partner. Therefore one strand may act as a template to create another.

DNA replication, shown in Figure 3.2, takes place within the cell *nucleus* and requires the double DNA strand to be split so that a nucleotide template can be exposed. The energy required to separate the entire length of DNA is huge due to the large collective strength of the hydrogen bonds between the strands. Consequently, only small portions are unstitched at a time, as individual bonds can be easily broken with little expenditure of energy.

The positions at which the DNA molecule may be opened are called *replication origins*, and are specifically marked by their own sequence of nucleotides. The human genome contains some 10 thousand of these origins. To start replication, a certain type of initiating protein binds to a replication origin and creates the initial opening. A collection of other proteins, called the *protein machine*, is then drawn towards the exposed strands and starts replication. The protein machine moves along the DNA molecule unzipping the strands, adding new nucleotides, then trailing two copied DNA molecules behind it. The Y shaped region formed by the separating strands is called the *replication fork*, and on human DNA it moves at a rate of 100 nucleotide pairs a second.

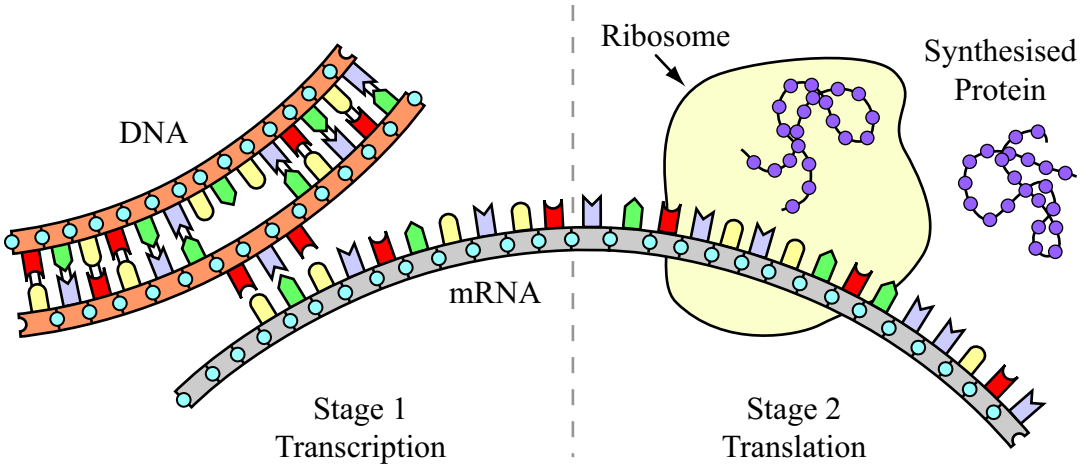


Figure 3.3: *The conversion of DNA into proteins involves two stages, the transcription of DNA into mRNA, and the translation of mRNA into the protein sequence of amino acids.*

In humans, two replication forks move apart from the same replication origin forming a bubble shaped gap in the DNA. This bidirectional process ensures that the complete DNA molecule is copied. In fact a number of replication bubbles normally occur concurrently during the same replication process, vastly increasing the overall speed of replication. The entire copying process is said to be *semiconservative* as the two resulting DNA molecules are formed with one original strand paired with one new strand.

3.1.2 Creating Proteins from DNA

The replication of DNA is important as it enables the stored information to be passed from cell to cell. However, equally important is a mechanism to retrieve the genetic information, and produce something with it. Section 3.1.1 introduced proteins as fundamental elements of cells, determining their structure and function. Each protein type is constructed from a unique linear sequence of amino acids. It is these sequences that are encoded in DNA.

The conversion of DNA to proteins, as illustrated in Figure 3.3, requires two stages. The first, called *transcription*, converts the area of DNA encoding a particular protein into a new nucleotide sequence called messenger ribonucleic acid (mRNA). RNA is chemically very similar to a single DNA strand but with two differences, the sugar in the sugar-phosphate backbone is ribose rather that deoxyribose, thus the difference in name. Secondly, RNA uses the base uracil instead of thymine.

mRNA is also produced in the nucleus in a process similar to DNA replication. However, instead of the new strand staying attached to the DNA template, they divide allowing

the DNA to join back together. In fact the almost immediate recombination of DNA strands means transcription can start on the same area of DNA before the previous mRNA molecule has been finished.

The second stage is to convert the transcribed mRNA into proteins. This stage, called *translation*, converts the code of DNA into the code of amino acids, the building blocks of proteins. However, as there are more amino acid types than the four RNA base types, each is described in a three nucleotide wide unit called a *codon*. This code allows $4^3 = 64$ different amino acids to be referenced. In fact only about 20 different amino acid types are found in proteins, so most codons represent the same thing.

The processing of the mRNA into proteins takes place outside the nucleus in factory blocks called ribosomes. Each ribosome attaches itself around the end of a single mRNA molecule allowing it to read through the code. With every codon read, a new amino acid is added to the growing protein sequence. Once the complete mRNA molecule has been decoded, the new protein is complete and is released.

3.1.3 Stem Cells

In the early stages of embryo development, the dividing cells must at some point specialise to a specific type. However, before this happens the cells, called *embryonic stem cells*, have the potential to become any cell type within that developing organism. In fact, their universal nature stretches beyond even this. Embryonic stem cells are said to be *totipotent*; each one is singularly capable of developing into a whole new organism.

This property has fuelled a great deal of research into embryonic stem cells. The hope is that, if the development of stem cells may be controlled, specific tissues or even organs may be grown to repair damaged or faulty body parts.

The controversy that surrounds this research is based on how embryonic stem cells are acquired. A living embryo is required for embryonic stem cells to be removed, a process that destroys the embryo. However, there are other versions of stem cells that may be retrieved without harm.

Adult stem cells are not as potentially universal as embryonic stem cells, however they are still believed to be *pluripotent*. They are capable of specialising into a number of different cell types. Within the body, these cells are available to replace certain cell types

that have died. The hurdle for researchers is to find out what conditions allow adult stem cells to specialise into the desired type.

Underlying the power of all stem cells is DNA. This complete set of structural and functional instructions is what ultimately allows stem cells to have the potential to specialise in any direction.

3.2 Cell Communication

Cells make use of a number of different techniques to accomplish intercellular communication. However, the type of players involved are generally the same. The *signalling cell* initiates communication by releasing a particular type of *signalling molecule*. By means of a *receptor protein*, the *target cell* is able to detect the signal, completing the communication [Alb02].

In the first stage of signalling, molecules that travel from signalling cell to target cell are called *first messengers* and may be one of a great number of different chemicals. However, the communication methods in which they are utilised are far fewer in number, with each method following the same sequence of stages. The stages are:

1. **Synthesis** of signalling molecules.
2. **Exposure** of these molecules to extracellular space.
3. **Transportation** of signals towards their target cell.
4. **Detection** by the target cell.
5. **Response** generated by the target cell.
6. **Removal** of the signalling molecule.

In fact, the way in which the first messengers propagate from source to target provides the primary means of classifying communication techniques. Sections 3.2.1 to 3.2.4 describe the five main intercellular communication mechanisms [Coo00; Alb98].

3.2.1 Endocrine Signalling

Endocrine signalling is by far the most public form of intercellular communication. Using this technique, messages can be broadcast throughout the whole body. The signals are

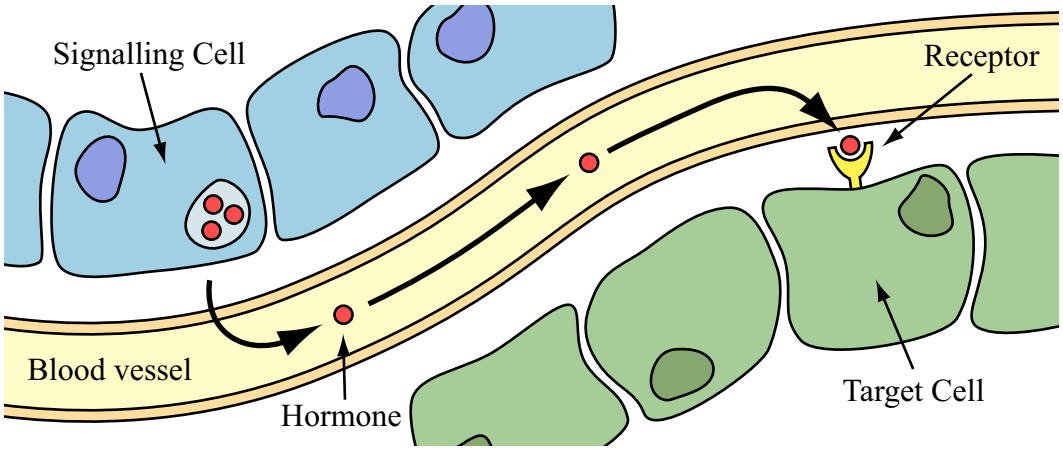


Figure 3.4: *Suited to long distance and public communication, endocrine signalling releases hormones into the bloodstream so they may be carried towards their target cells.*

synthesised within *endocrine cells*, from which they are released into the bloodstream. The signalling molecules used during endocrine communication are called *hormones*, and take their name from the Greek to 'excite' or 'arouse'.

By using the circulatory system as a transport medium, the hormones are able to travel great distances to reach their target cells. Figure 3.4 depicts how endocrine signalling is achieved.

3.2.2 Paracrine and Autocrine Signalling

In contrast to the very public endocrine communication mechanism, paracrine and autocrine signals operate locally. *Paracrine signalling* provides communication between a cell and its close neighbours. This is useful where the effect of the signal molecules must be restricted.

As an example, during wound healing, highly potent molecules called cytokines are released into the area of injury. These molecules cause the skin cells to express more adhesion molecules, helping them to rebind. Furthermore, they activate macrophage cells which remove pathogenic materials from the exposed area. They also initiate the production of growth-factors that help mend the wound. These clearly powerful signalling molecules are contained by the inflammation that they cause.

To maintain the locality of paracrine signals, in this case called *local mediators*, the signalling molecules are very shorted lived. If they are not rapidly taken up by their target cells they may be destroyed by enzymes in the extracellular space. Movement

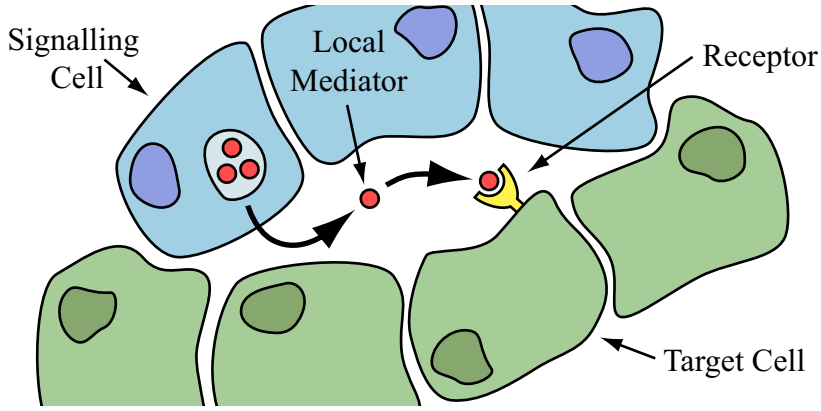


Figure 3.5: *Local mediators used in paracrine signalling provide short distance communication to neighbouring cells.*

may be further restricted by the *extracellular matrix*, the physical structure around the cell. Figure 3.5 depicts paracrine signalling.

Autocrine signalling, depicted in Figure 3.6, takes place when the released signalling molecules bind back to the source cell. Cells may use this approach to reinforce the reception of weak signals.

Groups of like cells can use autocrine signalling to coordinate collective decisions. A single cell surrounded by different cell types would receive a weak autocrine signal. However, a cell surrounded by like cells all releasing the same signal would receive a very much stronger signal. This effect is depicted in Figure 3.7a and 3.7b respectively.

The ability of a cell to stimulate itself via autocrine signalling can sometimes become out of control. Growth factors have the effect of stimulating cells to grow and proliferate. Cells that respond to their own growth factors can proliferate uncontrollably and may

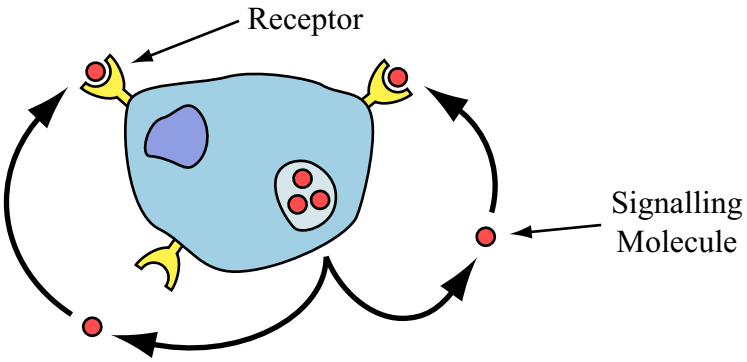
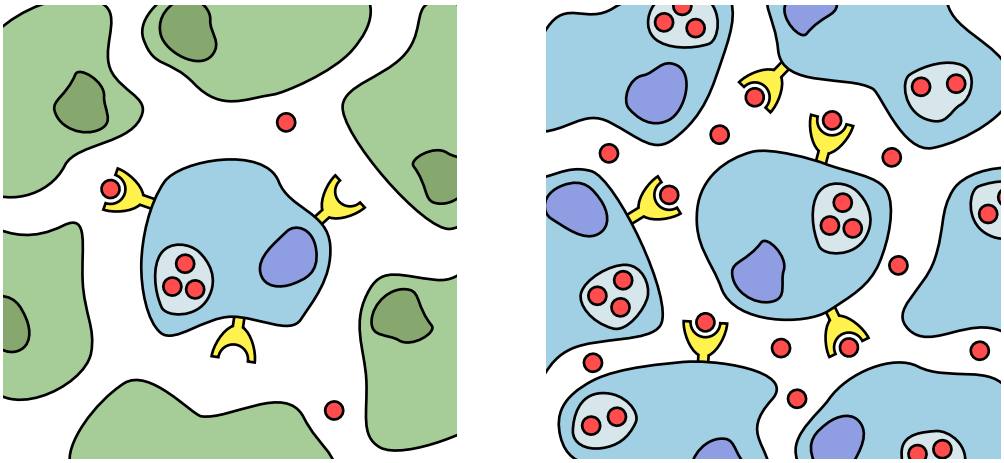


Figure 3.6: *Cells may also release signals destined to themselves. This is called autocrine signalling.*



(a) (b)

Figure 3.7: *(a) A cell surrounded by cells of a dissimilar type will receive a weak autocrine signal. (b) The autocrine signal for a group of like cells will be strong.*

also stimulate neighbouring cells to do likewise. This process can lead to the creation of tumours.

3.2.3 Contact Dependant Signalling

Not all communication techniques involve the release of signalling molecules. During contact dependant communication, signalling cells attach their signal molecules to their outer layer, called the *plasma membrane*, where they are exposed to the extracellular space. Target cells have to be in direct contact with the signalling cell so that their receptors may receive the signal.

Contact dependant signalling, as shown in Figure 3.8, clearly only operates over a very short range. It is therefore well suited to communication within cell tissues. During development, groups of unspecialised cells use this signalling technique to organise the direction in which each cell should specialise. An example of this behaviour is found in the formation of the nervous system. A cell that has specialised as a neuron will send inhibitory messages to its neighbours stopping them also becoming neurons, thus leaving them simply as structural cells to maintain the tissue shape.

Another area where contact dependant signals are used is the immune system. Cells that gather and bind antigens, fragments of pathogenic substances, present the material to other immune cells to determine if an immune response should be initiated.

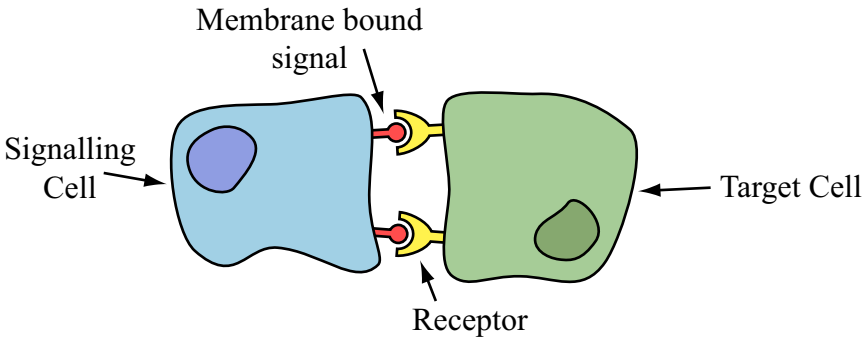


Figure 3.8: *During contact dependant communication, the signalling molecule remains attached to the signalling cell. The target cell has to be sufficiently close to receive the signal.*

3.2.3.1 Gap Junctions

A second contact dependant signalling technique involves the creation of a special *gap junction* between cells. When the plasma membranes of two cells are sufficiently close, the formation of a gap junction directly connects the cytoplasm of both cells. The narrow water-filled gap is only large enough for small signalling molecules called *intracellular mediators* to pass.

Gap junctions also form an electrical connection between joined cells. An interesting example where this is utilised is the heart. When receiving an electrical wave of excitation, the electrical connections between heart cells ensure that their contractions are synchronised.

Figure 3.9 shows the structural parts of a gap junction. A tubular structure called a connexon penetrates the cell plasma membrane creating the channel through which signalling molecules may pass. Each connexon is in turn made up of six parts called connexins. When the connexons of opposing cells are aligned the complete channel is formed.

However, when a connexon is not connected to another it closes itself, thus preventing particles from extracellular space entering the cell. Gap junctions are, in fact, made up of a number of connexons ranging from only a few to thousands. Almost all animal cells utilise gap junctions for communication.

3.2.4 Neuronal Signalling

The signals transmitted via neuronal signalling are delivered in a very direct manner. As shown in Figure 3.10, a long fibre called the *axon* extends from the signalling cell, in this

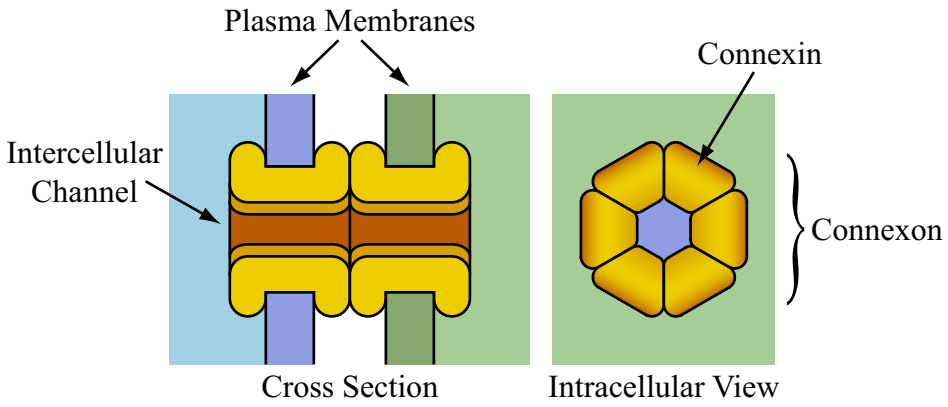


Figure 3.9: *Special connections called gap junctions allow neighbouring cells to directly join their cytoplasm's, creating a path for signalling molecules.*

case called a *neuron* directly towards the target cell. The axon can be many centimetres long allowing direct communication between distance cells.

The connection between the end of the axon and the target cell is made with a *synapse*. Electrical signals from the neuron cause signalling molecules to be released into the synapse. These molecules, called *neurotransmitters*, are then received at the target cell by receptors. As the majority of the signalling distance takes place electrically over the axon, communication is very fast and can reach speeds up to 100 metres per second.

Neuronal signalling initially seems quite different from the other communication methods presented so far due to the involvement of electrical signals over axons. However, it still makes use of signalling molecules released into extracellular space. In this way, neuronal signalling may be considered a remote form of paracrine signalling.

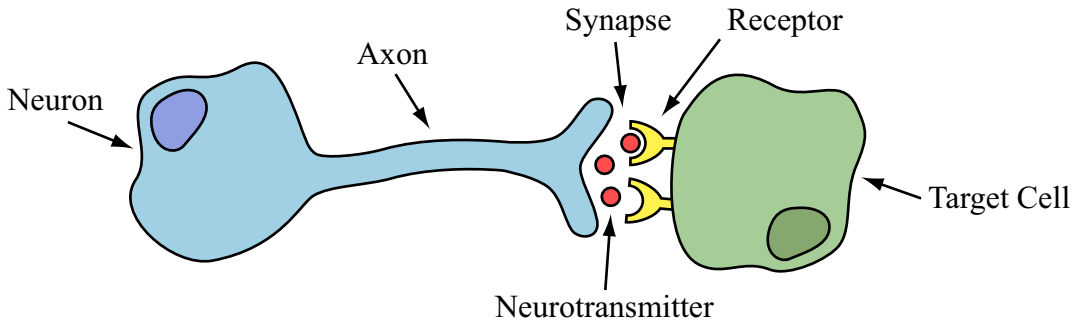


Figure 3.10: *The axon attached to a neuron means a signal can be channelled quickly and accurately to target cells. However, the final communication hop is a form of paracrine signalling.*

3.2.5 Signalling Summary

Biological organisms utilise a substantial variety of communication techniques, each type specialised to a particular function. Endocrine communication provides a signal broadcasting technique that can operate over long distances. Neuronal communication operates over both long and short distances, but is fast and directed to single cells. Where local communication is required, the short-lived paracrine mediators allow only neighbouring cells to be targeted. There is even provision for loop back signalling, where cells can activate their own receptors. All these techniques are summarised in Table 3.1.

The process of intercellular communication does not finish when signalling molecules reach their target. The following section covers the details of signal reception; how messengers are received, the intracellular processes invoked by message reception and how messages can combine to create a greater range of responses.

Type	Messenger Name	Range	Comments
Endocrine	Hormones	Long	A long distance broadcast type communication that uses blood flow to distribute the hormone messengers.
Paracrine	Local Mediators	Short	A local form of communication, allowing neighbouring cells to coordinate behaviour. Local Mediators are short lived to limit their range.
Autocrine	Local Mediators	Zero	When a cell releases messengers to bind to back itself. This is used as a positive feedback technique to reinforce the reception of weak signals.
Contact Dependant	Intercellular Mediators	Very Short	Requires direct cell contact. Messengers are membrane bound, or released through direct channels between cells. Allows for very specific communication.
Neuronal	Neuro-transmitters	Long	The signalling neuron, uses a long axon to release neurotransmitters right to the target. Communication is fast and can cover long distances.

Table 3.1: *A summary of the different types of intercell communication.*

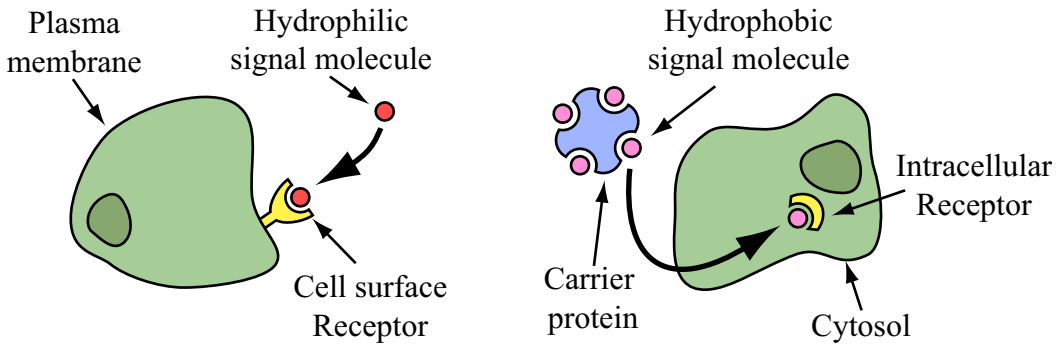


Figure 3.11: *Cell receptors may be located on the plasma membrane or internally to the cell. The water solubility of a signalling molecule impacts which receptors it can reach.*

3.3 Signal Reception

Many different forms of signalling molecule exist. One method of classifying them is by their level of solubility in water and the physical location of the receptors to which they bind. Receptors may exist either on the cell surface attached to the plasma membrane, or inside the cell within the nucleus or cytosol. The latter of these two locations requires that the signalling molecules are able to cross the target cell's plasma membrane.

The water solubility of a signalling molecule directly affects its ability to enter a cell and, therefore, the type of receptor to which it is able to bind. Most signalling molecules are *hydrophilic*, they readily dissolve in water, and are unable to pass through plasma membrane. However, *hydrophobic* or *lipophilic* molecules, those that do not readily dissolve in water, are able to pass through and reach internal receptors. Figure 3.11 illustrates the paths that these two messenger types take to reach a receptor. Because water is not inclined to interact with hydrophobic molecules, it is unlikely that these signalling molecules will be readily transported in the extracellular fluids. They therefore rely on carrier proteins to move them from source to target cell.

3.3.1 The Cell Plasma Membrane

The reason for water solubility determining the ability of a signalling molecule to enter a cell is due to the way in which the cell plasma membrane is constructed. The function of a cell's outer 'shell' is is to keep the internal cell parts together and separated from extracellular space. Keeping two watery areas apart requires a substance that will not

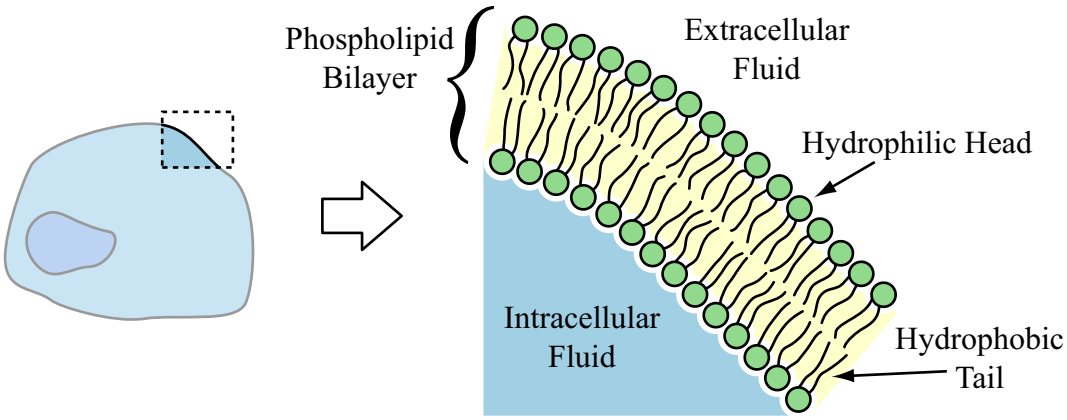


Figure 3.12: *The outer 'shell' of cells are formed by a phospholipid bilayer that directly affects the ability of signalling molecules to enter a cell. The attractive and repulsive forces that hold the layer together also account for the enclosed nature of cells.*

dissolve in water. In the case of cells, this is a phospholipid bilayer, made up of an organic oily substance.

Figure 3.12 shows a simplified cross-section of a cell plasma membrane. The phospholipids that form the membrane have a hydrophilic head which is attracted to water, and two hydrophilic tails that are repelled from water. The formation of a double layer of these phospholipids, called the *phospholipid bilayer* means a surface of hydrophilic heads is available for interfacing with the intracellular and extracellular fluids. The layers are then kept together by the hydrophilic tails that are attracted to each other, and importantly provide the membrane with a water repellent core that keeps the two outer areas separate.

The hydrophilic signalling molecules that are dissolved in the extracellular fluid can not directly pass through the plasma membrane because of the central water repellent core of the phospholipid bilayer. However, the lipophilic messengers may pass unrestricted through the layer, and move directly into the cell.

It is interesting to note that the enclosed and containing nature of cells is fundamentally dependant on their phospholipid bilayer. The bilayer finds itself in its lowest energy state when all hydrophilic parts are only interfacing with water, and all hydrophobic parts are connected to other lipids. To achieve this state the layer must close in upon itself so that no layer edges are exposed to water, and thus forming a self-sealing compartment.

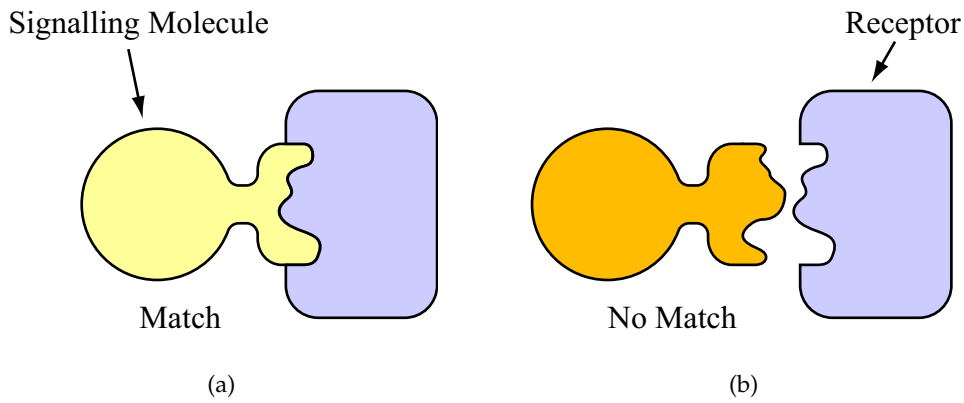


Figure 3.13: *The analogy of a lock and key can be used to describe the selective binding of signalling molecule and receptor.*

3.3.2 Receptors

The variety of different signalling molecules all require receptors to which they may bind and so invoke a reaction. There are consequently many hundreds of different receptor types. Even though it is believed that all cells have the ability to synthesise any receptor type, in general a single cell will present less than ten, although each type is presented in quantities of thousands.

So that a specific signalling molecule type is only able to bind to the receptors of their destined target cell, there must be a selectivity mechanism in the binding process. The lock and key principle, as shown in Figure 3.13, is a good analogy of this mechanism. It takes a signalling molecule, the key, of a specific shape to fit a particular receptor, the lock. Even a slight change in the molecular structure of either receptor or signal can make binding impossible. The closeness of fit between a receptor and molecule is called the *specificity* of the interaction.

An understanding of receptor binding specificity has allowed drug manufacturers to artificially interact with cell receptors. Tailored chemicals called *agonists* are able to fit specific receptors and invoke the same reaction as the naturally occurring signalling molecules. In contrast, another set of chemicals called *antagonists* can also bind with specific receptors but without producing a response. As a result the natural signals are blocked from binding, and the particular response is stopped or repressed.

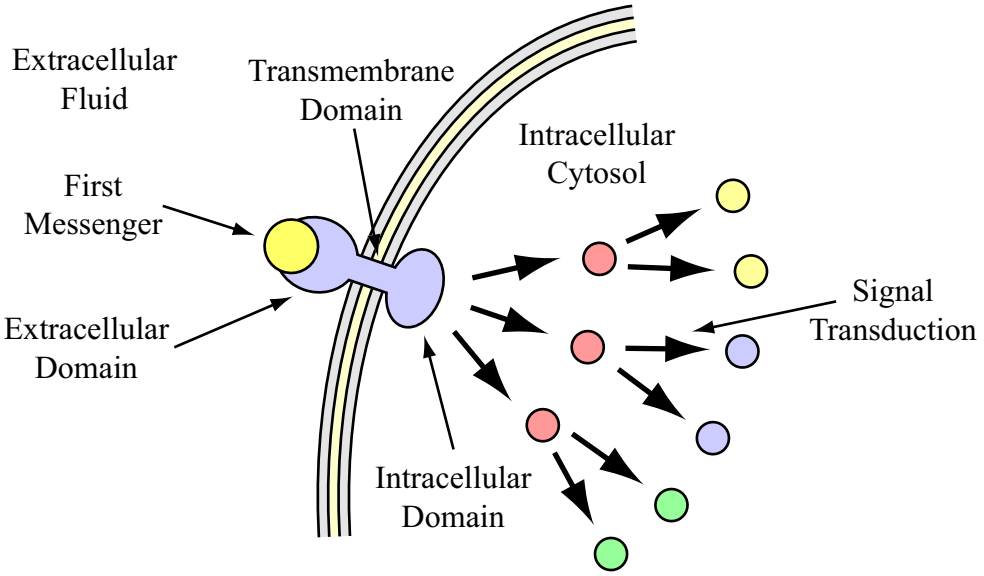


Figure 3.14: *Secondary Messengers relay messages from cell surface receptors into the cell. Multiple signal transduction stages further increases the effect of the message.*

3.3.3 Secondary Messengers

When a signalling molecule attaches to a receptor bound to a cell membrane, its progress is stopped and its task is complete. However, if a response is to occur, the message must be passed on into the cell. Cell surface receptors have three regions that enable this task to be performed. An extracellular domain provides an external point to which signalling molecules unable to enter the cell may bind. A hydrophobic transmembrane domain anchors the receptor to the cell membrane due to its affinity with the membrane's lipid core. Finally, an intracellular domain that protrudes into the cell cytosol is capable of initiating a new signalling pathway that relays the message into the cell.

The new intracellular signal is mediated by signalling molecules known as *secondary messengers*. These messengers, produced by the membrane bound receptors, are in fact only the first of a cascade of different messenger molecules. In a process known as *signal transduction*, one set of signalling molecules stimulates the production of a subsequent signal set, all passing on the same message. The process of first messenger reception to secondary messenger transduction is illustrated in Figure 3.14.

The cascade of intracellular signal transduction provides a number of important functions. These features are:

1. It allows the message to be finally converted into a molecular form that is able to invoke the desired response.

2. Each stage of signal transduction allows the message to be amplified, with each stage utilising increasing numbers of signalling molecules.

3. The multiple stages allow the intracellular signals to be distributed around the cell, thus allowing multiple targets to be reached.

4. The transduction stages are open to interference from other signalling pathways. Therefore the pathway of one signal may be modulated by another, allowing complex interactions and dependencies to be formed.

3.3.4 Responses

The overall objective of communication is to invoke some kind of response. However, the final stages of communication are not so clear cut. A single type of signalling molecule can initiate a different response depending upon the cell type to which it binds.

Figure 3.15 illustrates this point. The neurotransmitter acetylcholine can produce a number of different responses. When binding with heart muscles, it causes the muscles to relax, decreasing the force and rate of heart contractions. However, when the same signal binds with skeletal muscles, it makes them contract. These contrasting reactions can be explained by the difference in the acetylcholine protein receptors used by the two muscle types.

However, receptor difference is not always the cause of dissimilar responses. Salivary glands have an acetylcholine receptor that is very similar to that of heart muscles. In this case, on binding, the salivary glands are stimulated to produce saliva.

3.3.4.1 Message Combinations

The stimulation of cell responses is not limited to the reception of a single signalling pathway. Some cell reactions are dependant on the reception of a combination of signals. A very common example of this are the cell survival signals. Most cells have an in built mechanism that allows them to self destruct. The process, called *programmed cell death*, or *apoptosis*, requires the reception of multiple signals to be kept dormant. This suppressive approach means cells that stray from their correct position or circulatory route no longer

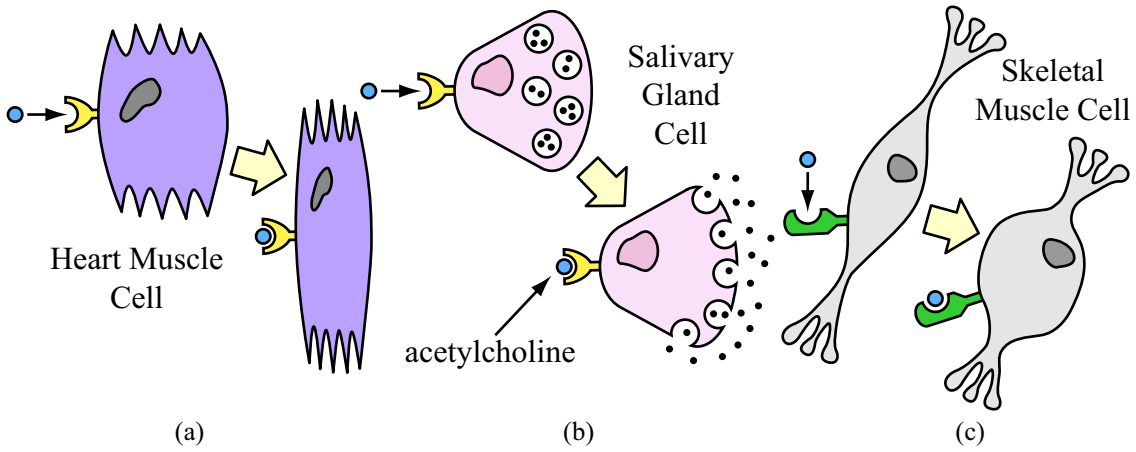


Figure 3.15: *The same signalling molecule type can invoke differing responses depending on the cell type it binds to. Neurotransmitter acetylcholine causes heart muscles to relax (a), salivary glands to release saliva (b) and skeletal muscles to contract (c).*

receive their survival signals. They then die so the chance of them becoming a potential problem is removed.

Figure 3.16 depicts how the same single cell will undergo a variety of responses depending on the combination of signals it receives. The combined 'A' group of signalling molecules simply keep the cell alive. These, along with the combination of 'B' signals cause the cell to differentiate, or specialise into a new cell type. Similarly, the 'C' signals along with the survival signals cause cell division. The absence of any signals initiates cell death.

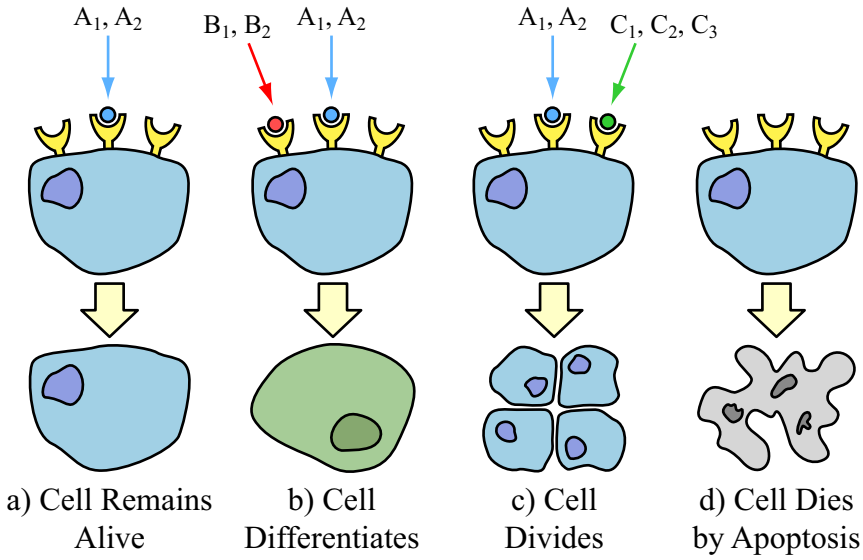


Figure 3.16: *Signals can operate in combinations upon a single cell allowing a richer set of responses to be initiated.*

3.4 The Endocrine System

In order to survive, our bodies must be capable of adjusting to the physiological changes that are inflicted upon it. Changes in environmental factors such as temperature, humidity and social situation all require responses to help maintain the equilibrium of our physiology. As well as these external influences, activities such as eating and exercise more directly alter our internal environment, which also requires subsequent action to make sure physiological balance is kept. The primary role of the endocrine system is to provide this maintenance system [Bro01; Gre94; Gar98]

3.4.1 Primary Endocrine Glands

The endocrine system operates using a set of glands capable of influencing the behaviour of the body's cells. Using the cell communication techniques described in Section 3.2 the primary endocrine glands release signals that invoke physiology altering reactions. This set of primary glands are shown in Figure 3.17. However, this illustration is not exhaustive. There are other cells in the body that contribute, to a lesser extent, to endocrine control.

3.4.2 Glucose Regulation

The brain is completely dependant on the supply of glucose for fuel. The dependency is due to neurons not being able to use any other energy source to operate effectively. Therefore, when glucose levels drop, it is extremely important that extra glucose is found and added to the blood stream, otherwise it can lead to coma, or even death.

The regulation of glucose levels within a narrow, allowable band is ultimately controlled by two hormones, *insulin* and *glucagon*. Special channels in a cell's plasma membrane allows glucose to enter. The state of these channels can be controlled by the hormone insulin. When the insulin receptor is bound, secondary messengers within the cell causes the opening of more channels, allowing more glucose into the cell.

As shown in Figure 3.18 when blood glucose levels are detected to be too high, insulin is released by the pancreas. The stimulation of cells to increase glucose uptake reduces it concentration back to safe levels. Insulin is the only hormone that has this effect. However, there are a number of hormones that will increase blood glucose levels. It

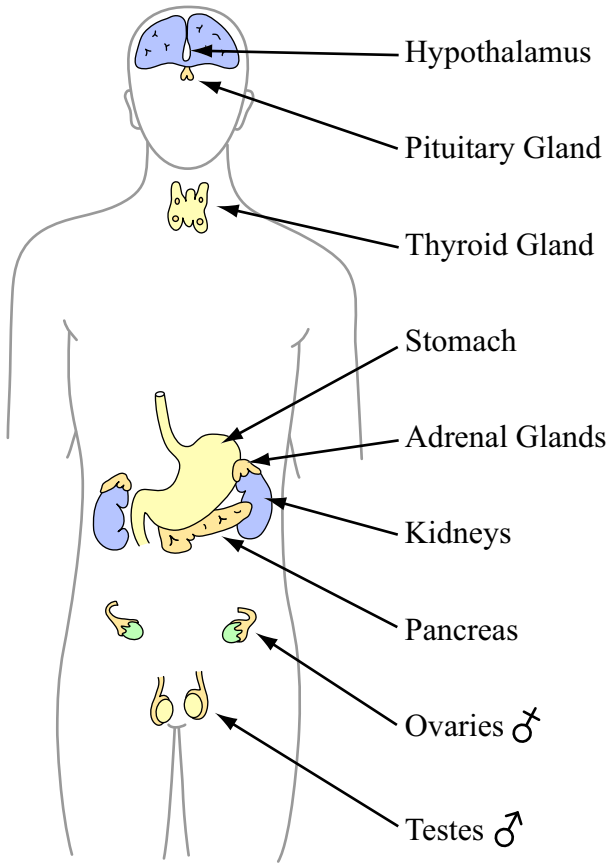


Figure 3.17: *The primary role of the endocrine system is to maintain physiological equilibrium. It achieves this using cell communication, the majority of which are instigated by the primary endocrine glands.*

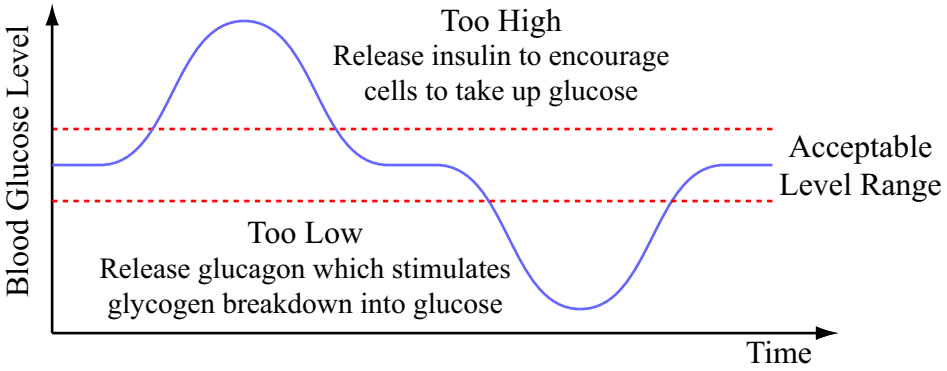


Figure 3.18: *Blood glucose levels must be kept within a narrow acceptable band. The hormones insulin and glucagon help maintain this level by stimulating glucose removal, or production.*

is believed this is because it is far more dangerous, in the short term, to have too little glucose, than too much.

One of the hormones responsible for increasing blood glucose concentration is glucagon. Also released from the pancreas, this hormone stimulates the liver to produce glucose in two different ways. The liver extracts glucose from the blood when it is in surplus and stores it in the form of *glycogen*. The glucagon hormone causes the release and breakdown of glycogen back into glucose. Secondly, glucagon activates a series of chemical reactions called the *gluconeogenesis metabolic pathway* which creates new glucose.

3.4.3 Endocrine Control

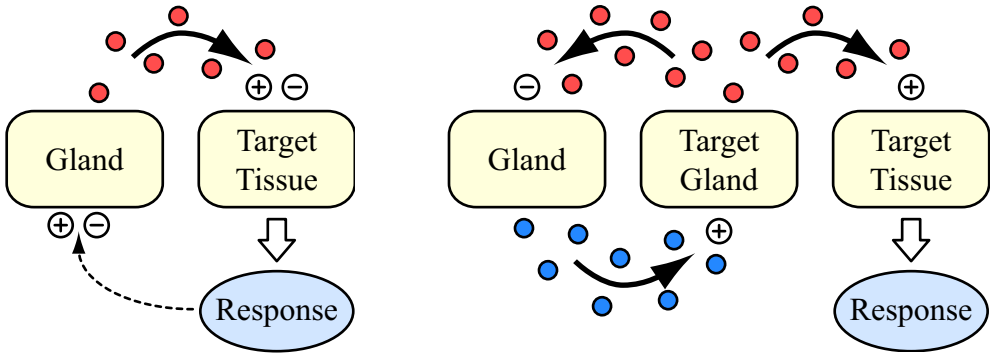
The glucose regulation example presented in Section 3.4.2 illustrated how two hormones can provide physiological control. In general, most hormones are in constant circulation and it is a change in their concentration that triggers a response. The three factors that determine the level of hormones in the body are as follows.

1. **Production Rate** is the most regulated influence of hormone concentration. The synthesis and secretion of hormones is controlled by positive and negative feedback loops, both of which are discussed later.

2. **Delivery Rate** can determine how quickly target receptors will notice a change in hormone concentration. Signalling cell location will determine circulation speeds and therefore signal transport time.

3. **Degradation and Elimination Rate** depends on whether hormones are actively removed from circulation by excretion or metabolism, or are allowed to degrade naturally. Hormones with long half-lives will remain in circulation long after their production has ceased. However, those with a shorter half-life will show much faster decay.

The feedback loops that regulate hormone production provide the endocrine system with robust control over its operation. Figure 3.19 shows two of these possible control scenarios [Bro01]. In Figure 3.19a, the gland releases hormones that may either stimulate or inhibit a response in the target tissue. This change in response is directly fed back to the originating gland which may be positively stimulated to produce more hormones and further exaggerate the response, or negatively inhibit the gland to stop the response.



(a) (b)

Figure 3.19: *The endocrine system uses both positive and negative feedback to provide physiological control. Feedback may come directly from the response, or via further signalling molecules.*

By far the most common scenario is for negative feedback to occur. Positive feedback is utilised, but rarely due to its unstable nature.

The second control scenario shown in Figure 3.19b involves two different hormone types. The initiating gland releases the first set of hormones which stimulates a second gland. The invoked reaction is the production of a second hormone type. These hormones serve a dual purpose, they are fed back to the initiating gland, inhibiting its production of the first hormone type. The same hormones also stimulate the final target tissue to initiate the response. Due to the inhibition of the original gland the target gland will eventually no longer receive the stimulatory hormones, and will therefore in turn stop stimulating the final target tissue.

3.4.4 Complex Signalling Pathways

The simplified control scenarios of Figure 3.19 do not take into account the fact that a whole sequence of communication stages may be in operation within in a single physiological control loop. As shown in Figure 3.20, many separate communication steps and types can form a single signalling pathway. In this case, via a long signalling pathway, a neuron stimulates the thyroid gland into hormone production and release. Such hormones are integral to the control of metabolism and growth [Har91].

The endocrine system is highly complex; many of the control systems are interdependent, with diverse contributing factors. A great deal remains to be learnt and discovered about the endocrine system, and how it keeps our physiology in the correct functioning state.

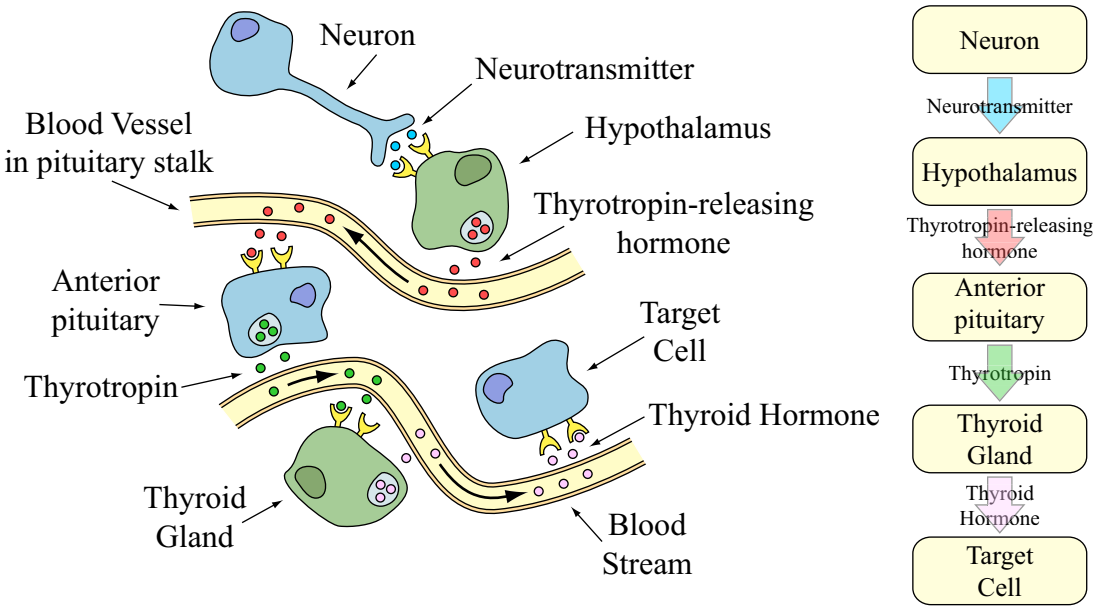


Figure 3.20: *The signalling pathway between a source cell and the required target cell which is capable of performing the desired response can be very complex.*

3.5 Summary

Biological cells are highly versatile units. Their abilities stretch far beyond those outlined within this chapter. Central to the operation of all animal cells is DNA. This information database holds, in its entirety and in every cell, the details required for a cell to develop and operate. Stem cells are a generic precursor to all cells, and are able to utilise the DNA blueprint to specialise into any cell type required by an organism.

Cells do not live an isolated life in multicellular organism. They must communicate to perform the functions required in order to keep that organism alive. Animals make use of a number of different communication techniques, each suited to a particular situation. Of particular interest is the long range, broadcasting communication made possible by endocrine signalling, and the more localised communication performed by paracrine signalling.

The complex nature of biological organisms requires a robust control mechanism to maintain a stable operating environment. The endocrine system is the major participant in this role. Using different cell communication techniques, it is able to control the operation of whole organs of cells, stimulating or inhibiting their activities.

The following chapter provides an analysis of these biological features, highlighting the characteristics that provide organisms with fault handling abilities, and how they may inspire an artificial fault tolerant electronic architecture.

Chapter 4

Analysis and Development

The fault handling characteristics of biological organisms have inspired a number of reliability engineered electronic systems such as those described in Chapter 1. This chapter describes the development of a new direction in this field, a fault tolerant electronic architecture whose main source of inspiration is the human endocrine system.

The full system development process, from conception to realisation, was undertaken in four stages. The first three stages, presented within this chapter, are concerned with the finding of a design concept, the engineering of that concept into a form more suitable to realisation and the completion of an initial design confidence test. The last stage, presented in Chapter 5, involves the hardware implementation of the proposed system. The full development process is shown Figure 4.1.

Figure 4.1: *The system development process.*

The first three development stages are documented as follows. Section 4.1 contains an analysis of the two biological features which form the foundation of this research: cellular redundancy and the endocrine system. This includes how these complex systems have been manipulated into simpler forms, more appropriate to electronics, but that still exhibit the required fault tolerant characteristics. The result is a conceptual design for an architecture that performs arbitrary computation on a data stream.

The second stage, documented in Section 4.2, describes the development of this design concept into a form more conducive to realisation as an electronic architecture. This includes the high-level design choices and operational differences from the real biology required to allow the feasible creation of the design in electronic circuitry.

Section 4.3 describes a software simulation written specifically to test the feasibility of the proposed architecture. This includes initial results that illustrate the computationally correct operation and fault tolerant nature of the system.

4.1 Biological Analysis

Redundancy is key to all fault tolerant architectures, a fact that was established in Chapter 3. This equally applies to biological systems where the incorporation of redundancy provides organisms with the ability to endure injury. For this reason, this analysis of biology in the context of its fault handling capabilities starts with a look into the organisation of biological redundancy.

In more complex organisms, biological redundancy is multi-tiered; it exists and operates at a number of separate structural levels as illustrated in Figure 4.2. The highest level is organ redundancy. Certain organ types operate in multiples, in a static redundancy form, simultaneously providing their particular bodily function. The benefit being, in the event of a single organ failure those that remain will continue in their role, providing the same service, albeit with potentially reduced efficiency. The duplication of lungs and kidneys in humans provides a good examples of organ redundancy. It is, however, not a universal trait, with some vital organs such as the heart or brain only existing in single quantities.

At a much lower level, redundancy is also found in DNA. This so called *genetic redundancy* involves the replication of information within the DNA sequence. In some organisms multiple genes that are functionally interchangeable are encoded within the same DNA strand. The consequence being an incorrect coding of one gene will not necessarily result in a completely incorrect expression of that gene function [Klu97].

However, it is the redundancy found at the cellular level that is of most interest to this research, in particular its organisation and the ways in which it is utilised. Approximations for the number of cells in the human body vary, but is thought to be in the region of 50×10^{12} to 100×10^{12} [Bla01; Mar00]. It is these massive cell quantities

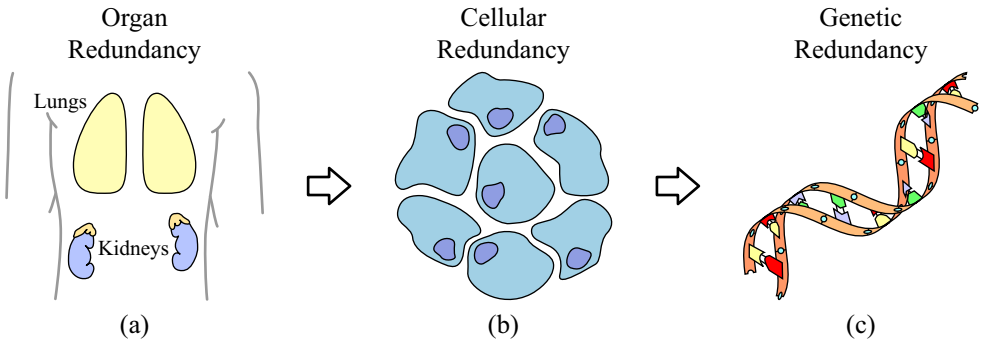


Figure 4.2: *Biological redundancy is multi-tiered; it exists and operates at separate structural levels. (a) the highest level, organ redundancy, (b) the massive redundancy of the cellular level and (c) information redundancy found in DNA.*

that are key to the fault handling characteristics of higher animals, and explains why humans can survive without replication of important organs.

What makes the cellular architecture so powerful, in terms of fault tolerance, is the cooperation of many cells as a group in the fulfilment of a single function. The loss or incorrect operation of a single cell within a group, or organ, is simply masked by the healthy operation of its many neighbours. Furthermore, the cellular structure is organised in a hybrid redundancy form. The immune system provides a detection mechanism that identifies and kills faulty cells [Kub97]. The process of regrowth then replenishes the loss with new cells. These detection and recovery mechanisms help to explain how such high cell quantities can continually enforce tolerance rather than eventually become a reliability problem.

4.1.1 A Simplified Cellular Architecture

The cellular structure of higher animals involves the complex arrangement of many cells, of many different cell types. Despite its complexity it still offers an enticing source of inspiration for an electronic architecture due to its inherent fault tolerant characteristics. In fact, the complexity is more due to the provision of organism functions than fault handling abilities. Therefore, it is possible to hone the overall architecture down to a much simpler form whilst still maintaining the key property of fault tolerance.

From a simplified viewpoint, the biological cellular structure is nothing more than a three dimensional lattice of separate cells. The individual functions required by the organism as a whole are provided by sub-sections of that lattice, these forming the organs. An

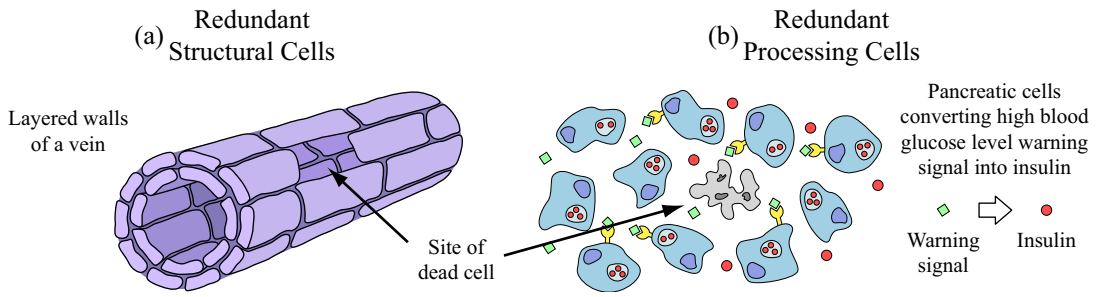


Figure 4.3: *Depending on the function of cells, redundancy is utilised in different ways.*

organ is simply a localised combination of cells, all specialised for the same task and working in coordinated operation. The fault tolerance of an organ is achieved by the underlying static redundancy of the organ's cellular structure.

The method by which cellular redundancy maintains organ operation in the event of cell failure is largely dependant on the function of the organ cells. The passive structural cells, such as those that form skin tissue or arteries and veins, provide a barrier or constructive function holding matter either in, out, together or apart. In these cases redundancy means limited cellular failure does not immediately result in a barrier breach or structural collapse, as depicted in Figure 4.3a.

In contrast, the fault tolerant property of an organ containing more active cells is attributed to the combined and concurrent processing efforts of redundant cells, as oppose to structural redundancy. The cells shown in Figure 4.3b represent those in the pancreas that produce insulin in response to a signal warning of a high blood glucose level. If a minority of these pancreatic cells fail, either not producing, or continually producing insulin irrespective of the incoming signal, the majority of cells will still process the signal correctly, providing the correct regulatory response.

It may be concluded that, in the creation of a simplified cellular architecture, the most important feature to maintain, in order to provide tolerance to cell failure, is the grouping of cells into single functional units or organs. It may also be assumed that, as the redundancy scheme utilised by active cells is far more appropriate to a data processing electronic system, cell placement is of secondary importance to cell cooperation. However, it will be shown later that the positional grouping of cells is also advantageous. Taking these points into consideration results in an initial basic architecture such as that shown in Figure 4.4.

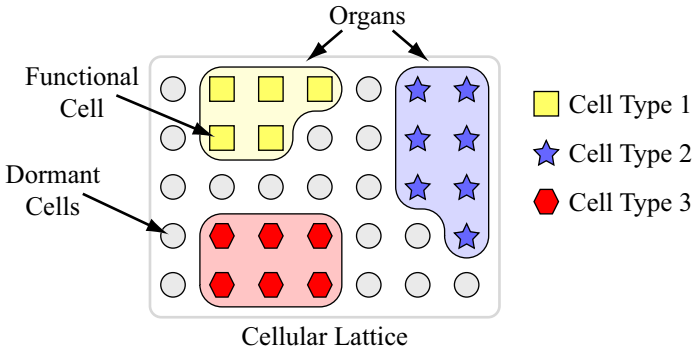


Figure 4.4: *The complexity of the biological cellular architecture can be reduced to a much simpler form more conducive to electronic realisation. An organism is represented by a single tissue that contains groups of like cells that represent organs.*

This simplified view paves the way for a realisable artificial cellular platform. However, in order for that platform to perform a useful task, some form of overlaying system is required that provides coordination between like cells, and links the operational functions of organs. Within real organisms, a biological mechanism that performs this role is the endocrine system.

4.1.2 Incorporating the Endocrine System

The endocrine system provides the inspiration for converting the inert cellular platform into a functional system. This is achieved by an implementation of the endocrine control scenarios described in Section 3.4.3, where the use of endocrine signalling allows organs to affect one another's activity.

Figure 4.5a shows an extended version of one of the control scenarios introduced in Chapter 3. The first organ releases its hormone messengers that go on to activate the second organ, which in turn releases the dual purpose hormones that activate (stimulate) the third organ and deactivate (inhibit) the first. This sequence of activation and deactivation creates a cascade effect with one organ after another becoming temporarily activated in a determinable sequence as shown in the plots of Figure 4.5b.

The conceptual leap that converts this process into a computationally useful system, is the assignment of a computational operation to each organ, see Figure 4.5c. As a result, the activation cascade becomes a sequence of separately activated operations. However, the operation cascade is purposeless without data to operate on. A mechanism is required that allows each organ to receive data on which to operate and then pass the result on to the next organ in the sequence. This is achieved by encoding data into the hormone

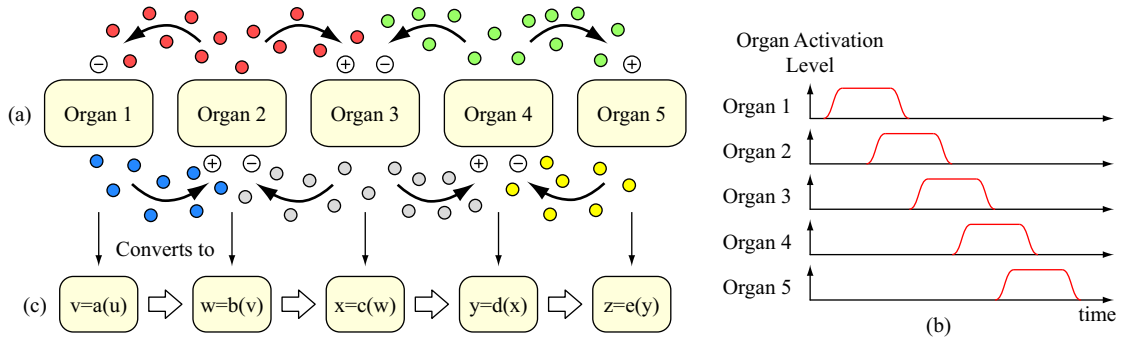


Figure 4.5: *Extending the endocrine control system results in a cascade of consecutively activated organs. If each organ represents a computational operation, a complete functional sequence can be performed.*

messages that control the activation of organs. In this way, when an organ releases stimulatory messages to the next organ in the cascade, it also passes on the required data.

4.1.2.1 Fault Tolerance in Endocrine Signalling

There are three aspects of endocrinologic signalling that make it an appropriate communication and control scheme for a fault tolerant system. The first is the lack of 'hardwired' communication links between source and target cells like those found in the nervous system. The failure of such a link results in complete communicative isolation from source to target and thus represents a single point weakness. In contrast, the signalling molecules that mediate messages of the endocrine system have the freedom to change their path, allowing the circumvention of problem areas within the cellular lattice. Furthermore, the use of multiple messenger molecules in a single communication means low level loss or corruption of molecules will not automatically result in the failure of the overall communication.

The second aspect is the feedback mechanisms employed during endocrine control. The more complex biological control scenarios use signalling molecules with a dual function. They not only stimulate the next target organ, but also inhibit the release of stimulatory signals from the previous organ. This form of feedback provides an acknowledge signal from target to source organs. In practise, the source organ will continue to release signalling molecules until it receives the acknowledgement that the target has been reached, making for a robust form of communication.

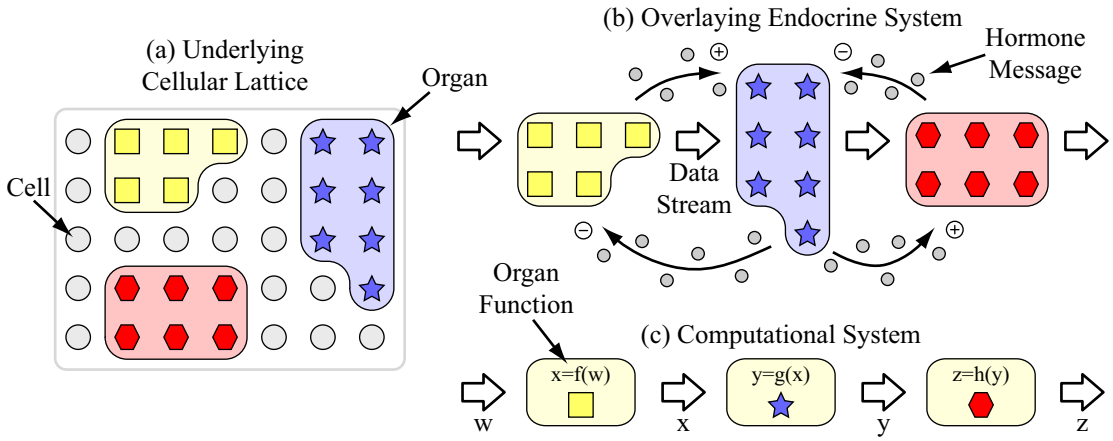


Figure 4.6: *A summary illustration of the proposed architecture. (a) An underlying cellular lattice divided into organs, (b) the overlaying endocrine system and (c) the computational sequence performed by organs, and controlled by endocrine signalling.*

The final fault tolerant aspect of the endocrine system is due to its utilisation of cellular redundancy. This occurs in the manner described in Section 4.1.1, where the coordinated and concurrent operation of active cells within a organ allows for cell failure to occur without loss of organ function. This last point is the most important as it represents the tolerance of actual physical faults that may occur in the underlying cellular architecture.

4.1.3 Analysis Summary

The result of this analysis section is a basic concept for a fault tolerant architecture that can perform arbitrary computation on a stream of data. In summary the organisation of this architecture is formed as follows:

1. The underlying physical structure of the architecture is a cell-based lattice, inspired by the cellular structure of biological organisms. Cells are grouped into organs in a static redundancy form, as shown in Figure 4.6a. The coordinated and concurrent operation of cells in organs provides tolerance to cellular failure.

2. Overlaying the cellular lattice is an artificial endocrine system that provides the overall system function, Figure 4.6b. Separate stages of the processing to be performed on the data stream are performed by individual organs that communicate data in an endocrine signalling fashion, Figure 4.6c. A feature that takes advantage of the redundancy contained in the underlying cellular architecture.

The basic operation of the system occurs as follows. Having established the system application, it must be divided into discrete computational sections that may be implemented

in separate organs. Data is directly fed into the cells of first organ which are immediately activated. Activation prompts these cells to operate on the received data and release the result, in the form of activating messages, into the electronic lattice. These messages travel until they find and activate a cell that is a member of the second organ in the computational sequence. These cells extract the data from the activating message and operate upon it. They in turn release the second set of messages that activate the third organ cells, but also deactivate the original cells, halting their message release. This cascade continues until the last organ in the sequence is activated from which the final data result is removed.

The greater detail of how this conceptual process can be realised are developed in the following section.

4.2 Developing a Computational Architecture

Section 4.1 established a design concept for a functional system. However, the design details still need to be developed in order that a realisable implementation can be achieved. The aim of this section is to present those details and describe more fully the operation of the proposed architecture.

4.2.1 An Electronic Messaging Space

One of the main difficulties when implementing biologically inspired techniques in electronic hardware is recreating the movement of organic matter. A lot of biological processes rely on the free movement of cells or proteins, processes that can prove difficult to replicate artificially due to the fixed structural nature of electronic circuitry.

In the case of the endocrine system, each mobile hormone messenger represents a static signalling state rather than a dynamic and functional unit such as a cell. This considerably simplifies the implementation of endocrine signalling, since the transport of hormones can be represented as the transmission of electronic data. However, the task of recreating the properties of hormone messenger propagation still needs to be addressed.

Two particularly important properties of hormone transport require replication. The first is to maintain the ability of hormones to diffuse around the cells of their target organ. The second is to avoid the use of single connections between source and target cells, instead allowing messages to take various routes. One way to replicate the required messaging

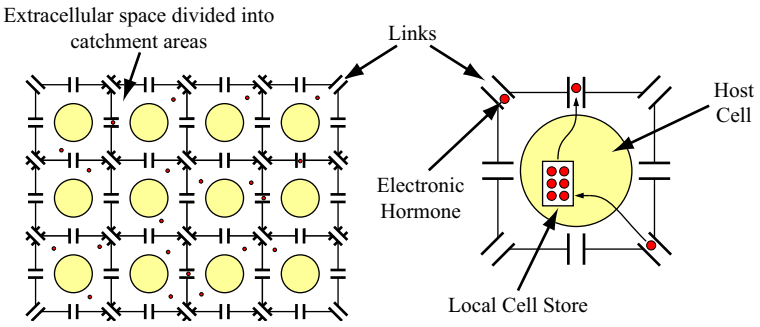


Figure 4.7: *The free propagation of hormone messengers is important to the correct function of the system. Dividing the signalling space into catchment areas, each governed by a cell, the property of free propagation can be artificially recreated.*

space is to divide the cellular lattice into catchment areas, centred around single cells. The hormone messengers within a particular catchment area are stored together locally, where they are able to present themselves to the host cell for binding. Hormones are free to move between neighbouring catchment areas via dedicated links between them. Such a scheme simulates the ability of hormones to coexist within the same space around a cell, and move freely around the lattice.

An implementation of this scheme is shown in Figure 4.7. In this case the cellular lattice is based upon a two dimensional square array. This design choice eases the implementation of the architecture in electronic circuitry. Each message catchment area has eight connections which allow hormones to enter and leave. These connections link the catchment area to its eight nearest neighbouring areas, providing hormones with a high level of freedom to migrate around the lattice.

4.2.1.1 Hormone Signatures

As hormones have the freedom to move around cells of all different types, a selection mechanism is required such that hormones only affect the correct cell type. This is described in Section 3.3.2 as a lock and key type process, where only the correct hormone key signature will fit the required cell receptor lock. Recreating this mechanism in the electronic system simply requires the inclusion of a digital signature in the hormone message that the target cell type can recognise. As a message enters a cell's catchment area, the cell can read the signature and determine if binding should occur.

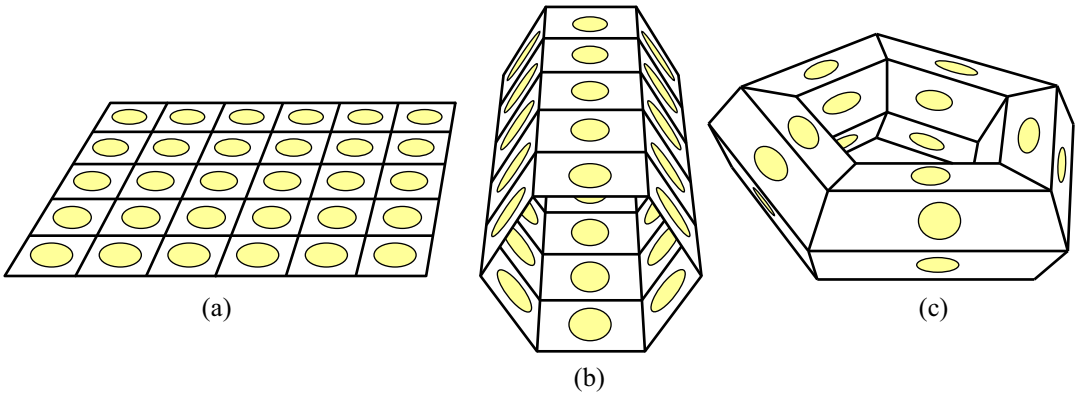


Figure 4.8: *A square array of cells (a) can be made boundless by joining two opposite boundaries into a cylinder (b) and then joining the two cylinder ends to create a toroid (c). Such a formation allows for an artificial circulatory system.*

4.2.1.2 Message Propagation

Hormones in biological organisms propagate around the body via the blood circulatory system. The flow of blood helps the spread of hormones, enabling them to reach their target cells. A similar mechanism is required to direct the movement of the electronic hormone messages around the cellular lattice. Producing such a flow can be achieved by letting cells control the movement of messages. When a message enters a cell's catchment area, if it is not of the correct type to bind to the host cell, it is simply passed onto the next catchment area.

To create a circulatory flow of messages, a simple biasing of transmission direction may be employed. When cells choose the direction in which to pass a message, a random choice is made from the neighbouring cells that are free to receive a message. By biasing the random choice more towards neighbours at one side, messages have a greater chance of flowing in their direction. If a universal biasing is used between all cells, an artificial circulatory message flow will result.

A circulatory system should generally have no boundaries in the direction of flow. For a simple square array this poses a problem, when hormone messages reach an edge where should they go? The solution is to remove the boundaries. As shown in Figure 4.8, in the case of a square based array this may be achieved by the conversion into a toroidal topology. The advantage of such a formation is that it lends itself very well to a circulatory flow of messages.

4.2.1.3 Hormone Ageing

The lifespan of biological hormones is finite. They either degrade, or are actively removed from the body. This process is important since it allows the endocrine system to maintain control over hormone concentrations. A similar measure is required in the electronic system to stop messages from continually circulating in the event that they repeatedly do not bind.

As there is to be no central point of coordination between cells, a single time reference is not available to age the messages. Therefore, the approach taken is to count the number of hops a message has taken between cell catchment areas, a value that can easily be encoded into the message data. When a host cell detects that the hop count has reached a preset value, the message is deemed to have expired, and is removed.

4.2.2 Dealing with Reduced Redundancy

As described in the earlier biological analysis, the level of redundancy at the cellular level is very high. The quantities of cells that make up even a single organ are far greater than is realistically reproducible in electronics with current technology. However, it is the high redundancy found in these structures that underpins the important characteristic of fault tolerance. Therefore, a method must be found for maintaining fault tolerance even with reduced cell quantities.

Low level cell death within organs is acceptable because of their healthy neighbours' continuing operation. Recreating this artificially requires the replication of cells. As long is there is a single healthy cell available per organ, operation will continue. However, this is highly dependant upon cells failing in a manner that does not interfere with the operation of healthy cells. In the event that a cell starts to release erroneous messages, receiving cells are unable to detect them as such and may react incorrectly.

Normally, in biological systems such a situation is exceptionally unlikely as the erroneous signals are overwhelmed by those released from the majority of correctly operating cells. However, with reduced cell numbers this is not the case. The solution is the employment of a NMR type scheme within the cell hormone binding process.

For a response to occur, a cell requires a number of hormones to bind, all requesting the same action. But, to counter the possibility that a single faulty cell releasing multiple

erroneous messages could trigger an incorrect reaction, cell activation requires the binding of hormones from different source cells. To allow cells to determine the original source of a message, the message data incorporates an identification field that represents the cell that created the message. The same majority vote binding approach is taken for both stimulatory and inhibitory responses.

4.2.3 Improving Data Flow

As the data stream flows through the system, it is important that each data item is operated upon in the correct order. If this does not occur, the data held by cells will become out of sequence and subsequent data operations will provide incorrect results.

What makes this a potential problem is that organs have no concept of the correct sequence of incoming data. In systems where consecutive processing stages are directly connected, ordering is maintained. However, in this system there are no direct connections between the organs. The nature of hormone propagation is nondeterministic since messages are not constrained to a set route from one organ to the next. This leads to variations in the time between hormone release and binding. It is therefore possible for waves of hormones, and the data they hold, to become disordered.

One solution to this problem is to allow the organ cascade to settle back to a non-activated state before more data are passed into the system. If a pause of a suitable length is left between the input of new data, organs will be activated in the correct sequence, data are operated upon in the correct order, and the data retrieved at the output are guaranteed to be in the correct order. However, this makes for very inefficient use of the system hardware as organs would spend most of their time in a quiescent state, not performing any useful operations.

An alternate approach that involves encoding a sample time stamp into hormones allows the data input rate to be increased to a level that matches the natural speed of the organ activation sequence. At this speed a pipelining effect occurs allowing organs to begin operating on the next sample as soon is it had finished with the previous.

To avoid any confusion with data ordering, cells utilise the sample time stamp of a received message to determine the position of the message data relative to the currently held data. Cells simply ignore any data that does not refer to the time stamp of the

currently activated operation. However, when the cell becomes deactivated it then allows hormones with a different sample time to bind, and activation specific to that time occurs.

The activation level of the first organ provides a guide to the data input rate. If samples are supplied when the organ reaches a non-activated state, the result is that the data throughput matches the natural propagation rate of the system. Such a scheme makes for much better utilisation of system hardware.

4.2.4 Improving Cell Coordination

Due to the reduction in the number of cells that make up an organ in the electronic system, there is greater importance for all organ cells to be activated during inter organ communication. If this does not occur, there may not be enough hormone messages from distinct sources to activate the next stage in the organ cascade.

The biological approach that has been adopted to help address this problem is an implementation of paracrine communication. To help improve the impact of a single message, and thus increase the chances of total organ activation, on the binding of a hormone message the receiving cell emits a paracrine message to its direct neighbours containing the exact same message data. The neighbouring cells recognise the message as a paracrine signal and do not pass it any further, maintaining the local nature of paracrine communication. Due to the positional grouping of like cell types, individual cells are likely to have neighbours of the same type. So any paracrine messages passed to a neighbour will have a good chance of enforcing the receiving cell's activation or inhibition process.

4.2.5 Dormant cells

A feature that makes the biological cellular level so robust is its ability to replenish cell loss. This ability converts an otherwise static redundancy architecture into a hybrid redundancy form. However, artificially growing new electronic circuitry to replace failed parts is not possible with current electronics technology. Instead, a spare parts scheme must be employed.

This is achieved within the artificial cellular architecture by the inclusion of dormant cells in the lattice. The static redundancy scheme already in place allows the seamless integration of these cells into an organ to replace those that have failed. The choice

to leave these cells as spares rather than using them immediately to boost an organ's static redundancy maximises their potential for replacing failed cells. Their dormant state means they can be integrated into whichever neighbouring organ needs them.

This 'one cell fits all' scheme is dependant on the ability of cells to specialise into any cell type found within the system. This requires that all cells have both the information and the ability to perform the operations of any cell type. This concept is also found in biology, where pluripotent stem cells are able into specialise to any cell type. Each artificial cell must contain the equivalent of DNA, from which it can extract the required operational information. The automation of cell replacement is yet to be fully implemented, but the required infrastructure is present. This provides scope for further work, see Section 7.3.3.

The spare cells are, in fact, not completely dormant. They must participate in the passing of messages, otherwise they would present major obstacles to message circulation.

4.2.6 System Inputs and Outputs

The task of supplying input data to the system has already been discussed. Passing data into the system involves the input of stimulatory messages directly to the cells of the first organ in the computational sequence. These messages contain the required input data and immediately place the receiving cells into an activated state. The rate of data input is timed by the activation cycles of the first organ's cells. Only when the input organ has finished operating upon the current data and has become inactive, is the data for the next sample time input.

Retrieving data from the system involves extracting output data from the cells of the system's last organ. This may be done actively by those cells, transmitting data via alternate communication means to an external system. Once extracted, the data from the multiple output cells will need combining into a single data stream. If this combining process is implemented with a voting stage then any erroneous data from output cells can be masked.

Retrieving data from the system involves combining the output data from the cells of the last organ in the system. This requires a separate external system to sort the data into a single output stream. If this combining process is implemented with a voting stage then any erroneous data from output cells can be masked.

The cells of the last system organ are not subjected to any specific inhibitory hormones from other organs. Without any such signals there is no way for the output organ to become deactivated and move on from the very first set of data. The solution is to provide the organ cells with a naturally decaying activation level. Rather than deactivation occurring under hormone control, the cells automatically return to a deactivated state, from which they can then be stimulated to operate on the next set of data.

Utilising the same automated activation decay process in all system cells provides a useful system recovery process. If a cell fails to receive enough inhibitory messages before their source stops releasing them, the cell would remain activated and would suffer the same fate as the output cells. Therefore, the same decay process can be used to ensure that cells always eventually become available for reactivation and processing new data.

4.2.7 Cell Operation Assignment

The assignment of operations to cells takes the form of assigning cells to organs, and the placement of like cells into spatially localised cell groups. In biology this occurs as part of the development process. During embryo growth cell specialisation is determined by position and environmental factors. A similar artificial process for automated cell operation assignment in the cellular lattice would be an interesting direction for further work, and could perhaps follow the work of [Liu04] or [Mil04]. However, currently the assignment of cell operations must be performed manually.

4.2.8 Development Summary

Converting a biological system into a form that may be implemented in electronic circuitry, whilst still maintaining the original system's operational characteristics, requires a number of careful compromises and implementation alterations. This section has documented the changes required to convert a conceptual endocrinology-based architecture into a realisable form. The result is a high-level description of system operation, engineered for electronic implementation.

The final stage of the development process takes the high-level instructions and produces a hardware system. This process, along with implementation-specific details are presented in Chapter 5. However, before this last developmental step was made and designs were committed to hardware, the proposed system was tested to provide confidence in

its correct operation. This penultimate stage of the development process is described in the following section.

4.3 Initial Testing

In order to test the feasibility of the proposed architecture as a functional and fault tolerant system, a software model was written that simulated its operation [Gre03]. When the model was written, the architecture was in early development and therefore had a number of operational differences from those described in Section 4.2. These differences are as follows:

- Cells use a random selection mechanism when choosing the direction in which to pass messages. Therefore, there is no circulatory message flow.
- Cell activation is determined on a threshold basis. The level of cell activation increases as more stimulatory messages are received, and decreased with the reception of inhibitory messages. Above a certain activation threshold, the cell is deemed operational. Furthermore, message sources are not discriminated.
- Messages do not include a sample time stamp. Therefore correct data ordering is maintained by allowing the organ cascade to settle between message inputs.
- Cells are not placed in positional groups and paracrine signalling is not employed to reinforce message impact.

Despite the differences between the system modelled by the simulator and the final proposed system, the simulation results presented in this section provide a good demonstration of the architecture's performance.

4.3.1 The Test System

The simulated system was configured to perform a very simple computational function, the equation of which is given in Equation 4.1. The equation was chosen for ease of determining correct system operation rather than computational capability. The system used to implement this equation contains two organs, one performs the subtraction of two, the other the multiplication by three. This configuration is shown in Figure 4.9.

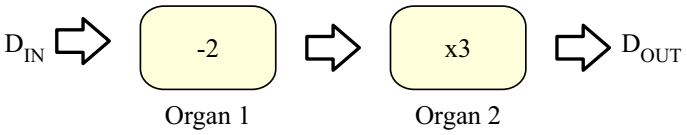
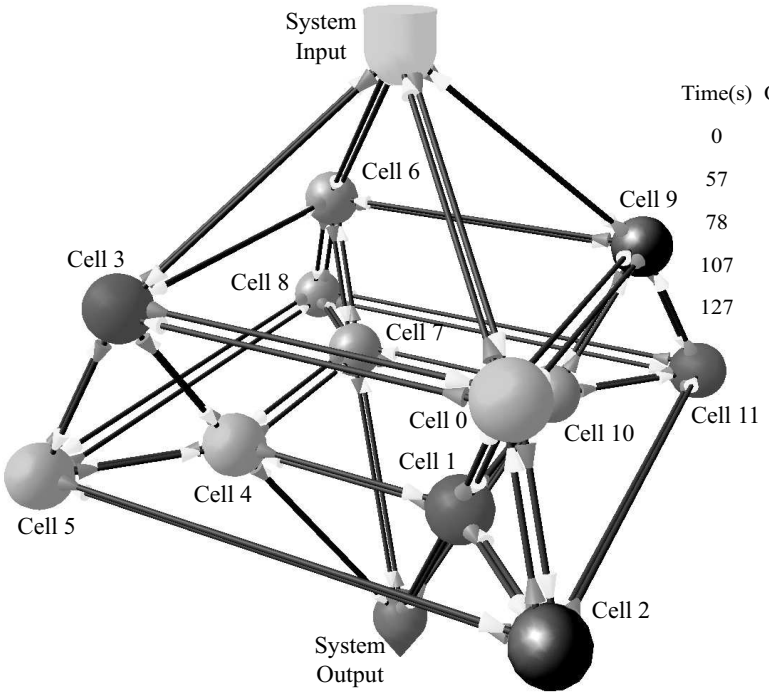


Figure 4.9: *The simple computational sequence simulated by the system software model.*



Changes in cell function during experiment

Time(s)	Organ 1	Organ 2	Dormant	Killed
0	9, 4, 5	2, 3, 11	0, 1, 6, 7, 8, 10	
57	4, 5	2, 3, 11	0, 1, 6, 7, 8, 10	9
78	0, 4, 5	2, 3, 11	1, 6, 7, 8, 10	9
107	0, 4, 5	3, 11	1, 6, 7, 8, 10	2, 9
127	0, 4, 5	1, 3, 11	6, 7, 8, 10	2, 9

Figure 4.10: *The cellular network used in the simulation of the test system. The table shows the state and organ assignment of each cell as the experiment is conducted.*

$$D_{OUT} = (D_{IN} - 2) * 3 \tag{4.1}$$

The cellular network used in the simulation is shown in Figure 4.10. This network is based on a boundless square array, but there are no diagonal connections between cells.

4.3.2 Simulation Operation

The input and output streams of the system are shown in Figure 4.11. The input numbers are integer values, spaced in time to allow the system to settle and maintain the integrity of output order. Inspection of the output values show that they are true for Equation 4.1. The replication of output values per input sample is due to the release of multiple output hormones from the cells of the second organ.

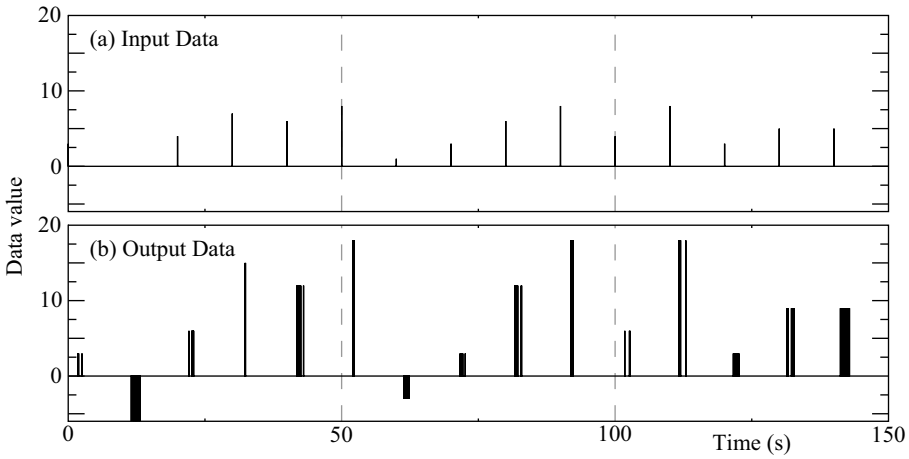


Figure 4.11: *Plots of the system input and output streams. The output data is the correct result for the implemented system function. The replication of output values at each sample instance is due to the release of multiple hormones from the final stage cells.*

The data presented in the plots of Figure 4.11 provide assurance of the systems correct computational operation. However, and importantly, these results were achieved under the simulated failure of two cells. Figure 4.12 shows the activation levels of the cells affected by the cell failure process. Each of the two separate plots represents the killing of a cell and the activation of a dormant cell to take over its task. For example, as shown in Figure 4.12a, Cell 9 is killed at time equals $57s$. However, because of the two remaining cells in the organ, the system continues to operate correctly, as shown in 4.11. At time equals $78s$, Cell 0 is made operational from dormancy and assigned to the currently deficient organ. The participation of the new cell in organ function is immediate, restoring the organ to its full complement of cells. The same process, with the same resulting display of fault tolerance, is shown in Figure 4.12b, however, Cell 2 is killed and Cell 1 is made operational.

4.4 Summary

An analysis of biological organism robustness led to the cellular architecture being highlighted as a major contributor to biological fault tolerance. The endocrine system was then chosen as a method of utilising this underlying architecture to create a computationally useful system.

The following stage involved the conversion of the developed conceptual design into a form suitable for electronic implementation, whilst maintaining the key fault tolerant

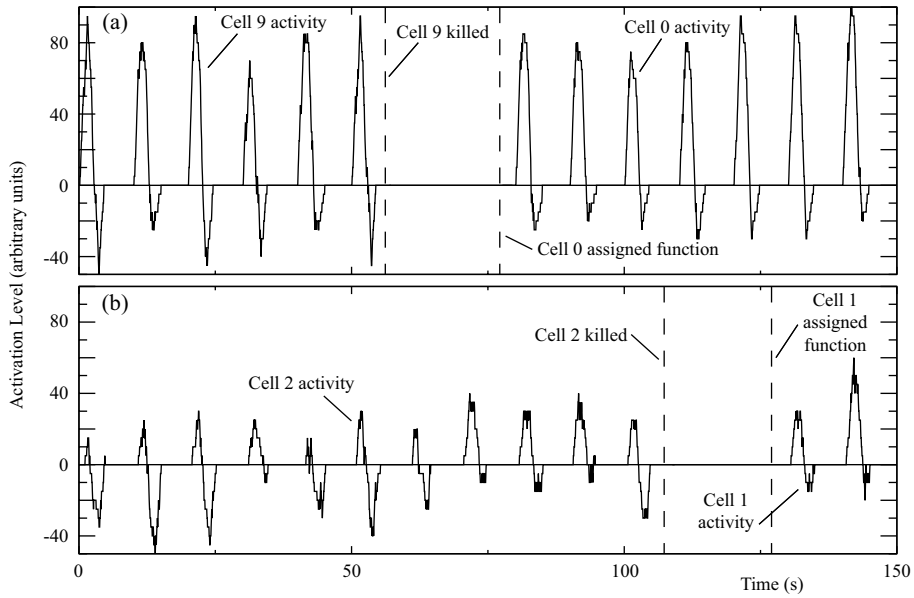
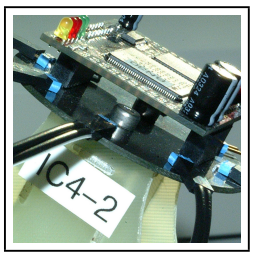


Figure 4.12: *When cells are failed and new cells are introduced into the organ, the new cells can be seen to become activated, contributing to the operation of the system.*

properties. A simulation was used to gain a level of confidence in the correct operation of the system, testing both for computational correctness and fault tolerance.

The next stage in the development process is the creation of an actual hardware implementation of the proposed system, this being the topic of the next chapter.



Chapter 5

The BioNode System

Hardware simulation tools are useful for acquiring confidence that a system design will operate correctly. However, they are not a substitute for testing with a real physical implementation. For this reason a custom, hardware platform was built that allows the proposed architecture to be implemented and its computational and fault tolerant abilities tested. This chapter documents the construction and operation details of this platform, called the BioNode System.

Section 5.1 introduces the bioNode system. The initial system design aims are presented, followed by an explanation of the bioNode module's and network's construction. Section 5.2 explains how the bioNode system can be used to implement the proposed endocrinology inspired system. The bioNode system also includes a set of supporting systems. These are documented in Section 5.3.

5.1 System Introduction

At the initial design stages of the bioNode system, a number of implementation aims were established to maximise the potential of the resulting hardware. They were:

- To create a highly visual system whose structural form represents the nature of the underlying cellular system, and in this way, provide a demonstrator that enables an understanding of the system's operation to be easily achieved.
- To incorporate a high level of reconfigurability such that the system may used to implement as many array based experiments as possible.

- To provide the means for faults to be injected into the system, both programmatically under software control, and physically by system users.

The electronic hardware that fulfils these aims, but more importantly supports the implementation of the endocrinology inspired system is now introduced.

5.1.1 The BioNode Module

The bioNode module was designed as a flexible self-contained unit that can be used to implement a number of electronic systems. Rather than tailoring the bioNode's architecture specifically to the role of implementing the endocrine based system, its functional abilities are far more general in purpose. To achieve this, as much reconfigurability was included as possible.

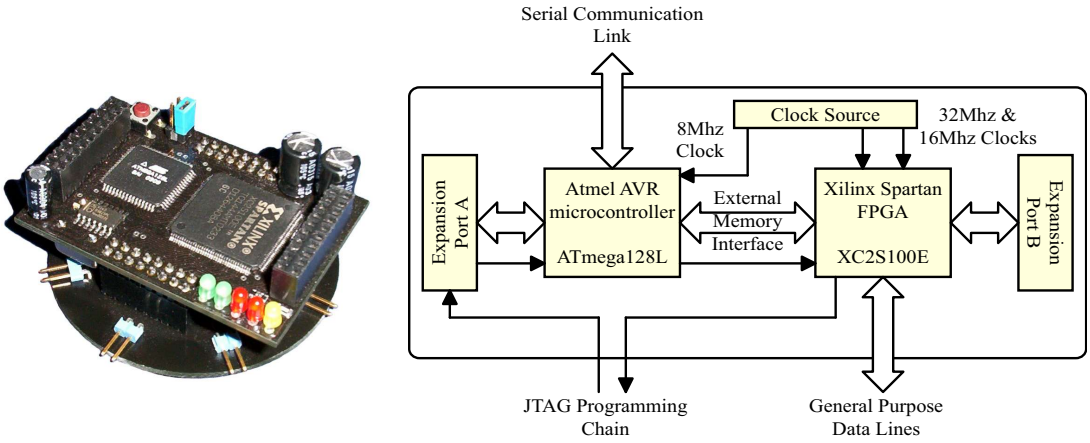
The core functional elements of the bioNode are a microcontroller and a *FPGA*. These provide scope for reconfiguration of both software and hardware. The microcontroller is an Atmel® AVR ATMega128L [Atm04], an 8-bit *RISC* device that contains a substantial 128Kbyte of program space. The FPGA is a Xilinx® Spartan™-IIE XC2S100E [Xil03], containing 100×10^3 programmable gates. Brief specification summaries of these devices may be found in Tables C.1 and C.2 respectively.

A picture of a bioNode and a diagram of its electronic structure are shown in Figure 5.1. The module has two main headers that connect it to a hosting system. These headers provide power and programming connections, a full duplex serial communications line to the microcontroller and sixteen general purpose data lines connected to the FPGA. There are also two expansion ports that provide extra connections to the microcontroller and FPGA. These can be used to add further functionality to the bioNode.

The AVR microcontroller supports a special external memory interface such that peripheral devices may be directly mapped into the microcontroller's memory space. The FPGA is connected to this interface such that configured hardware can be easily accessed using the mircocontroller's standard memory access instructions. A high-level schematic of the bioNode illustrating this arrangement is shown in Circuit B.5.

5.1.2 The BioNode Network

The bioNode network contains thirty separate bioNodes positioned into a five by six square based grid. The network is formed by connecting each bioNode to its eight nearest



(a) (b)

Figure 5.1: *A bioNode module (a) and a diagrammatic representation of its main functional parts (b). Also visible in (a) is the network interconnection board that provides extra system flexibility, see Section 5.1.2.*

neighbours. These connections are made using the sixteen FPGA data lines available on each module.

The network is made boundless by creating a toroidal topology. The three dimensional structure this forms is fully recreated in the final hardware structure. The toroid is created with five vertically aligned hexagonal platers arranged in a circle. Each plater holds six nodes, producing the total of thirty. The resulting structure is shown in Figure 5.2.

BioNodes are connected into the network using interconnection boards. One of these is visible in Figure 5.1a. These boards carry the actual connections of the system network. This scheme has two main purposes, it allows bioNodes to be removed from the network without disrupting the many network connections. It also allows the network to be partially populated for running smaller experiments. To aid this scenario, the connecting wires between the interconnect boards are removable, so that network subsections can be fully isolated. A further use of both these system hardware features is that faults can be manually created in the system by removing nodes, or disconnecting inter-bioNode wires.

The bioNode modules and the network in which they are held provide the perfect platform upon which the endocrinology inspired system may be implemented. The following section describes how this implementation is achieved.

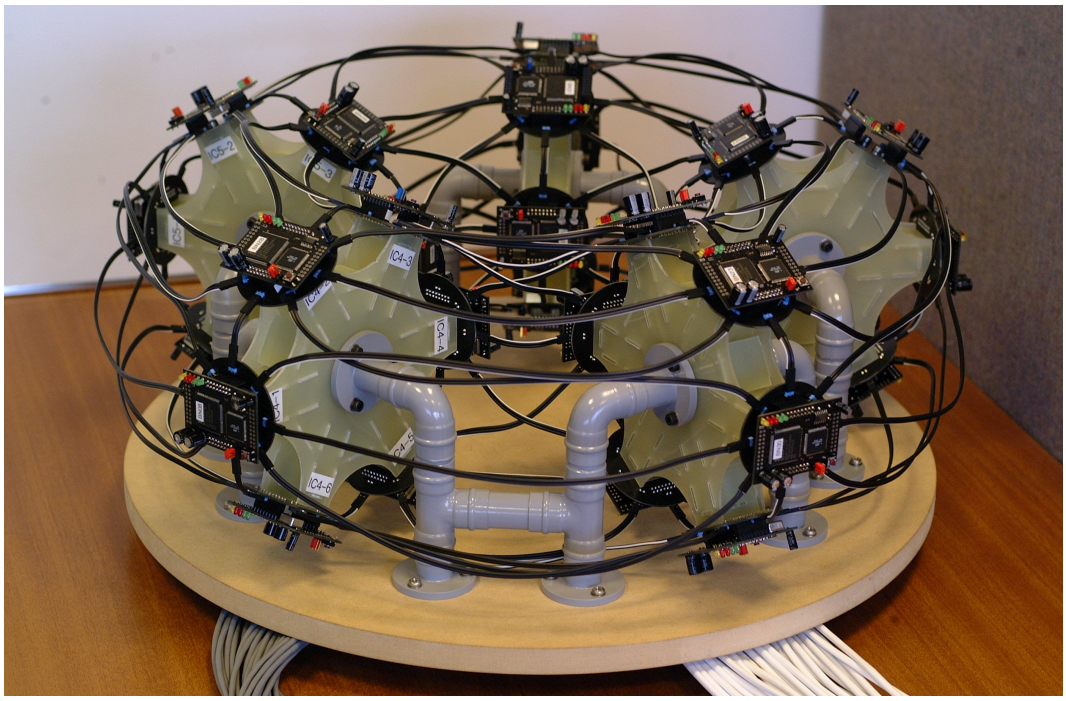


Figure 5.2: *The bioNode system is created from thirty bioNode modules connected in a boundless network. Each node has a direct connection to its eight nearest neighbours.*

5.2 Implementing the Endocrinologic System

The bioNode network is used to represent the cellular architecture that underpins the endocrinology inspired system. Each individual bioNode performs two vital architectural roles, the first is to emulate the function of the cell, this is achieved in software on the bioNode's microcontroller. The second is to control the cell's messaging system and catchment area, both of which are performed in hardware within the bioNode's FPGA. The first of these two roles is described in Section 5.2.1, the second in Section 5.2.2.

5.2.1 Cell Emulation Software

Emulating the behaviour of the cells used within the proposed system requires the implementation of two main functions. The first is to perform the designated operation of the organ of which the cell is a member. The second is to determine which hormones, the electronic equivalent being messages, should bind and how they affect the cells activation. To describe how these closely linked functions are achieved the following sections contain a set of flow charts that represent the program flow of the cell emulation software.

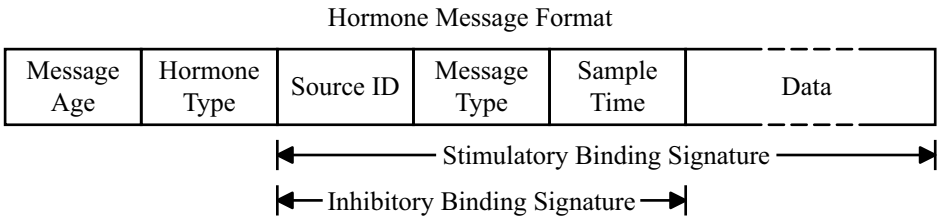


Figure 5.3: *The format of a hormone message. The sections used for stimulatory and inhibitory signatures is shown.*

5.2.1.1 Initial Binding

Each message contains a binding signature that is set by the cell that initially created the message. Figure 5.3 shows the format of a hormone message, and the sections used for binding.

The message age field, as explained in section 4.2.1.3, is used to establish a finite lifetime for a message. The hormone type field is used to identify messages as either a local paracrine type message, or a longer range endocrine message. The message source ID holds the number of the bioNode that originally created the message, this value being unique to each bioNode. The sample time field represents the position of a message's data within the system data stream.

The message type field distinguishes different sets of messages from each other. It is analogous to identifying hormones as either insulin or glucagon, for example. The initial stages of the binding process undertaken by a cell involves determining if the message type of a received message is relevant to that cell. All the cells of a specific organ have a particular stimulatory and inhibitory message type associated with them, using these values, a cell can determine what response, if any, a received message should invoke. It is important to note that messages themselves are neither inhibitory nor stimulatory, but are determined as such by the cell type they bind to. This distinction allows the same message type to invoke both stimulatory and inhibitory reactions

The initial binding process occurs as follows. If a received message is deemed to be stimulatory and the cell is currently active, the message is not required and is passed on to a neighbouring cell. If however, the cell is currently inactive the message should be considered as part of a stimulatory response. The message data is therefore passed to a handling function that coordinates cell stimulation.

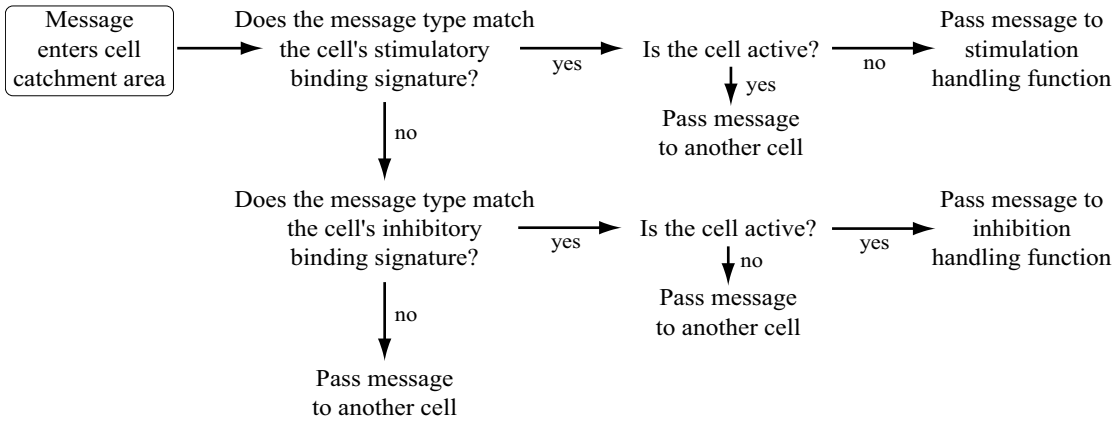


Figure 5.4: *This flow chart represents the initial binding procedure carried out by cells to determine if a message is of the correct type to bind.*

Processing the reception of inhibitory messages occurs in a similar way. If the cell is currently inactive then inhibitory messages are not required and are passed on to another cell. However, if the cell is active, the message may be useful, and is passed to an inhibition handling function. This complete filtering process is depicted in the flowchart shown in Figure 5.4.

The result of this initial binding stage is to filter out all the messages not destined to the cell, passing them on to neighbouring cells. Those that remain have been identified as relevant stimulatory and inhibitory messages, and are passed on to handler functions. These functions are described in the next two sections.

5.2.1.2 Stimulation Handler

Stimulatory messages passed to the stimulation handling function have arrived whilst the cell is inactive, and therefore can be involved in cell activation. The main purpose of this function is to monitor these incoming stimulatory requests, and when a number of matching requests from different source cells have arrived, activate the cell to operate on the data contained within the messages. The operation of this handling function is shown in Figure 5.5.

At the heart of this process is a table used to record the incoming messages. Each line, or entry, in the table represents all the received requests to perform a particular data operation. A request being the combination of message's data and the data sample time. Stored in each entry are the source IDs of the cells that have so far made the same request, and a count of the number of sources held.

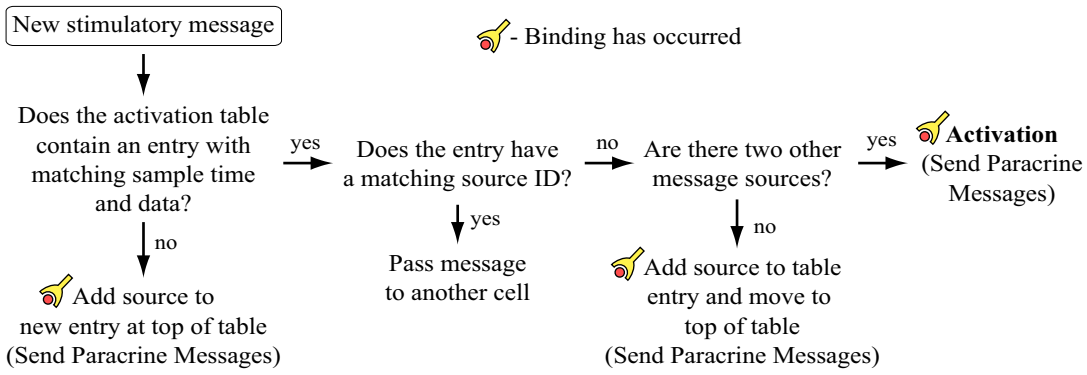


Figure 5.5: *The stimulation handling function monitors incoming stimulatory messages, and determines when the cell should be activated. The cell holds a table of bound messages, which is used to find a set of matching messages form diverse sources.*

When a request arrives, if the table does not already hold a matching request a new entry containing the message's data and source ID is added to the top of the table. If a matching request is found, the ID of the cell that sent the message is compared with those already held in the entry. If a source ID match is not found the new ID is added to the entry, which is then moved to the top of the table. If however, a request from a cell has already been logged, the requesting message must not participate in further activation and is passed on to a neighbouring cell. As more messages are received, the older requests are removed from the end of the table, leaving the new, and therefore currently more active requests, towards the top.

When a table entry is found to contain three requests from different source cells, then the cell is activated to perform its designated operation upon the data held in that table entry.

5.2.1.3 Inhibition Handler

The inhibition handling function operates in a similar way to the stimulation handler, but in a simpler form. Once a cell has become activated it has specialised its operation to a particular data sample time. Therefore, inhibitory messages need only be considered if they represent a signal from the same sample time.

Rather than using a whole table to record incoming inhibitory messages, a simple list of the message source IDs is used. When an inhibitory message is received with the correct sample time, the message source ID is compared with those in the list. If it has not already been logged the ID is added to the list. When three unique message sources have been acquired the cell is fully deactivated. This inhibition handling function is depicted in

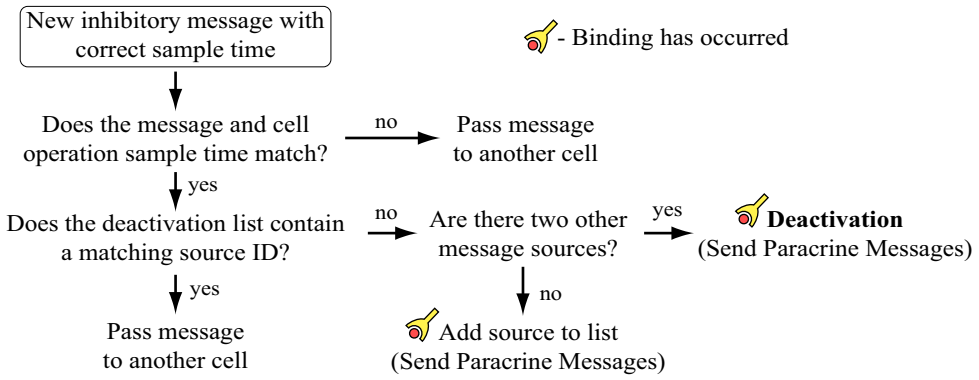


Figure 5.6: *The inhibition handling function monitors incoming inhibitory messages with a data sample time that corresponds to the current cell's operation. Cell deactivation occurs when three inhibitory messages from unique sources have been received.*

Figure 5.6. All other received messages, those with different data sample times, and those originating from source cells already logged, are passed on to other cells.

5.2.1.4 Activation Loop

When a cell becomes activated it performs the operation determined by the organ of which it is a member. The result must then be released within new messages back into the cellular network. The release of new messages is controlled by the activation loop shown in Figure 5.7. On activation, the loop is started and a new message is released every cycle. The rate of message release is constrained by a delay, to stop the network becoming over saturated.

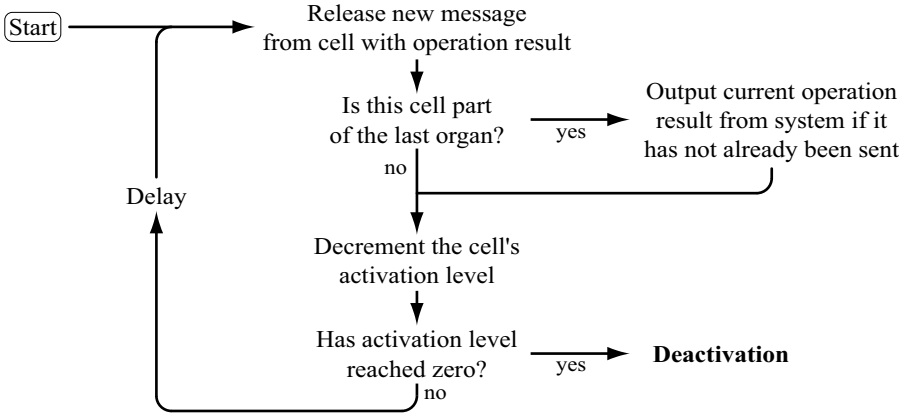


Figure 5.7: *When activated, the cell releases messages into the network. The release is controlled by a timed loop, an overview of which is shown here. An additional part of the loop is the cells auto deactivation control.*

Another function of the activation loop is the implementation of the cells automated activation decay process. When a cell is activated it is set with a predetermined level of activation. Every time the activation loop cycles, the activation level is decremented. When the level reaches zero, the cell is fully deactivated and ready to start operation on the next sample time of data.

5.2.1.5 Cell Data Operation

To enable a cell to perform the function of any cell type within the system, it must contain the information and capabilities to perform every operation that the system requires. Implementing cellular behaviour within software makes this task considerably easier. When a cell is activated to perform a data operation is simply needs to call one of a set of sub-routines, the selection being based on the cells type.

5.2.2 Cell Communication

The method used to artificially recreate the hormone message signalling space requires a considerable number of interconnects between cells. The use of a complex wiring scheme for each inter-bioNode link would result in a unwieldy network of prohibitive implementation cost. Therefore the links have been kept very simple, and are implemented with a two wire connection, making one wire available per direction of transmission.

The utilising of such a simple wiring scheme does not directly allow for data synchronisation or flow control. Therefore, some interesting techniques have been employed to enable these important features. Furthermore, due to the hot-plugging of bioNodes within the network, extra steps must be taken to maintain the integrity of node communication during plugging.

Data synchronisation is achieved by the standard approach of dividing the data stream into packets that include a start pulse with which receivers can synchronise [Rum93, Page 7]. This is necessary as each bioNode uses its own clock source, and there are not enough data lines for a dedicated inter-bioNode clock signal.

Extra steps must be taken to ensure that receivers can clearly distinguish the start pulse from other packet data. This is due to the chance that network wiring may be removed, causing the reception of half packets and the loss of receiver synchronisation. The start pulse is made distinct by utilising *Manchester coding* [Skl88, Page 80] on the packet data,

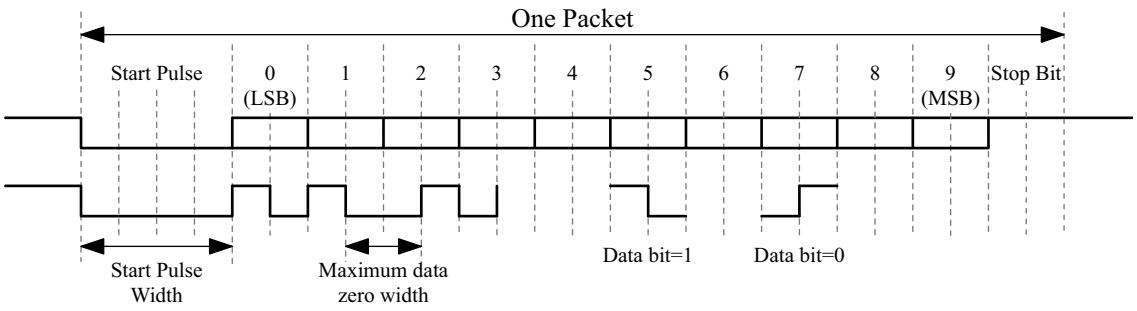


Figure 5.8: *Manchester coding is used to encode the data held in inter-bioNode communication packets, this allows receivers to easily detect start bits even when packet runts occur.*

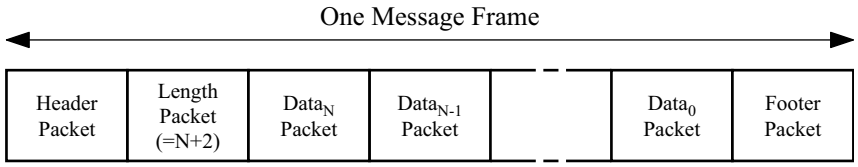


Figure 5.9: *The frame format enables receiving bioNodes to correctly determine whole packets from half frames created by faults.*

and increasing the length of the start pulse to two bit widths. The resulting packet format is shown in Figure 5.8 where it can be seen that the longest 'zero' time for data is half that of the now distinct start pulse.

The amount of data required to encode a hormone message exceeds the capacity of a single packet. Therefore, messages are transmitted in multiple packets, creating message frames. Frames also need special attention so that receivers can distinguish whole frames from runt frames. The removal of a bioNode, or the disconnection of an interconnect, can lead to the reception of half frames, these being either front or back halves.

The frame format that is used is shown in Figure 5.9. The header and footer packets hold a matching frame identification tag that is incremented with each frame transmission. Receiving bioNodes use this tag to determine the integrity of a frame. If the tag of a footer does not match that of the pending header, the frame is considered to be corrupt and ignored. The length packet is a further measure for detecting frame corruption. Receivers can check the number of data packets received between a header and footer matches this value.

The packet and frame formats take care of message synchronisation and the handling of message corruption. However, provisions for flow control are still required. To a certain extent this can be achieved with packet buffering by the receiving bioNode, but without feedback to the transmitting bioNode, buffer overrun and consequently message

corruption can occur. But how can feedback be achieved with the minimal wiring scheme? The solution to this problem is inspired by the function of a feature found in a number of the Ethernet protocols [Ste94].

The 10Base-T version of the Ethernet protocol [Spu00] uses the transmission of a link pulse, repeated at a predefined rate, to allow devices to inform each other of their existence. If link pulse fails to arrive at a devices receiver then it is assumed that no device is connected to the opposite end of the link.

The same scheme is utilised by the inter-bioNode communication system. It allows bioNodes to determine which communication links are viable choices for messages transmission. However, the actual implementation involves a modification that allows bioNodes to transfer flow control information. Rather than just transmitting a single pulse, bioNodes transmit a whole packet that contains status information. Included is the fill level of the transmitting bioNode's receiver buffer. The bioNode at the opposite end of the link can then decide whether there is enough buffer space at the destination to accommodate a new message. The status packet is transparently multiplexed between message data packets, allowing a regular status packet transmission rate to be maintained.

To avoid the problem that transmission of a frame may begin before the transmitting bioNode has received warning of a full buffer, the receiver buffer is considered full when all but one frame space is filled. As the minimum frame transmission time is greater than the status packet repeat time, this scheme avoids buffer overrun.

5.2.3 Fault Injection

To test the fault tolerant abilities of the developed endocrinology inspired architecture, the bioNode system incorporates two distinct methods of introducing faults. The first, already introduced, is manually, via the removal of actual bioNodes or the interconnecting wires that join them. The second is under the control of software. Using a monitor communication link, introduced in Section 5.3.1, it is possible to send commands to specific bioNodes, causing them to introduce operation faults.

The faults that can be programmatically introduced into a bioNode are as follows:

- Complete shut down. The bioNode does not release any new messages, or respond to received messages.

- Garbage Messages. The bioNode is forced to output correctly formed messages that contain random information.
- Prolonged Activation. Once activated the bioNode ignores inhibitory messages, and uses a longer auto deactivation time.

5.3 Support Hardware

The bioNode system incorporates a number of sub-systems that provide supporting roles. These include power regulation for the bioNodes, programming control for the bioNode's microcontrollers and FPGAs, and a set of communication links to each bioNode microcontroller.

These functions are supplied to each bioNode via a pair of umbilical cables, one provides power and the other programming and communication data lines. The supporting systems that connect to these cables are described in this section. It should be noted that these systems have not been been designed to include fault tolerant characteristics, but purely to support the operation of the bioNode system.

5.3.1 BioNode Control and Monitoring

The bioNode's microcontroller has a dedication *USART* that is utilised to provide a communication link between the bioNode and a system management program. This link has three purposes, to input and output data from the system, to provide a method of monitoring the node's operation and to programmatically induce faults into the system.

The management software runs on a personal computer connected through a custom built interface that enables a direct and concurrent connection to each bioNode. The full communication interface is depicted in Figure 5.10.

The interface between the nodes and the management software entails a number of different stages. However, these can be grouped into two main blocks. The first is a hardware multiplexer and demultiplexer that channels the data between the thirty bioNode links and a single data channel. This block is implemented on a Xilinx® Virtex™ FPGA. The second block transfers data between the FPGA block and a dedicated PC via a digital IO card. The data is then made available to the management software via a custom written network based data server.

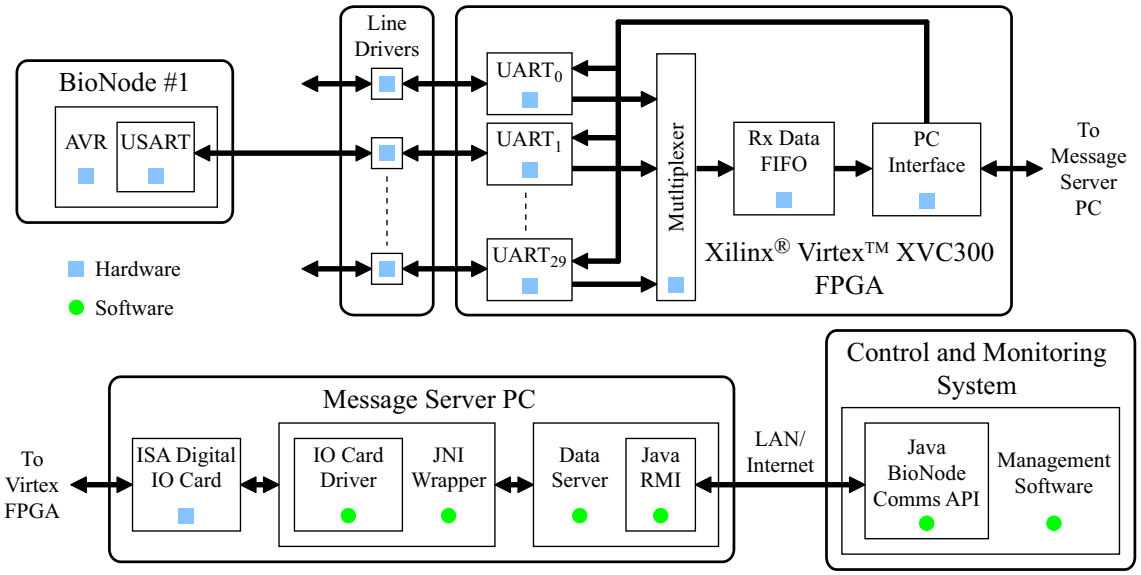


Figure 5.10: *The bioNode control and monitoring system comprises a number of stages, the result is an interface which can be connected to over a standard computer network.*

The main advantage of this set up is its flexibility and ease with which it may be connected to. By using a dedicated machine to channel the data into a networked software domain, the only special requirements needed by a monitoring computer is a common network card. Furthermore, the use of Java [Fla02; Shi00] to implement the RMI [Ste00] based network connection provides system access to a number of different operating systems. Using this system, it is also possible to connect to the data server via the Internet. Although this is outside the requirements of the implemented endocrinology based system, it provides extra functionality for future systems.

5.3.2 BioNode Programming

Programming the bioNodes is achieved via a JTAG chain. The main function of the JTAG scheme is to provide a debugging and fault finding system for integrated circuits. However, the same scan path that is used to read out a device's internal state can also be used to feed programming information in. An useful feature of the JTAG scheme is the ability to chain devices together to form a longer scan, or programming, path. The advantage is that a single set of programming lines can be used to program many devices.

Each bioNode has a dedicated JTAG path connected to it. This allows bioNodes to be reprogrammed separately from all the others. The chaining capability of the JTAG scheme is taken advantage of by connecting the programming ports of the

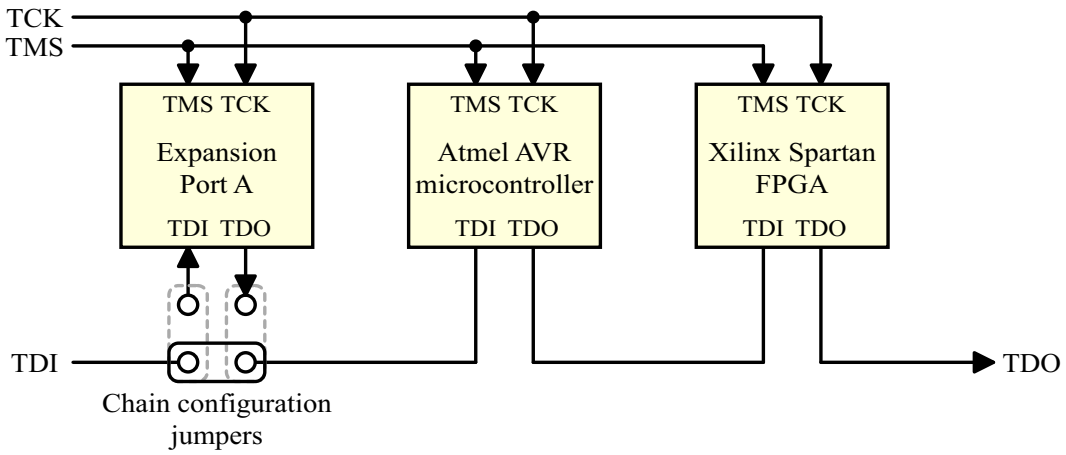


Figure 5.11: *The bioNode uses a JTAG chain to program the on board devices. A jumper allows the chain to incorporate hardware attached to expansion port A.*

microcontroller and FPGA. The chain can also be configured to incorporate one of the bioNodes expansion ports. The complete scan path is shown in Figure 5.11.

This scheme allows a single bioNode module to be individually targeted for programming. This is an important capability, as the FPGAs hold there configuration data in volatile memory. When a bioNode is removed and subsequently replaced its FPGA needs to be reprogrammed without halting the operation of the other bioNodes.

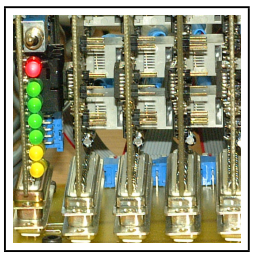
The software and configuration data that are used to program a bioNode's microcontroller and FPGA are held in the same PC used to control the bioNode monitoring communication links. The same hardware that channels the monitoring data is also used to direct the JTAG data to specific bioNodes. This enables the same system interface hosted by the dedicated support PC to provide remote control of bioNode programming.

5.4 Summary

The bioNode system provides a hardware platform capable of implementing the endocrinology inspired system developed in Chapter 4. Individual hardware modules implement the function of a cell. The module interconnects provide channels for hormone messages to propagate. The main system input and output is achieved through a set of dedicated communication links that connect each bioNode to a PC.

The implementation aims that were established prior to the construction of the bioNode system have all been achieved. The system's toroidal shape and visible interconnects form a true representation of the underlying cellular structure. The pairing of a

microcontroller and a FPGA into each bioNode provides a very flexible architecture which can be used to implement many different electronic systems. Finally, allowing bioNodes and their interconnects to be removed provides a method of injecting faults into the system. Further faults can be injected via the bioNode monitor communication links.



Chapter 6

Experiments and Results

The bioNode system provides a hardware platform on which the proposed endocrinology inspired architecture can be tested. This chapter presents the results of the first experiments carried out on this new hardware. Section 6.1 details the first experiments, demonstrating the architecture's ability to correctly perform computational operations and highlighting the importance of paracrine communication. Following these experiments, Section 6.2 describes the tests and results that demonstrate the fault tolerant abilities of the architecture. Finally, the effect of increasing the number of a system organs is tested in Section 6.3.

6.1 Basic System Operation

The experiments detailed in this section were used to determine if the proposed endocrinology inspired system, implemented on the bioNode platform, would correctly produce the organ activation cascade, execute data operations, and perform the operation sequence.

To simplify this task a very simple computational system was devised that would allow the correctness of output data to be easily checked. The system used is depicted in Figure 6.1. The computational operation is divided across three organs, each concurrently performing operations on three streams of data held within the inter cell messages. This system is initially operated without paracrine messages.

The assignment of cells into organs is shown in Figure 6.2a. Each organs contains a set of six positionally grouped cells. The message redirection bias and consequently, general

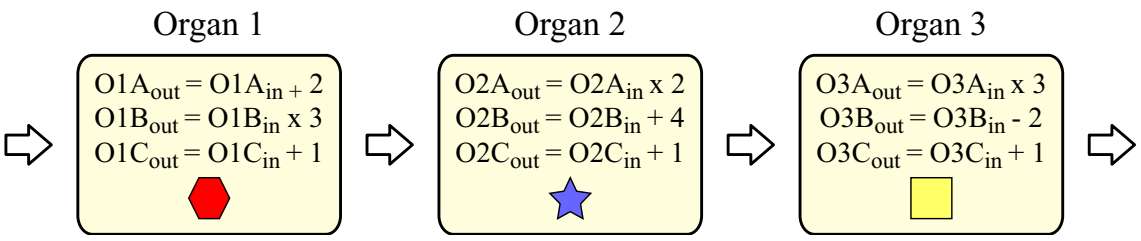


Figure 6.1: *The basic test system performs a very simple computational task. This helps to determine the system's correct computational operation.*

message flow is shown in Figure 6.2c. Figure 6.2b shows the network position numbers. These are used to identify particular cell locations within the network, and will be used throughout this chapter. It should be noted that these numbers are different from those used to generate message source IDs, which are unique to each physical bioNode, and are not related to network position.

Table 6.1 shows the message types associated with the cells of each organ. For example, cell 10, a member of organ 2, releases messages of type `0x20` when activated. The cell is stimulated into activation by messages of type `0x10`, and deactivated by messages of type `0x30`. Cells of organ 1 do not have an associated stimulatory message type as they are activated when data is directly input into them. As there is no source of inhibitory messages that target the cells of organ 3, they do not have a inhibitory message type associated with them.

6.1.1 Initial Results

The computational operation of the first test system is shown in Figure 6.3. This plot shows the input and output data values for the three streams of data that the system

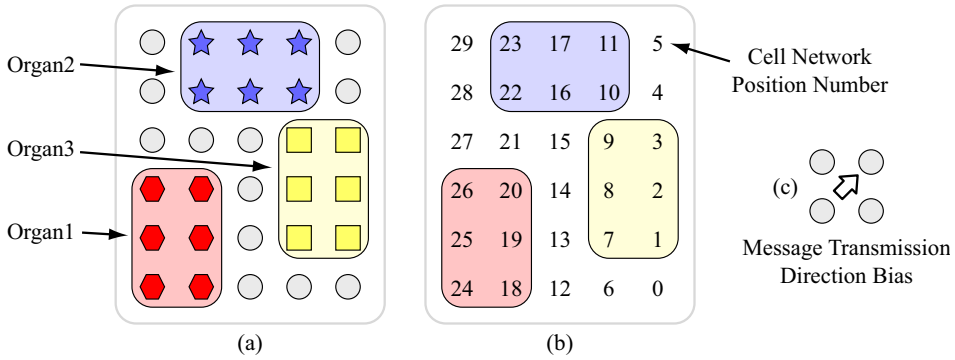


Figure 6.2: *The cells within the lattice are organised into three organs, each containing six positionally grouped cells (a). The network cell position numbers are also shown (b). The message transmission bias is shown in (c).*

Organ Number	Constituent Cells	Stimulatory Message Type	Inhibitory Message Type	Released Message Type
1	18,19,20,24,25,26	-	0x20	0x10
2	10,11,16,17,22,23	0x10	0x30	0x20
3	1,2,3,7,8,9	0x20	-	0x30

Table 6.1: *The message types relevant to each cell type. Shown are the message types that have a stimulatory or inhibitory effect, and the message type released by a cell.*

operates on. The data values are shown with respect to sample time rather than actual experiment runtime. This allows the correlation between input and output values to be more readily seen. The output data values are derived from a majority vote of the data retrieved from the cells of the last organ. The number of cells contributing to one sample of output values is shown above each output sample. To allow the system's correct computational operation to be more readily discerned, Table 6.2 shows the input and output data values for the first nine samples. These values show that the system did perform the correct computational operation on each data stream.

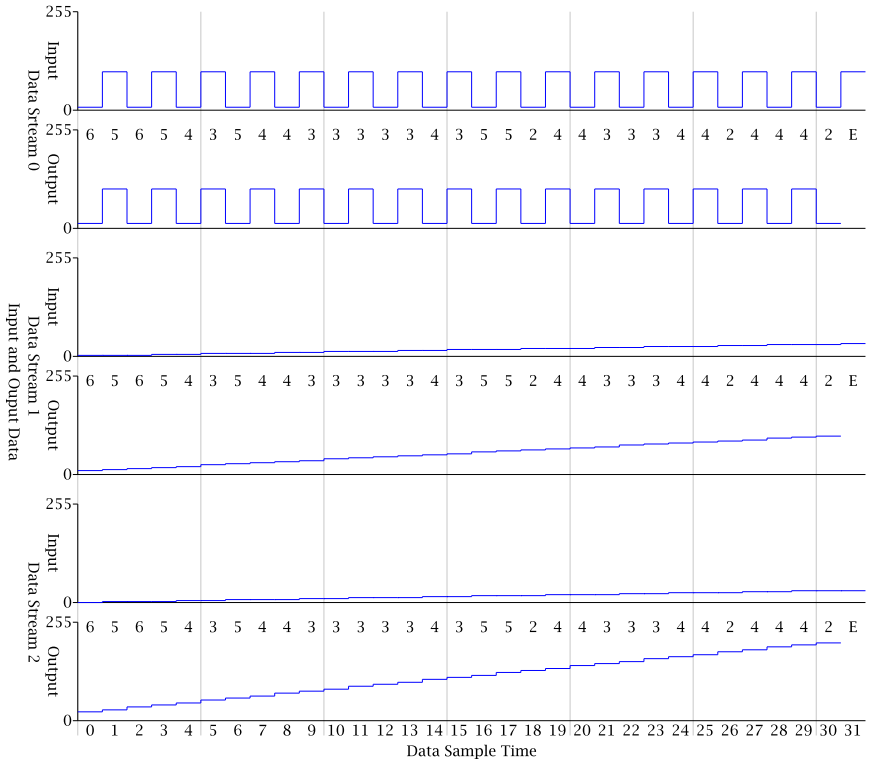


Figure 6.3: *A plot of the input and output data for the three different system data streams. The numbers following above each output trace represent the number of last organ cells whose matching output data has been combined to create the final output value.*

Sample Time	# of Contributing Output Cells	Input Data			Output Data		
		2	1	0	2	1	0
0	6	2	3	10	24	11	13
1	5	3	4	100	30	14	103
2	6	4	5	10	36	17	13
3	5	5	6	100	42	20	103
4	4	6	7	10	48	23	13
5	3	7	8	100	54	26	103
6	5	8	9	10	60	29	13
7	4	9	10	100	66	32	103
8	4	10	11	10	72	35	13

Table 6.2: *Shown are the first nine data samples of the three data streams that the system operated upon. The number of last organ cells contributing to the final output data values is also shown.*

The actual timing of input and output messages is shown in Figure 6.4. From this plot it is possible to see the delay between data input, and the output of data from the cells of the last organ. To provide a clearer view of how the cells are operating on the data streams, Figure 6.6 shows the activation levels of all the non-dormant cells within the system. Note an explanation for this plot is shown in Figure 6.5.

The traces of Figure 6.6 are arranged into three groups of six, one for each organ within the system. The first six traces, cells 18, 19, 20, 24, 25 and 26, are for organ one. Each of these cells can be seen to activate at the same point of time, as they simultaneously receive data from the external source. As the messages these cells release propagate through the system, the cells of organ two, cells 10, 11, 16, 17, 22 and 23, become activated. The same then occurs for organ three. It should be noted that the timing discrepancies, particularly visible in the difference in activation time of organ three cells, is due to the discretised nature of the cell status monitoring.

There are three features of system operation, shown by this plot, that are of particular interest, all related to cell deactivation. First is the inhibitory effect messages can have on cells. It is possible to see that the cells of organ one, operating on sample zero, are being deactivated by messages well before their activation level is automatically reduced to zero. This shows that messages released by organ two cells are performing both a stimulatory and inhibitory function.

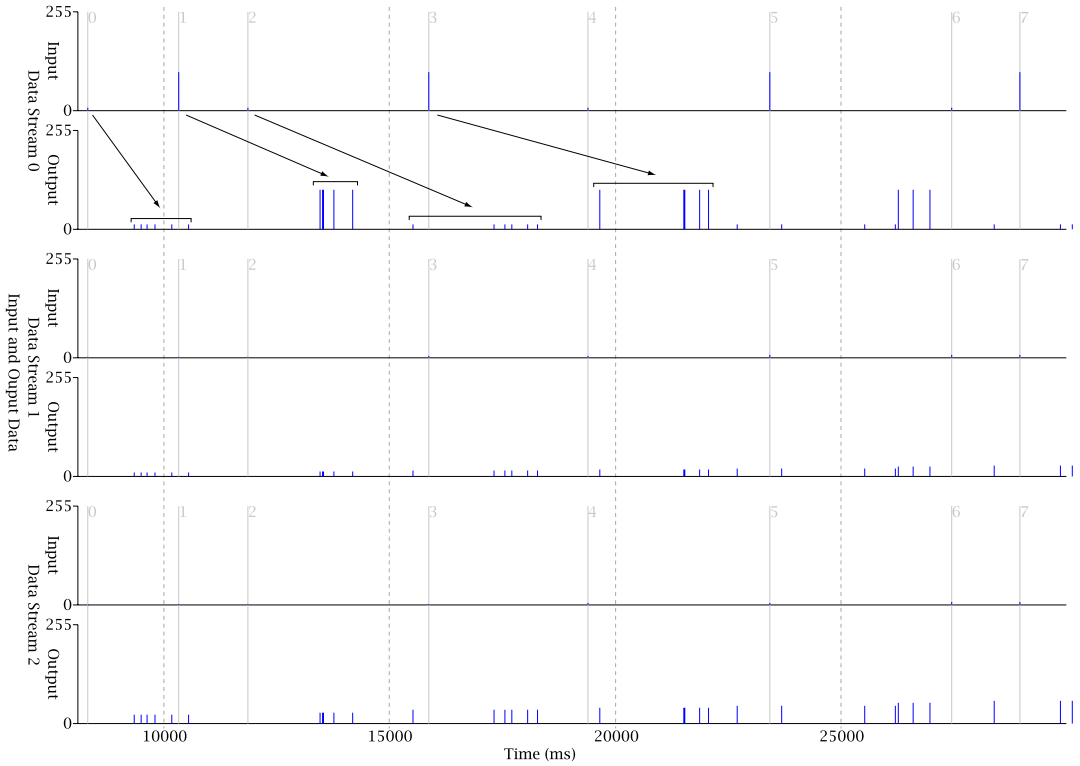


Figure 6.4: *A plot of the overall system input and output data shown as data messages are input to first organ cells, and received from last organ cells. The input data sample time is shown for each input message.*

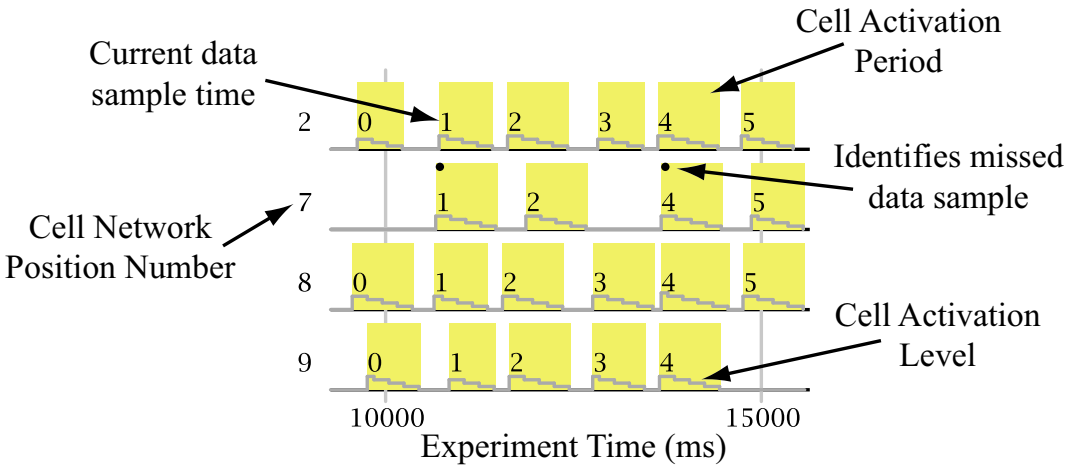


Figure 6.5: *An explanation of the cell activation level plots. Each horizontal plot shows the activation level of a single cell. The sample time of data being operated upon is shown, as well as an indication of missed samples.*

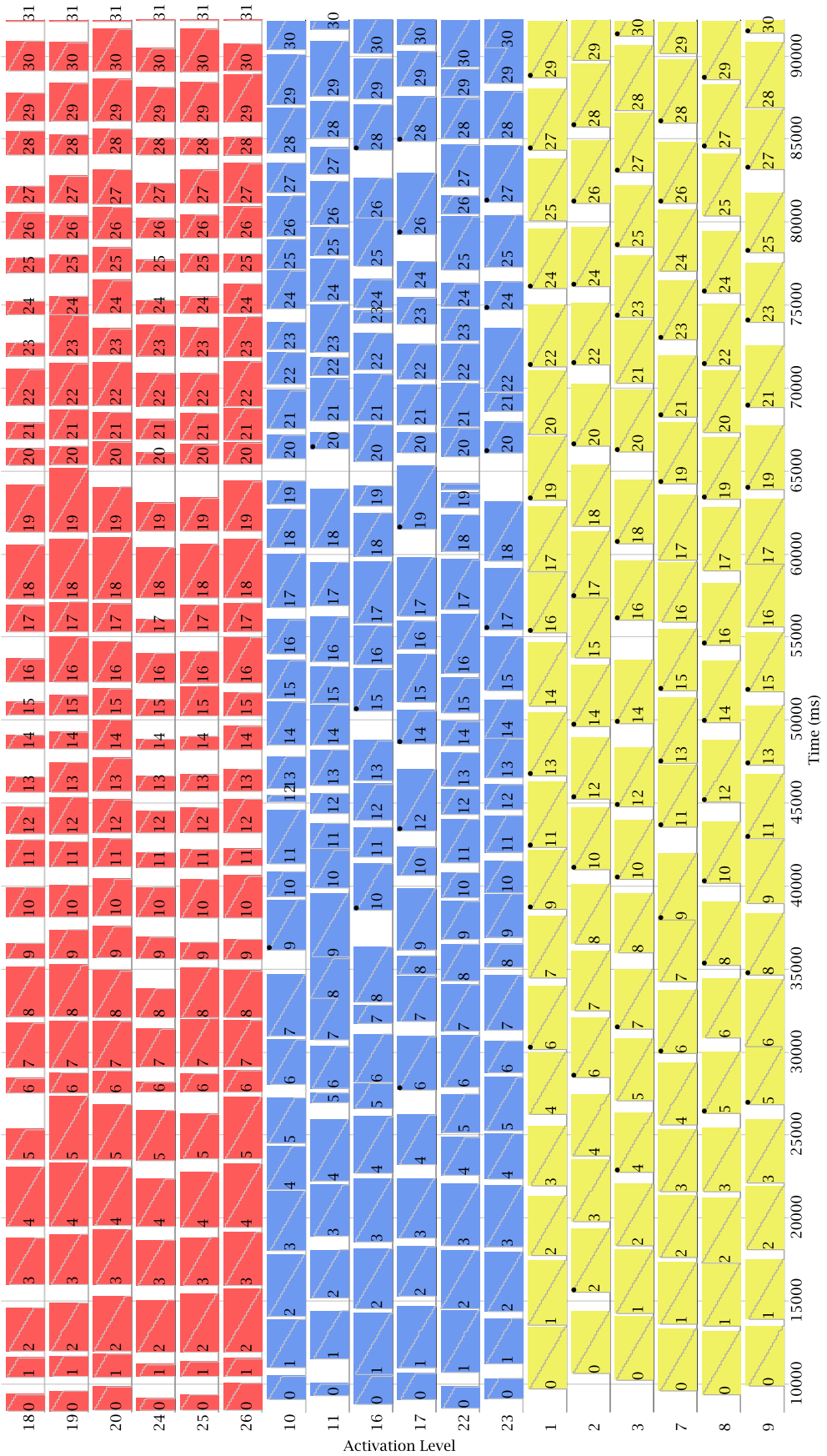


Figure 6.6: *A plot of cell activation levels. The cells without activity plots are dormant cells and do not undergo activation. See Figure 6.5 for a plot explanation.*

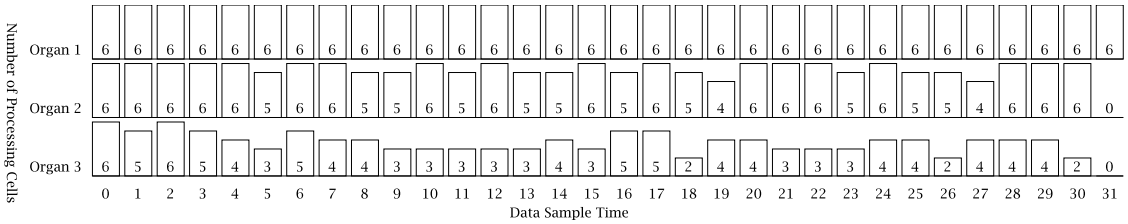


Figure 6.7: *A plot of the number of cells within each organ that operated on a particular sample. For example, only three organ three cells operated on the data of sample five.*

However, this message mediated deactivation is not consistent. The second point of interest is how some non-last organ cells are having to wait for a longer time before they receive inhibitory messages, some even automatically reach deactivation before any inhibitory messages arrive. This problem is shown by the activation level of organ one cells operating on sample two. All are remaining activated longer than when they operated on sample 0 or 1. The reason is the length of automatic deactivation time used by the cells. For example, whilst organ three cells are activated to operate on sample zero, the cells of organ two, processing sample one, are waiting for inhibitory messages. These will not be released until the organ three cells are deactivated and starts processing sample one also. The result is that the organ two cells are waiting for inhibitory messages, some reaching deactivation under automatic control. The same delay is passed on to the organ one cells, which also suffer the extended activation.

It can be seen by sample six, the cells of organ one are once again being deactivated by inhibitory messages. This is a result of the third interesting operational feature. Figure 6.7 shows the number of cells within each organ that operated on a particular sample. Due to the data input mechanism, all the cells of the first organ operate on every sample. However, after sample four, the number of cells within organ three operating on the same sample reduces. A closer inspection of the relevant traces of Figure 6.6 show that the organ's cells are dividing their operation between alternating samples. This emergent behaviour results in an increased likelihood that inhibitory messages will reach their targets before automatic deactivation occurs.

Clearly the data throughput of the first system is restricted by the long automatic deactivation time enforced upon the cells of the last organ. To address this problem, a second experiment was undertaken that used a maximum activation level set to a quarter of the previous maximum level. The consequence this has on cell activation is shown in Figure 6.8. The short activation time means there are far less messages being released into

the system, the result is that a reduced number of organ two cells are becoming activated, and even less for organ three. Figure 6.9 shows the effect on system data output. Output data for samples 13 and 21 could not be formed as no organ three cells operated on this data. Furthermore, a number of samples were only operated on by two or less organ three cells, which does not provide enough data to perform a majority vote on the output.

This situation must be avoided as the integrity of data output is lost. However, as the following section will show, the introduction of paracrine messages avoids the need to return to higher maximum activation levels.

6.1.2 Introducing Paracrine Messages

The effect of paracrine messages is to increase the impact of a single endocrine message. The area of increased message reception is localised to cells that are likely to be specific to the original endocrine message. The result this has on the system is to increase the coordination of cells such that they operate together on the same sample of data.

A third experiment based on the same three organ system was undertaken. The same reduced maximum activation level was used, but paracrine messages were enabled. The result on cell activation levels is shown in Figure 6.11. Many more organ cells are operating on each data sample. Furthermore, as can be seen from Figure 6.10, the system is able to generate an output data value for every sample.

The requirement for cells to operate on every data sample is dependant on the type of operation cells are performing. The system used for the experiments so far comprised cell operations that were only a function of the data held within the activating messages. For more complex operations, such as integrators, or filters, past sample data will also be required. Cells will need to store data from previous samples to perform these operations, and consequently, if cells do not receive all data samples, they will not have a correct set of past data on which to perform their operation.

The presence of discontinuities in the sequence of samples operated upon by cells heavily limits the types of system the architecture can implement. This is a matter for further work and is discussed in Section 7.3.5.

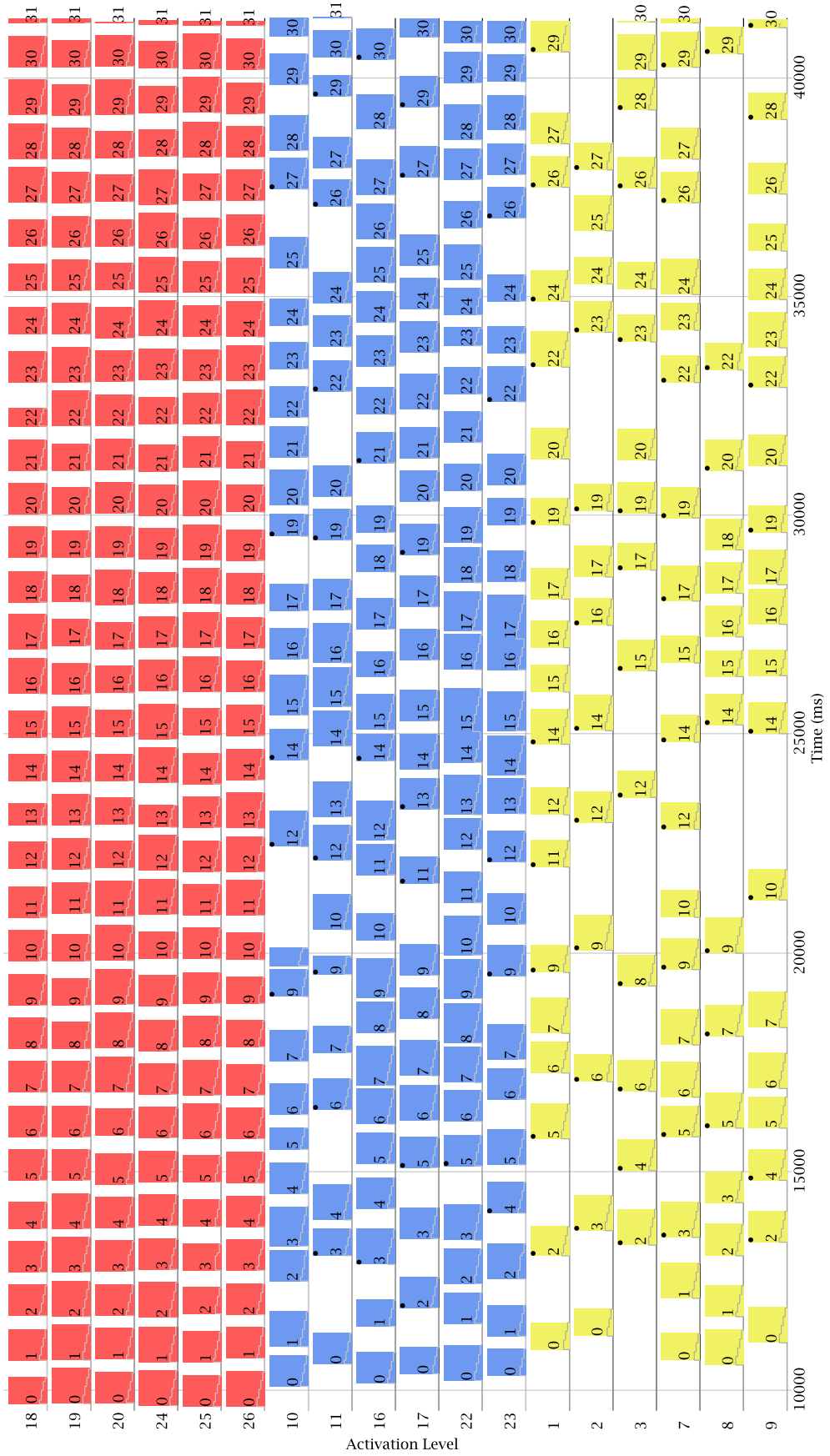


Figure 6.8: *A plot of cell activation levels. The same system is used, but the maximum activation level is set to a quarter of the previous maximum level.*

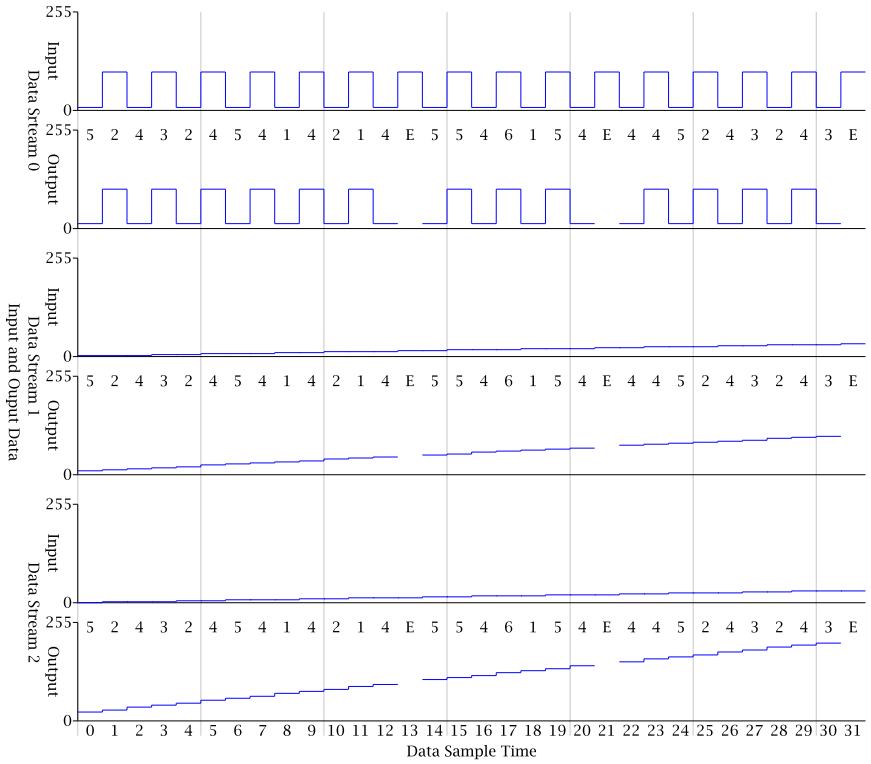


Figure 6.9: *A plot of the input and output data for the three different system data streams. The missing sections within the output traces show where an output data value could not be formed.*

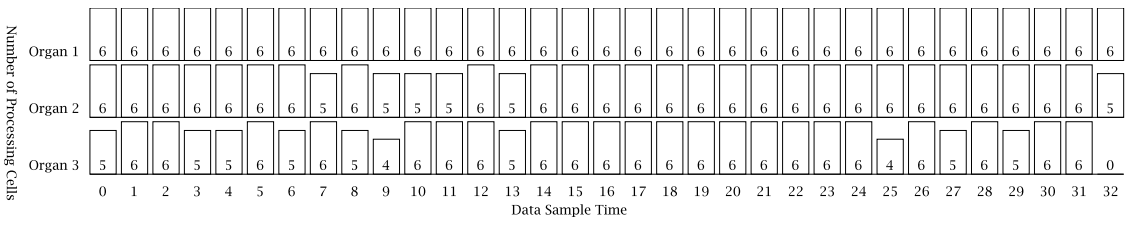


Figure 6.10: *A plot of the number of cells within each organ that operated on a particular sample. The introduction of paracrine messages greatly increases cell coordination.*

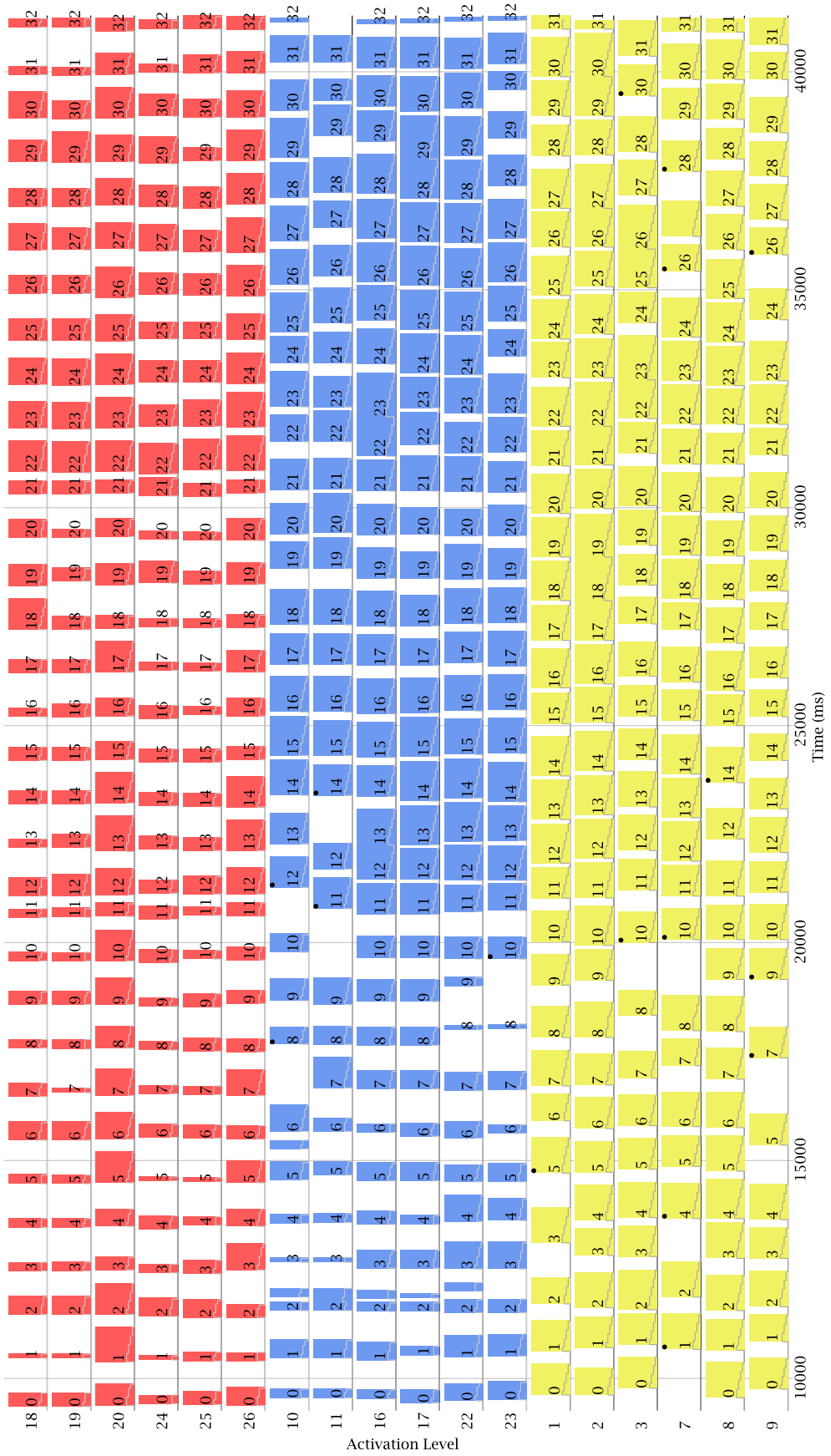


Figure 6.11: *A plot of cell activation levels. The same system is used, but with paracrine messages.*

6.2 Testing Fault Tolerance

To test the fault tolerance abilities of the architecture, faults were programmatically injected into the system. The system used for testing was the same as that described so far. The lower maximum activation level was used, and the vital paracrine messages were enabled. Table 6.3 shows the cells affected and the time of fault injection. In this test, the only fault type injected was complete cell shutdown. This stops the affected cell from receiving or transmitting any data, and therefore effects both operational and dormant cells.

Sample Time When Fault Introduced	Cell Affected	Fault Type	Cell Type
5	20	Shutdown	Organ 1
8	16	Shutdown	Organ 2
12	6	Shutdown	Dormant
15	1	Shutdown	Organ 3
18	25	Shutdown	Organ 1
21	9	Shutdown	Organ 3
24	29	Shutdown	Dormant

Table 6.3: *The type and time of faults injected in the first fault tolerance test.*

The input and output data plots shown in Figure 6.12 show that the system's computational operation is unaffected by the introduction of faults. Figure 6.14 shows the cell activation levels for this experiment. The permanent deactivation of cells induced by fault injection can be clearly seen.

The first fault tolerance test demonstrates the architecture's ability to make use of its cellular redundancy. However, all the cells were failed in such a manner that they would not affect the operation of other cells. The second fault tolerance test makes use of two other fault types. As described in Section 5.2.3, the three faults that may currently be injected are complete cell shutdown, stopping the cell from receiving or transmitting messages; inducing cells to transmit message that contain all garbage data; and disabling the effect of inhibitory messages whilst increasing the cells maximum activation level. Table 6.4 shows the type and introduction time of the faults injected. Note, the faults injected into Cell 8 are removed after sample time 18, this is to demonstrate the architecture's ability for repair by incorporate new cells.

Sample Time When Fault Introduced	Cell Affected	Fault Type	Cell Type
5	8	Shutdown	Organ 3
8	23	Garbage Output Messages	Organ 2
12	19	Shutdown	Organ 1
15	21	Shutdown	Dormant
18	8	Cell Repaired	Organ 3
21	3	Inhibition Ignored	Organ 3
24	24	Garbage Output Messages	Organ 1
27	1	Shutdown	Organ 3
30	17	Inhibition Ignored	Organ 2
33	5	Shutdown	Dormant
36	11	Shutdown	Organ 2

Table 6.4: *The type and time of faults injected in the second fault tolerance test.*

Figure 6.13 shows that the system is able to produce correct output data whilst harbouring a considerable number of faults. It is not until sample time 37, after the introduction of 10 faults, that system is unable to output a valid output data sample. At sample time 18, the faults injected into Cell 8 are removed. It can be seen from Figure 6.15 that the cell successfully reintegrates into organ 3, and continues to operate on message data.

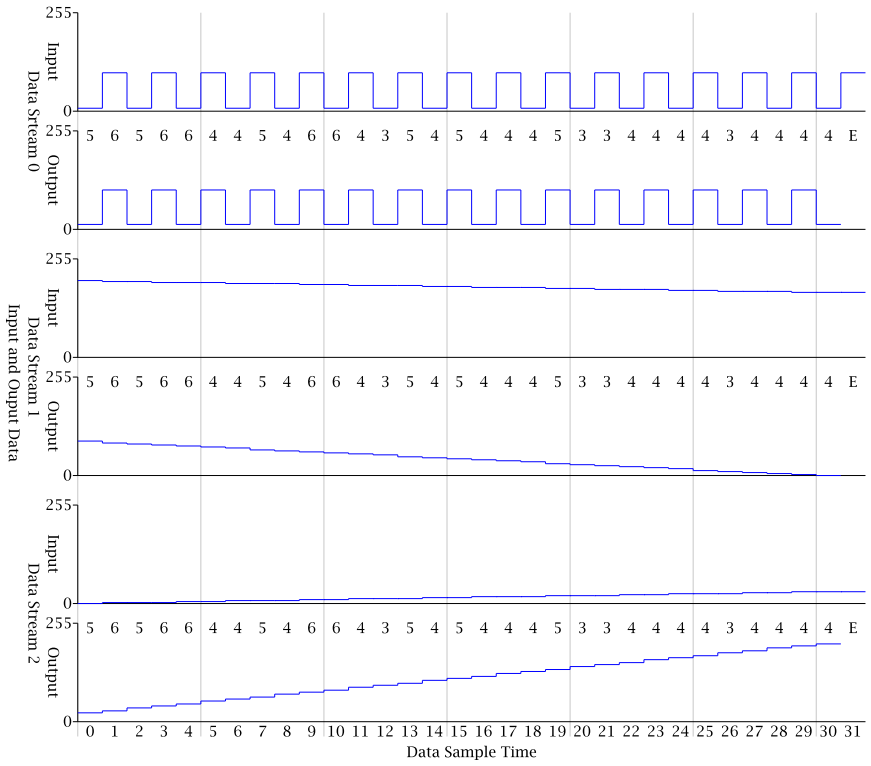


Figure 6.12: *A plot of the input and output data for the first fault tolerance test system.*

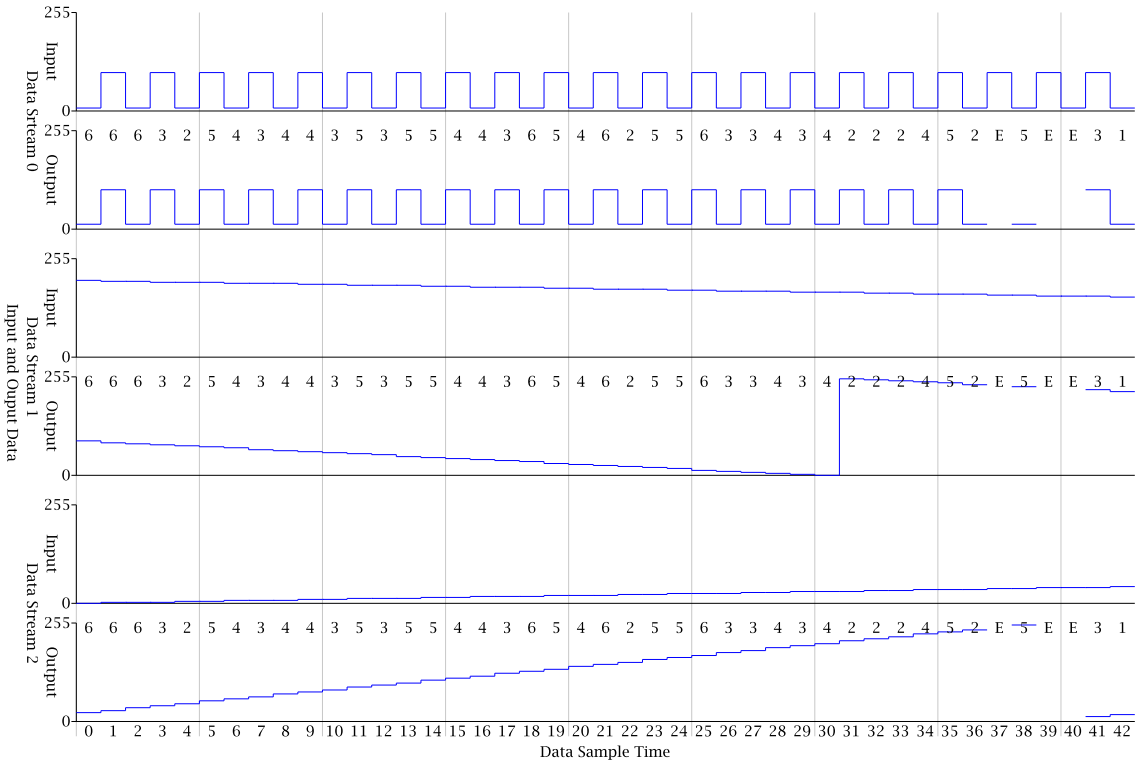


Figure 6.13: *A plot of the input and output data for the second fault tolerance test system.*

Figure 6.14: *A plot of cell activation levels for the first fault tolerance test system.*

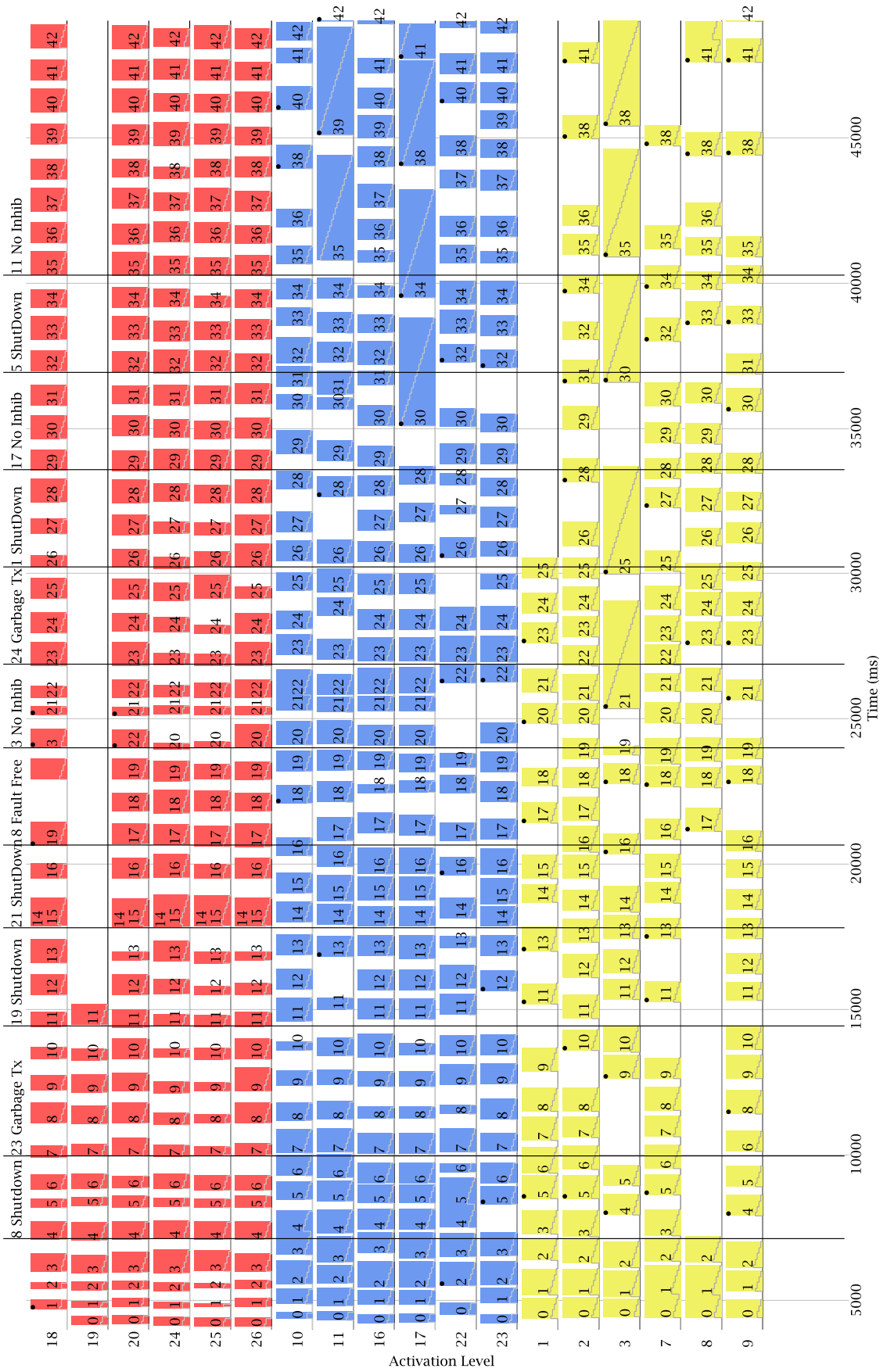


Figure 6.15: *A plot of cell activation levels for the second fault tolerance test system. The superimposition of data samples 14 and 15 for Organ 1 cells occurs due to a fast deactivation for sample 14.*

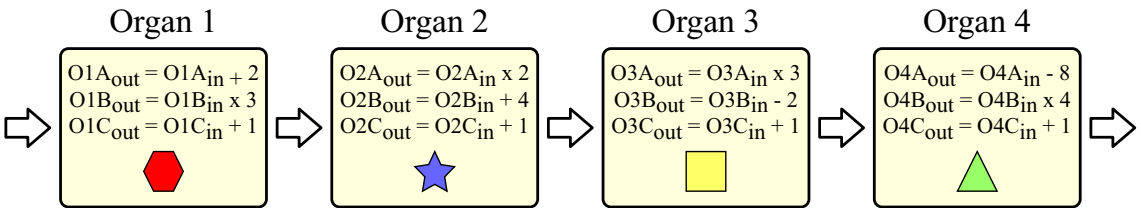


Figure 6.16: *A test system that contains four organs.*

6.3 Four Organ System

To test the architectures ability to correctly operate with a different number of organs, a four organ system was tested. Figure 6.16 shows this second test system. The assignment of cells to organs is shown in Figure 6.17. The message types relevant to each cell are shown in Table 6.5.

Organ Number	Constituent Cells	Stimulatory Message Type	Inhibitory Message Type	Released Message Type
1	8,9,14,15,21	-	0x20	0x10
2	18,19,24,25,26	0x10	0x30	0x20
3	17,22,23,28,29	0x20	0x40	0x30
4	0,1,2,6,7	0x30	-	0x40

Table 6.5: *The message types relevant to each cell type. Shown are the message types that have a stimulatory or inhibitory effect, and the message type released be a cell.*

The longer activation cascade can be seen to operate correctly from the input and output data plots shown in Figure 6.18. However, the number of last organ cells that are providing output data per sample is lower than that of the three organ system. This

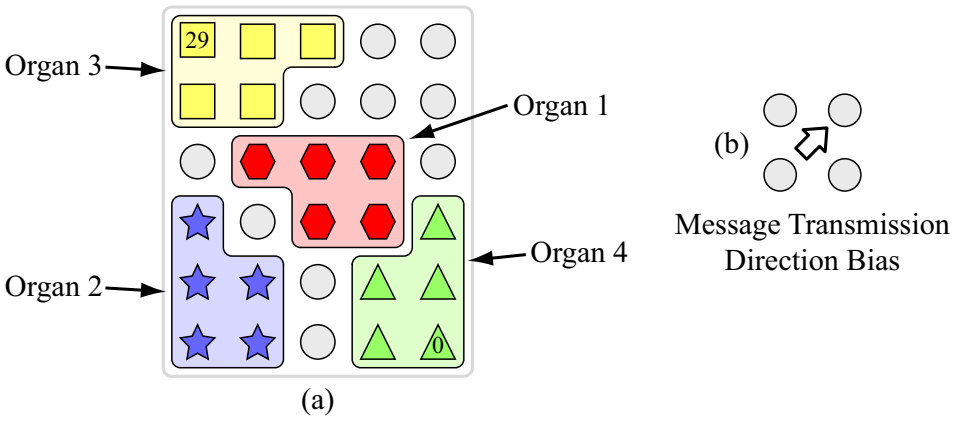


Figure 6.17: *(a) The assignment of cells within the four organ system. The same message transmission direction bias is used (b).*

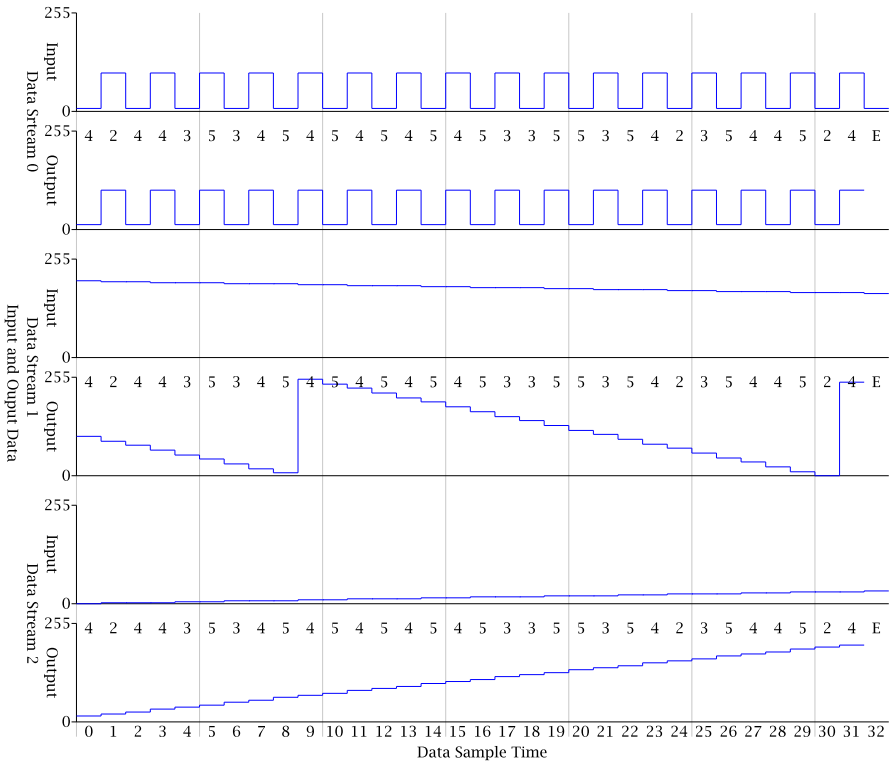


Figure 6.18: *A plot of the input and output data for the three system data streams of the four organ system. The number of last organ cells contributing to an output value is noticeably lower than those of the three organ system.*

is to be expected due to organs having five constituent cells rather than six. However, it can be seen from Figure 6.19 that the number of cells operating on each sample of data decreases further through the organ cascade. This suggests that it is the cascade length that affects cell coordination rather than organ size.

The importance of good cell coordination within an organ has already been mentioned. In this case the progressive lack of coordination is leading to samples being missed by the cells of organs at later positions in the cascade. Although the system provides correct output data, the system operation is further from the ideal of all organ cells operating

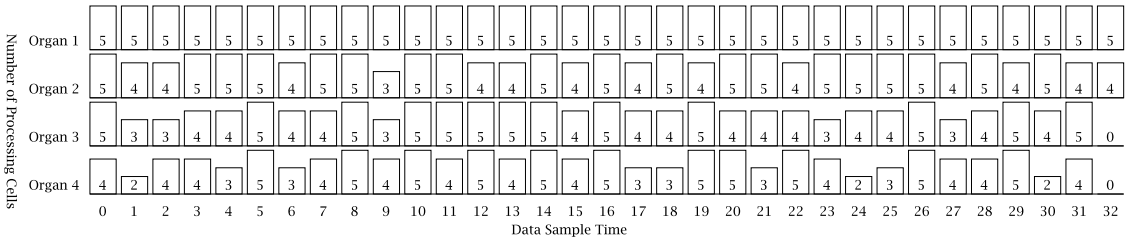


Figure 6.19: *A plot of the number of cells within each organ that operated on a particular sample. The introduction of paracrine messages greatly increases cell coordination.*

together on every data sample. This situation strongly points towards a need to increase the utilisation of cell paracrine messages.

6.4 Summary

A number of experiments have been performed using the bioNode system to test the operation of the endocrinology inspired architecture. These experiments have verified the architecture's ability to correctly perform computational operations on multiple concurrent data streams. The required sequence of activated organs has been demonstrated, as well as the stimulatory and inhibitory effects of the hormone messages.

The importance of paracrine messages has been highlighted. The extra coordination that the inclusion of this secondary messaging system provides, promotes an organ's cells to operate on a greater number of data samples. This is particularly important if cell operations requiring a data history are implemented, however, this is a field for further work.

The architecture's ability to tolerate faults has been demonstrated. The static redundancy present in organs is able to tolerate both cell loss and the presence of malfunctioning cells. Furthermore, the demonstrated ability to seamlessly incorporate new cells into an organ show the architecture's capacity for repair.

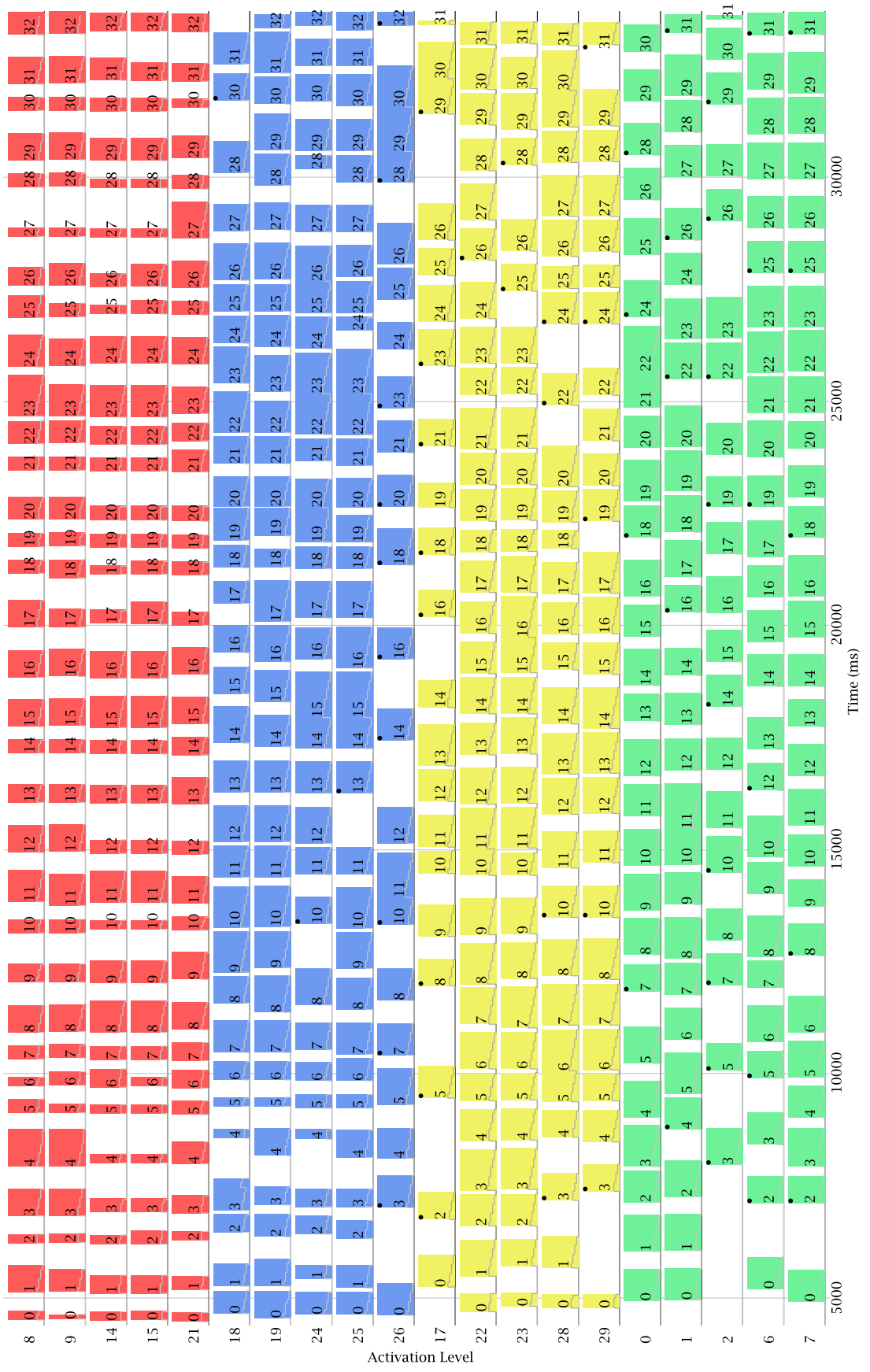
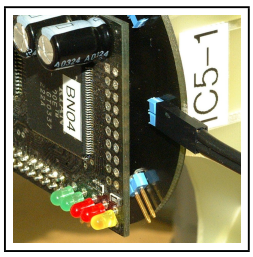


Figure 6.20: *A plot of cell activation levels for the four organ system.*



Chapter 7

Conclusions

This final chapter contains a summary of the presented research and the conclusions that may be drawn from it. Following these sections is a list of suggestions for further work. The chapter is finished with a discussion of the potential future of bio-inspired systems.

7.1 Research Summary

The continual improvement in electronic device fabrication techniques is providing the means to produce increasingly complex integrated circuits. A situation whose momentum is fuelled by the desire of engineers to create ever more capable electronic systems. Generally, increased complexity is realised in the form of greater numbers of device transistors. The growth of which, as predicted by Moore, has been consistent, and shows no signs of diminishing. A consequence, and problem, of increasing levels of transistor integration is electronic devices becoming increasingly likely to suffer failure of internal features.

Electronic systems are vulnerable to many different types of fault. Chapter 2 provided an introduction to a selection of these and the failure mechanisms that lead to their creation. As highlighted by the chapter, many faults such as SEUs or electromigration occur once a system has been placed into service. The only course of action available to handling these post-production faults is the employment of fault tolerance techniques.

The research documented by this thesis involved the development of a novel fault tolerant electronic architecture, designed to maintain correct system operation even with the presence, and occurrence, of faults. The inspiration for this architecture was taken

from animal biology, the rationale being, the underlying processes that enable biological organisms to endure injury are applicable to the design of fault tolerance techniques. As a further incentive, organisms exhibit fault handling abilities despite being both structurally and operationally complex, suggesting that the biological approaches to fault tolerance are appropriate for inspiring those of complex artificial systems.

An analysis of biological organisms revealed the utilisation and organisation of redundancy found in their underlying cellular structure as a key factor in their ability to survive internal faults. This observation led to the concept of a simple electronic cellular lattice that could be used to artificially implement higher level biological systems in a fault tolerant manner. The choice of system to overlay the lattice was driven by the need to create an electronic system capable of computation. The biological process highlighted to fulfil this need was the endocrine system.

The endocrine system, as described in Chapter 3, is a main contributor to an organism's control of homoeostasis. The reasons that make this particular biological process so suitable for inspiring a fault tolerant computational system are twofold. The first is the endocrine system's inherent fault tolerance. Operation involves the utilisation of organs, large groups of cells that work in a coordinated fashion which can tolerate cell failure. Inter-organ communication involves no single-point weaknesses, and contains a form of reception acknowledgement. The second reason is the system's ability to activate the processing abilities of organs in a controlled and ordered manner.

The development of these biological features into a conceptual system, described in Chapter 4, is summarised pictorially in Figure 7.1. The fundamental artificial development is the assignment of computational operations to organs and the encoding of data into hormones. The result being a pipeline of processing stages that operate on a data stream.

A number of changes from the true operation of the biological system were required in order to feasibly implement the proposed system in electronic hardware. These developmental changes, described in the Section 4.2, involved recreating an artificial hormone propagation space, dealing with reduced redundancy and providing overall system inputs and outputs. In order to gain an initial level of confidence in the correct performance of the proposed system, simulations, using a custom software model, were undertaken. These provided positive results, demonstrating that an endocrine inspired

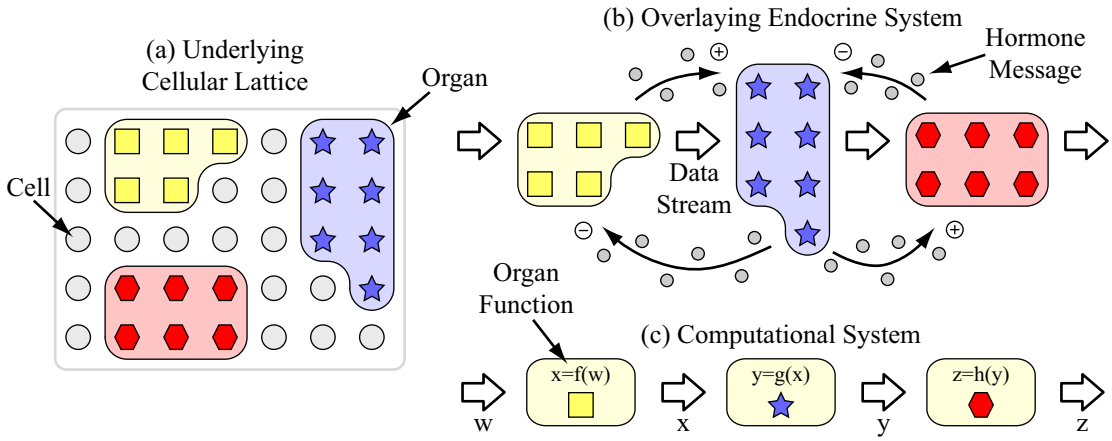


Figure 7.1: *A simplified cellular lattice (a) provides a fault tolerant architecture upon which an artificial endocrine system (b) may function. The true biological system has been altered to create a cascade of activated organs (c) that perform processing operations on a data stream.*

system could be used to correctly perform computational operations whilst suffering cell loss.

In order that more rigorous tests of the system's effectiveness could be performed, a hardware platform capable of implementing the proposed system was required. As described in Chapter 5, a multi-purpose platform, called the bioNode system, was built containing thirty highly reconfigurable nodes capable of modelling cell behaviour. The bioNode network is ideal for implementing the artificial cellular lattice and the required hormone message propagation space. A set of supporting sub-systems, allow data to be input and output from the system, and monitor the operation of the bioNodes. In order to simulate hardware faults, bioNodes and their interconnects are removable. Furthermore, faults may be programmatically injected into bioNodes, via the monitor communication links.

A number of experiments were undertaken using the bioNode system, the results of which were detailed in Chapter 6. The first set of experiments were used to ascertain the proper operation of the organ activation cascade. They also served to show correct computation of data by the organ's assigned functions. A second set of experiments tested the system's ability to tolerate faults. These tests showed that the system was able to tolerate a considerable number of faults before a correct data output stream could not be formed.

7.2 Final Analysis and Conclusions

The sections that follow present an analysis of the architecture developed by this research and the conclusions that may be drawn as a result.

7.2.1 A Solution for IC Complexity

The need for further research into fault tolerance architectures was presented in Chapter 1 as a response to the increase in IC complexity. In particular, the trend for greater numbers of transistors being integrated into single devices. Therefore, an important question is, does the proposed endocrinology inspired system provide a fault tolerant architecture applicable to protecting the operation of complex ICs?

The answer is very much dependant on how the architecture is implemented. As a single device solution, the circuitry required to create the endocrine signalling system may dramatically out-proportion, or leave little physical space for the the circuitry required to perform the main system operations. Furthermore, the structure of the main system circuitry may not be suitable for dividing into cellular units. These potential obstacles do not rule out the use of the new architecture within a single device, but do limit its potential for use.

The architecture is far more appropriate for implementation at higher system levels, where cells encapsulate whole ICs, or even multiple ICs. In this scenario, the internal circuitry of a cell's functional hardware is unimportant. Also, a smaller proportion of a cell's total hardware will be used to implement the endocrine signalling.

In conclusion, the presented architecture is most suited to incorporating fault tolerance into systems that use highly complex ICs rather than into the individual devices themselves.

7.2.2 A Comparison with Embryonics

The limited number of bio-inspired fault tolerant architectures leaves few systems with which the presented system may be compared. However, the embryonic architecture does contain similarities and deserves a comparison. Both systems are multi-cellular and artificially replicate the concept of pluripotent cells; each cell is able to specialise to any cell type required by the implemented system. Furthermore, dormant cells are

used to replenish cell loss. However, the underlying biological premise of each cellular architecture is quite different.

The embryonic architecture is based on the process of embryo development. Fault tolerance is achieved by replicating cellular growth, controlled by a coordinate system. The endocrinology inspired system utilises an architecture that mimics the cellular structure of a developed organism. In this case, fault tolerance is an inherent property of the organisation and utilisation of that structure. Fundamentally, the difference is the former uses a dynamic redundancy scheme, whereas the latter utilises static redundancy.

There are a number of disadvantages to the embryonic architecture that are not present in the endocrinology inspired system. The dynamic redundancy scheme of a basic embryonic architecture implementation can suffer erroneous output data during system reconfiguration. This can be addressed by limiting the system's data rate to a sample time greater than the reconfiguration latency [Can03; Jac03], however, the consequence is a considerable restriction of the systems functional potential. In contrast, the static redundancy of the endocrinology inspired system means erroneous data is masked, and new cells can be seamlessly reintegrated into the system.

The reconfiguration scheme used in embryonics is heavily dependant on maintaining data connections through failed cells. Unfortunately, this rules out tolerance of faults that might occur on these interconnections. A possibility due to failure mechanisms such as electromigration. The inter-cell signalling scheme used by the endocrinology based system does not have this dependency as data is able to take alternate routes between cells.

Another problem of embryonics is the wastage of redundancy that occurs during reconfiguration. Removing an entire row, or column, of cells to eliminate a single failed cell, generally leaves many healthy cells unused. This is a result of the architecture's strict coordinate system that requires whole lines of cells to be removed to maintain correct coordinate information. The organisation of redundancy in the endocrine inspired system is far less restrictive, failed cells are eliminated without removing healthy neighbours from operation.

In conclusion, although the architecture developed by this research was not created as a replacement for, or in competition of embryonics, it does provides an alternative cell-

based fault tolerance scheme that does not suffer from many of the problems of what is considered a well established bio-inspired fault tolerant architecture.

7.2.3 System Limitations

For computationally simple systems the overhead of implementing the controlling endocrine system could be considered relatively large. However, as the computational complexity of the cells increases this overhead is reduced, even to the point of insignificance. This is possible due to the good abstraction between a cell's computational operations and the underlying signalling system. The only system variable that links the two is the amount of data that needs to be passed between organs per data sample time. Larger messages can take longer to propagate, but if the system is not time critical, then this is not a problem. Furthermore, with the high speed network technologies currently available, message size need not impose a real constraint.

A weakness of this and other static redundancy schemes is the requirement that at some point a single system output must be generated. The final stage of the endocrinology based system requires that output data from the last organ's cells are combined within an external system. The problem being this represents a single point weakness. Furthermore, due to the possible differences in data output times between different cells, a simple passive voter is not an appropriate solution for the task. However, with extra fault preventative efforts concentrated on this data combining stage the weakness may be reduced.

A further limitation of the proposed system is the restriction on the types of system application that may be implemented. The architecture's functional operation is suited to performing a sequence of processing stages on data streams. A requirement of both digital signal processing, and physical plant modelling tasks. However, until further work is taken to improve the coordination of organ cells, the implementation of operations that require operations on past data samples is not possible.

In conclusion, the proposed system does contain a number of operational limitations, however, these do not completely invalidate the usefulness of the architecture. The limitations impose restrictions on how, and what systems can be implemented, but future work may loosen these restrictions, opening the architecture's potential for use.

7.2.4 Achievement of Fault Tolerance

The initial software simulations provided a high level of confidence that a hardware implementation of the endocrine inspired system would be able to tolerate faults. The experiments presented in Chapter 6 proved this to be the case. The redundancy contained in each organ was able to maintain correct computational operation of the systems data steam even whilst harbouring faults. However, to fully establish the extent of the architectures fault tolerant abilities further experiments will be necessary, as suggested in Section 7.3.1.

7.2.5 Final Conclusion

The experiments used to test the operation and fault tolerant abilities of the endocrinology inspired system have provided positive results. They have demonstrated the architecture's ability to correctly perform computational operations, whilst harbouring hardware faults that would have normally caused system failure. These results support the final conclusion that the hypothesis given at the start of this thesis, reproduced below, has been proved by the undertaking of the presented research.

Hypothesis: The operation and underlying architecture of the endocrine system can provide inspiration for the structure and operation of a reliability engineered electronic system.

7.3 Further Work

There now follows a list of suggestions for further work, providing a variety of directions in which the course of this research may be continued.

7.3.1 Further Experiments and Tests

The system data presented in Chapter 6 represents the results of a preliminary investigation into the performance of the endocrinology based system. This data established the system's computationally correct and fault tolerant operation. However, subsequent experiments may be used to further explore the particular modes of cell failure that the system can tolerate, and those that could still lead to complete system failure.

7.3.2 Optimisation of System Parameters

A number of system parameters that control the operation of the endocrinology based system were selected through limited experimentation. These variables include the lifetime of system messages, the number of messages a cell releases before automatic deactivation occurs, and the number of unique message sources required to activate and deactivate a cell.

Further experimentation to optimise these values may result in a system with improved performance. It is possible that the majority of messages are binding a long time before they reach removal age. Finding the correct message lifetime could reduce the number of messages within the network at any one time, and reduce the overhead of passing on irrelevant messages.

The cells of the last organ are dependant on their automated deactivation to ensure that they may move on to processing subsequent data samples. When selecting the number of hormones to release before deactivation occurs, there is a trade off between slowing the flow of data through the system when the cells remain active for too long, and not providing enough messages to inhibit the previous organ's cells.

7.3.3 Automated Cellular Regrowth

The static redundancy contained in organs allows the failure of cells to be tolerated. This is true of both biological and the artificially recreated system. However, in biology, cell deficiency is replenished by the growth of new replacement cells. As this is not possible in electronic hardware, the cellular architecture of the bioNode system allows for a number of spare dormant cells to be included in the network. The activation and joining of which into deficient organs represents an artificial form of cellular regrowth. Automating the process of activating the dormant cells that neighbour an organ suffering cell loss would represent a major system development.

7.3.4 Error Detection

To implement an automated cell replacement scheme that maintains organs with a full complement of healthy cells, a fault detection mechanism would be required. Cells would need to able to identify a cell as failed, invoke a form of apoptosis upon it, and then switch in a replacement of the same type.

The detection mechanism could be achieved by monitoring message data. A receiving cell could check the integrity of a message and detect if the transmitting cell is producing erroneous data. As cells are able to perform any operation required by a system implementation, it is even possible for cells to check the computational result stored in a message. Once a cell has been detected as failed, it can be easily isolated from the network if all the neighbouring cells simply ignoring its presence. No data would be passed to it, and received data would be rejected.

7.3.5 Improve Cell Coordination

The correct coordination of an organ's constituent cells is vital for implementing cell operations that require the processing of past data. If cells are not activated for every data sample they loose past data and would be operating on an incomplete data sequence. Operations that use stored past data, such as integrators, differentiators and filters, are vital system functions, required by most digital signal processing or plant modelling tasks.

Therefore, improving the coordination of cells within an organ is a major direction for further work. The positive outcome of which would make the proposed architecture a far better candidate for real world applications.

7.4 Final Thoughts

The development of electronic components seems to move further forward with alarming pace everyday. As development obstacles are met, the implementation of new technology maintains a stream of increasingly capable devices. The continuation of Moore's Law is repeatedly stated as having reached its end, but is then pushed further forward with new technology. The ongoing development process looks to continue the stream of new and improved components that will in turn be put to wider and more advanced uses. The question this situation provokes is where, if at all, do bio-inspired systems fit within this continued electronics development, and if so, how?

There are a number of areas where biological systems or characteristics far outperform their artificial counterparts. It is where these biological features would represent a useful artificial attribute that bio-inspired work is likely to occur. Biological fault handling techniques is only one such area, with abilities such as object and pattern recognition,

emergent behaviour, or even consciousness and thought being other bio-inspired holy-grails.

There is little that may form an argument against the desire for bio-inspiration in electronics, so what is holding back its further exploration? Perhaps the main obstacle is knowledge. Electronic engineering techniques generally remain in the electronic knowledge domain. Tradition holds a lot of value in an industry that requires quick and accurate solutions. However, this and an increasing number of other bio-inspired works demonstrate that the borders between different knowledge domains are being crossed. As electronic engineers discover more about alternate subjects, the pool of knowledge from which solutions to electronic problems can be drawn broadens.

A more practical obstacle is a matter of implementation difficulties. Biological systems operate using a completely different infrastructure to electronic circuitry. Many biological systems are dependant on the movement and regrowth of complex units of physical matter, characteristics which are unrealisable with current electronic techniques. Engineers have found themselves in a situation similar to Leonardo Da Vinci during his design of ornithopters. The biological knowledge has been identified and understood, but the materials and tools are not yet available to fully convert them into a functional artificial system. However, like the electronic fabrication problems that have come before, these yet unrealised electronic abilities may too become another realised part in the continual stream of electronic device development, allowing the growth of bio-inspired systems to flourish.

Appendix A

Derivations and Data

This appendix contains derivations of the equations and terms used within the text. Also included, where appropriate, is data for included graphs in tabular form.

A.1 Reliability

Calculating a measure of dependability for a component placed in a particular in-service environment may be achieved as follows [Lal85]. First consider a quantity of N component samples, all placed in a environment representative of that whilst in service. The number of components that have survived failure at time t is given by $S(t)$. The probability of survival, also called reliability, is given by:

$$R(t) = \frac{S(t)}{N} \tag{A.1}$$

If $F(t)$ is the number of failed components at time t, then the unreliability is:

$$Q(t) = \frac{F(t)}{N} \tag{A.2}$$

The rate of failure $Z(t)$, normally quoted in units of failures per hour, is given by the change in failed components compared to the current number of surviving components.

$$Z(t) = \frac{1}{S(t)} \frac{dF(t)}{dt} \tag{A.3}$$

Figure A.1 shows the normal change in failure rate for a set of components as they age. The initial decline, called the *burn-in period* is due to inevitable severe manufacturing defects effecting a subset of components. There follows a stretch of relatively constant

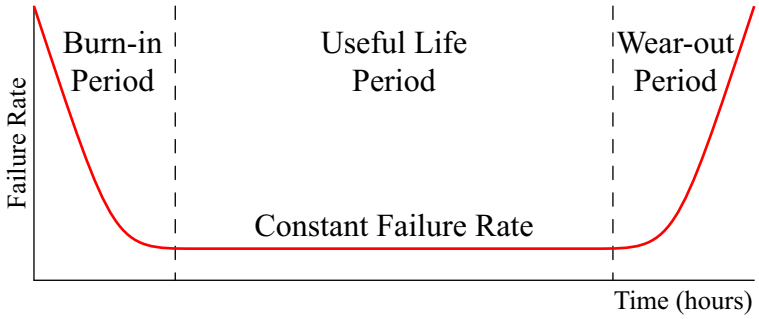


Figure A.1: *The usual change in component failure rate as time passes fits into three distinct phases.*

failure rate, called the *useful life period*. The final increase is due to components reaching the end of the lifetime, this is the *wear-out period*. By concentrating on the useful life period we can set the failure rate to a constant:

$$Z(t)_{\text{useful life period}} = \lambda \tag{A.4}$$

After rearrangement, an equation for the component failure rate in terms of component reliability may be achieved, Equation A.7.

$$R(t) = \frac{S(t)}{N} = \frac{N - F(t)}{N} = 1 - \frac{F(t)}{N} \qquad \text{since } S(t) + F(t) = N \tag{A.5}$$

$$\frac{dR(t)}{dt} = -\frac{1}{N}\frac{dF(t)}{dt} \qquad \text{rearranges to} \qquad \frac{dF(t)}{dt} = -N\frac{dR(t)}{dt} \tag{A.6}$$

$$\lambda = \frac{N}{S(t)}\frac{dR(t)}{dt} = -\frac{1}{R(t)}\frac{dR(t)}{dt} \qquad \text{since } R(t) = \frac{S(t)}{N} \tag{A.7}$$

To find the reliability in terms of failure rate it is necessary to rearrange, Equation A.8, then integrate, Equation A.9. The limits of integration represent the experiment start time, $t = 0$ when reliability is 1, to the point t when by definition the reliability is $R(t)$.

$$\lambda.dt = -\frac{dR(t)}{R(t)} \tag{A.8}$$

$$\lambda \int_0^t dt = -\int_1^{R(t)} \frac{dR(t)}{R(t)} \qquad \Rightarrow \qquad \lambda = -\frac{\log_e R(t)}{t} \tag{A.9}$$

$$R(t) = e^{-\lambda t} \tag{A.10}$$

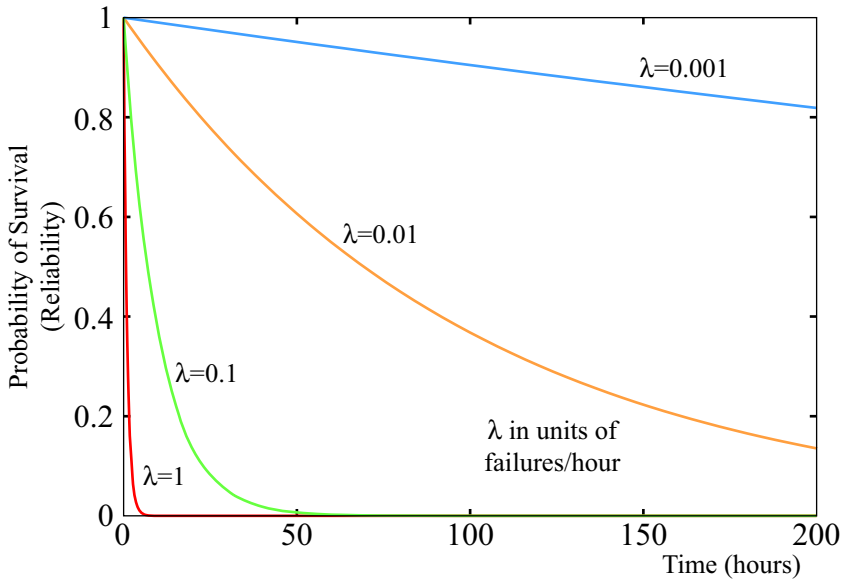


Figure A.2: *The failure rate of a component directly effects the rate at which its reliability declines with time.*

Equation A.10 is known as the *exponential failure law*. It shows hows the reliability of a component exponentially declines in time, the rate of the decline being dependant on the failure rate, usually measured in failures per hour. Figure A.2 shows the decline in reliability for different values of failure rate.

A.2 MTBF

Although the reliability measure provides the probability that a component will fail at a particular time, it doesn't directly give a guide to how long that component will operate without failure. For this, the mean time between failure (MTBF) measure is used.

MTBF is given by the area underneath the reliability curve of the component in question. For the exponential failure law, the MTBF can be derived as follows:

$$MTBF = \int_0^\infty e^{-\lambda t} dt = \frac{1}{\lambda} \qquad \text{when } R(t) = e^{-\lambda t} \tag{A.11}$$

If the failure rate is measured in units of failures/hour, MTBF has units of hours.

A.3 NMR Reliability

The reliability equation for the fault tolerance architecture NMR [Lal01] is:

$$R_{NMR} = \sum_{i=0}^{n} \binom{N}{i} (1 - R_M)^i \, R_M^{N-i} \qquad \text{where } n = (N-1)/2 \tag{A.12}$$

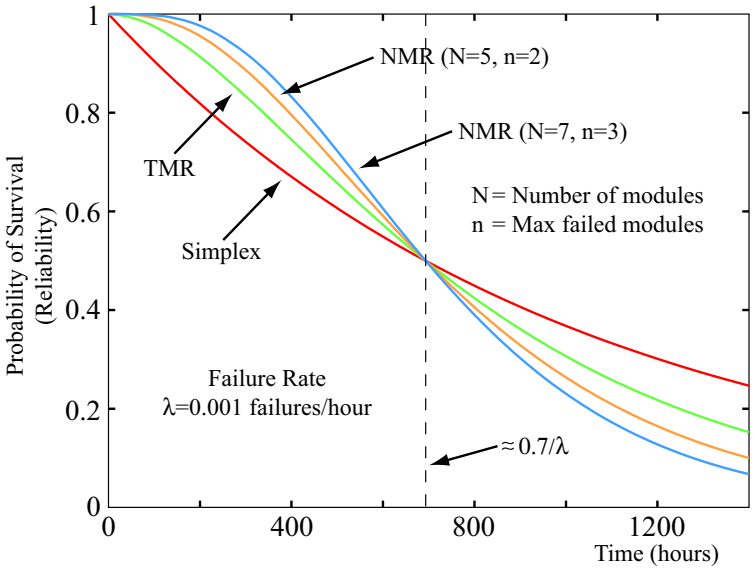


Figure A.3: *Reliability of the TMR architecture for various numbers of redundant modules. The reliability curve for a simplex system is included for comparison.*

Where N is the total number of modules (usually an odd number), n is the number of failed systems that may be successfully masked, and R_M is the reliability of the simplex system. Using this formula it is possible to derive the reliability of the TMR form of NMR, when $N = 3$ and $n = 1$

$$R_{TMR} = 3R_M^2 - 2R_M^3 \qquad \text{when } N = 3, n = 1 \tag{A.13}$$

By substituting the standard exponential failure law, Appendix A.1, for the simplex system we can derive the reliability of TMR in terms of time.

$$R_{TMR}(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t} \qquad \text{when } R_M(t) = e^{-\lambda t} \tag{A.14}$$

The same process may be used to find the reliability of NMR when $N = 5$:

$$R_{NMR} = 6R_M^5 - 15R_M^4 + 10R_M^3 \qquad \text{when } N = 5, n = 2 \tag{A.15}$$

$$R_{NMR}(t) = 6e^{-5\lambda t} - 15e^{-4\lambda t} + 10e^{-3\lambda t} \qquad \text{when } R_M(t) = e^{-\lambda t} \tag{A.16}$$

and when $N = 7$:

$$R_{NMR} = -20R_M^7 + 70R_M^6 - 84R_M^5 + 35R_M^4 \qquad \text{when } N = 7, n = 3 \tag{A.17}$$

$$R_{NMR}(t) = -20e^{-7\lambda t} + 70e^{-6\lambda t} - 84e^{-5\lambda t} + 35e^{-4\lambda t} \qquad \text{when } R_M(t) = e^{-\lambda t} \tag{A.18}$$

Plots of all three NMR reliability curves compared to that of the simplex system may be seen in Figure 2.15, reproduced in Figure A.3 for clarity.

A.4 Hybrid Redundancy Reliability

The general form of the reliability equation for the hybrid redundancy architecture depicted in Figure 2.17 is shown in Equation A.19 [Lal85]. S is the number of spare modules, n is the number of modules that may simultaneously fail and N is the number of modules in the NMR structure. It is assumed that the voter, switch and disagreement detector will not fail.

$$R_{Hybrid} = \sum_{i=0}^{n+S} \binom{N+S}{i} (1 - R_M)^i \, R_M^{N+S-i} \qquad \text{where } N = 2n + 1 \tag{A.19}$$

For a hybrid redundancy architecture using TMR and 2 spares (N=3, S=2), the reliability is given by:

$$R_{Hybrid} = 1 - (1 - R_M)^4 \, (1 + 4R_M) \tag{A.20}$$

Substituting the exponential failure law gives:

$$R_{Hybrid}(t) = 1 - (1 - e^{-\lambda t})^4 \, (1 + 4e^{-\lambda t}) \qquad \text{when } R_M(t) = e^{-\lambda t} \tag{A.21}$$

and for 3 spares (N=3, S=3):

$$R_{Hybrid} = 1 - (1 - R_M)^5 \, (1 + 5R_M) \tag{A.22}$$

$$R_{Hybrid}(t) = 1 - (1 - e^{-\lambda t})^5 \, (1 + 5e^{-\lambda t}) \qquad \text{when } R_M(t) = e^{-\lambda t} \tag{A.23}$$

Figure 2.18, reproduced in Figure A.4, show plots of the change in reliability as a function of time for these two cases.

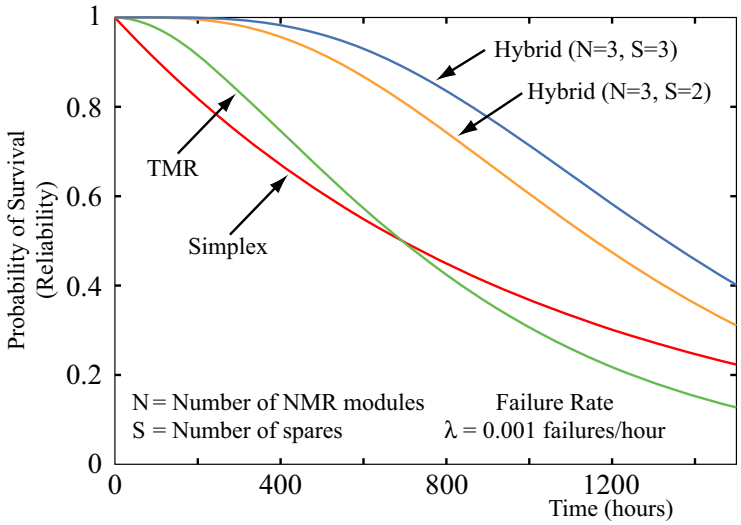


Figure A.4: *This graph show the change in reliability with time of two hybrid redundancy systems compared to that of simplex and TMR systems.*

A.5 Intel® Microprocessor Data

Intel® has a long history of manufacturing microprocessors. An analysis of how the features of their microprocessor have changed with time provides an interesting insight into the development of intergrated circuits in general. A summary of transitor quantity, feature size and clock speed for selected processors may be found in Table A.1, [Intb].

Date Introduced	Device	Clock Speed (MHz)	Feature Size (µm)	Transistor Quantity
Nov 1971	4004	0.8	10.0	2 300
Apr 1972	8008	0.8	10.0	3 500
Apr 1974	8080	2	6.0	4 500
Mar 1976	8085	5	3.0	6 500
Jun 1978	8086	10	3.0	29 000
Feb 1982	80286	12	1.5	134 000
Apr 1989	80386DX	33	1.0	275 000
Apr 1991	80486DX	50	1.0	1 200 000
Mar 1993	Pentium	66	0.8	3 100 000
Mar 1994	Pentium	100	0.6	3 200 000
Mar 1995	Pentium	120	0.6	3 200 000
Jun 1995	Pentium	133	0.35	3 300 000
Jan 1997	Pentium	200	0.35	4 500 000
May 1997	Pentium II	300	0.35	7 500 000
Apr 1998	Pentium II	400	0.25	7 500 000
May 1999	Pentium III	550	0.25	9 500 000
Oct 1999	Pentium III	733	0.18	28 000 000
Mar 2000	Pentium III	1 000	0.18	28 000 000
Nov 2000	Pentium 4	1 500	0.18	42 000 000
Aug 2001	Pentium 4	2 000	0.13	55 000 000
Apr 2002	Pentium 4	2 400	0.13	55 000 000
Apr 2003	Pentium 4	3 000	0.13	55 000 000
Feb 2004	Pentium 4	3 400	0.09	125 000 000

Table A.1: *A feature history of Intel® microprocessors*

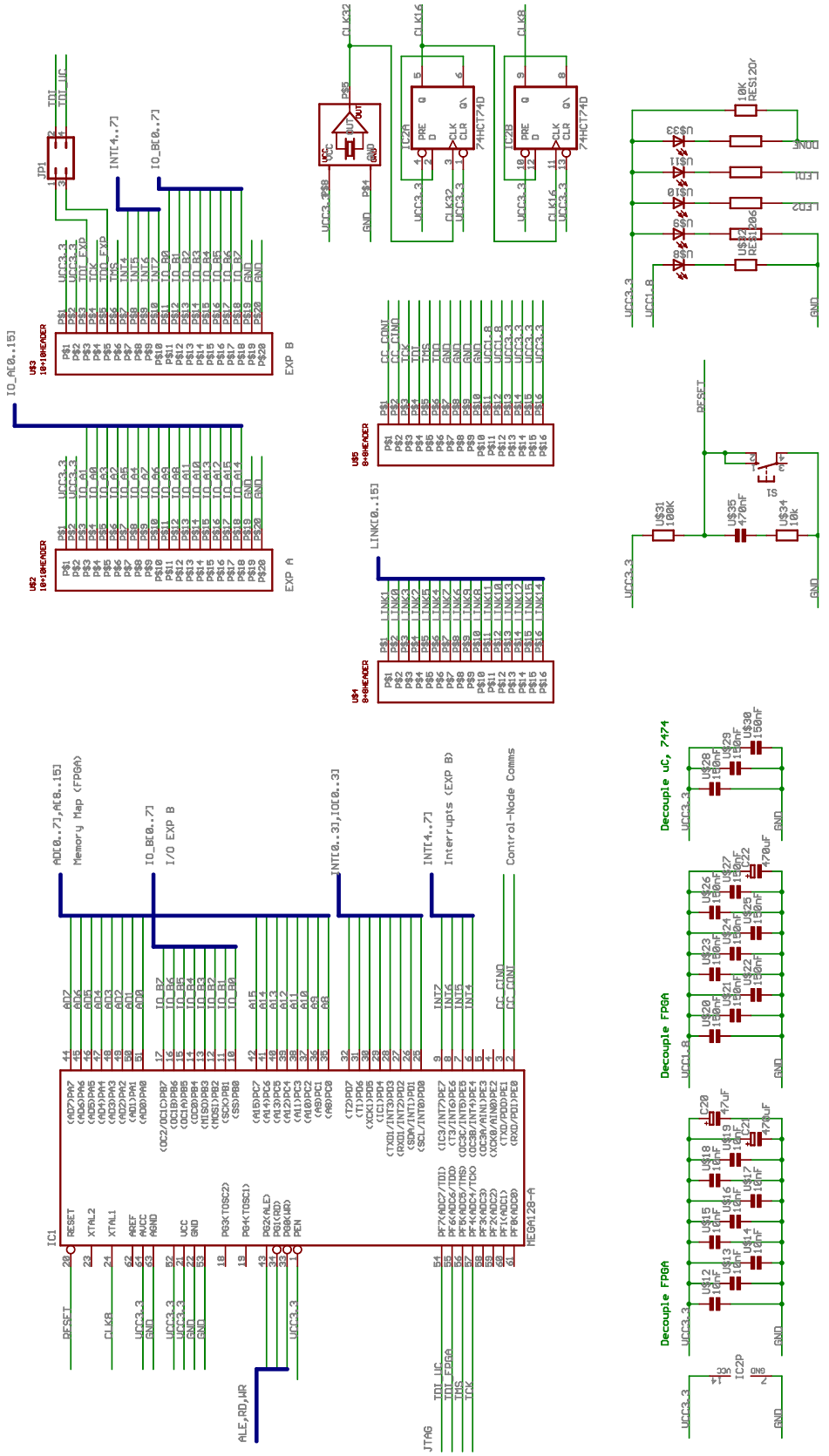
Appendix B

Circuit Diagrams

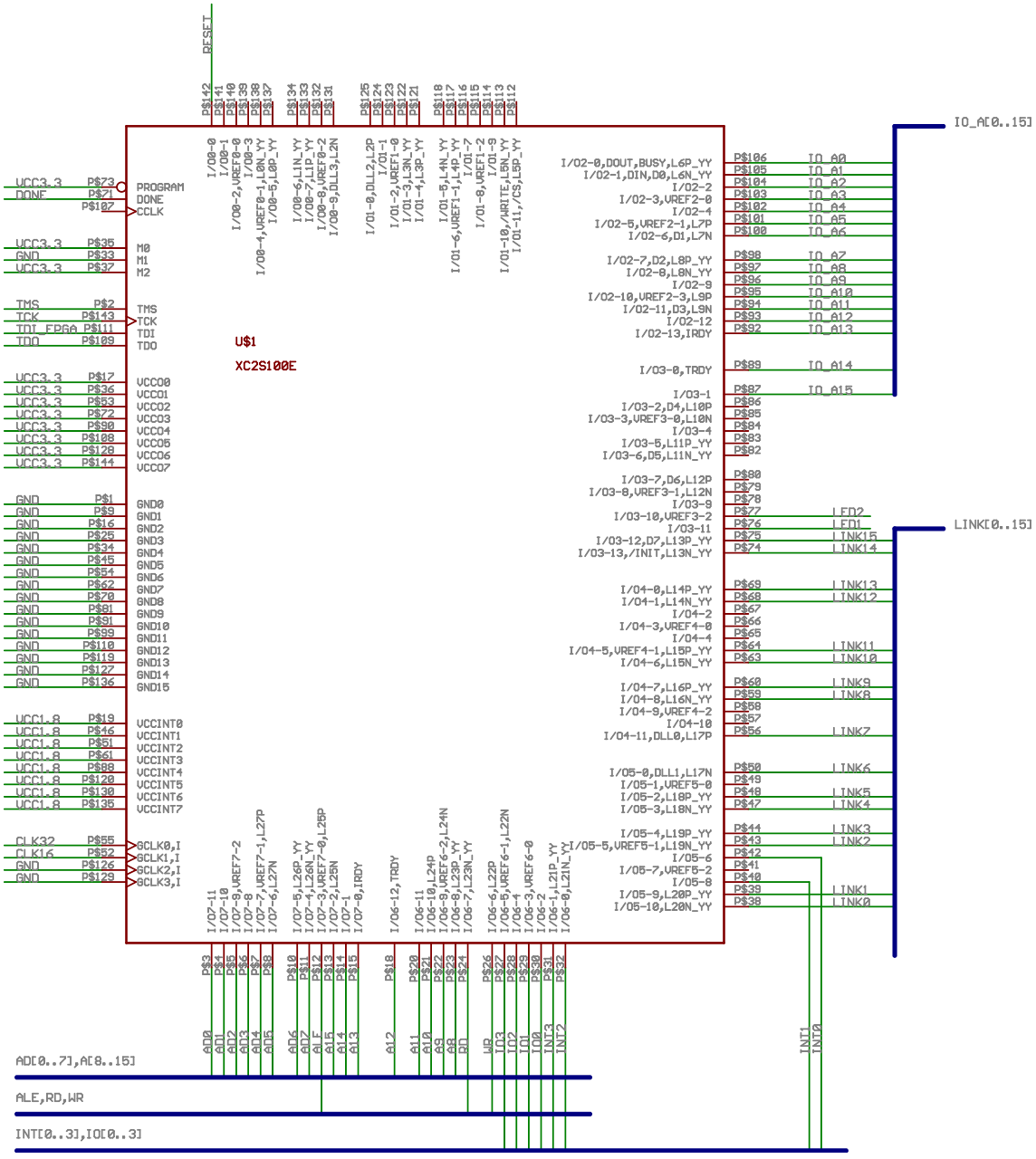
This appendix contains copies of the schematic and PCB layouts of the bioNode hardware. Table B.1, summarises these.

Circuit	Detail	Number
BioNode	Schematic	Circuit B.1 and B.2
	PCB Top Layer	Circuit B.3
	PCB Bottom Layer	Circuit B.4
	Pin Connections	Circuit B.5
Interconnect Board	PCB Bottom Layer	Circuit B.6
	Pin Connections	Circuit B.7
JTAG Multiplexer	Schematic	Circuit B.8
	PCB Top Layer	Circuit B.9
	PCB Bottom Layer	Circuit B.10
JTAG Controller	Schematic	Circuit B.11
	PCB Top Layer	Circuit B.12
	PCB Bottom Layer	Circuit B.13
Power Regulator	Schematic	Circuit B.15
	PCB Top Layer	Circuit B.16
	PCB Bottom Layer	Circuit B.17

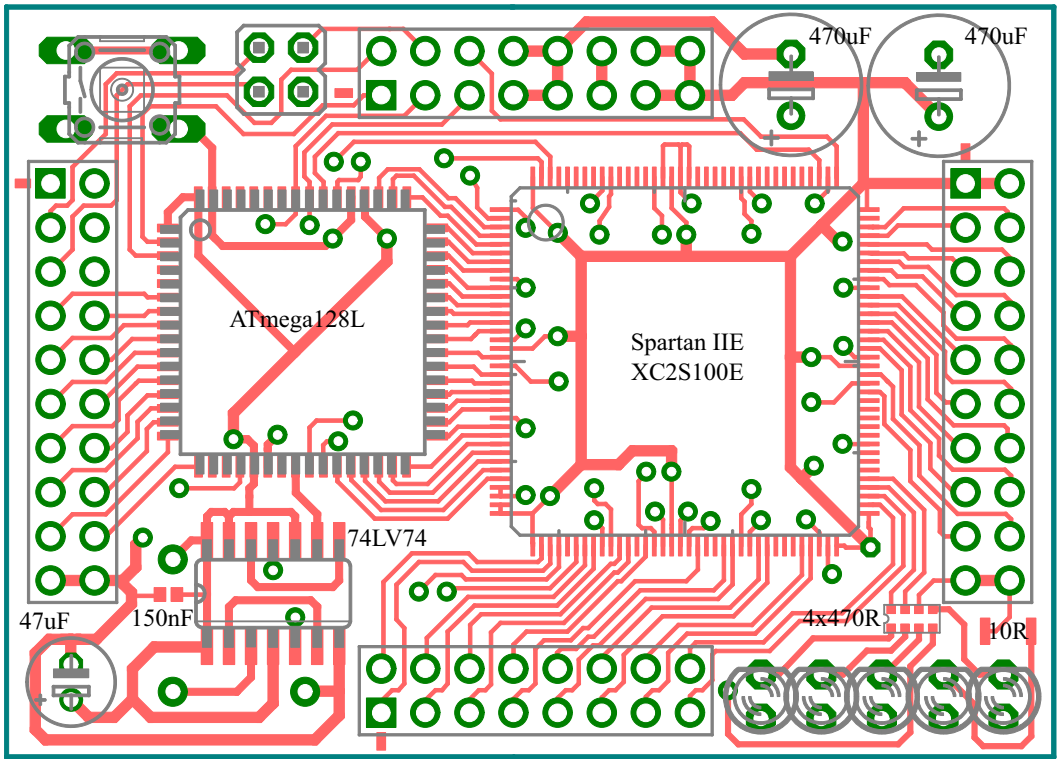
Table B.1: *An index of the BioNode System Circuit Diagrams and PCB Layouts.*



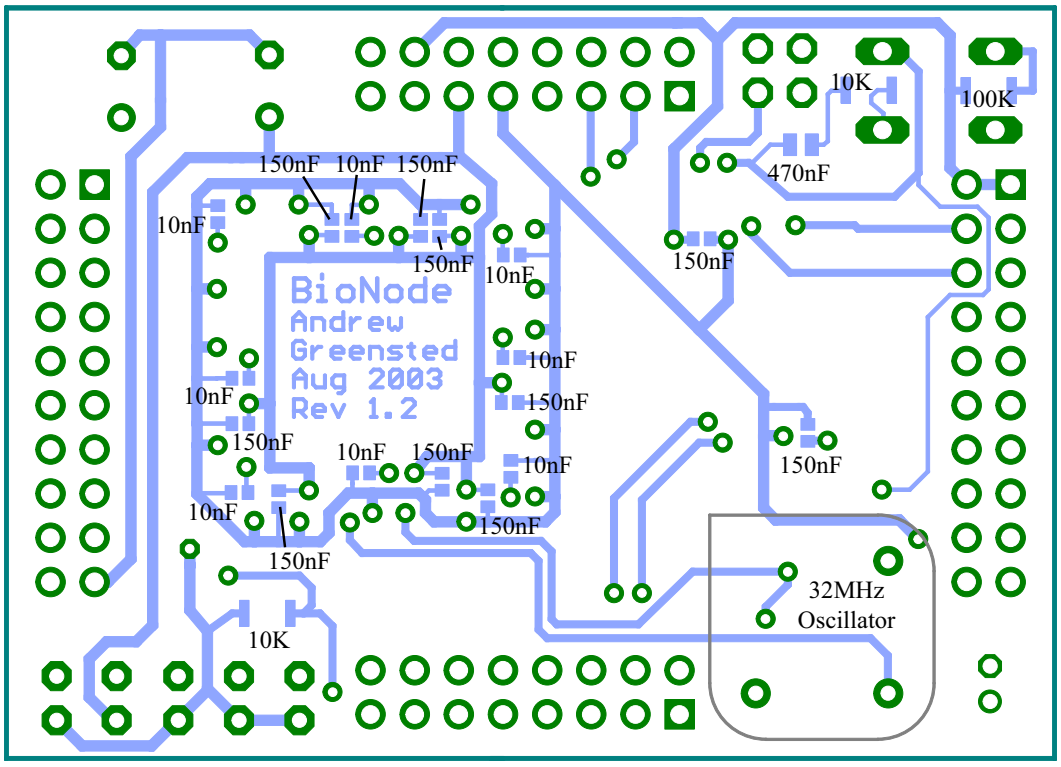
Circuit B.1: *The first of the two bioNode circuit diagrams. This contains the Atmel® AVR microcontroller, and the supporting hardware.*



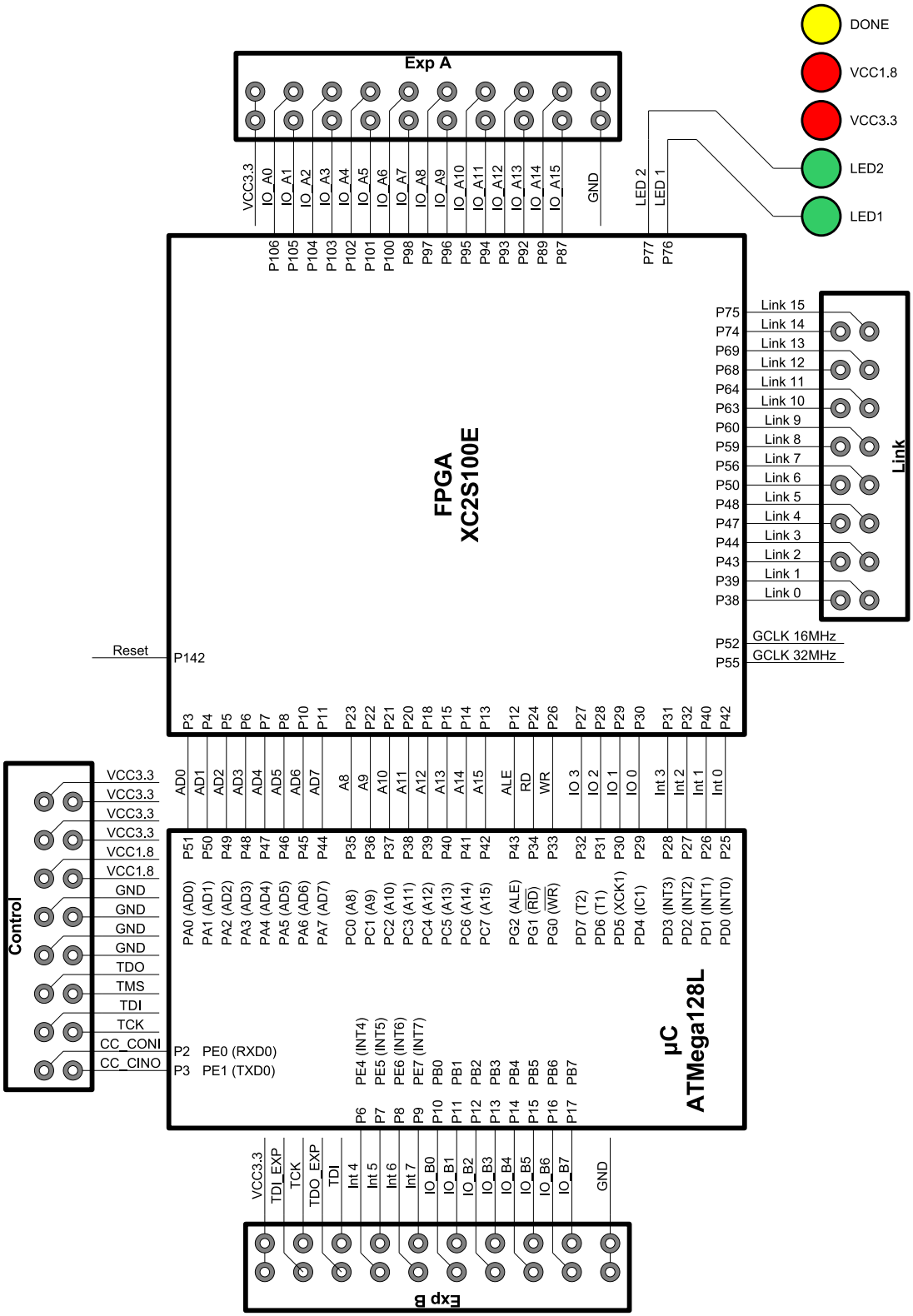
Circuit B.2: *The second bioNode circuit Diagram. This shows the connections for the Xilinx® Spartan™ IIE FPGA.*



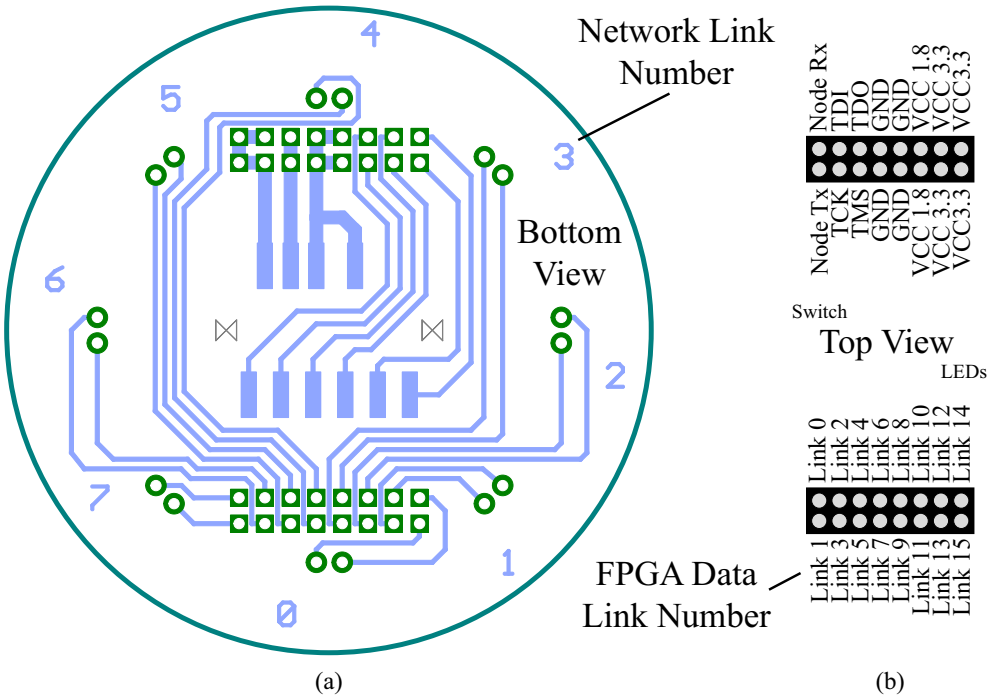
Circuit B.3: *The top layer of the bioNode PCB. The main features are; the FPGA to the right, the microcontroller to the left, the board reset switch on the top left, the clock divider on the bottom left and the power and debug LEDs on the bottom right.*



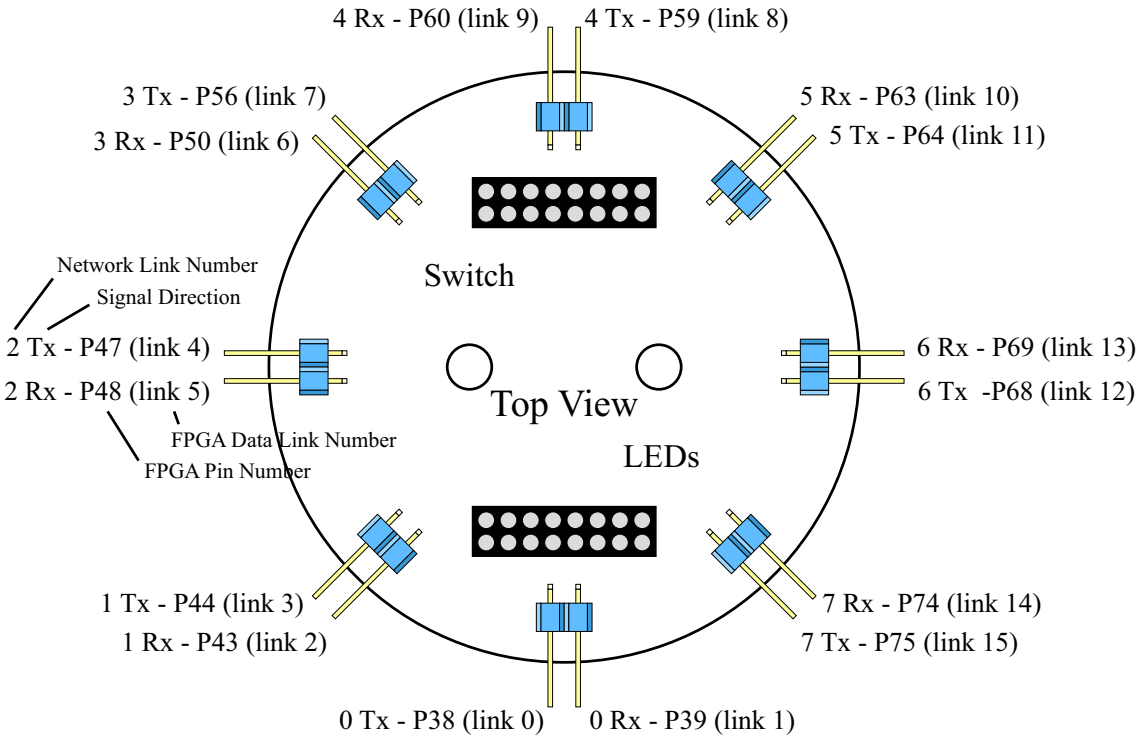
Circuit B.4: *The bottom layer of the bioNode PCB. The main features are; the main clock oscillator on the bottom right and the FPGA decoupling resistors to the left.*



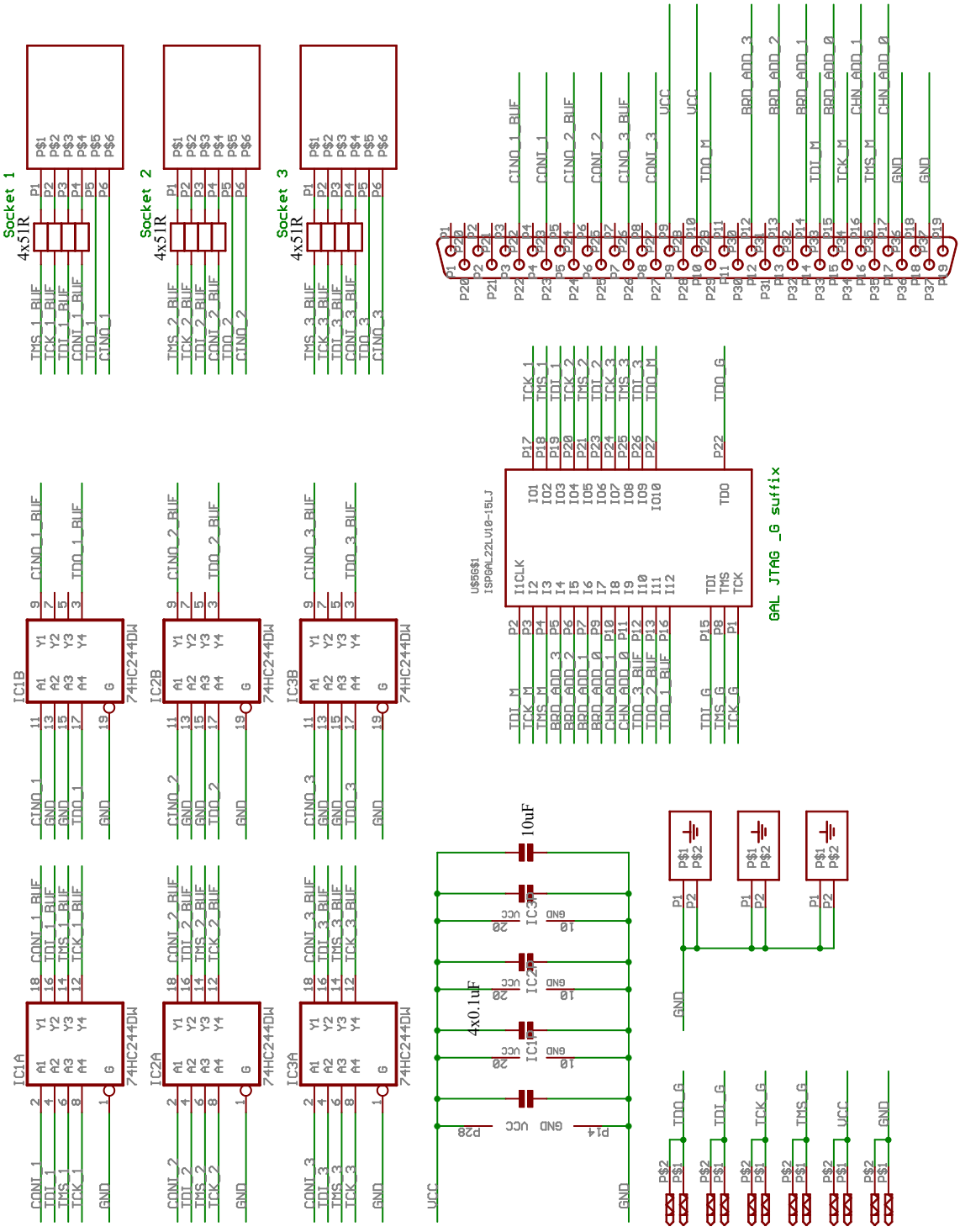
Circuit B.5: *The pin numbers of the connections between the main bioNode module devices.*



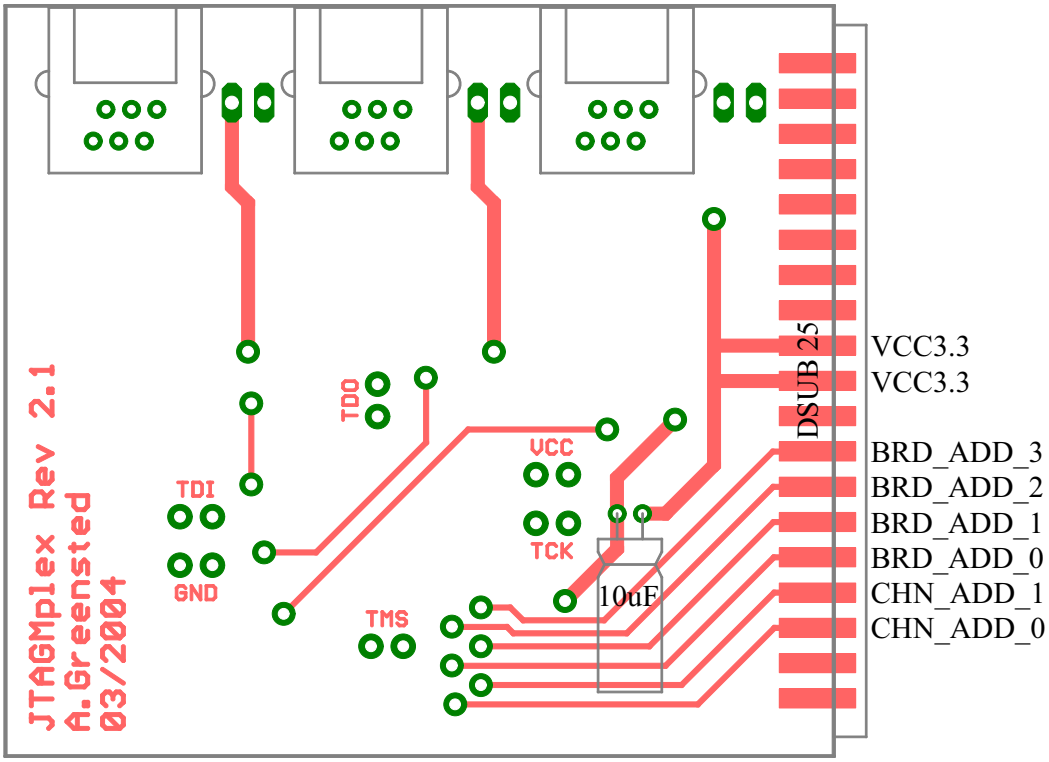
Circuit B.6: *(a)The bottom layer of the bioNode network interconnection board. This PCB shows the umbilical connection pads at the boards centre. The numbers refer to the bioNode network link numbers. (b) The interconnection board connection assignments.*



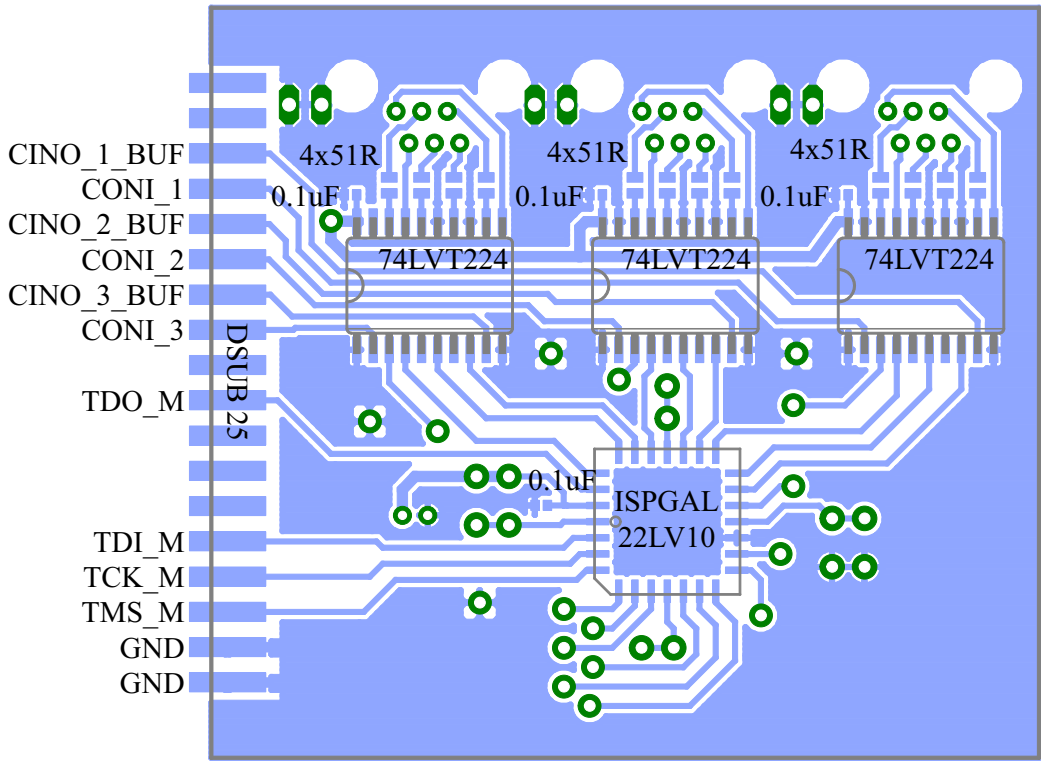
Circuit B.7: *The pin assignment of the bioNode network interconnection board. This view, from the top, shows the eight network connections, and how each link relates to the sixteen FPGA data lines. The 'switch' and 'LEDs' labels show the alignment of the connecting bioNode.*



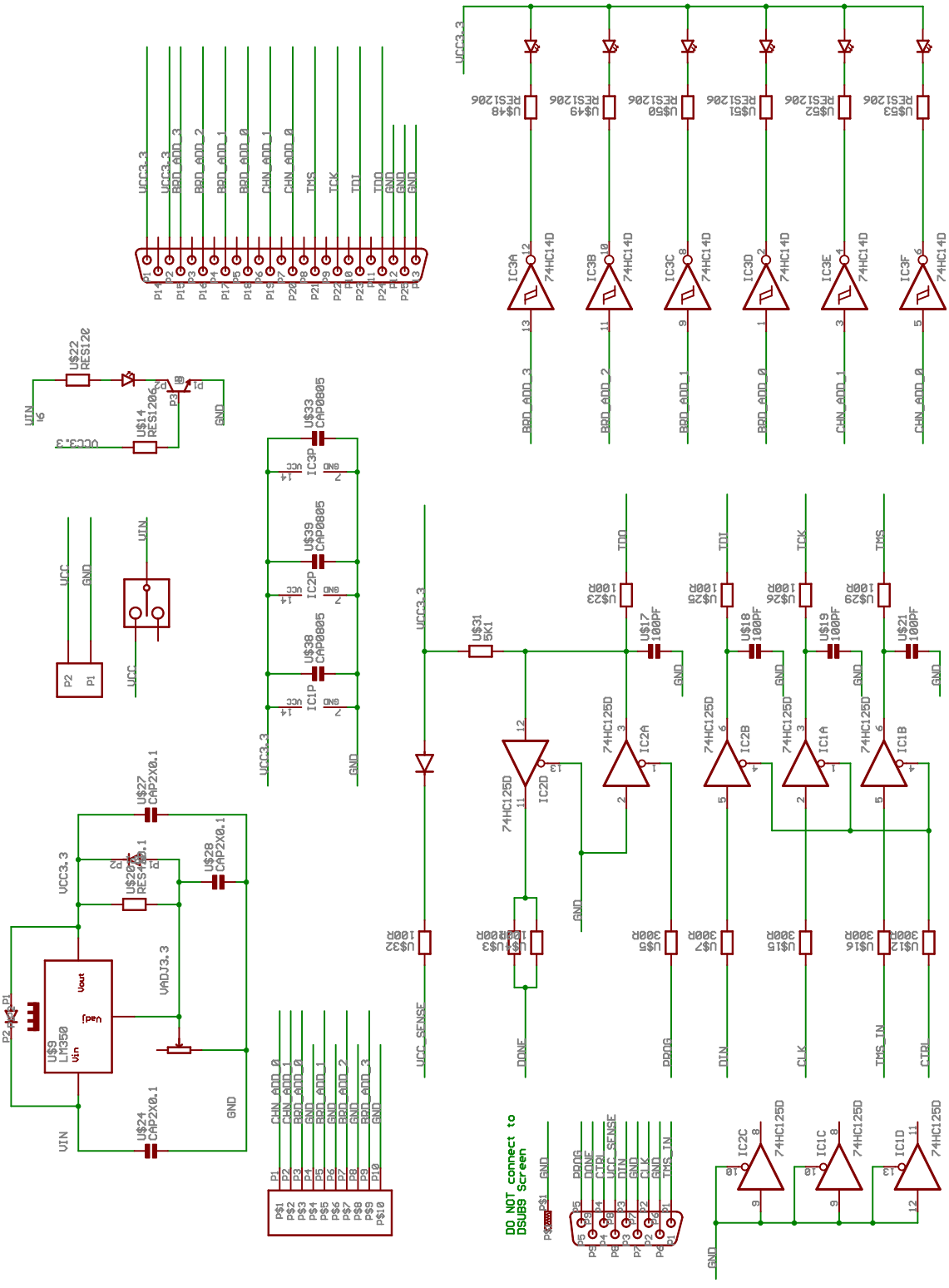
Circuit B.8: *The JTAG multiplexer and the system communications driver circuit. Ten of these circuits channel the JTAG programming signals to the correct bioNode, and provide drivers for the system communications serial lines.*



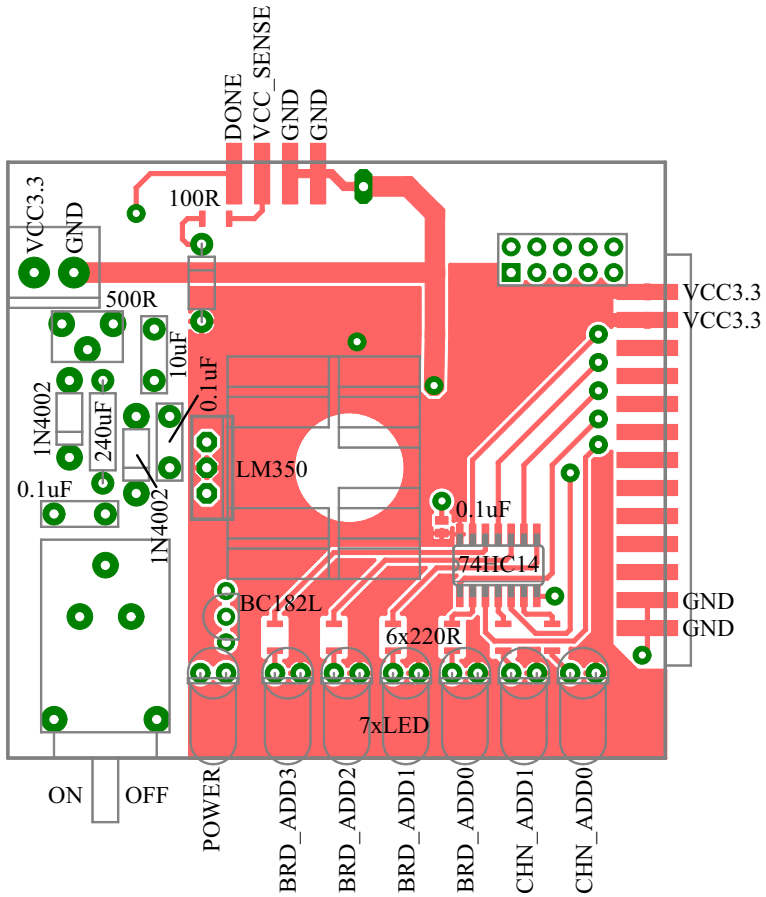
Circuit B.9: *The top Layer of the JTAG and System comms PCB. The main features are the three umbilical sockets that connect to individual bioNodes.*



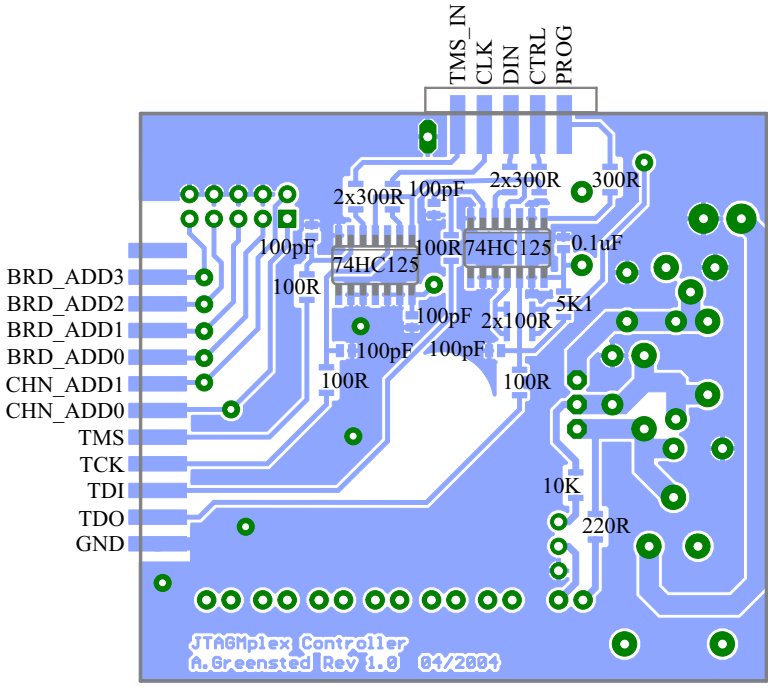
Circuit B.10: *The bottom Layer of the JTAG and System comms PCB. The main features are; the three line driver ICs, one for each bioNode connection, and in the centre a reprogrammable GAL that controls the channelling of the JTAG signals to and from the bioNodes. The large edge connector provides power, control and signal lines.*



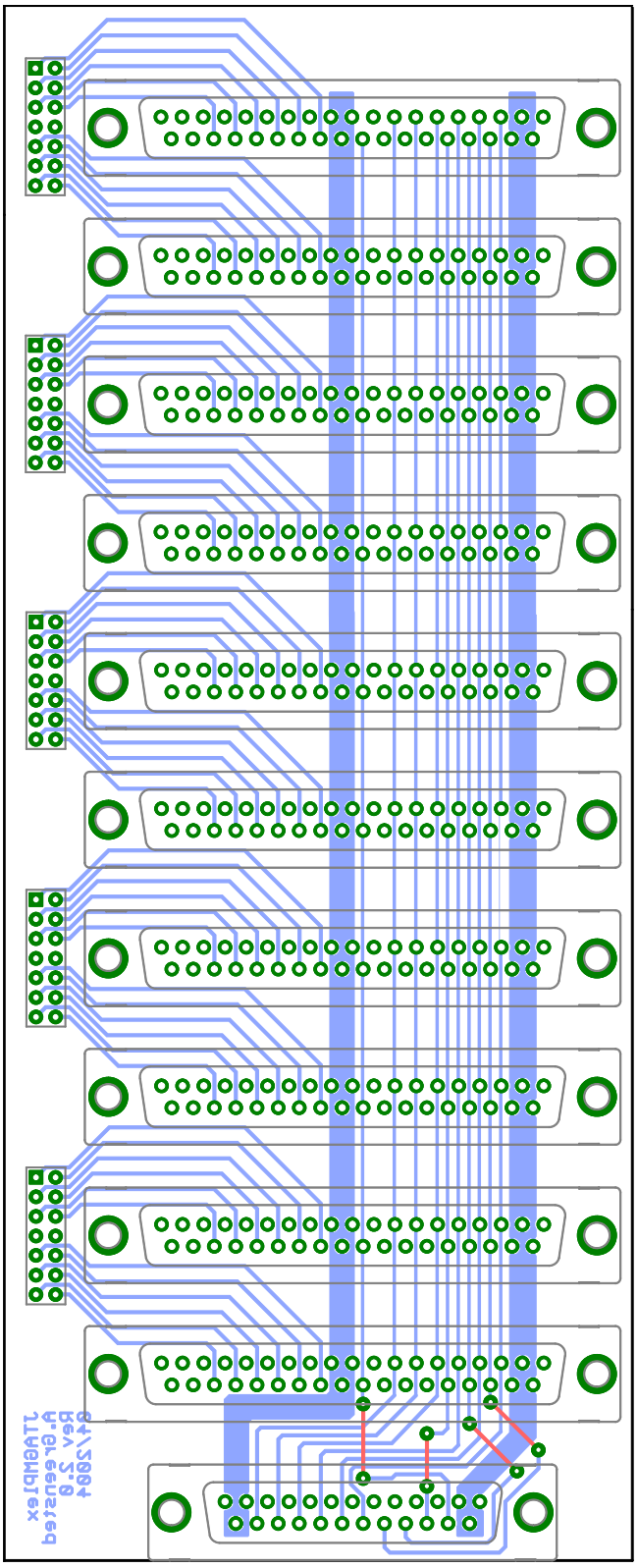
Circuit B.11: *The JTAG controller schematic. This circuit provides power regulation for the all the JTAG multiplexer and system communication line driver boards.*



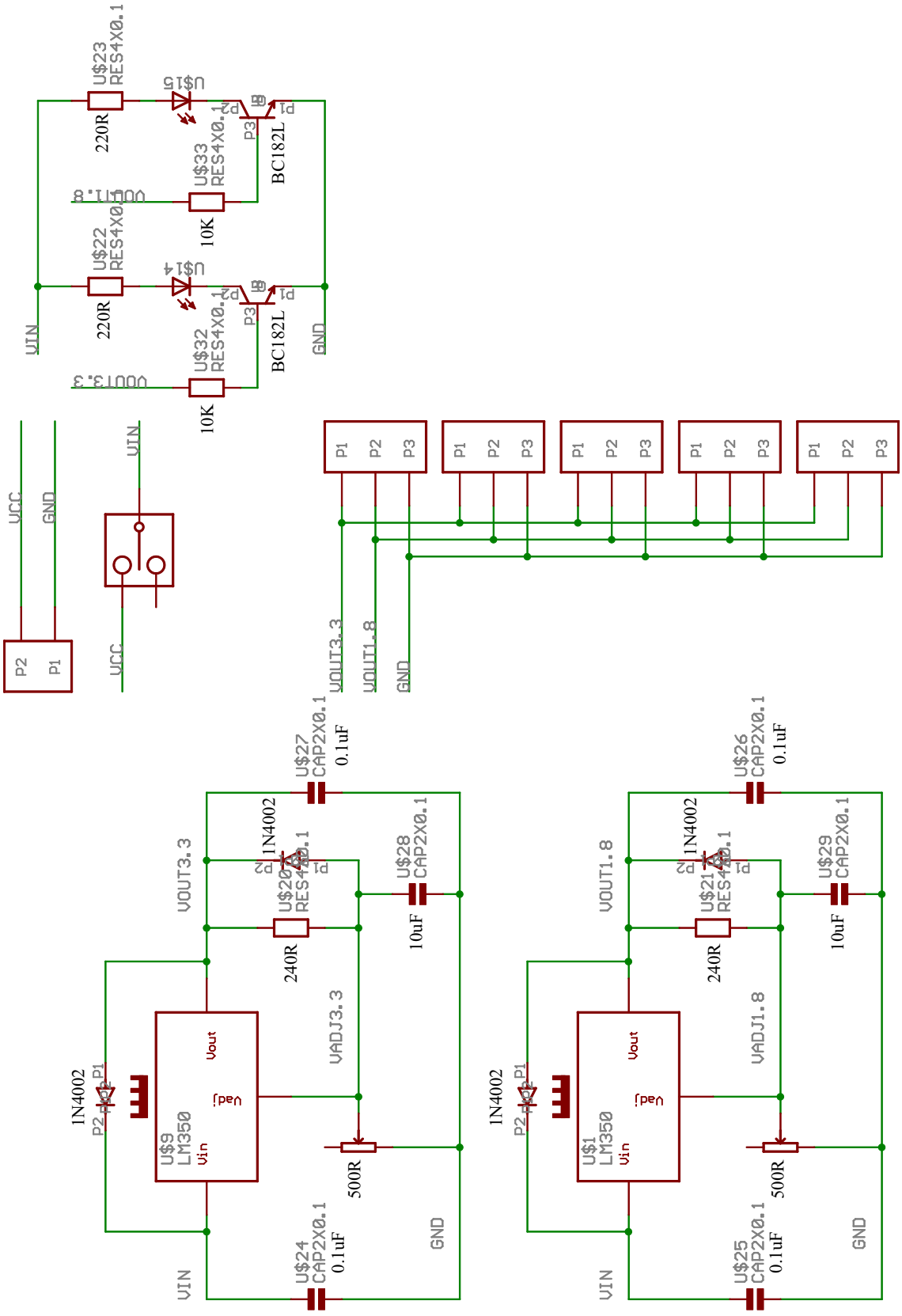
Circuit B.12: *The top layer of the JTAG controller PCB. The main features are; the top edge connector that connects the JTAG signals to the programming PC and the edge connector to the right which connects to the system back plane.*



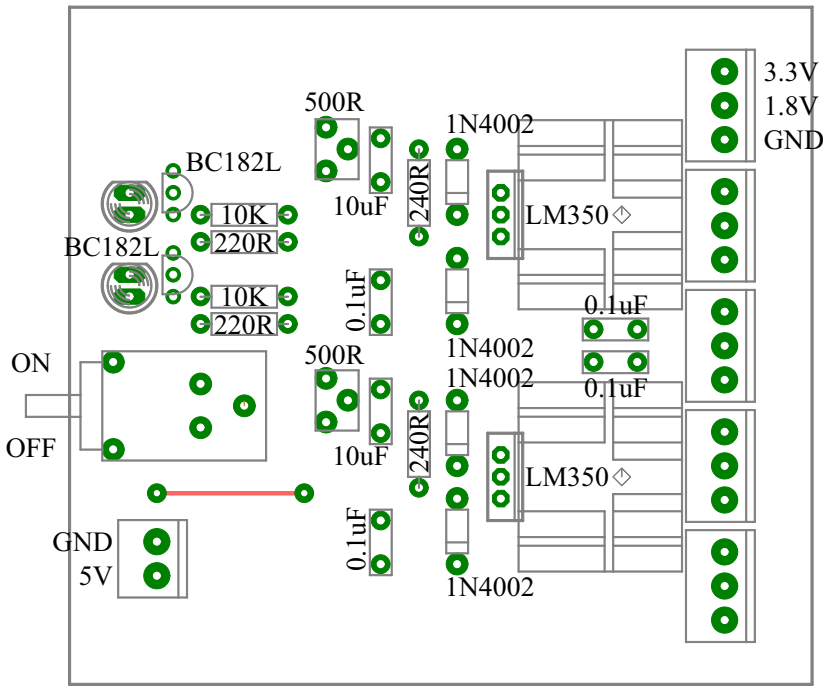
Circuit B.13: *The bottom layer of the JTAG controller PCB.*



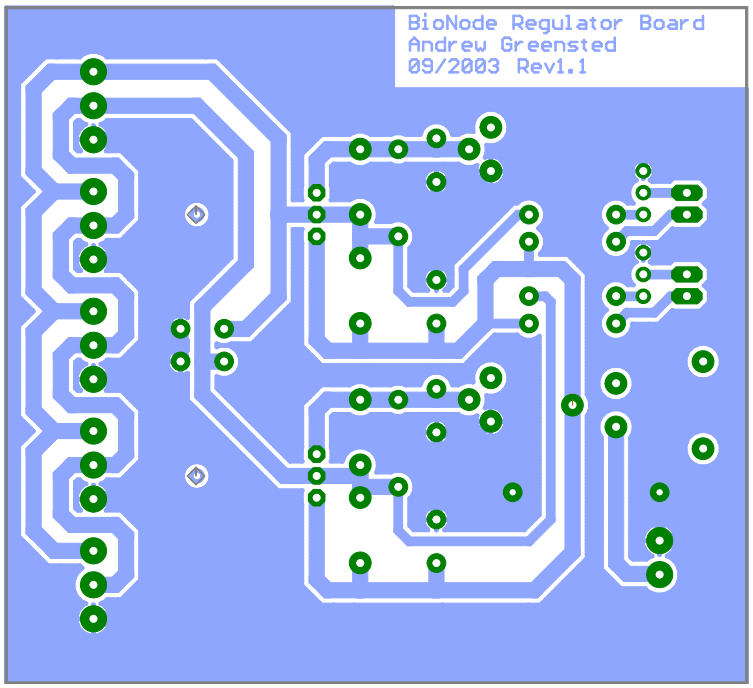
Circuit B.14: *Top view of the JTAG and system communication board back plane. The bottom most connector is for the JTAG controller board, the remaining ten connectors are for the JTAG multiplexer and system communication driver boards. The five connectors along the left hand side connect the system communication serial lines to the Virtex™ FPGA multiplexer system, see Section 5.3.1*



Circuit B.15: *The power regulator schematic. This circuit provides power regulation for the five system bioNodes. Two voltages are supplied one main 3.3V line and a 1.8V line for the FPGA core.*



Circuit B.16: *The top layer of the power regulator PCB. The main features are; the five power outputs on the right and the main power input on the left.*



Circuit B.17: *The bottom layer of the power regulator PCB.*

Appendix C

BioNode System Datasheets

This appendix includes operational data for those who wish to make use of the bioNode system.

C.1 Device Specification Overview

Specification	Value
Program Memory (Flash)	128 Kbytes
EEPROM	4 Kbytes
Static RAM	4 Kbytes
Max IO	53 pins
Maf operating Frequency	8MHz
Operating Voltage	2.7V - 5.5V

Table C.1: *A summary of the Atmel® ATMega128L microcontroller specification. A full description may be found within the referenced datasheet [Atm04].*

Specification	Value
System Gates	100 000
Logic Cells	2700
Block RAM	40 Kbits
Max IO	202 pins
IO (TQ144 Package)	102 pins

Table C.2: *A summary of the Xilinx® Spartan™ IIE XC2S100E FPGA specification. A full description may be found within the referenced datasheet [Xil03].*

C.2 BioNode Communication Link Manager

Table C.3 is a full memory map of the bioNode link manger.

Address (Hex)	Address (Binary)	Function
0x2000 ► 0x21FF	10 0000 0000 0000 ► 10 0001 1111 1111	Link Unit 0 Access
0x2200 ► 0x23FF	10 0010 0000 0000 ► 10 0011 1111 1111	Link Unit 1 Access
0x2400 ► 0x27FF	10 0100 0000 0000 ► 10 0101 1111 1111	Link Unit 2 Access
0x2600 ► 0x27FF	10 0110 0000 0000 ► 10 0111 1111 1111	Link Unit 3 Access
0x2800 ► 0x29FF	10 1000 0000 0000 ► 10 1001 1111 1111	Link Unit 4 Access
0x2A00 ► 0x2BFF	10 1010 0000 0000 ► 10 1011 1111 1111	Link Unit 5 Access
0x2C00 ► 0x2DFF	10 1100 0000 0000 ► 10 1101 1111 1111	Link Unit 6 Access
0x2E00 ► 0x2FFF	10 1110 0000 0000 ► 10 1111 1111 1111	Link Unit 7 Access
0x3000	11 0000 0000 0000	FPGA Configuration check number (Read Only)
0x3001	11 0000 0000 0001	Tx Unit Free Flags (Read Only)
0x3002	11 0000 0000 0010	Link Established Flags (Read Only)
0x3003	11 0000 0000 0011	Rx Data Waiting Flags (Read Only)
0x3004	11 0000 0000 0100	Remote Rx Buffer Full Flags (Read Only)
0x3005	11 0000 0000 0101	Random Number Generator Output (Read Only)

Table C.3: *The bioNode communication system memory map. The FPGA general purpose data lines are used in pairs to create eight inter-bioNode link, each controlled by a Link Unit. These units are memory mapped onto the microcontroller.*

Address (Hex)	Address (Binary)	Read Function	Write Function
0x2000 ► 0x203F	10 nnn0 0000 0000 ► 10 nnn0 0011 1111	Direct access to Rx Buffer 0 Data	-
0x2040 ► 0x207F	10 nnn0 0100 0000 ► 10 nnn0 0111 1111	Direct access to Rx Buffer 1 Data	-
0x2080 ► 0x20BF	10 nnn0 1000 0000 ► 10 nnn0 1011 1111	Direct access to Rx Buffer 2 Data	-
0x20C0 ► 0x20FF	10 nnn0 1100 0000 ► 10 nnn0 1111 1111	Direct access to Rx Buffer 3 Data	-
0x2100	10 nnn1 0000 0000	Link Unit Status 0	User Status Data
0x2101	10 nnn1 0000 0001	Link Unit Status 1	Link Unit Command
0x2102	10 nnn1 0000 0010	Rx Data	Tx Data
0x2103 ► 0x21FF	10 nnn1 0000 0011 ► 10 nnn1 1111 1111	-	-

Table C.4: *Each inter bioNode link is controlled by a Link Unit. Eight of these can be accessed from the AVR. This table shows the memory map of a single linkUnit, where n is the binary form of the link number.*

7	6	5	4	3	2	1	0
Rx Data Available	Tx Busy	Link Established	Current Rx Buffer Number		Local Buffer Fill Level		
7	6	5	4	3	2	1	0
Remote User Status Data					Remote Buffer Fill Level		

The contents of the link unit status registers.

Command Number (Hex)	Command	Description
0x00	Send Frame	Starts the transmission of the buffered data to the connected node
0x01	Reset Tx Pointer	Resets the Tx buffer address pointer, next write enters start of buffer
0x02	Reset Rx Pointer	Resets the Rx buffer address pointer, next read is from start of buffer
0x03	Free Rx Buffer	Release the data in the Rx Buffer, allow more frames to be received

Table C.5: *These command values are written to a link unit command address to control the units operation.*

Appendix D

Photographs

Photograph	Number
The complete bioNode System	Photo D.1
A structural plater used to create the system toroid	Photo D.2
A bioNode module in the network	Photo D.3
The network interconnection board	Photo D.4
The supporting hardware	Photo D.5
The data channelling FPGA board	Photo D.6
The power regulator unit	Photo D.7
The main system power supply	Photo D.8
The meeting of bioNode umbilicals	Photo D.9
The monitor and programming hardware	Photo D.10

Table D.1: *An index of photographs of the BioNode System.*

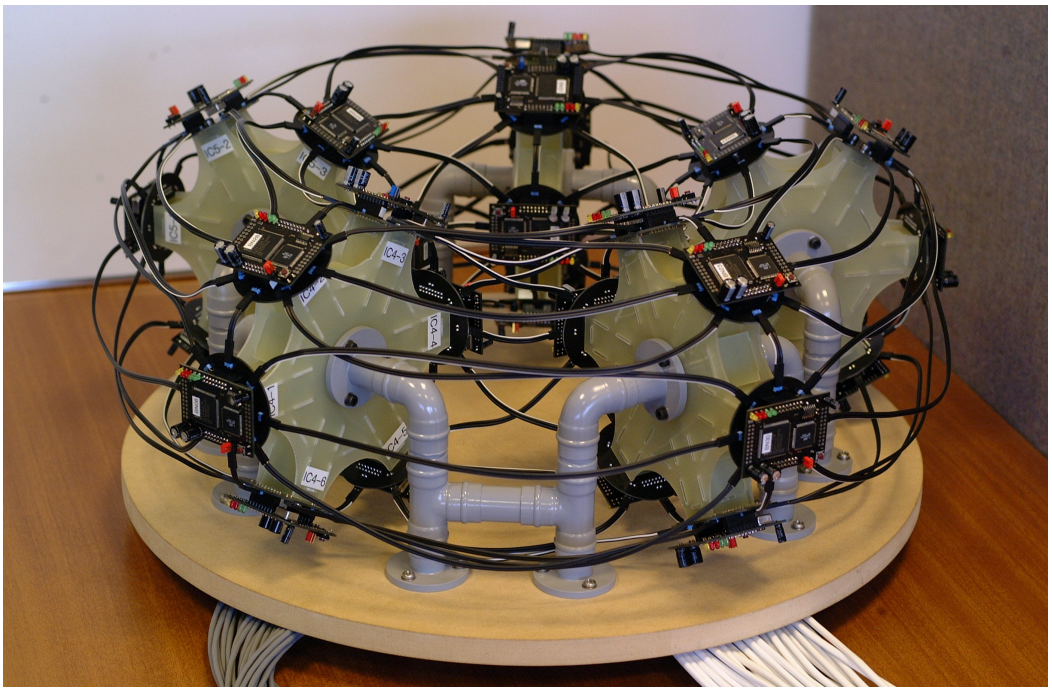


Photo D.1: *The finished bioNode system.*

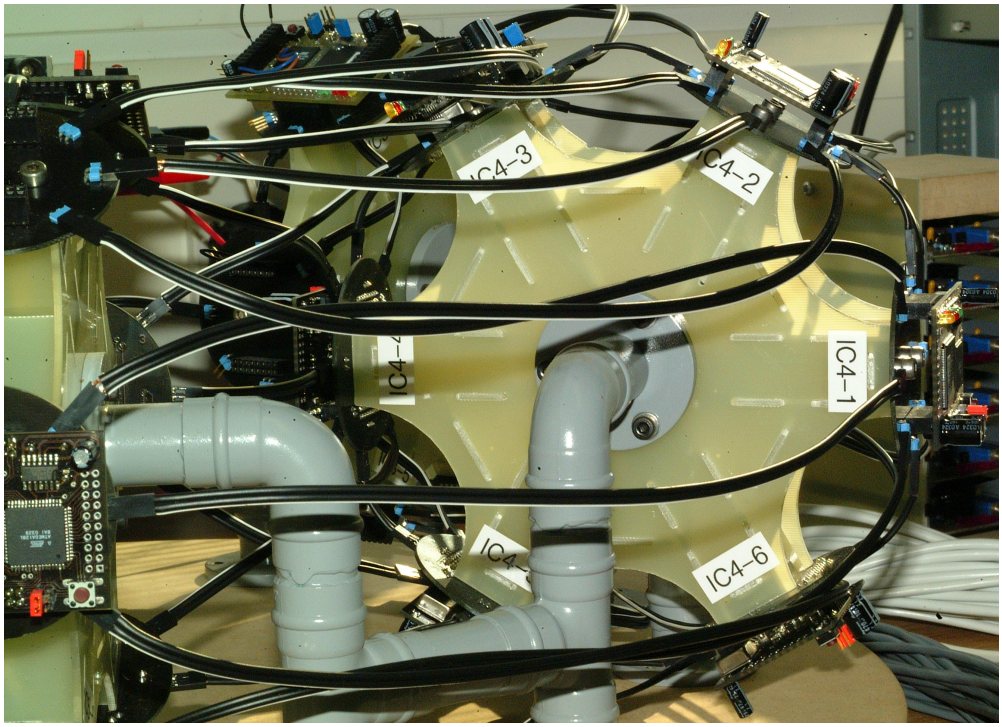


Photo D.2: *A close up of one of the platers that form the bioNode system's toroidal shape.*

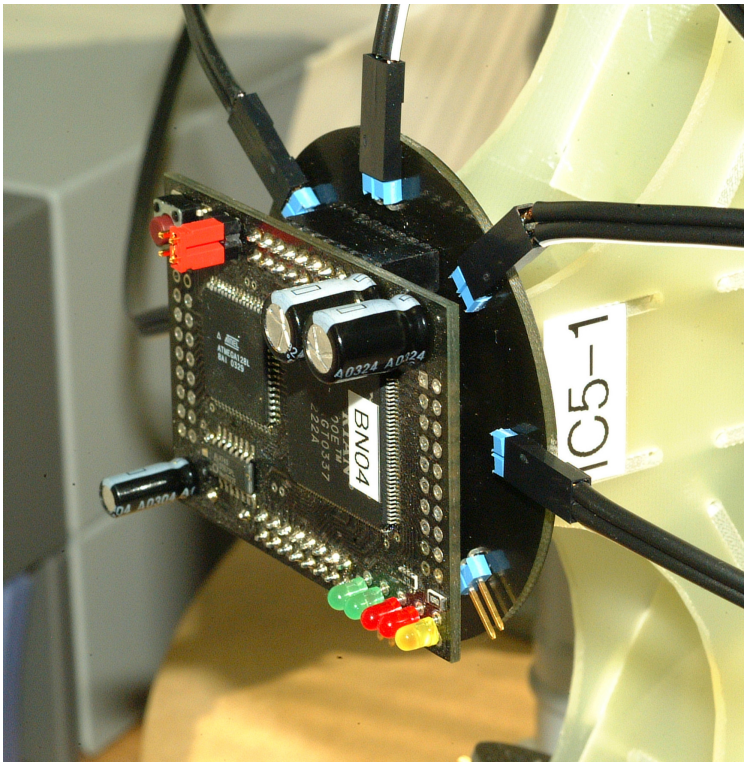


Photo D.3: *A single bioNode module plugged into the system network*

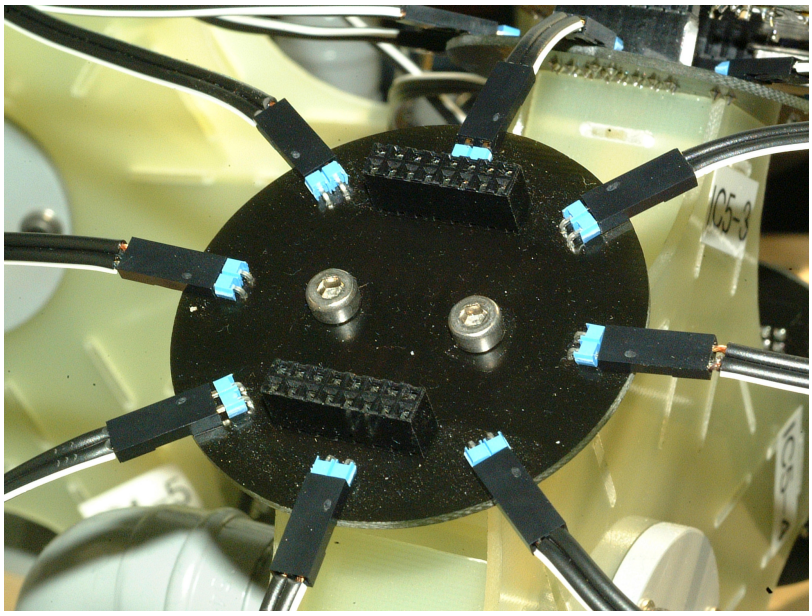


Photo D.4: *The interconnecting board that allows bioNodes to be removed without disrupting the underling network, and also enables network interconnects to be removed.*

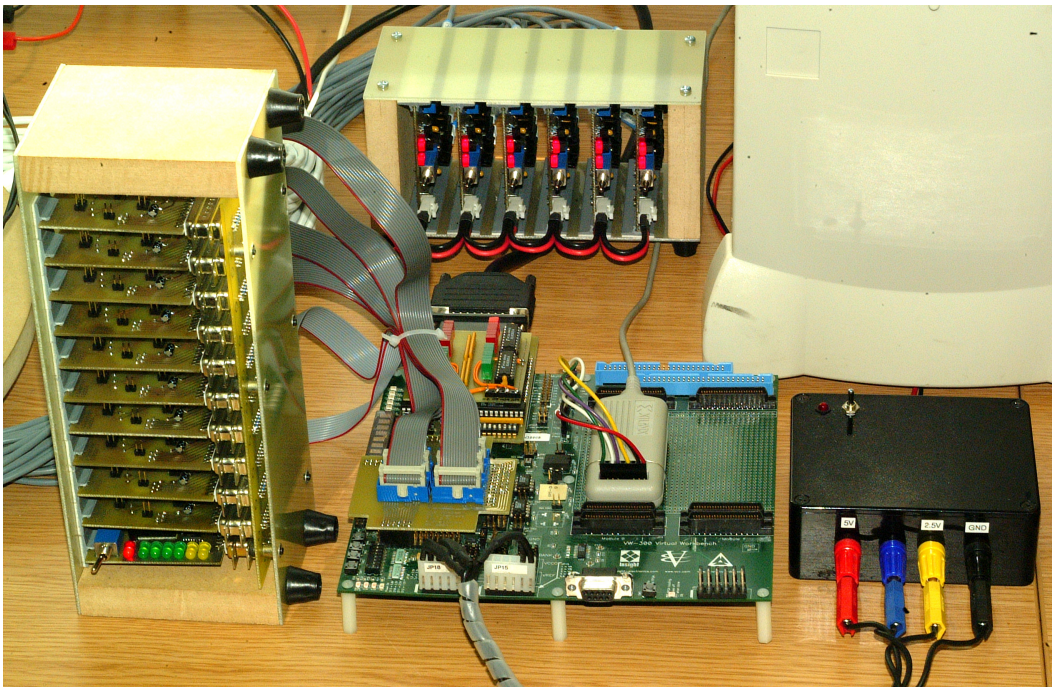


Photo D.5: *The hardware that supports the operation of the bioNode system.*

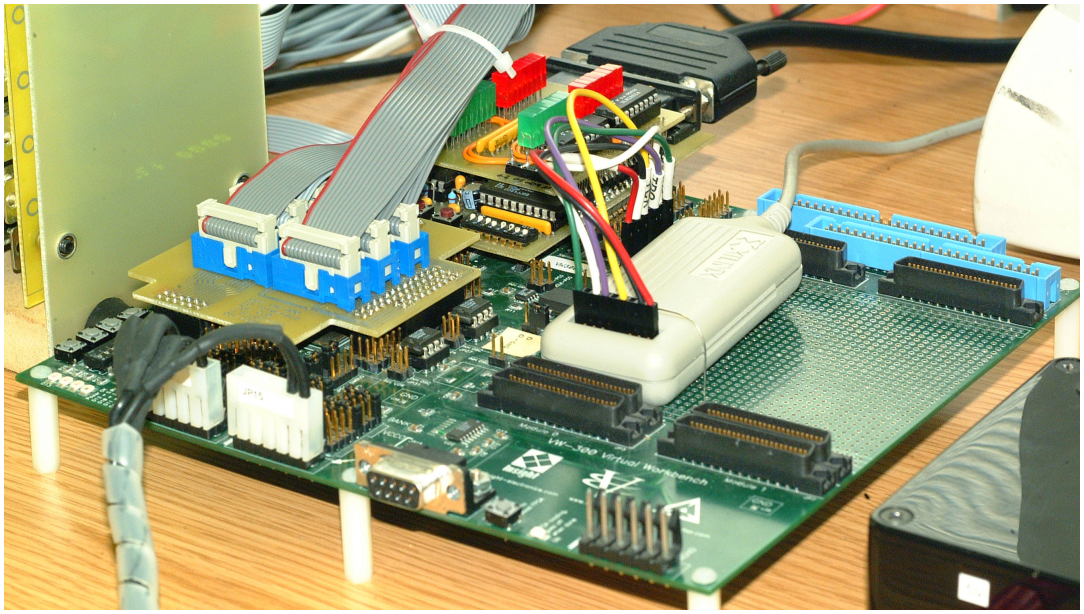


Photo D.6: *One of only two non-custom made PCBs. This FPGA development board is used to channel monitoring data between the bioNode system and the data server PC.*

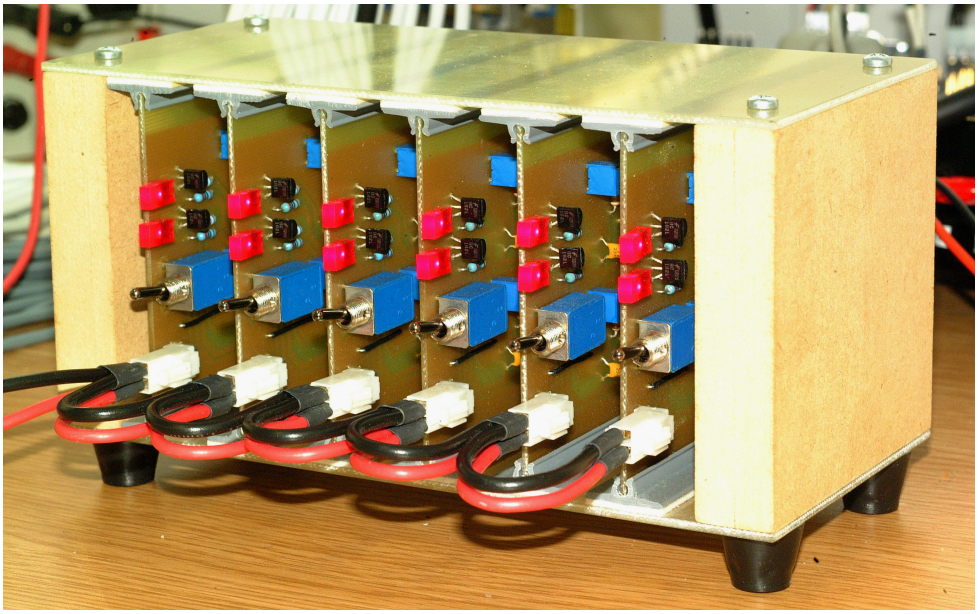


Photo D.7: *The bank of power regulator boards, that provide power to each bioNode.*

Photo D.8: *The main system power supply. This is a converted PC PSU, that provides a very high power source.*

Photo D.9: *The thirty bioNode data umbilicals connect to a bank of boards that channel monitoring and programming to and from the main supporting PC.*

Photo D.10: *The front view of the monitor and programming data channelling boards. The column of LEDs to the left show which bioNode is currently selected for programming.*

Glossary

Agonists A group of chemicals tailored to fit a specific receptor type and artificially invoke a reaction. Page 68

Antagonists A group of chemicals tailored to fit a specific receptor type and block natural signalling molecules from invoking a reaction. Page 68

Apoptosis A cell self destruct mechanism that occurs when a cell does not receive the appropriate survival signals. Page 71

Autocrine Signalling A form of cellular communication where a cell releases signalling molecules that bind back to itself. Page 61

Axon A long fibrous projection from a nerve cell that directs communication to a specific target cell. Page 64

Bio-Inspired Electronics Electronic systems whose design or operation has been inspired by biology. Page 18

Chromosomes A molecule of DNA and other structural proteins that keep it in a compact form. Page 55

Codon A small sequence of RNA, three nucleotides wide that encodes an amino acid type. Page 58

Deoxyribonucleic Acid or DNA, the genetic blueprint of living things. A double strand in a helix formation that encodes the information for organism construction and operation. Page 54

Die Overcoat A protective layer used in IC fabrication to stop moisture ingress or corrossion. Page 43

Dynamic Redundancy A fault tolerance scheme that uses reconfiguration to remove faulty circuitry from operation. Page 46

Embryonic Stem Cells A cell type that has the ability to specialise into other organism cell types. Page 58

Embryonics Electronic systems whose structure and/or operation is based upon embryo development. Page 21

Endocrine Cell The name given to a cell involved in endocrine signalling. Page 60

Endocrine Signalling A long distance and public form of inter cell communication mediated by hormones. Page 60

Evolutionary Algorithm A problem solving tool based on Darwin's theory of evolution. A survival of the fittest approach is taken to improve succesive generations of a solution population. Page 18

Evolvable Hardware Electronic hardware designed or modified by an evolutionary algorithm. Page 19

Extracellular Matrix The physical structure around cells that helps to hold cellular formations. Page 61

Failed A system is said to have failed if its operation deviates from the system's specification. Page 27

Fault A physical defect or abnormality in system circuitry. Page 28

Fault Tolerance Techniques Techniques incorporated into a system such that it may continue correct operation despite harbouring faults. Page 45

First Messenger The signalling molecule used between source and target cells, as appose to secondary messenger that operates internal to a cell. Page 59

Fitness Function When applied to evolutionary algorithms, a test that determines how closely an individual solves the required problem. Page 18

FPGA Field Programmable Gate Array, a versatile electronic device whose internal circuitry can be reconfigured to the users needs. Page 97

Gap Junction A group of openings in a cell's plasma membrane that can be used to pass signalling molecules to a connected cell. Page 63

Gene A sub sequence of DNA that determines a particular characteristic of an organism. Page 55

Genetic Algorithm A form of evolutionary algorithm that represents an individuals properties in a form similar to DNA. Page 19

Genetic Redundancy The replication of DNA sections, enabling correct gene expression despite DNA errors. Page 79

Genome The complete DNA sequence of all the chromosomes of an organism. Page 55

Glucagon A hormone that stimulates the liver to release stored glucose. Page 72

Glucogen The stored form of glucose, held in the liver. Page 74

Gluconeogenesis Metabolic Pathway A process that occurs in the liver to create glucose. Page 74

Higher Animal Mammals or other vertebrates with advanced characteristics. Page 54

Hormone The name of the messenger molecule used during endocrine signalling. Page 60

Hybrid Redundancy A combination of static and dynamic redundancy, providing both masking and removal of circuit faults. Page 46

Hydrophilic Something that readily dissolves in water. Page 66

Hydrophobic Something that does not readily dissolves in water. Page 66

Immunotronics The name for hardware electronic systems that use an artificial immune system. Page 20

Insulin A hormone that stimulates cells to take up glucose. Page 72

Intracellular Mediators The name of the signalling molecules that pass through cell gap junctions. Page 63

JTAG Joint Test Action Group, a method used both for debugging and programming integrated circuits. Page 108

Kirkendall Voids Gaps that occur in bonding wire pads due to a mismatch of metal migration rates between a pad and bond. Page 43

Lipophilic Something that will combine with lipids, but does not readily dissolves in water. Page 66

Local Mediator The name of the messenger molecule used during paracrine and autocrine signalling. Page 61

Manchester Coding A form of serial data encoding that uses logic transitions to represent data bits. Page 105

Masking Redundancy A redundunccy scheme that leads to the masking of errors caused by faults. Another name for static redundancy. Page 46

Moore's Law An informal name given to Gordon Moore's prediction that the number of transistors in ICs will double every two years. Page 17

Neuron A special type of cell that is able to communicate long distance with great speed to specific cells. Page 64

Neurotransmitters The name of the signalling molecules used in neuronal communication. They are released by the synapse, and travel very short distances to the target cell. Page 64

Nucleotide The combination of a base and a sugar-phosphate that form a single link in a single strand of DNA. Page 54

Nucleus A sub unit of a cell which contains most of its genetic material. Page 56

Paracrine Signalling A local form of inter cell communication that operates between neighbouring cells. Page 60

Phospholipid Bilayer A double layer of specially orientated phospholipids that form the outer shell of a cell. Page 67

Plasma Membrane The outer membrane of a cell that holds the cells contents together. Page 62

Pluripotent The property of stem cells to develop into any cell type within an organism. Page 59

Programmed Cell Death A cell self destruct mechanism that occurs when a cell does not receive the appropriate survival signals. Also called apoptosis. Page 71

Protein Machine In the context of DNA replication, this is the collection of proteins that are drawn to a replication fork to perform the task of DNA replication. Page 56

Receptor Protein A specially shaped protein used by cells to provide a point to which signalling molecules may bind. Page 59

Reliability Engineering A branch of engineering that deals with improving system reliability via fault handling techniques. Page 26

Replication Fork The name given to the point where two DNA strands are split during the process of replication. Page 56

Replication Origin A specific point within DNA that a replication fork may be created and replication may begin. Page 56

RISC Reduced Instruction Set Computer, a type of microprocessor that uses a small set of instructions. Page 97

RMI Remote Method Invocation, a method of transparently connecting Java software over a network. Page 108

Secondary Messengers The signalling molecules that operate internally to a cell. Normally generated by the activation of a membrane bound receptor. Page 69

Semiconservative In the context of DNA replication, this refers to the fact that the result of replication is that the two new strands both contain one old strand. Page 57

Signal Transduction The process of signal molecule conversion in the transmission of a single signalling pathway. Page 69

Signalling Cell A cell that initiates intercell communication. Page 59

Signalling Molecule One of a variety of substances that relays a message between cells. Page 59

Specificity How closely a molecule fits a receptor, the higher the specificity, the better the fit. Page 68

Static Redundancy A fault tolerance scheme that provides masking of circuit faults. Page 46

Synapse The end of the axon, where neurotransmitters are released to form the final part of neuronal communication. Page 64

Target Cell The cell on the receiving end of intercellular communication. Page 59

Totipotent The property of stem cells to develop into a whole new organism. Page 58

Transcription The conversion of DNA into mRNA that is the first stage of creating proteins from DNA information. Page 57

Translation The conversion of mRNA into proteins, the second stage of creating proteins from DNA information. Page 58

USART Universal Synchronous and Asynchronous serial Receiver and Transmitter, converts data between parallel and serial form for the serial transmission and reception of data along single, or low a quantity of wires. Page 107

Zygote The single fertilised cell that is the structural starting point of organisms . Page 54

Bibliography

[Act02] Actel. *Understanding Soft and Firm Errors in Semiconductor Devices*, December 2002.

[Ahm86] Syed Sajid AHMAD, Richard C. BLISH, Timothy J. CORBETT, Jerrold L. KING and C. Glenn SHIRLEY. *Effect of bromine in molding compounds on gold-aluminium bonds*. *IEEE Transactions on Computers, Hybrids and Manufacturing Technology*, volume 9, no. 4, pages 379–385, December 1986.

[Alb98] Bruce ALBERTS, Dennis BRAY, Alexander JOHNSON, Julian LEWIS, Martin RAFF, Keith ROBERTS and Peter WALTER. *Essential Cell Biology, An Introduction to the Molecular Biology of the Cell*. Garland Publishing, Inc., 1998. ISBN 0815329717.

[Alb02] Bruce ALBERTS, Alexander JOHNSON, Julian LEWIS, Martin RAFF, Keith ROBERTS and Peter WALTER. *Molecular Biology of the Cell*. Garland Science, 4th edition, 2002. ISBN 0815340729.

[Atm04] Atmel. *Atmel 8-bit AVR Microcontroller with 128K In-System Programmable Flash, ATmega128, ATmega128l*, 2467L-AVR-04/04 edition, 2004. URL `http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf`.

[Avi76] Algirdas AVIŽIENIS. *Fault tolerant systems*. *IEEE Transaction on Computers*, volume C-25, no. 12, pages 1304–1311, December 1976.

[Avi00] Algirdas AVIŽIENIS, Jean-Claude LAPRIE and Brian RANDELL. *Fundamental concepts of dependability*. In *Proceedings of ISW 2000. 34th Information Survivability Workshop*, pages 7–12. 2000.

[Ben99] John P BENTLEY. *Reliability and Quality Engineering*. Addison Wesley, 1999. ISBN 0201331322.

[Bla69] James R. BLACK. *Electromigration failure modes in aluminium metallization for semiconductor devices*. *Proceedings of the IEEE*, volume 57, no. 9, pages 1587–1594, September 1969.

[Bla01] Colin BLAKEMORE and Sheila JENNETT, editors. *The Oxford Companion to The Body*. Oxford University Press, 2001. ISBN 019852403.

[Bra99] Tom BRANCA. *How to add features and fix bugs - remotely*. *XCell*, volume 33, pages 12–13, 1999.

[Bra01] Daryl BRADLEY and Andy TYRRELL. *Multi-layered defence mechanisms: Architecture, implementation and demonstration of a hardware immune system*. In *Proceedings of 4th International Conference on Evolvable Systems*, pages 140–150. Springer Verlag, October 2001.

[Bra02] Daryl BRADLEY and Andy TYRRELL. *Immunotronics: Novel finite-state-machine architectures with built-in self-test using self-nonself differentiation*. *IEEE Transaction on Evolutionary Computation*, volume 6, no. 3, pages 227–38, June 2002.

[Bro01] Charles G. D. BROOK and Nicholas J. MARSHALL. *Essential Endocrinology*. Blackwell Science, 4th edition, 2001. ISBN 0632056150.

[Can02] Richard CANHAM and Andy TYRRELL. *Evolved fault tolerance in evolvable hardware*. In *Proceedings of CEC 2002, The Congress on Evolutionary Computation*, pages 1267–1271. IEEE Press, 2002.

[Can03] Richard CANHAM and Andy TYRRELL. *An embryonic array with improved efficiency and fault tolerance*. In *Proceedings of EH 2003, 5th NASA/DoD Conference on Evolvable Hardware*, pages 275–282. IEEE Computer Society, July 2003.

[Coo00] Geoffrey M. COOPER. *The Cell, A Molecular Approach*. Sinauer Associates, 2nd edition, 2000. ISBN 0878931066.

[Cut56] Franseco CUTRY. *The flight of birds*. In *Leonardo da Vinci*, page 337. Reynal and Company, 1956.

[Dar59] Charles DARWIN. *On The Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray, London, 1859.

[ESD99] ESD Association. *ANSI/ESD-S20.20-1999 - Standard for the Development of an Electrostatic Discharge Control Program for: Protection of Electrical and Electronic*

Parts, Assemblies and Equipment (Excluding Electrically Initiated Explosive Devices), August 1999.

[Fla02] David FLANAGAN. *Java in a Nutshell*. O'Reilly, 4th edition, 2002. ISBN 0596002831.

[Fog94] David B. FOGEL. *An introduction to simulated evolutionary optimization*. *IEEE Transactions on Neural Networks*, volume 5, no. 1, pages 5–14, January 1994.

[Foo87] P.K. FOOTNER, B.P. RICHARDS and R.B. YATES. *Purple plague: Eliminated or just forgotten? Quality and Reliability Engineering International*, volume 3, pages 177–184, 1987.

[For96] Stephanie FORREST, Steven A. HOFMEYR, Anil SOMAYAJI and Thomas A. LONGSTAFF. *A sense of self for Unix processes*. In *Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy*, pages 120–128. IEEE Computer Society Press, 1996.

[Gar98] Paul GARD. *Human Endocrinology*. Taylor Francis Ltd, 1998. ISBN 0748406557.

[Gre94] Ben GREENSTEIN. *Endocrinology at a Glance*. Blackwell Science, 1994. ISBN 0632038357.

[Gre03] Andrew GREENSTED and Andy TYRRELL. *Fault tolerance via endocrinologic based communication*. In *Proceedings of ICES 2003, 5th International Conference on Evolvable Hardware*, number 2606 in LNCS, pages 24–34. Springer Verlag, March 2003.

[Gre04] Andrew GREENSTED and Andy TYRRELL. *An endocrinologic-inspired hardware implementation of a multicellular system*. In *Proceedings of EH 2004, 6th NASA/DoD Conference on Evolvable Hardware*, pages 245–252. IEEE Computer Society, June 2004.

[Har91] D. G. HARDIE. *Biochemical Messengers, Hormones, Neurotransmitters and Growth Factors*. Chapman and Hall, 4th edition, 1991. ISBN 0412303507.

[Har04] Simon HARDING and Julian MILLER. *Evolution in materio: A tone discriminator in liquid crystal*. In *Proceedings of CEC 2004, The Congress on Evolutionary Computation*, volume 2, pages 1800–1807. June 2004.

[Hof99] Steven A. HOFMEYR and Stephanie FORREST. *Immunity by design: An artificial immune system*. In *Proceedings of the Genetic and Evolutionary Computation Conference* (edited by Wolfgang BANZHAF, Jason DAIDA, Agoston E. EIBEN, Max H. GARZON, Vasant HONAVAR, Mark JAKIELA and Robert E. SMITH), volume 2, pages 1289–1296. Morgan Kaufmann, Orlando, Florida, USA, July 1999.

[Inta] INTEL®. *Intel Research - Silicon - Moore's law*. URL `http://www.intel.com/research/silicon/mooreslaw.htm`.

[Intb] INTEL®. *Microprocessor quick reference guide*. URL `http://www.intel.com/pressroom/kits/quickreffam.htm`.

[Jac03] Alex JACKSON. *Asynchronous Embryonics, self-timed biologically-inspired fault-tolerant computing arrays*. Ph.D. thesis, The University of York, 2003.

[Jan99] Charles A. JANEWAY, Paul TRAVERS, Mark WALPORT and J. Donald CAPRA. *Immunobiology, The Immune System in Health and Disease*. Garland Publishing, 4th edition, 1999. ISBN 0443062757.

[Jen95] Finn JENSEN. *Electronic Component Reliability - Fundamentals, Modelling, Evaluation and Assurance*. John Wiley & Sons, 1995. ISBN 0471952966.

[Kea03] Martin A. KEANE, John R. KOZA and Matthew J. STREETER. *Evolving inventions*. *Scientific American*, pages 40–47, February 2003.

[Key00] Didier KEYMEULEN, Ricardo Salem ZEBULUM, Yili JIN and Adrian STOICA. *Fault-tolerant evolvable hardware using field-programmable transistor arrays*. *IEEE Transaction on Reliability*, volume 49, no. 3, pages 305–316, September 2000.

[Klu97] William S. KLUG and Michael R. CUMMINGS. *Concepts of Genetics*. Prentice Hall, 5th edition, 1997. ISBN 0135310628.

[Koz04] John R. KOZA, Martin A. KEANE and Matthew J. STREETER. *Routine high-return human-competitive evolvable hardware*. In *Proceedings of EH 2004, 6th NASA/DoD Conference on Evolvable Hardware*, pages 3–17. IEEE Computer Society, June 2004.

[Kub97] Janis KUBY. *Immunology*. W.H. Freeman, 3rd edition, 1997. ISBN 0716728680.

[Lal85] Parag K. LALA. *Fault Tolerant and Fault Testable Hardware Design*. Prentice Hall, 1985. ISBN 0133082482.

[Lal01] Parag K. LALA. *Self-Checking and Fault-Tolerant Digital Design*. Academic Press, 2001. ISBN 0124343708.

[Lap89] Jean-Claude LAPRIE. *Dependability: a unifying concept for reliable computing and fault tolerance*. In *Dependability of Resilient Computer* (edited by Tom ANDERSON), chapter 1, pages 1–28. Blackwell Scientific Publications, 1989.

[Lee90] Pete LEE and Tom ANDERSON. *Fault Tolerance, Principles and Practice*. Springer Verlag, 2nd edition, 1990. ISBN 0387820779.

[Liu04] Heng LIU, Julian MILLER and Andy TYRRELL. *An intrinsic robust transient fault-tolerant developmental model for digital systems*. In *GECCO 2004 workshop, Genetic and Evolutionary Computation Conference*. July 2004.

[Llo97] J R. LLOYD. *Electromigration in thin film conductors*. *Semiconductor Science and Technology*, volume 12, no. 10, pages 1177–1185, October 1997.

[Lod00] Harvey LODISH, Arnold BERK, S. Lawrence ZIPURSKY, Paul MATSUDAIRA, David BALTIMORE and James DARNELL. *Molecular Cell Biology*. W. H. Freeman and Company, 4th edition, 2000. ISBN 0716731363.

[Man96] Daniel MANGE, M. GOEKE, D. MADON, A. STAUFFER, Gianluca TEMPESTI, S. DURAND, P. MARCHAL and P. NUSSBAUM. *Embryonics: A new family of coarse-grained field-programmable gate array with self-repair and self-reproducing properties*. In *Towards Evolvable Hardware: The evolutionary engineering approach* (edited by Eduardo SANCHEZ and Marco TOMASSINI), number 1062 in LNCS, pages 197–220. Springer-Verlag, 1996.

[Man00] Daniel MANGE, Moshe SIPPER, André STAUFFER and Gianluca TEMPESTI. *Towards robust integrated circuits: The embryonics approach*. *Proceedings of the IEEE*, volume 88, no. 4, pages 516–541, April 2000.

[Mar00] Elaine N. MARIEB. *Human Anatomy and Physiology*. Benjamin Cummings, 5th edition, 2000. ISBN 085349898.

[Mil04] Julian F. MILLER. *Evolving a self-repairing, self-regulating, french flag organism*. In *Proceedings of GECCO 2004, Genetic and Evolutionary Computation Conference*. Springer Verlag, July 2004.

[Mit96] Melanie MITCHELL. *An introduction to Genetic Algorithms*. MIT Press, 1996.

[Mon87] MONTREAL MUSEUM OF FINE ARTS, editor. *Leonardo da Vinci : Engineer and Architect*. Montreal Museum of Fine Arts, 1987. ISBN 2891920848.

[Moo65] Gordon E MOORE. *Cramming more components onto integrated circuits*. *Electronics*, volume 38, no. 8, April 1965.

[Oku81] Katsuya OKUMARA. *Degradation of bonding strength (al wire-au film) by kirkendall voids*. *Journal of the Electrochemical Society*, volume 128, no. 3, pages 571–575, March 1981.

[OS98] Cesar ORTEGA-SÁNCHEZ and Andy TYRRELL. *Design of a basic cell to construct embryonic arrays*. *IEEE transactions on Computers and Digital Techniques*, volume 145, no. 3, pages 242–248, May 1998.

[OS00] Cesar ORTEGA-SÁNCHEZ and Andy TYRRELL. *A hardware implementation of an embryonic architecture using Virtex® FPGAs*. In *Proceedings of ICES 2000, 3rd International Conference on Evolvable Hardware*, number 1801 in LNCS, pages 155–164. Springer Verlag, 2000.

[Rid04] Mark RIDLEY. *Evolution*. Blackwell, 3rd edition, 2004.

[Rum93] Francis RUMSEY and John WATKINSON. *The Digital Interface Handbook*. Focal Press, 1993. ISBN 0240513339.

[Sel69] Bernard SELIKSON. *Void formation failure mechanisms in integrated circuits*. *Proceedings of the IEEE*, volume 57, no. 9, pages 1594–1598, 1969.

[Ser91] G. E. SERVAIS and S.D. BRANDENBURG. *Wire bonding - a closer look*. In *Proceedings of ISTFA 1991, 17th International Symposium for Testing & Failure Analysis*, page 525. November 1991.

[Sha97] Ashok K. SHARMA. *Semiconductor Memories: Technology, Testing and Reliability*. IEEE Press, 1997. ISBN 0780310004.

[Shi00] Jack SHIRAZI. *Java Performance Timing*. O'Reilly, 2000. ISBN 0596000154.

[Skl88] Bernard SKLAR. *Digital Communications: Fundamentals and Applications*. PTR Prentice Hall, 1988. ISBN 0132119390.

[Sma99] R.E. SMALLMAN and R.J. BISHOP. *Modern Physical Metallurgy and Metals and Materials: Science, Process, Applications*. Elsevier, 6th edition, 1999. ISBN 0750645644.

[Spu00] Charles E. SPURGEON. *Ethernet: The Definitive Guide*. O'Reilly, 2000. ISBN 1565926609.

[Ste94] Richard W. STEVENS. *TCP/IP Illustrated*, volume 1. Addison-Wesley, 1994. ISBN 0201633469.

[Ste00] Dick STEFLIK and Prashant SRIDHARAN. *Advanced Java Networking*. Prentice Hall PTR, 2nd edition, 2000. ISBN 0130844667.

[Str95] Ben G. STREETMAN. *Solid State Electronic Devices*. Solid State Physical Electronics. Prentice Hall, 4th edition, 1995. ISBN 0131587676.

[Tho97] Adrian THOMPSON. *An evolved circuit intrinsic in silicon entwined with physics*. In *Proceedings of ICES 1996, 1st International Conference on Evolvable Hardware*, number 1259 in LNCS, pages 390–405. Springer Verlag, 1997.

[Tsc01] Thomas TSCHAN. *An overview of flip-chip technology*. *Chip Scale Review*, May-June 2001.

[vN56] John Louis VON NEUMANN. *Probabilistic logics and the synthesis of reliable organisms from unreliable components*. In *Automata Studies* (edited by C. E. SHANNON and J MCCARTHY), page 43. Princeton University Press, 1956.

[Wan96] Weiqing WANG and Z. SUO. *A simulation of electromigration-induced transgranular slits*. *Journal of Applied Physics*, volume 79, no. 5, pages 2394–2403, March 1996.

[Wol91] Lewis WOLPERT. *The Triumph of the Embryo*. Oxford University Press, 1991. ISBN 0198542437.

[Wol02] Lewis WOLPERT, Rosa BEDDINGTON, Thomas JESSELL, Peter LAWRENCE, Elliot MEYEROWITZ and Jim SMITH. *Principles of Development*. Oxford University Press, 2nd edition, 2002. ISBN 0198792913.

[Xil03] Xilinx. *Spartan-IIE 1.8V FPGA Detailed Functional Description*, 2.1 edition, 2003. URL `http://direct.xilinx.com/bvdocs/publications/ds077_2.pdf`.

[Zeb02] Ricardo Salem ZEBULUM, Marco Aurélio C. PACHECO and Marley Maria B.R. VALLASCO. *Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms*. CRC Press, 2002. ISBN 0849308658.

[Zie79] J. F. ZIEGLER and W. A. LANFORD. *Effect of cosmic rays on computer memories*. *Science*, volume 206, pages 776–788, November 1979.

[Zie96] J. F. ZIEGLER, H. W. CURTIS, H. P. MUHLFELD, C. J. MONTROSE, B. CHIN, M. NICEWICZ, C. A. RUSSELL, W. Y. WANG, L. B. FREEMAN, P. HOSIER, L. E. LAFAVE, J. L. WALSH, J. M. ORRO, G. J. UNGER, J. M. ROSS, T. J. O'GORMAN, B. MESSINA, T. D. SULLIVAN, A. J. SYKES, H. YOURKE, T. A. ENGER, V. TOLAT, T. S. SCOTT, A. H. TABER, R. J. SUSSMAN, W. A. KLEIN and C. W. WAHAUS. *IBM experiments in soft fails in computer electronics (1978-1994)*. *IBM Journal of Research and Development - Terrestrial cosmic rays and soft errors*, volume 40, no. 1, pages 3–18, 1996.

Colophon

The original of this document is printed in colour. However, at no point is colour used as a primary method of infering information.

This document was created using the LaTeX typesetting system, version 2e. The text was edited using Vim, version 6.2. Illustrations were created using using Adobe Illustrator, version 10. The document typeface is 11pt. Palatino.