

The 'Evolvable Motherboard'  
A Test Platform for the Research of  
Intrinsic Hardware Evolution

**Paul Layzell**

**CSRP 479**

**January 1998**

School of Cognitive and Computing Sciences, University of Sussex,  
Brighton, Sussex. BN1 9QH, UK.

E-mail: [paulla@cogs.susx.ac.uk](mailto:paulla@cogs.susx.ac.uk)

## Abstract

The study of intrinsic hardware evolution relies heavily on commercial FPGA devices which can be configured in real time. Use of these devices presents certain drawbacks to the researcher desirous of studying the fundamentals of hardware evolution, since he has no control over the architecture or type of basic configurable element. Furthermore, analysis is difficult as only a small region of FPGAs is accessible to test equipment. After discussing current issues arising in evolvable hardware, this paper presents a test platform designed specifically for research into intrinsic hardware evolution, together with experimental results exemplifying its use.

## 1 Introduction

Research into the use of evolutionary algorithms (EAs) for the optimisation and solution of complex problems has been extensively carried out by many institutions world-wide for some twenty years, and is now a firmly established field. In recent years, EAs have been applied to the design of electronic circuitry with significant results being attained using both computer simulations and physical hardware. The possibility of the latter is largely due to the advent of new Field Programmable Gate Array (FPGA) chips consisting of many small circuit elements which can be configured virtually instantaneously to produce a huge variety of different circuits. Any particular configuration can be specified by a genotype, allowing FPGA circuits to be evolved with basic EAs such as genetic algorithms. Furthermore there is good reason to do so, for (as will be shown later) evolved circuits do not suffer from the constraints imposed by conventional design, and can exploit the silicon medium more effectively than human-designed circuitry can. It is tempting to think therefore, that FPGAs are *ideal* for hardware evolution, however this is certainly not the case. For one thing the vast majority of commercial FPGAs use digital configurable logic blocks (CLBs) as the basic element. While evolution can produce analogue behaviours from these elements, such circuits may suffer from a number of undesirable characteristics such as dependence on temperature and lack of portability. Analogue equivalents (e.g. the Motorola/Pilkington FPAA)

are starting to become available using operational amplifiers as basic elements, but we do not yet know exactly what the most appropriate basic element for hardware evolution might be - it could be extremely basic, such as a transistor; some higher level multi-functional unit; or combinations of different components including passive resistors or capacitors. Also the interconnection highways between circuit elements in FPGAs are designed with the conventional, modular circuit design methodology in mind. Evolution may be able to exploit a different system of interconnections or architecture - perhaps a more arbitrary one - much more effectively, but once again not enough is known at this stage to specify exactly what that system might be. Finally circuits which have evolved on an FPGA to produce some desired behaviour are extremely difficult to analyse, since it is not possible to access individual circuit elements with test equipment, and computer simulations cannot practically reproduce all of the physical properties of silicon that may be being exploited.

In this article a new discrete configurable testbed is presented which has been specifically designed and built to illuminate the above issues, as well as others arising from current research in the field of Evolvable Hardware (EHW). The article is structured as follows: After a description of the basic principles of hardware design by evolution (Section 2), an exploration of the most promising target domains is made, with the aid of examples drawn from recent literature (Section 3). Some conclusions about the benefits and pitfalls of current techniques are then presented as motivation for future research (Section 4), followed by a description of the testbed and discussion of its place in EHW (Section 5). Finally, results of initial experimentation, together with a short discussion on their implications is presented (Chapter 6).

## 2 Hardware design using evolutionary techniques

### 2.1 Basic principles of EHW

EHW is a recent field which applies evolutionary algorithms (EA) to the design of electronic and electrical systems. The basic approach is as follows: A circuit is represented by a (usually binary) array or string (the *genotype*), whose contents may describe for example the connections within the circuit, types of

component to be used, logical functions, amplification factors etc. An initial population of strings is created at random, and each individual string is evaluated according to how well the circuit it represents (the *phenotype*) achieves some behavioural specification, or *fitness function*. A breeding phase follows, using a variety of genetic operators depending on the particular EA used, in which fitter individuals are likely to survive and/or have offspring, and less fit individuals are likely to die out. The circle of evaluation-breeding continues until a fitness level equating to the solution of the problem is achieved.

## 2.2 Choice of EA

At present there is little consensus among researchers as to the most appropriate EA for hardware evolution. Among those currently being explored are Genetic Algorithms (GA) (Holland, 1975), Genetic Programming (GP) (Koza, 1992), and Evolutionary Programming (EP) (Fogel et al., 1966). The selection of a particular EA often depends on the researcher's desire to incorporate domain specific knowledge into the evolutionary process. Koza has pointed out differences in GP trees and electronic circuit graphs (Koza et al., 1996a) and proposed a circuit constructing tree which biases evolution in favour of producing circuit topologies similar to those produced by conventional design. Hemmi et al. also use a tree-like genotype representation in their Hardware Description Language (HDL) which promotes the repetition of evolved substructures or building blocks (Hemmi et al., 1996), considered by most authors to be a good heuristic for the evolution of large circuits, and prevalent in conventional design. However as Thompson (1996a) points out concerning the use of domain-specific knowledge :

“This *is* sensible if this information inevitably would have to be rediscovered, but if the information represents just *some* ways of setting about solving the problem - perhaps ways suitable for human designers, or for evolution in biology, but not suitable for the electronic medium - then forcing evolution to use this information unnecessarily restricts the space of possible solutions. Even worse, it could steer evolution in ways incompatible with the nature of the evolutionary process or of the reconfigurable medium.”

The complexity of circuits produced so far has been largely independent of the EA used. Whilst it may be important to prevent useable circuits from being isolated within the search space, this is arguably more dependent on the genotype - phenotype mapping than any particular evolutionary algorithm. I shall propose in section 5.7 one mapping possibility that can confine the search space to useable circuits using traditional GAs.

## 2.3 Evaluation

The evaluation phase can be categorised into two important approaches, dubbed by De Garis (1993) as:

*Extrinsic EHW*, in which the individuals are simulated on computer until completion of the evolutionary process, whereupon the best individual is instantiated in hardware.

*Intrinsic EHW*, in which each individual is physically instantiated in hardware.

Extrinsic EHW is the simpler in that only a computer and appropriate simulation software is necessary to carry out the evolutionary process. The circuit produced is easy to analyse, since the physical properties modelled by the simulation are accurately known, and most simulation software allows the user to view the waveforms present at any desired point within the circuit. However, the computing power required to model even a small subset of the physical properties of semiconductors poses severe limitations on the speed of evolution. Another potential drawback with extrinsic EHW is that proprietary simulation software such as SPICE and SPLASH are intended as conventional design tools. In general they assume that for example transistors will be used in conventional configurations and the physical properties they model are optimised for these configurations. This poses severe constraints on evolution since evolved circuits can be most interesting when they use components in non-standard ways (see sections 3.2, 5.4). Such circuits achieved in simulation are unlikely to work when realised in hardware since the properties of real devices differ considerably from those modelled by software in such configurations.

The fact that physical modelling is not necessary with intrinsic EHW offers two major advantages over extrinsic EHW: The speed of evaluation can be much higher, and more importantly continuous dynamics arising from say, differing gate propagation-times; the proximity of two

components; or fluctuations on the power lines can be exploited. Unfortunately, since circuits are usually instantiated on configurable integrated circuits such as FPGAs, analysis is difficult or impossible because the internal parts of these devices are inaccessible to measuring equipment. Furthermore, the variety of currently available configurable devices suitable for EHW is severely limited, mainly to digital functions.

### 3 Target domains for EHW

#### 3.1 EHW versus conventional design techniques

Electronic circuitry is currently designed using a methodology that is now firmly established - successive divisions or abstractions of the main task into sub-tasks which can be tackled using industry-standard building blocks. The building blocks used in this process are either custom-designed integrated circuits (for example oscillators, multiplexers, amplifiers, logical functions, etc.) or configurations of discrete components whose properties have already been analysed in depth, and are well-understood (for example butterworth filter, class B amplifier, wein-bridge oscillator (Horowitz & Hill, (1989))). In recent years, the variety and complexity of standard building blocks has increased, while their production cost has reduced. The properties of many larger building blocks can be configured in real time by varying the voltage levels present on particular pins, and systems incorporating large numbers of such building blocks generally require a synchronous microprocessor to co-ordinate events. Hence items such as car stereos, washing machines, video recorders etc., whose functionality has changed little over the past twenty years have actually increased in complexity by several orders of magnitude (complexity here referring to the quantity of active components present in the item) over the same period. The overall pay-off to the designer is of questionable value: Whilst he or she need no longer have a thorough background in the fundamental principles of electronics or semiconductor physics, an engineer capable of designing medium sized systems must be conversant with computer architecture and programming, and have an encyclopaedic knowledge of currently available building blocks and 'standard' circuit configurations. Large systems (for example the control system of an aircraft) are beyond the scope of a single designer, and must be tackled by teams, each having very domain-specific

expertise. Hence despite the abstraction inherent in conventional design, the development of large complex systems is very costly and requires many specialist engineers.

This chapter discusses how the use of evolutionary algorithms to design hardware offers potential advantages as an alternative to the conventional approach, and may also be able to complement it by producing more building blocks which make use of electronic properties not yet explored by conventional designers. Analysis of evolved circuits by electronics engineers may lead to such properties being exploited to create new and innovative designs in the conventional manner. It is tempting to envisage the advancement of EHW research to such a stage that it is able to replace the conventional design methodology altogether for many tasks, particularly complex ones. To do this however would be to ignore other significant benefits that EHW can offer electronic engineering in the near future. Below is a classification of general targets for the field which I consider to be most worthwhile.

1. Functional circuits which can be produced using the evolutionary approach requiring little or no specialist knowledge of electronics, and which may or may not offer significant benefits compared to those produced using conventional design
2. Fault tolerant systems
3. Low power circuitry
4. Adaptive systems, which are able to change their configuration in the operational phase to cope with unexpected challenges in a dynamic environment
5. An entirely new methodology for the design of complex behaviours not requiring the abstraction inherent in the 'Divide and Conquer' approach

#### 3.2 Functional circuits

There are already many examples in the literature where EHW has produced small functional circuits which could be used as building blocks in conventional design. So far they have been presented not as an end in themselves, but as evidence to support some particular evolutionary strategy or principle. Notable digital examples are: 6-input multiplexer and multiple XOR circuits (Kitano, 1996); 4-bit comparator (Higuchi et al., 1996); Sequential Adder (Hemmi et al., 1996); and a digital string generator (Zebulum et al., 1996). Examples of analogue circuits are : 60dB

operational amplifier; two-band crossover filter for loudspeakers (Koza et al, 1996b); Slow oscillator and Tone discriminator (Thompson, 1996b). Note however, that of these examples, only the slow oscillator and Tone discriminator have been realised in hardware.

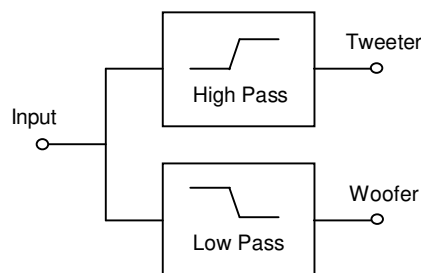


Figure 3.1  
Conventional Crossover filter

Some of these circuits highlight important characteristics of hardware evolution, and are worth mentioning in more detail. Firstly, Koza's two-band crossover filter: Figure 3.1 shows how such filters are conventionally abstracted into two simpler filters independent of one another, which simplifies design. Evolution has no need of such decomposition, and Koza's final circuit is holistic in that there are numerous internal connections between the parts feeding the tweeter and those feeding the woofer.

Thompson's two circuits show the ability of evolution to achieve some function even when denied seemingly essential components. The slow oscillator was evolved in a computer simulation of a small portion of a Xilinx X6216 FPGA consisting of 100 configurable logic gates, each with propagation delays of  $\sim 1\text{ns}$ . Using this set-up, Thompson was able to evolve a 4kHz oscillator, i.e. with a period of 0.25ms :- 5 orders of magnitude longer than the gate delays. For a circuit composed of such fast gates, some form of capacitor/resistor network would normally be necessary to provide a delay, however evolution was able to cope without this. Analysis of the simulated circuit showed that the slow oscillation was obtained by using two closely matched high frequency oscillations to create a beat frequency.

The Tone discriminator circuit was required to distinguish between two frequencies of 1ms and 0.1ms period, again much longer than the

FPGA's gate delays. It is not known precisely how the discriminator works, but the output waveforms for intermediate generations showed voltages of more analogue than digital nature. Whilst FPGA's are designed for digital use, they are essentially composed of high-gain analogue amplifiers, and evolution has exploited the chip as a continuous-time dynamical analogue system. Furthermore, once the 'junk' parts of the circuit were determined and removed, the component count was significantly lower than would be expected from a conventional approach. When ascertaining which parts of the circuit were functional, and which were junk, Thompson noticed that certain blocks, whilst not part of the connection path, were nonetheless essential to the circuit's operation. Why this should be so is not certain, however it is clear that intrinsic evolution has exploited some very subtle physical property that would certainly not be envisaged by a human designer. Perhaps for this reason, the circuit is very sensitive to temperature variation, having an operational range of  $\pm 5^\circ\text{C}$ , small by conventional standards.

### 3.3 Fault tolerance

Fault tolerance is not an inherent characteristic of conventional design - the failure of a single transistor or connection very often results in breakdown of the entire system. The most common conventional approach to improve fault tolerance is to use redundancy, particularly in safety-critical systems such as aircraft, although innovative methods for graceful degradation using multi-processor arrays have achieved some success in the spacecraft industry (Castro, 1995). Many evolved hardware strategies use a one-to-one mapping between genotype bits and circuit connection or logic function. The type of fault produced by bit mutation is therefore similar to those that would lead to system failure for conventional circuits. Mutation is a common operator in evolutionary algorithms used to introduce variation, and a characteristic of evolved systems is that fitter individuals in later generations are relatively insensitive to bit mutation. This has already been observed in molecular evolution (Eigen 1987, Huynen & Hogeweg 1994) for a problem whose fitness landscape was such that the optimum peak suffered severe fitness degradation from single mutations, whereas mutations on another slightly sub-optimal peak would result in only small degradation. The problem has recently been repeated in the context of a GA (Thompson, 1996a). In almost all cases the

population moved away from the isolated global optimum in favour of the slightly inferior fitness peak. After investigating the effect further with NK fitness landscapes (Kaufmann, 1993) Thompson concluded that although limited in magnitude and range of faults to which it can apply, for certain types of EA:

“ [graceful degradation] arises ‘for free’ out of the nature of the evolutionary process, without any special measures having to be taken.”

That a degree of fault tolerance is inherent in EHW may be of profound benefit to engineering. A further amount can be potentially achieved by

1. Incorporating fault tolerance into the fitness function, and deliberately introducing serious faults when the hardware is configured. Some success has already been achieved in this area (Thompson, 1995a)
2. For larger systems, determining serious faults requires testing for every possible fault, which is prohibitive. An alternative is Co-evolving (Hillis, 1992) a population of faults that concentrate on the weak spots of the evolving circuits. This has not yet born fruit in EHW, although some notable results are to be found in Rosin (1997).

### 3.4 Adaptive systems

Closely related to fault tolerance, adaptive systems can change their configuration or behaviour during the operational phase to cope with unexpected challenges from their environment or indeed, faults occurring within them. An adaptive robot, for example might be able to change its gait if a leg joint seized; right itself if it were placed upside-down; or find some way of walking around or over some obstacle. There are two obvious approaches to achieving this in EHW - firstly evolving the system in a noisy environment in which examples of each event are likely to occur, and secondly carrying on evolution during the operational phase so that the system is in a continual state of improvement, an ‘intrinsic environment’. Both of these methods are fraught with problems: the first method could indeed exhibit adaptive behaviour, but all the challenges it could meet would be to some extent expected, since they have been introduced during evolution. The second method would require some opportunity for newly evolved behaviours to be evaluated. Not only would the evaluation mechanisms be unclear, but a system of which continual

operation was expected would simply not be able to stop so that new behaviours can be tried out.

The word ‘adaptation’ is spread liberally throughout the literature, however EHW has not yet achieved the definition applied here i.e. adaptation in the *operational phase*. However there are a number of precedents which may be applied to EHW. An example of the first method is Jakobi’s work on minimal simulations for evolutionary robotics (Jakobi, 1997), which has shown that a reliable transition from simulation to reality is possible by modelling only a small subset of the agent/environmental properties. He proposes a general set of equations that describe the way in which an agent-environment system changes over time, and with careful analysis of what it means for an agent to exhibit a particular behaviour within such a system, establishes a minimally sufficient set of conditions under which a controller that reliably displays a particular behaviour in simulation will continue to do so in reality. Jakobi’s system has produced examples of 8-legged walking with obstacle avoidance, visual shape recognition, and a visually-guided movement tracker, all of which exhibit extremely robust performance under very noisy conditions. Furthermore, due to minimal simulation, this behaviour is evolvable in practical timescales, with basic walking behaviour produced in a matter of hours.

A possible precursor of the second method is Grefenstette’s Anytime Learning Model (GrefenStette, 1996) in which, while operating in a real environment, an agent maintains a simulated environmental model which can be updated whenever new features occur either in the environment, or the agent’s own sensory capabilities. Learning continues throughout the operational phase, however new strategies for performance improvement are concurrently tested in simulation so that, should they fail, no dangerous or time-consuming behaviour is exhibited in reality. Successful strategies can then be switched from simulation to reality once reliable improvements are discovered. Some success has already been achieved in EHW with successful transfer of behaviour from simulation to real, albeit simple environments (Thompson, 1995b; Keymeulen et al, 1997). The application of Jakobi and Grefenstette’s ideas to on-line adaptation using EHW would make an interesting study.

### **3.5 Complex behaviour**

Complexity is a relative term. EHW is nowhere near producing its first aircraft control system, hence I use the term here to describe behaviours which would require extensive design were they to be implemented using conventional techniques. Current work on autonomous robots fits loosely into this category, one notable example being Thompson's wall-avoiding robot, 'Mr Chips', whose task was to move continuously in the centre of an arena (Thompson, 1995b). The robot was equipped with two ultrasonic transducers and was propelled by two wheels powered by d.c. motors, however the hardware normally associated with these facilities (i.e. reflection timers and p.w.m. controllers, respectively) were denied. Using 'genetic latches' evolution was able to choose which part of the circuit, if any, was synchronised to a programmable clock, and the resulting circuit using a mixture of synchronous/asynchronous control was able to achieve the required behaviour using only 32-bits of RAM and a few flip-flops. The combination of synchronous and asynchronous operation allowed the system to explore a wider range of dynamics than would have been possible under synchronous control, this being demonstrated in further experiments where synchronicity was enforced, resulting in failure.

Another domain which arguably falls under this heading is that of signal processing. EHW has been successfully applied to lossless image compression by classification of neighbouring patterns within the image and defining a set of evolved functions to establish an adaptive prediction coding system, a method commonly used in image compression (Salami, 1997). Using simulation of a modified Xilinx coarse-grained FPGA, results comparable to JPEG compression have been achieved.

## **4 Foundations and Stumbling blocks for EHW - a focus on further work**

With a view to determining future research directions, this section presents a few conclusions from some of the work described above, on areas where EHW either particularly excels over conventional design, or where significant obstacles lie in the way of further success.

### **4.1 Benefits of the evolutionary approach**

1. Through relaxing constraints which human designers deem necessary to cope with circuits of high behavioural complexity, evolution can produce functional circuits with richer dynamical and spatial structure than is within the scope of conventional design.
2. When implemented intrinsically, evolution is capable of exploiting subtle physical properties of the medium that cannot be predicted using the conventional approach, thereby increasing further the spectrum of behaviours that the medium can produce.
3. Complex behaviours can be achieved with smaller component count than normally expected of the conventional approach.
4. Evolution is capable of producing functionality in the absence of components that would be necessary under conventional design maxims. This has significant implications for VLSI where components such as capacitors, resistors and inductors are difficult to implement in small size.
5. The relative insensitivity of certain types of EA to mutation can be exploited to produce systems which are inherently robust to noise and certain types of fault.
6. The circuits produced using evolution may reveal electronic properties not yet exploited by conventional designers. Analysis of such circuits by electronics engineers may lead to such properties being used to create new and innovative designs in the conventional manner.
7. The human requirement to the evolutionary approach is to produce a behavioural specification, which can be achieved with little or no knowledge of electronics, thereby opening up the field of hardware design to non-specialists.

### **4.2 Difficulties of the evolutionary approach**

#### **4.2.1 Transferral from simulation to physical hardware**

So far most of the examples described in chapter 3 have been produced in simulation only and many cannot be yet be physically instantiated in hardware, either because they require values of resistance, capacitance etc. that are not readily available, or because the simulations do not take account of manufacturing variations in physical devices. An important example of this is the  $h_{FE}$

parameter (sometimes loosely referred to as the *DC current gain*) for bipolar transistors. One commonly used transistor, the BC109 has a typical value of 350 for this parameter, the value adopted by most simulations. However, in physical BC109s  $h_{FE}$  varies from 200 to 800 (SGS-Thompson, 1989). The value is further dependent on factors such as collector current and temperature. Thus evolved circuits which assume a consistent value for  $h_{FE}$  are extremely unlikely to work in practice. Conventional engineering has already surmounted this particular problem - Whilst an amplifier can be constructed using only two external resistors around a BC109 whose  $h_{FE}$  is known, the circuit would fail altogether if a different BC109 were to be substituted. In practice the amplifier is designed so that its functionality is dependent only on  $h_{FE}$  being above a certain minimum value, however such amplifiers require approximately four times as many external components (Horowitz & Hill, 1989).  $h_{FE}$  is just one of many parameters associated with semiconductor devices whose values vary significantly in reality. Conventional engineering has coped by using years of analysis to produce circuit topologies whose behaviour is independent of parameters known to vary within component batches, these standard topologies then being employed universally as building blocks. If EHW is to be used to design circuit topologies which can be reliably instantiated in hardware then some method for evolution to take manufacturing tolerance into account must be found.

#### 4.2.2 Portability of intrinsically evolved circuits

When it comes to producing a circuit diagram to be used as a blueprint in order to reproduce evolved circuits, intrinsic EHW suffers from similar problems to those encountered by extrinsic EHW, particularly when unconstrained. The tone discriminator circuit exploited some properties inherent only in the particular area of silicon on which it evolved, and failed to work when instantiated on another chip, or area of the same chip. However, only a relatively small number of further generations were necessary to resume functionality upon transferral (Thompson 1996c).

#### 4.2.3 Susceptibility to temperature variation

Semiconductor physics is rife with temperature dependent properties, which can be detrimental for certain types of circuit (for example

oscillators, reference voltages, d.c. amplification & measurement). Conventional electronics has countered the problem by (a) using analysis to find equations describing those properties, and using combinations of components with both positive and negative temperature coefficients, such that the overall configuration has zero tempco (for example Bandgap Reference (Horowitz & Hill, 1989); (b) by using digital techniques to adapt non temperature-dependent components to the application, for example stable quartz crystal oscillators can be made into timers or slow clocks by using flip-flops to successively divide the frequency by several orders of magnitude. By using such techniques, most proprietary integrated circuits have a functional operating range of around  $\pm 60^\circ\text{C}$ , and sometimes more.

If evolution exploits temperature dependent properties, then the resulting circuit is likely to function only at the approximate temperature in which it was evolved. Producing evolved circuits with operating ranges similar to conventional electronics is a tall order. Here are some suggestions:

- Allow evolution to use only non-temperature dependent building blocks. Not really practical - such an approach would severely confine the search space of useable circuits, and in any case, it is difficult to envisage how to apply such constraints: the gates in a Xilinx FPGA have commercial operating ranges, yet the tone discriminator exploited unexpected properties, resulting in a much smaller operating range for that circuit.
- Vary the temperature during evolution. This suggestion is also impractical - not only is it difficult to provide variations comparable to those under which a commercial chip would be expected to perform, but the temperature would have to be varied during the evaluation of *each individual*, probably several times. Even if each evaluation lasts as long as one second, it is virtually impossible to produce large temperature variations in so short an interval, and in any case, the packaging for commercial devices are unable to withstand large, rapid variations.
- Evaluate each individual on several configurable devices, each at a different temperature. This is the approach currently being taken by Thompson (1997), one which may also cope with the reproducibility issue described in the previous section. It seems the only practical method of evolving



reproducible circuits with large operating temperature ranges without imposing pre-defined constraints, however it will inevitably constrain the exploitable resources available in the medium. At this early stage of research, my own feeling is that there is much yet to discover of the useful physical properties that evolution should be capable of harnessing. Is such a constraint too limiting?

#### 4.2.4 Scalability and complexity

A commonly levelled criticism of much evolutionary research is that success can only be achieved with small genotypes and thus, in the case of EHW, small circuits. So far, the largest genotype length resulting in a physical useful circuit has been 3500 bits long (Thompson, 1996b), however one that say used the entire surface of an X6216 FPGA would require roughly 200,000 bits based on existing mapping techniques. The scalability issue is not only dependent on genotype length. Considering what has already been achieved with only 1% of the X6216 being exploited, one would only expect to use the entire surface for a complex problem, for which the major difficulty would be defining the evaluation procedure and fitness function. Fortunately this is a problem which applies to the EA field as a whole, and a good deal of research has already been carried out, for example (Blickle, 1996) for multi-parameter optimisation, (Harvey et al, 1994) for incremental evolution, and (Worden, 1995) for analysis of a speed limit for evolution.

#### 4.2.5 Analysis of evolved circuits

Since evolution does not need to use abstraction to solve a problem, evolved circuits generally include numerous feedback paths making their analysis very difficult, especially as conventional tools such as functional decomposition are inappropriate. One school of thought is that analysis should be confined to the circuit's external behaviour, whilst acknowledging the internal circuit as a 'black box'. It is true that determining exactly how for example the tone discriminator works may well be impossible, due to difficulty of measurement and the inability of current software models to simulate it. Time spent on analysis of such circuits may well be time wasted if the results are no more than speculative. On the other hand, as touched on in section 3.2, much could be gained by electronic engineering in knowing exactly what properties are being exploited by evolution, and at what scale. Are intrinsically evolved FPGA circuits capable of exploiting the medium down to the

molecular level, as has been found in biological systems, or would this result in ludicrous dependency on for example temperature? The question is also of importance to those trying to establish fundamental differences between silicon and organic material as evolutionary media.

#### 4.2.6 Building blocks for EHW

Whilst not so much a difficulty in itself, the question of which building blocks EHW should use has implications for all of the above topics. Large, configurable functions such as those found in coarse-grained FPGAs would allow complex circuits to be defined by relatively small genotypes, and would possibly make analysis easier, but may be too restrictive for interesting continuous dynamics to be fully exploited. On the other hand, transistors, the fundamental active component of conventional electronics may prove impractical due to their temperature dependency and manufacturing variations. Up until now, researchers have had little choice in the matter, since very few commercially available configurable devices are suitable for EHW, however the number is steadily increasing and among devices to be on the market in the near future, Motorola/Pilkington's FPAA (Bratt & Macbeth, 1996) - which uses operational amplifiers as the basic block - looks particularly applicable.

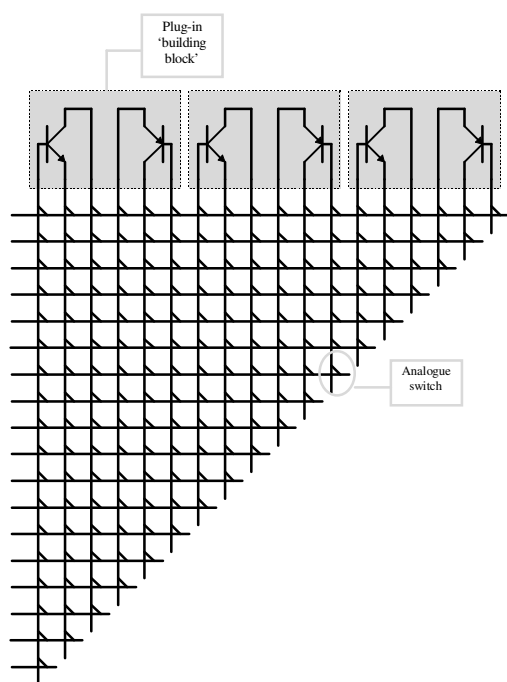


Figure 5.1 A testbed for the study of intrinsic hardware evolution

## 5 The Evolvable Motherboard - A testbed for hardware evolution

### 5.1 Surmounting obstacles in unconstrained intrinsic EHW

With its rich dynamics, efficient use of silicon and ability to realise directly in hardware new ways of exploiting electronic components, unconstrained intrinsic EHW has much potential as a practical alternative/extension to conventional design. For this potential to be realised, then the difficulties outlined in the previous section must be investigated, and solutions proposed for them. Those issues of particular importance are analysis; scalability; building-blocks; architecture; susceptibility to manufacturing variation or temperature.

Analysis of evolved circuits is difficult enough, but can only be indirectly achieved unless internal parts of the circuit can be accessed with measuring equipment. This is not possible with existing configurable devices, implying the design of a new testbed to allow for it. The test bed must allow various types of genotype-phenotype mapping including direct mapping, so that any trade-off between reduced genotype length to aid scalability, and corresponding reduction in functionality can be assessed. The

issues of building blocks and component variation can be also studied if the testbed allows for a variety of different types, and if identical circuits can be implemented in a number of different ways.

Figure 5.1 is a simplified representation of the testbed designed and built for this purpose. It is essentially a diagonal matrix of analogue switches, connected to up to 6 plug-in boards, which contain the desired building-blocks for experimentation. Each board takes up to eight lines on the switch matrix, and a further eight connections (not shown) to allow for various power lines and i/o, which may be required by certain components on the plug-in boards.

#### 5.1.2 Allowing for different genotype - phenotype mapping

The matrix is designed to provide the minimum number of switches necessary so that every combination of possible circuits can be configured. In total there are approximately 1500 switches, giving a search space of  $10^{420}$  possible circuits. Most of these circuits will be useless, since with this configuration of switches, there are many combinations which result in every wire connected to every other wire. However the software interface together with this switch arrangement allows many different mapping techniques, such as GP trees etc. to be investigated. Also, via software, the board can be subdivided (for example 2 matrices each containing three plug-in boards) to allow for mapping individuals to two sets of components thereby allowing the impact of manufacturing tolerance to be assessed. The testbed incorporates additional connectors so that several can be daisy-chained together, should the need arise.

#### 5.1.3 Circuit analysis

The prime function of the testbed is to allow any point of the circuit to be accessible for measurement. If evolution exploits the components allowed it in an unexpected way, it should be simpler to deduce what properties are being used than is the case for an FPGA. Note however that the analogue switches are themselves semiconductor devices, and are contained within integrated circuits each containing 128 switches. These switches behave like low-value resistors, but they also have other properties which may be exploited making analysis difficult. Fortunately, each circuit can

be configured with various different switches to determine whether subtle properties of particular switches are indeed being exploited, and part of the analytical procedure will include duplicating the circuit on conventional platforms, substituting the switches with resistors. If for example switch capacitance is being exploited, it should show up at this stage.

#### 5.1.4 Dealing with illegal configurations

Of the many circuit configurations possible with this testbed, certain of them could lead to its self-destruction. An obvious example of this is if the power lines become shorted together. Each switch can only handle a relatively small current (around 25mA) and if this current is exceeded, the switch will blow-up. One way of preventing this would be to check for such configurations in software, but there are so many that the risk of omitting some is too high. Likewise, there is no point in putting a current limiter on the power supply - many circuits are likely to exceed 25mA current drain without any single switch having to cope with more than half this value - a great many legal circuits would trip the current limiter, and not be evaluated.

The simplest solution to this dilemma can be found by using the fact that the switches behave like resistors whose value is known ( $> 50$  ohms). Furthermore, the power supply is arranged such that any path leading to direct shorts requires at least two switches. By using Ohm's law, it can be shown that if the power supply does not exceed 2.5 Volts, then the testbed itself cannot be blown up. Unfortunately the problem does not end there: certain devices that could be used as building blocks may be destroyed with much smaller currents, however this can be compensated for either by avoiding the use of such components (this is not a major limitation) or by incorporating additional resistors in the plug-in boards.

## 6 Experimentation and results

To illustrate the capabilities of the motherboard, and to highlight its potential as a tool for investigating many issues current in EHW, three experiments are presented in this section. While the tasks are relatively simple, the following experiments are significant in that they are the first ever circuits evolved intrinsically at the transistor level.

### 6.1 An intrinsically evolved NOT Gate

As a starting point for experimentation, bipolar transistors were used as the evolutionary building block, and the task was to evolve a NOT gate. Whilst in itself a fairly trivial function, the NOT gate is an interesting test, firstly because as with many digital functions, evolution can easily tend to a local optimum which does not achieve the required behaviour, and secondly because it is difficult to envisage how gradual improvement can be catered for by the fitness function.

### 6.2 The experimental set-up

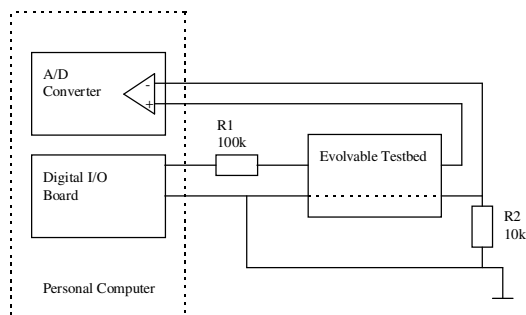


Figure 6.1 Experimental set-up for evolution of digital circuits

Figure 6.1 shows the complete experimental set-up. The digital input to the testbed is provided by a personal computer via a digital I/O board, and the output is connected to an A/D converter card in differential mode to minimise external noise. R1 prevents the I/O card being damaged should its output be shorted to the power supply lines, and is of a high value to encourage the evolved circuit to have a high input impedance. A high value of R1 also prevents configurations that would use the input to power the circuit - undesirable for digital devices. R2 is a necessary requirement when using the A/D converter in differential mode. The testbed is powered by two separate supplies (not shown): one to

provide a +5V supply for the analogue switches, and the other to provide the lower voltage supply to the evolved circuit. By setting the appropriate switches, evolution can make whatever use of this supply it requires, including none, but it does not have access to the +5V switch supply.

The evolutionary algorithm used was a generational GA with single-point crossover, rank-based selection, and elitism. Only a small portion of the testbed was available for this simple circuit, allowing the use of just one NPN transistor and one PNP. To evaluate the circuit, a series of 100 test inputs containing 50 '1s' and 50 '0s' (Logical Highs and Lows, respectively) was applied sequentially in random order. For each test input, the output voltage was measured five times, with random delays between measurements, and summed. Fitness was scored according to equation 6.1:

$$\frac{1}{5} \left\{ \left( \sum_{t \in S_L} v_t \right) - \left( \sum_{t \in S_H} v_t \right) \right\} \quad (6.1)$$

where  $t$  signifies the test input number,  $S_L$  and  $S_H$  the set of Low and High test inputs respectively, and  $v_t$  ( $t = 1, 2, \dots, 100$ ) the summed output voltage in millivolts of the circuit corresponding to test number  $t$ . The power supply to the circuit was set at 2.8V, thus with the above fitness function, ranges of fitness from -140,000 to 140,000 are expected.

### 6.3 Tailoring the GA to the testbed

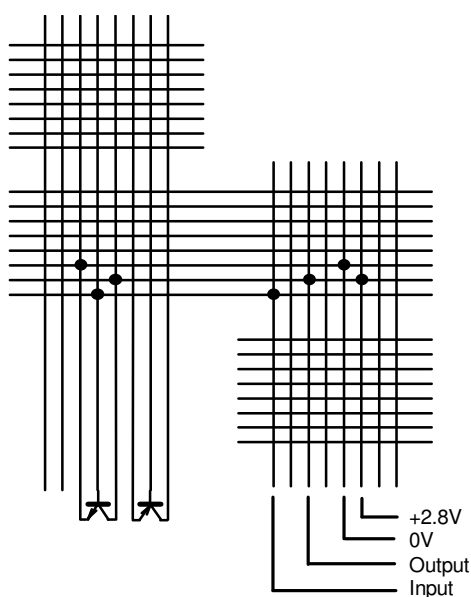
All the experiments detailed in this report use direct mapping from genotype to phenotype in order to maximise the search space, i.e. each switch is represented by a unique bit in the genotype. There is a potential problem using this type of mapping with the testbed: GAs are usually seeded with random populations containing approximately equal numbers of '1's and '0's. This equates to half the testbed switches being on at any one time - a configuration which is almost certain to result in all the wires connected to each other, thereby shorting out all the active components. The population could be seeded with a biased random population corresponding to a much smaller proportion of switches on, but mutation would lead to the balance being equal again after a few hundred generations.

The adopted solution to this problem was to bias the mutation rate so that flips from '1' to '0' are more likely to occur than vice versa. It is

acknowledged that this solution is far from ideal, firstly because it biases evolution away from certain regions of the search space, but more importantly it requires the user to use domain-specific knowledge to set a figure for mutation bias, i.e. effectively imposing a human constraint - undesirable. An alternative mapping is presented in section 6.6, but as the following sections show, direct mapping reveals useful information on how the non-active elements of the circuit may be exploited.

#### 6.4 A hand-seeded NOT gate.

A potentially inescapable local optimum for the fitness function is the trivial circuit giving constant output. Depending on noise, such a circuit would have a fitness of approximately zero, or half marks. This fact combined with the other uncertainties regarding mapping and mutation bias made *hand-seeding* the initial population seem a sensible first step. It was



achieved as follows: An initial population of 50 individuals was seeded randomly with genotypes containing 3% '1's, and biased mutation rates averaging 2 ( $1 \rightarrow 0$ ) and 0.4 ( $0 \rightarrow 1$ ) bits per genotype. The point of the experiment was to see if evolution could improve on an existing prototype circuit, as well as learning more about the other uncertainties involving the testbed. Hence a single individual of the initial random population was replaced by the 'poor' NOT gate circuit shown in figure 6.2.

This circuit conforms to the NOT function in that its output corresponding to a '0' input is of slightly higher voltage than that corresponding to a '1' input, however this difference is too small to be of any practical use. In electronic parlance, its swing is not great enough to cross the low-voltage digital logic threshold. However, the circuit gives a better fitness score than the local optimum described above, hence evolution has something to work on.

#### 6.4.1 Making the best of a bad job

Over 470 generations the elite fitness score increased threefold - from 32,000 to 97,000 - a dramatic improvement in performance when observed on the oscilloscope. While still not ideal, the circuit could now function as a practical NOT gate. Figure 6.3 (overleaf) shows the best individual. In essence the circuit is the same as the seeded one - a single NPN transistor

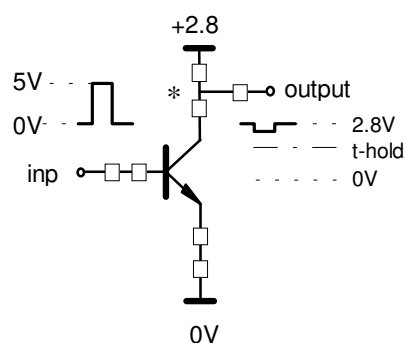


Figure 6.2 Hand-seeded NOT Gate  
(left): Instantiation of the gate on the evolvable testbed. Connections are shown by dark circles.  
(above): The equivalent circuit diagram. Connections are shown by square boxes, each equivalent to resistors of approx.  $50\Omega$  value.

with power and i/o lines connected at the same points, so how has such an improvement been achieved? The increased performance is due to the way evolution has exploited the resistance of the analogue switches. This configuration will give a good voltage swing if the path between the positive supply and the point marked '\*' on the diagram is of high resistance compared with the path between that point and 0V, when the transistor is switched on. Evolution has achieved this in a small measure by moving the output connection slightly, but mainly by exploiting a fundamental law concerning

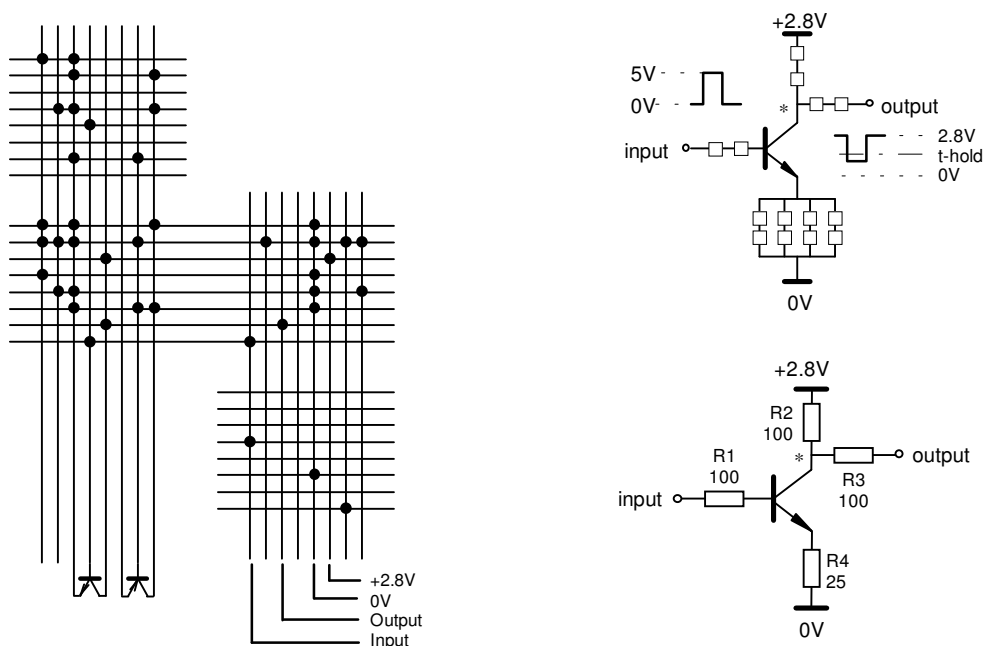


Figure 6.3 Evolved NOT Gate after 467 generations

(Above left): Instantiation on the evolved testbed

(Top right): The equivalent circuit. By connecting the emitter to ground via several switches, evolution has effectively lowered their resistance. In fact there are also several indirect paths from emitter to ground. They are omitted on this representation for clarity.

(Lower right): A conventional circuit diagram representation where the switches are replaced by their equivalent combined resistance.

resistance i.e. that resistors placed in series increases the combined resistance, but placing them in parallel *reduces* it.

Referring to the left diagram in figure 6.3 it can be seen that the portion of the testbed used for this experiment allows up to eight paths between external connections (i/o and power) and the transistors. The evolved circuit has had to use three for connecting input, output and +2.8V, but it has dedicated all remaining paths to the 0V connection in order to achieve parallel resistance. This has important implications concerning genotype-phenotype mapping. Evolution would not have been able to make such good use of parallel resistance if some topology mapping were used in place of direct mapping.

Figure 6.4 shows plots of both average population fitness and best individual fitness as evolution proceeds. While a gradual - though jagged - improvement is observed in the best individual, the population is of less average fitness than the original hand-seeded design. There are two possible explanations for this - either the biased mutation rate is badly chosen, or more likely, the hand-seeded gate belongs to a

very rugged region in the fitness landscape which is very sensitive to mutation. Whether this poor graph is due to NOT gates being an unsuitable function for evolution, or whether the design I have imposed amounts to a bad 'human' constraint can only be answered by further experimentation, where evolution must be allowed to find its own prototype.

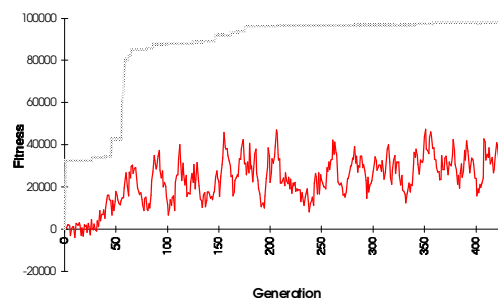


Figure 6.4

Fitness scores for the hand-seeded NOT gate. The upper plot (grey) represents best individual fitness, and the lower (black) represents the average population fitness.

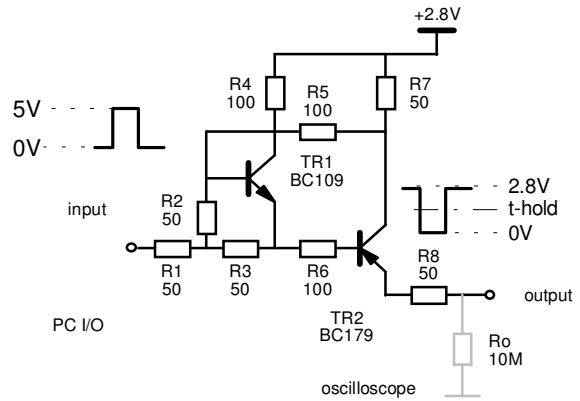
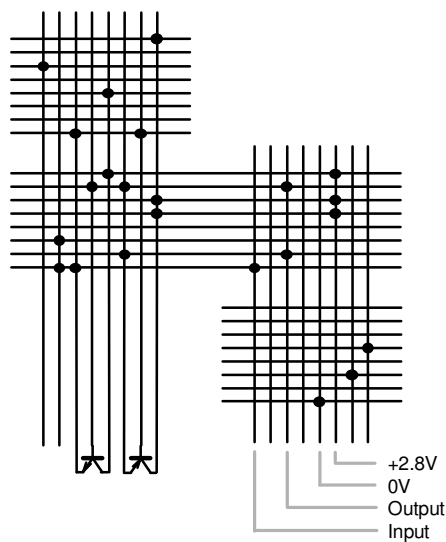


Figure 6.5  
The NOT gate evolved from scratch after 2500 generations. TR2 is configured in an entirely unconventional manner, and the circuit's only path to 0V is via the oscilloscope.

### 6.5 Evolving a NOT gate from scratch

One unexpected observation from various experimental runs with the hand-seeded gate was that the fitness function discourages circuits with many switches on. Although such circuits tend to short everything out leading in principle to the local optimum of constant output, the resistance of the switches invariably ensures that some small proportion of the input signal appears on the output line, resulting in slightly negative fitness scores. This observation was sufficiently encouraging to believe that evolving a NOT gate from scratch was a distinct possibility, since by the time evolution arrives at constant output, the proportion of switches off should be conducive to employing the transistors usefully, rather than just shorting them out. In this experiment, the initial population was randomly-seeded in the traditional manner, i.e. an equal proportion of '1's and '0's. However biased mutation was retained. Evolution continued for many generations with no result, but eventually a circuit giving fitnesses consistently greater than zero (but only just) arose, after which only a few more generations were required to produce fitness scores of around 133,000 - significantly better than had been achieved with the last experiment. Figure 6.5 shows the circuit.

This circuit is certainly unconventional. Although near-perfect behaviour was observed on the oscilloscope, it is not at all clear what is going on at first glance. Evolution has chosen *not* to make use of the 0V line, implying that the circuit should not work at all! In fact a path to 0V is achieved via the 10M input impedance of the oscilloscope ( $R_o$  on the diagram). TR2 is operating in reverse mode, and for best

operation,  $R_7$  should be small compared to  $R_o$ , which is evidently why evolution has exploited the oscilloscope in this manner. If the scope is unplugged, the circuit ceases to work. While TR1 seems meaningfully connected, it has no obvious function, and indeed unplugging it has no apparent effect on fitness scores.

The circuit is not exploiting any subtle properties particular to the transistors with which it evolved. It gave almost identical fitness scores when the transistor board was unplugged and substituted with other boards. The fitness plot of figure 6.6 shows some significant differences compared with that of the hand-seeded gate.

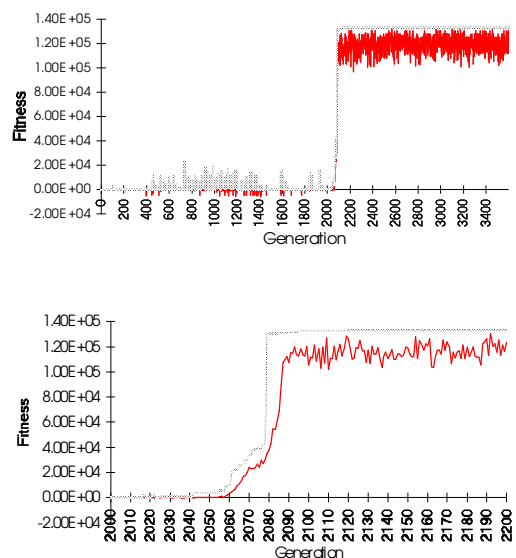


Figure 6.6 Fitness plots for the NOT gate.  
(upper): The complete evolutionary plot;  
(lower): The 200 generations which lead from prototype to maximum fitness.

Generations 0 - 400 consist of circuits which shorted out the transistors, and had average values of -350. Up to generation 2000, individuals appear with higher fitness values than is consistent with noise, but these elite individuals did not perform well when evaluated in the next generation. Around generation 2030 the best individual gets consistent fitness scores just above zero, and near-perfect fitness is achieved just 50 generations later. The average population fitness increases gradually over this period, and remains fairly close to the best fitness. This was not the case with the hand-seeded NOT gate. It seems that when left unconstrained, evolution has settled on a much wider plateau in the fitness landscape than it was able to when forced with the human-designed local optimum, resulting in better tolerance to mutation. It may be that TR1 is instrumental in this tolerance, though more analysis is necessary to confirm this.

### 6.6 An alternative mapping

The circuit was re-evolved several times using generation 2040 as the initial population. Each time, near-perfect fitness was achieved after about 50 generations. 2000 generations, i.e. up to 100000 different attempts may seem excessively many before the first stable prototype solution is found, but even with the small portion of the testbed used, the search space is very large at  $10^{77}$  possible circuits. One reason for the amount of generations required before positive fitness values are achieved is that a NOT gate cannot be produced with resistive elements alone - it requires the transistors, but these are shorted out at the start. The final experiment investigates an alternative genotype-phenotype mapping intended to prevent shorting occurring at the outset. This mapping allows only a limited number of switches per row to be set.

Figure 6.7 shows how the new mapping is encoded, and how this translates to the physical circuit. The basic idea is to limit the quantity of switches on per row, so that the pins of active components are not highly connected. This is consistent with many conventional circuits where each pin is usually only connected to two or three other pins, however it is not a knowledge-specific constraint, since the mapping does not limit the number of switches per column. In the encoding, each column is assigned a corresponding row. This can be observed in figure 6.7: taking the first row/column to be the top right, then Vcc (col 2) is assigned to row 2, Out (col 4) row 4, etc.. The

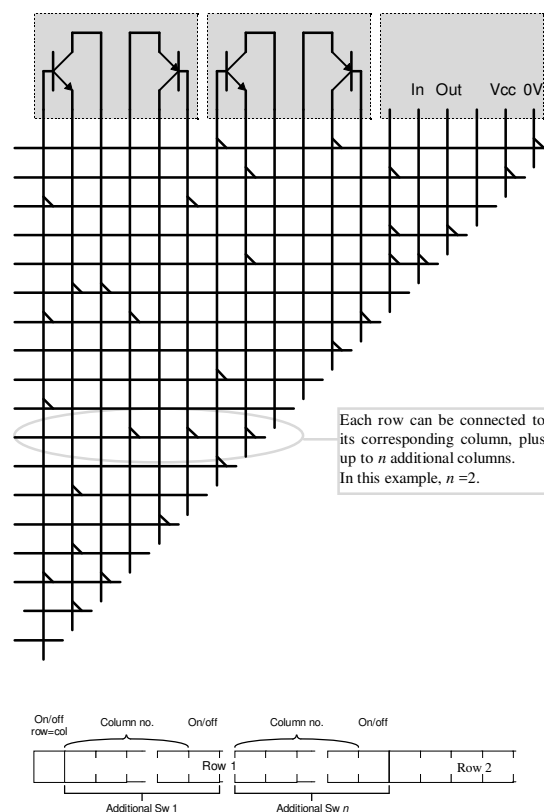


Figure 6.7. An alternative to direct mapping. In this scheme, a switch is specified by a number of address bits signifying the column, plus an additional bit signifying the state of that switch.

genotype represents the switches a row at a time. For each row, one bit specifies connection to the corresponding column, followed by column address and connection bits for up to  $n$  additional switches.

While the task remains the fairly simple NOT function, the third experiment uses the whole complement of the evolvable motherboard, with ten transistors on a 48 x 48 wire matrix. Thus, as well as exploring the new encoding, the experiment will give an indication of whether or not evolution will exploit all the resources open to it, since ten transistors would normally be considered an excessive amount for a NOT gate. Coding with  $n = 3$  requires a genotype length of 1056. Although fairly large, such gene lengths have not presented problems on the FPGA experiments detailed in section 3, and it is important to determine that the same is true for the evolvable motherboard. A standard GA was used with rank-based selection, single-point crossover, elitism and mutation rate of 8 bits per genotype. The fitness function was the same as



for previous experiments, however the normalisation factor was changed from 1/5 to 1/250 to reflect the maximum voltage swing in millivolts.

Figure 6.8 shows the fitness plots over 4000 generations and the final elite evolved circuit. The first prototype gate is evolved after only 200 generations, and by 600 generations, a high fitness value is attained. Thereafter a slow, gradual improvement occurs.

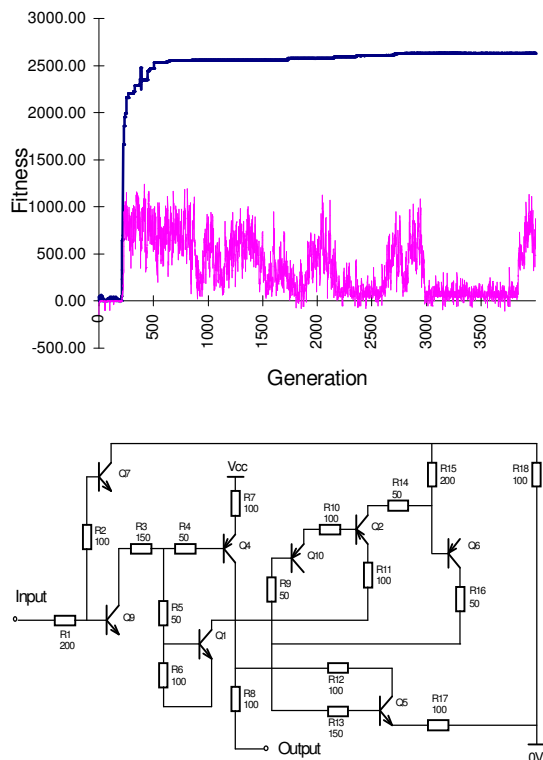


Figure 6.8  
Fitness plots and circuit diagram for a NOT gate evolved with a new genotype encoding. In the diagram, resistors represent analogue switches.

Given that the mutation rate as a percentage of genotype length is similar to that of the previous two experiments, it is interesting that the average population fitness is low, particularly so between generations 3000-3700 - probably because a mutation with the new encoding has a far more drastic effect than with direct mapping. The final circuit exploits eight out of the ten transistors. Each of these when unplugged results in circuit failure. The circuit has many feedback loops, with most of the transistors only partially connected and acting as diodes.

### 6.6.1 Fault tolerance in the population

Certain aspects of the evolved circuit run counter to intuition, in particular the fact that so many transistors are necessary. This is not due to increased performance, which is no better than for the previous circuit (although it does not rely on the oscilloscope!). Neither does the circuit exhibit any apparent tolerance to mutations or faults. Indeed one would expect that a circuit using as few components as possible would emerge, since mutation should have fewer deleterious effects on such a circuit. One possible reason for the evolved topology is suggested by a follow-up experiment performed with the population of 50 individuals in the generation that produced the circuit. This experiment consisted simply of removing one of the transistors from the motherboard, and re-evaluating all individuals. The result was that no matter which transistor was removed, at least one individual from the population scored higher than 75% of maximum possible fitness. Indeed the same result occurred in almost every case when two transistors at a time were removed. Since the population has reached convergence, all individuals have a semblance to the circuit of figure 6.8, and this must be a major factor in its topology. The population is fault/mutation tolerant while the elite individual is not. If this is true of evolved circuits in general, rather than being due to the encoding or the search space for this one task, then there are implications for the portability issue discussed in section 4.2.2. Evolved circuits may be made to be portable for a manufacturing process which implements a population, rather than a single individual.

## 7 Conclusion

The experiments of section 6 show that the evolvable motherboard can be used to investigate many important issues arising in current EHW research, including analysis; fault-tolerance; genotype encoding; portability; building blocks, and evolved topologies. The genotype-phenotype mapping of section 6.6 is one of many that are possible on the motherboard, due to the array of wires allowing any combination of pin connections. Mapping is analogous to the architecture of some configurable device, and experimentation/analysis of various mappings is likely to lead to a picture of the sort of architecture that an FPGA ideally suited to evolution should have. While section 6 includes suggestions to account for the phenomena observed and their possible implications, it should be noted that they are not, nor are they intended to be, conclusive. The experiments are presented primarily to illustrate the capabilities of the evolvable motherboard as a research tool. Firm conclusions from ongoing experimentation will be presented in a future publication.

Research detailed in this report continues to reveal positive and negative characteristics that seem to be shared by all evolved circuits. Whatever the benefits they offer, one principal characteristic is that they are very difficult to analyse due to the non-standard ways in which components are exploited and numerous feedback loops that make functional decomposition difficult or impossible. Whilst I must concur that analysis of very large complex evolved circuits should probably not be attempted at least until convenient methods become available, a methodical approach with indubitable conclusions must be taken if this field is to progress as a credible engineering tool. It seems that much of the current literature, though yielding impressive results, is supported by arguments which are at best inconclusive, and at worst somewhat 'hand-waving'. At such an early stage this is understandable, and it is true that getting bogged down in detailed analysis may slow down the acquisition of useful results without necessarily revealing any firm principles of hardware evolution. Thus continued research on simple, albeit unimpressive circuits is a major factor on the development of EHW. Analysis of such circuits is far from impractical, and is likely to contribute to the understanding of the properties that evolution can and cannot exploit, and why.

## Bibliography

- Bennet, F. H., Koza, J. R., Andre, D., Keane, M. A. (1996), "Evolution of a 60 Decibel Op Amp Using Genetic Programming". *International Conference on Evolvable Systems: from Biology to Hardware*. Japan.
- Bratt, S. & Macbeth, I. (1996) "*Design and Implementation of a Field Programmable Analogue Array*", Technical report, ACM 0-89791-773-1.
- Castro, H, Gough, M.P., "Dynamic Reconfigurability with Fault Tolerance", ENGGRP, University of Sussex.
- De Garis, H. (1993), "Growing an artificial brain with a million neural net modules inside a trillion cell cellular automaton machine". *Proc. 4<sup>th</sup> International Symposium on Microe Machine and Human Science*, pp. 211-214.
- Eigen, M. (1987), "New concepts for dealing with the evolution of nucleic acids". In *Cold Spring Harbor Symposia on Quantitative Biology*, Vol LII. Cold Spring Harbor Laboratory.
- Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966), "*Artificial Intelligence Through Simulated Evolution*". John Wiley & Sons, Inc.
- Grefenstette, J. J. (1996), "Genetic Learning for Adaptation in Autonomous Robots". *Robotics and manufacturing: Recent Trends in Research and Applications*. Vol. 6, ASME Press, New York.
- Harvey, I., Husbands, P., Cliff, D. (1994), "Seeing the light: Artificial Evolution, real vision. In Cliff, D. et al. (Eds.), *From Animals to Animats 3: Proc of 3<sup>rd</sup> Int. Conference on simulation of adaptive behaviour*, pp392-401. MIT Press.
- Hemmi, H., Mizoguchi, J. Shimonara. (1996), "Development and Evolution of Hardware Behaviors", *Lecture Notes in Computer Science - Towards Evolvable Hardware*, Vol. 1062, pp. 250 - 265, Springer-Verlag.
- Higuchi, T., Iwata, M.Kajitani, L., Iba, H., Hirao, Y.,Furuya, T., Manderick, B.(1996), "Evolvable Hardware and Its Applications to Pattern Recognition and Fault-Tolerant Systems", *Lecture Notes in Computer Science -Towards Evolvable Hardware*, Vol. 1062, pp. 118-135, Springer-Verlag.

- Hillis, W. D. (1992), "Co-evolving parasites improve simulated evolution as an optimization procedure". In Langton, C. et al, (Eds.) *Artificial Life II*, pp. 313-324. Addison-Wesley.
- Holland, J. H. (1975), "*Adaptation in Natural and Artificial Systems*", Ann Arbor: The University of Michigan Press.
- Horowitz, P., & Hill, W. (1989), "*The Art of Electronics*" 2<sup>nd</sup> Ed. Cambridge University Press.
- Huynen, M. A., & Hogeweg, P. (1994), "Circle in the round: State space attractors for evolved sighted robots". *Robotics and Autonomous Systems*, 15, 83-106.
- Jakobi, N. "Half-baked, Ad-Hoc, and Noisy: Minimal Simulations for Evolutionary Robotics", In Husbands, P., & Harvey, I., (Eds.) *Proc, Fourth European Conference on Artificial Life*, MIT Press.
- Kaufmann, S. A. (1993). "*The Origins of Order*". Oxford University Press.
- Keymeulen, D., Konaka, K., Iwata, M., Kuniyoshi, Y., Higuchi, T. (1997), "*Off-line Evolution for a Robot Navigation System based on a Gate-Level Evolvable Hardware*", Technical report, Electrotechnical Laboratory, Ibaraki 305, Japan.
- Kitano, H. (1996), "Morphogenesis of Evolvable Systems", *Lecture Notes in Computer Science - Towards Evolvable Hardware*, Vol. 1062, pp. 99-117, Springer-Verlag.
- Koza, J. R. (1992), "*Genetic Programming: On the programming of computers by means of natural selection*". MIT Press, Cambridge, Mass.
- Koza, J. R., Bennet, F. H., Andre, D., Keane, M. A. (1996a), "Towards Evolution of Electronic Animals Using Genetic Programming", *Artificial Life V Conference* in Nara, Japan, May 16-18.
- Koza, J. R., Bennet, F. H., Andre, D., Keane, M. A. (1996b), "Reuse, Parameterized Reuse, and Hierarchical Reuse of Substructures in Evolving Electrical Circuits Using Genetic Programming", *International Conference on Evolvable Systems: from Biology to Hardware*. Japan.
- SGS-Thompson (1989) *BC107, BC108-BC109 Low noise general purpose audio amplifiers*, datasheet, SGS-Thompson Microelectronics,
- Salami, M., Iwata, M., Higuchi, T. (1997), "Lossless ImageCompression by Evolvable Hardware". In Husbands, P., & Harvey, I., (Eds.) *Proc, Fourth European Conference on Artificial Life*, MIT Press.
- Thompson, A. (1995a) Evolving fault tolerant systems. In *Proc. 1<sup>st</sup> IEE/IEEE Int. Conf. On Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA '95)*, pp 524-529. IEE Conf. Publication No. 414.
- Thompson, A. (1995b), "Evolving electronic robot controllers that exploit hardware resources". In Moran, F., et al. (Eds.), *Advances in Artificial Life: Proc. 3<sup>rd</sup> Eur. Conf on Artificial Life (ECAL95)*, Vol.929 of LNAI, pp. 640-656. Springer-Verlag.
- Thompson, A. (1996a) "*Hardware Evolution: Automatic design of electronic circuits in reconfigurable hardware by artificial evolution*", DPhil Thesis, University of Sussex.
- Thompson, A. (1996b) "Silicon Evolution", In Koza, J. R. et al (Eds.), *Genetic Programming 1996: Proc. 1<sup>st</sup> Annual conference*, pp 444-452. Cambridge, AM: MIT Press.
- Thompson, A. (1996b) "An evolved circuit, intrinsic in silicon, entwined with physics", *International Conference on Evolvable Systems (ICES96)*, Springer LNCS.
- Thompson, A. (1997) "Temperature in Natural and Artificial Systems", In Husbands, P., & Harvey, I., (Eds.) *Proc, Fourth European Conference on Artificial Life*, MIT Press.
- Worden, R. P. (1995) "A Speed Limit for Evolution", *Journal of Theoretical Biology*, 176, pp. 137 - 152.
- Zebulum, R., Pacheco, M. A., Vellasco, M (1996) "Evolvable Systems in Hardware Design: Taxonomy, Survey and Applications", *International Conference on Evolvable Systems: from Biology to Hardware*. Japan.