On the Practical Limits of the Evolutionary Digital Filter Design at the Gate Level

Lukáš Sekanina and Zdeněk Vašíček

Faculty of Information Technology, Brno University of Technology, Božetěchova 2, Brno 612 66, Czech Republic sekanina@fit.vutbr.cz, xvasic11@stud.fit.vutbr.cz

Abstract. Simple digital FIR filters have recently been evolved directly in the reconfigurable gate array, ignoring thus a classical method based on multiply–and–accumulate structures. This work indicates that the method is very problematic. In this paper, the gate-level approach is extended to IIR filters, a new approach is proposed to the fitness calculation based on the impulse response evaluation and a comparison is performed between the evolutionary FIR filter design utilizing a full set and a reduced set of gates. The objective of these experiments is to show that the evolutionary design of digital filters at the gate level does not produce filters that are useful in practice when linearity of filters is not guaranteed by the evolutionary design method.

1 Introduction

FIR (finite impulse response) filters and IIR (infinite impulse response) filters represent two important classes of digital filters that are utilized in many applications. For these filters, a rich theoretical understanding as well as practical design experience have been gained in the recent decades [6]. Typically, their implementation is based on *multiply-and-accumulate* structures (regardless on software or hardware implementation). Alternative design paradigms (such us multiplierless designs) have also been formulated [7].

With the development of real-world applications of evolutionary algorithms, researchers have started to evolve digital filters. Miller has introduced probably the most radical idea for their design [9, 10, 11]: In evolutionary design process, target filters are composed from elementary gates, ignoring thus completely the well-developed techniques based on multiply-and-accumulate structures. The main practical potential innovation of this approach could be that the evolved filters are extremely area-efficient in comparison with the standard approach. We should understand this approach as a demonstration that the evolution is capable to put some gates together in order to perform a very simple filtering task. Definitely, the approach is not able to compete against the standard methods. A similar approach has been adopted for functional level evolution of IIR filters [3].

In contrast to an optimistic view presented in the mentioned papers, this work indicates that the approach is very problematic. In this paper, Miller's

[©] Springer-Verlag Berlin Heidelberg 2006

gate-level approach is extended to IIR filters, a new approach to the fitness calculation is proposed based on the impulse response evaluation and a comparison is performed between the evolutionary FIR filter design utilizing a full set of gates and reduced set of gates. The objective of these experiments is to support the following hypothesis: "Evolutionary design of digital filters at the gate level does not produce filters that are useful in practice when linearity of filters is not guaranteed." Two approaches that could ensure the linear behavior of evolved filters will be discussed.

The rest of the paper is organized as follows. Section 2 introduces the area of digital filter design and the use of evolutionary techniques in this area. In Section 3, the proposed approach is described to the evolutionary design of FIR and IIR filters. Results of experiments are reported in Section 4. Section 5 discusses the advantages and disadvantages of the proposed approach. Conclusions are given in Section 6.

2 Conventional and Evolutionary Design of Digital Filters

A discrete-time system is essentially a mathematical algorithm that takes an input sequence, x(n), and produces an output sequence, y(n) [6]. A digital filter is an example of discrete-time system. A discrete-time system may be linear or nonlinear, time invariant or time varying. Linear time-invariant (LTI) systems form an important class of systems used in digital signal processing.

A discrete-time system is said to be linear if it obeys the principle of superposition. Consider that $x_1(n)$ and $x_2(n)$ are two input signals and $y_1(n)$ and $y_2(n)$ are corresponding responses of the filter. The filter is *linear* if the following holds:

$$a_1 x_1(n) + a_2 x_2(n) \to a_1 y_1(n) + a_2 y_2(n)$$
 (1)

where a_1 and a_2 are arbitrary constants.

A discrete-time system is said to be *time-invariant* if its output is independent of the time the input is applied, i.e. a delay in the input causes a delay by the same amount in the output signal.

The input–output relationship of an LTI system is given by the convolution sum

$$y(n) = \sum_{k=-\infty}^{+\infty} h(k)x(n-k)$$
(2)

where h(k) is the *impulse response* of the system. The values of h(k) completely define the discrete-time system in the time domain.

A general IIR (infinite impulse response) digital filter is described by equation

$$y(n) = \sum_{k=0}^{N} b_k x(n-k) - \sum_{k=1}^{M} a_k y(n-k).$$
 (3)

The output samples y(n) are derived from current and past input samples x(n) as well as from current and past output samples. Designer's task is to

propose values of coefficients a_k and b_k and size of vectors N and M. In FIR (finite impulse response) filters, the current output sample, y(n), is a function only of past and present values of the input, i.e.

$$y(n) = \sum_{k=0}^{N} b_k x(n-k).$$
 (4)

The stability and linear phase are main advantages of FIR filters. On the other hand, in order to get a really good filter many coefficients have to be considered in contrast to IIR filters. In general, IIR filters are not stable (because of feedback). FIR filters are algebraically more difficult to synthesize.

Various methods have been proposed to design digital filters (such as frequency sampling method and window method for FIR filters and pole/zero placement and bilinear z-transform for IIR filters). These methods are well developed and represent an approach to digital filter design widely adopted by industry. Digital filters are usually implemented either on DSPs or as custom circuits. Their implementation is based on multipliers and adders. The quality of output signal, speed of operations and cost of hardware implementation are important factors in the design of digital filters. The multiplier is the primary performance bottleneck when implementing filters in hardware as it is costly in terms of area, power and signal delay. Hence multiplierless filters were introduced in which multiplication is reduced to a series of bitshifts, additions and subtractions [12, 7].

Evolutionary algorithms have been utilized either to optimize filter coefficients [4] or to design a complete filter from chosen components. In particular, structures of multiplierless filters were sought by many authors [12, 5, 1]. As these filters are typically composed of adders, subtracters and shifters (implementing multiplication/division by the powers of two) they exhibit "linear" behavior for the required inputs.

Miller has pioneered the evolutionary approach in which FIR filters are constructed from logic gates [9, 10, 11]. He has used an array of programmable gates to evolve simple low-pass, high-pass and band-pass filters that are able to filter simple sine waves and their compositions. The unique feature of these filters is that they are composed of a few tens of gates; thus reducing the implementation costs significantly in comparison with other approaches. The evolved filters do not work perfectly and they are far from the practical use; however, Miller has demonstrated that *quasi-linear* behavior can be obtained for some particular problems. The gate arrays are carrying out filtering without directly implementing a *difference equation* – an abstract model utilized for filter design. The fitness function can be constructed either in the frequency domain or time domain. In both cases Miller has obtained similar results. However, he mentioned that: "Experience suggests that gate arrays that are evolved using a fitness function which looks at the frequency spectrum of the circuit output appear to be more linear in behavior than using an error based measure of fitness" [10]. Recently Gwaltney and Dutton have utilized similar approach to evolve IIR filters at the functional level (for 16bit data samples) [3]. Their filters are composed of adders, multipliers and some logic functions; therefore, they are non-linear. They have evolved low-pass IIR filters using a couple of components. The filter fails to function properly when the input is changed to a signal that is "significantly" different from that used during evolution.

3 Gate-Level Evolution of Digital Filters

Similarly to Miller [9, 10, 11], Cartesian genetic programming (CGP) is utilized in this paper to evolve simple digital filters at the gate level. Although all internal data are at 1 bit and gates perform elementary logic operations, the inputs as well as outputs are interpreted as 8 bit values.

3.1 Cartesian Genetic Programming

CGP models a reconfigurable circuit, in which digital circuits are evolved, as an array of u (columns) $\times v$ (rows) of programmable elements (gates) [8]. The number of circuit inputs n_i and outputs n_o are fixed. Each gate input can be connected to the output of some gate placed in the previous columns or to some of the circuit inputs. *L*-back parameter defines the level of connectivity and thus reduces/extends the search space. For example, if L=1 only neighboring columns may be connected; if L=u, the full connectivity is enabled. A circuit configuration is defined using $3.u.v + n_o$ integers: the three integers describe the connection and function of a single gate and n_o integers specify the connection of outputs. Every gate performs one of functions specified in function set F. Figure 1 provides an example.

Miller has originally used a very simple variant of evolutionary algorithm to produce configurations for the programmable circuit [8]. Our algorithm is very similar. It operates with the population of 5 individuals; every new population consists of mutants of the best individual. Only the mutation operator has been utilized that modifies 1–3 randomly selected genes of an individual.



Fig. 1. An example of a 3-input circuit. CGP parameters are as follows: L = 3, u = 3, v = 2, $F = \{AND (0), OR (1)\}$. Gates 5 and 7 are not utilized. Chromosome: 1,2,1, 0,0,1, 2,3,0, 3,4,0 1,6,0, 0,6,1, 6, 8. The last two integers indicate the outputs of the circuit.

3.2 Gate-Level Digital Filters and CGP

Figure 2a shows our modification of CGP to design FIR filters. A delay chain was created using two registers. We can observe that the three w-bit samples are processed by the gate array (w = 8). Before simulation is started, delay registers are cleared. The following operations are repeated N-times (N is the number of samples): the *i*th sample is right-shifted by w bits and the (i + 1) sample possess its position. Then the output value is calculated and interpreted as an integer value.

Figure 2b shows the approach utilized to evolve IIR filters using CGP. In addition to the previous approach, the output delay registers have to be shifted to send the obtained output value back to the circuit. Because of the feedback, the IIR filter simulation is much slower than the FIR filter simulation.

The proposed fitness function works in the time domain. The objective is to minimize the difference between measured signal y(n) and target signal $y_{ref}(n)$, i.e.

$$fitness_{MSE} = -\sqrt{\sum_{i=0}^{N-1} (y(i) - y_{ref}(i))^2}.$$
 (5)

The main problem is to determine which signals should be included into the training set. Ideally, all frequencies and shapes should be testes; however, it is not tractable.

An alternative approach could be to apply the unit impulse (i.e. the signal that contains all frequencies) at the input and to measure the impulse response. This



Fig. 2. CGP utilized to design (a) FIR filters and (b) IIR filters



Fig. 3. Comparison of 2's complement encoding (a) and fraction arithmetic (b)

approach will be utilized to design IIR filters. The role of CGP is to find such the filter whose impulse response is as close as possible to the required impulse response. We have to work with real numbers because the values of impulse responses range from -1 to +1. In our case we will represent real numbers in the fraction arithmetic (which is based on 2's complement encoding as shown in Figure 3).

4 Results

The following CGP parameters represent the basic setup for experiments: $u = 15, v = 15, L = 15, n_i = 24, n_o = 8$, population size 5, 15 million generations, function set: $F = \{c = a, c = a \text{ and } b, c = a \text{ or } b, c = a \text{ xor } b, c = \text{ not } a, c = \text{ not } b, c = a \text{ and } (\text{not } b), c = a \text{ nand } b, c = a \text{ nor } b\}$. CGP was implemented in C++. The evolved filters were analyzed using Matlab.

4.1 Low-Pass Filter I

The training input signal consists of composition of frequencies f_1 and $f_3 = 3f_1$. As the circuit should carry out low-pass filter, the output should contain f_1 only (which will be expressed in this paper as: $f_1 + f_3 \rightarrow f_1$). In particular, we utilized N = 128 samples and $x_1(n) = 127 + 100.\sin(2\pi n/128)$ and $x_3(n) = 127 + 100.\sin(2\pi .3n/128)$.



Fig. 4. Behavior of the evolved low-pass filter: input signal (left), required output signal (center), output signal (right); (a) training signal, (b) test signal



Fig. 5. Behavior of the evolved high-pass filter: input signal (left), required output signal (center), output signal (right); (a) training signal, (b) test signal

Figure 4 shows the behavior of the best evolved filter. We utilized the signal with f_1 as a test signal and observed that the evolved circuit modifies the signal although it should transmit the signal without any change. Therefore, the circuit cannot be understood as a perfect filter.

4.2 High-Pass Filters

Figure 5 shows behavior of an evolved high-pass filter whose function was specified as $f_1 + f_{10} \rightarrow f_{10}$. Although the result for the training signal seems to be correct, the filter does not work for other signals at all.

4.3 Low-Pass Filter II

In order to evolve more robust filters, we included more requirements to the fitness function. A low-pass filter was specified as: $f_1 + f_3 \rightarrow f_1$, $f_1 + f_5 \rightarrow f_1$ and $f_5 \rightarrow f_0$. The evolved filter exhibits an acceptable (but not perfect) behavior for training signals as well as test signals (see Figure 6). This approach could eventually be utilized to design an extremely cheap filter which is supposed to work for a limited amount of input signals.

4.4 The Impulse Response in Fitness Calculation

In this experiment, we defined the required impulse response and were interested whether CGP is able to find an IIR filter with the same impulse response. Figure



Fig. 6. Behavior of the evolved low-pass filter II: input signal (left), required output signal (center), output signal (right)



Fig. 7. The impulse response: (a) required (b) evolved



Fig. 8. The input signal (left) and the "filtered" signal obtained using the evolved filter (right)

7a and 7b show that it is possible and furthermore, the corresponding frequency characteristic is also very close to the perfect one.

Unfortunately, the evolved circuit is not a filter at all. Figure 8 shows its response to a simple sine input signal. The output signal should exhibit a very small amplitude; however, its behavior is completely random.

4.5 A Reduced Set of Gates

In cellular automata and other circuits, a "linear" variant of the model is sometimes introduced (see, for example, linear cellular automata [2]). Then, logic circuits are composed using the gates *xor* and *not* because their analysis is amenable to algebraic methods. In order to explore whether this type of linearity could be useful for gate-level filters, we arranged the following experiments. The objective was to evolve a high-pass FIR filter specified as $f_1 + f_{10} \rightarrow f_{10}, f_3 + f_{10} \rightarrow$ $f_{10}, f_5 + f_{10} \rightarrow f_5 + f_{10}, f_{10} \rightarrow f_{10}$. We utilized CGP with u = 15 and v = 15and produced 20 millions of generations. The evolved filter was tested using signals: $f_{10}, f_1, f_3, f_4, f_5, f_0$. We have considered two scenarios: (1) a complete set of gates, F, and (2) a reduced set of gates, F', containing {xor, not}. Table 1 summarizes the obtained results. We can observe that the circuits composed of *xor* and *not* gates (scenario 2) are much smaller and more general than those obtained in case (1). On the other hand, the filter evolved using a complete set

Table 1. Filters evolved using the complete set of gates and reduced set of gates

Filter	MSE (training data)	MSE (test data)	# of used gates
Complete set (scenario 1)	132,583	738,075	197
Reduced set (scenario 2)	$369,\!275$	525,700	44



Fig. 9. Signal $f_4 = x(n)$: The outputs of filters evolved with the compelte set of gates (EvoFilter 1) and reduced set of gates (EvoFilter 2)

of gates is more adapted to training signals. However, Figure 9 shows that the output response is acceptable neither for scenario 1 nor 2.

5 Discussion

The common result of experiments performed herein, by Miller [9, 10, 11] and by Gwaltney and Dutton [3] is that the evolved filters do not work when they are required to filter signals *different* from training signals. Moreover, the evolved filters do not generate perfect responses either for training signals. The evolved circuits are not, in fact, filters. In most cases they are combinational circuits trained on some data that are not able to generalize. In order to obtain real filters, the design process must guarantee that the evolved circuits are *linear*. There are two ways how to ensure that: (1) The circuit is composed of components that are linear and the process of composition always ensures a linear behavoir. This approach is adopted by many researchers (e.g., [1]) but not by the methods discussed in this paper. (2) Linearity is evaluated in the fitness calculation. Unfortunately, that is practically impossible because all possible input signals should be considered, which is intractable. Note that Miller's fitness function [11] has promoted the filters exhibiting the quasi-linear behavior; however, it does not guarantee (in principle) that a candidate filter is linear although the filter has obtained a maximum fitness score.

In this paper, we have evolved FIR as well as IIR "filters" and proposed the approaches based on the impulse response and the reduced set of (linear) gates. However, none of them have led to satisfactory results. On the basis of experiments performed in this paper and results presented in [9, 10, 11], we are claiming that the gate level evolutionary design of digital filters is not able to produce filters "useful" in practice if the linearity is not guaranteed by the evolutionary design process. As we do not know now how to ensure the linear behavior, the approach should be considered as curious if one is going to design a digital filter.

There could be some benefits coming with this "unconventional" filter design. It was shown that circuits can be evolved to perform filtering task when sufficient resources are not available (e.g. a part of chip is damaged) [5] or when some noise in presented in input signals [10]. Furthermore, as Miller has noted, "The origin of the quasi-linearity is at present quite mysterious. ... Currently there is no known mathematical way of designing filters directly at this level." Possibly we could discover novel design principles by analyzing the evolved circuits.

The evolutionary design is very time consuming. In order to produce 20 millions of generations with a five-member population, the evolutionary design requires 29.5 hours for IIR filter and 6 hours for FIR filter (on a 2.8 GHz processor).

6 Conclusions

In this paper, the gate-level approach to the digital filter design was extended to IIR filters, a new approach was proposed to the fitness calculation based on the impulse response evaluation and a comparison was performed between the full set of gates and reduced set of gates for the evolutionary FIR filter design. On the basis of experiments performed herein and the results presented in literature we have recognized that the gate level evolutionary design of digital filters is not able to produce "real" filters. Therefore, this approach remains a curiosity rather than a design practice.

Acknowledgment

The research was performed with the Grant Agency of the Czech Republic under No. 102/04/0737 Modern Methods of Digital Systems Synthesis.

References

- Erba, M., et al.: An Evolutionary Approach to Automatic Generation of VHDL Code for Low-Power Digital Filters. In: Proc. of the 4th European Conference on Genetic Programming, LNCS 2038, Springer-Verlag, 2001, p. 36–50
- [2] Ganguly, N. et al. A survey on cellular automata. Technical report, Centre for High Performance Computing, Dresden University of Technology, December 2003
- [3] Gwaltney, D., Dutton, K.: A VHDL Core for Intrinsic Evolution of Discrete Time Filters with Signal Feedback. In Proc. of 2005 NASA/DoD Conference on Evolvable Hardware, IEEE Comp. Society Press, 2005, p. 43–50

- [4] Harris, S. P., Ifeachor, E. C.: Automating IIR filter design by genetic algorithm. Proc. of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95), No. 414, IEE, London, 1995, p. 271–275
- [5] Hounsell, B. I., Arslan, T., Thomson, R.: Evolutionary design and adaptation of high performance digital filters within an embedded reconfigurable fault tolerant hardware platform. Soft Computing. Vol. 8, No. 5, 2004, p. 307–317
- [6] Ifeachor, E. C., Jervis, B. W.: Digital Signal Processing: A Practical Approach, 2nd edition, Pearson Education, 2002
- [7] Martinez-Peiro, M., Boemo, E. I., Wanhammar, L.: Design of High-Speed Multiplierless Filters Using a Nonrecursive Signed Common Subexpression Algorithm. IEEE Trans. Circuits Syst. II, Vol. 49, No. 3, 2002, p. 196-203
- [8] Miller, J., Job, D., Vassilev, V.: Principles in the Evolutionary Design of Digital Circuits – Part I. Genetic Programming and Evolvable Machines, Vol. 1, No. 1, 2000, p. 8–35
- [9] Miller, J.: Evolution of Digital Filters Using a Gate Array Model. In: Proc. of the Evolutionary Image Analysis, Signal Processing and Telecommunications Workshop. LNCS 1596, Springer-Verlag, 1999, p. 121–132
- [10] Miller, J.: On the filtering properties of evolved gate arrays. In Proc. of NASA/DOD Workshop on Evolvable Hardware, IEEE Comp. Society, 1999, p. 2–11
- [11] Miller, J.: Digital Filter Design at Gate-level using Evolutionary Algorithms. In Proc. of the Genetic and Evolutionary Computation Conference (GECCO'99). Morgan Kaufmann, 1999, p. 1127–1134
- [12] Wade, G., Roberts, A., Williams, G.: Multiplier-less FIR filter design using a genetic algorithm. IEE Proceedings in Vision, Image and Signal Processing, Vol. 141, No. 3, 1994, p. 175–180