Hardware Implementation of Evolutionary Digital Filters

doi: 10.1109/ISCAS.2003.1205857

HARDWARE IMPLEMENTATION OF EVOLUTIONARY DIGITAL FILTERS

Masahide Abe and Masayuki Kawamata

Department of Electronic Engineering,
Graduate School of Engineering, Tohoku University,
Aoba-yama 05, Sendai, 980-8579, Japan

ABSTRACT

This paper designs and implements a hardware-based evolutionary digital filter (EDF). The EDF is an adaptive digital filter which is controlled by adaptive algorithm based on evolutionary computation. The hardware-based EDF consists of two submodules, that is, a filtering and fitness calculation (FFC) module and a reproduction and selection (RS) module. The FFC module has high computational ability to calculate the output and the fitness value since its submodules run in parallel. A synthesis result of the designed chip shows the clock frequency is 20.0MHz and the maximum sampling rate of the EDF is 3.7kHz. Moreover, the hardware-based EDF with 21 submodules of the FFC is 2.2 times faster than the software-based EDF.

1. INTRODUCTION

Several researchers have proposed adaptive algorithms for digital filtering, which are all based on the Darwinian concept of "natural selection." These include the adaptive algorithm based on the genetic algorithm (GA) [1–3], the new learning adaptive algorithm [4], and Darwinian approach to adaptive notch filters [5].

The authors have already proposed evolutionary digital filters (EDFs) [6–8]. The EDF is an adaptive digital filter (ADF) which is controlled by adaptive algorithm based on evolutionary computation. The advantages of the EDF are summarized as follows:

1. The adaptive algorithm of the EDF is a population-based and robust optimization method, especially used to tackle high-dimensional and multi-modal search space problems. It is a non-gradient and multi-point search algorithm. Thus, it is not susceptible to local minimum problems that arise from a multiple-peak surface.
2. The EDF can adopt the various error functions as the fitness function according to application, for example, the p-power norm error function, the maximum error function and so on.
3. The adaptive algorithm of the EDF has a self-stabilizing feature whereby unstable poles have a tendency to migrate back into the stable region. In addition, the EDF can search the poles which are near the unit circle.

Numerical examples in Refs. [6–8] show that the EDF has a higher convergence rate and smaller steady-state value of the square error than the LMS adaptive digital filter (LMS-ADF).

However, the EDF has the following disadvantage: the number of multiplication of the EDF is greater than that of the LMS-ADF, since the EDF consists of many inner digital filters. Thus, we implement the EDF on parallel processors.

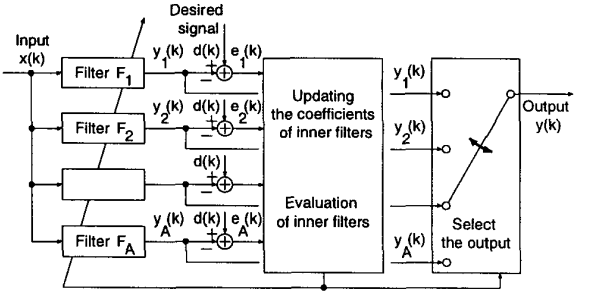

Figure 1: Block diagram of an evolutionary digital filter.

In order to implement the EDF in parallel, we present a hardware implementation of the distributed EDF, which consists of the modified structure and adaptive algorithm.

This paper is organized as follows: Section 2 summarizes the overall structure and the adaptive algorithm of EDFs. Section 3 describes the detailed structure of the proposed hardware-based EDF and its synthesis result. Section 4 gives concluding remarks.

2. EVOLUTIONARY DIGITAL FILTERING

In this section, we summarize the filter structure and the adaptive algorithm of EDFs. Figure 1 shows the block diagram of an EDF. The EDF consists of many linear/time-variant inner digital filters F_i's which correspond to individuals. Inner digital filter coefficients $\boldsymbol{W}$ which correspond to the feature of individuals are controlled by the following adaptive algorithm.

2.1. Adaptive Algorithm of Evolutionary Digital Filters

The adaptive algorithm of the EDF is similar in concept to GA. These concepts are based on the mechanics of natural selection and genetics to emulate the evolutionary behavior of biological systems. However, the adaptive algorithm of the EDF is different from the GA in the genetic operator and the representation of strings.

In the following sections, we use the following notations:

P population of individuals,
N the number of individuals.

The subscripts in the symbols P, N and $\boldsymbol{W}$ are denoted as follows:

0-7803-7761-3/03/$17.00 ©2003 IEEE

Table 1: Numbers of multiplications per iteration in the EDF and the LMS-ADF.

Algorithm (structure)	Number of multiplications for the filtering process	Number of multiplications for the adaptive process
EDF	$A(N+M+1)$	$(A - N_{ap} - \frac{1}{2}N_{sp})(N+M+1)/T_0$
LMS-ADF	$N+M+1$	$3N+M+2$

a the cloning method (the asexual reproduction),
s the mating method (the sexual reproduction),
p parent,
c offspring (child).

In the EDF, the adaptive algorithm updates the inner digital filter coefficients every T_0 samples. Thus, the relation between the generation t and the time k is given by

$$k = T_0 t,\ T_0 t + 1,\ \cdots,\ T_0 t + (T_0 - 1) \tag{1}$$

where k denotes the time in the filtering operation and T_0 denotes the period of the evaluation of one generation.

2.1.1. *Cloning Method*

Each parent in the population P_{ap}, with high fitness value within the population $P(t)$, creates the offspring population P_{ac} using the cloning method. In the cloning method, one parent creates N_{ac} offsprings, and forms a family $P_{af,i}$ which contains itself and its offsprings, where $i = 1, 2, \cdots, N_{ap}$. N_{ap} is the number of parents which use the cloning method. We assume that the proposed cloning method corresponds to transcribing the coefficient vector $\boldsymbol{W}_{ap,i}$ as the parent feature into coefficient vectors as the offspring feature $\boldsymbol{W}_{ac,i,j}$, where $i = 1, 2, \cdots, N_{ap}$, and $j = 1, 2, \cdots, N_{ac}$. Thus, the proposed cloning method updates the inner digital filter coefficients as individual feature according to

$$\boldsymbol{W}_{ac,i,j} \quad = \quad \boldsymbol{W}_{ap,i} + r \cdot \boldsymbol{n}_{i,j} \tag{2}$$

where the scalar r denotes the cloning fluctuation, and $\boldsymbol{n}_{i,j}$ is a Gaussian random variable vector with zero mean and unit variance.

In this algorithm, the cloning method corresponds to the local search. Therefore, this method is provided with the following strategy to select the candidate population for the next generation. In this method, one individual, of which fitness is maximum in each family $P_{af,i}$, is selected. These individuals form the candidate population P_a for the next generation. The population P_a of the best individuals is selected among each family $P_{af,i}$, that is, the coefficient vector of the inner filter with the highest fitness is selected among the $(N_{ac}+1)$ coefficient vectors. These coefficients are scattered on the narrow area. Thus, this operation corresponds to the local search.

2.1.2. *Mating Method*

If parents with low fitness value in population create the offsprings using the above cloning method, these offsprings may have low fitness value and can not be selected as candidates for the next generation. Therefore, parents in the population P_{sp}, with low fitness value within the population $P(t)$, create the offspring population P_{sc} using the mating method. $N_{sp}/2$ pairs among the N_{sp} parents are randomly selected for mating. In the mating method,

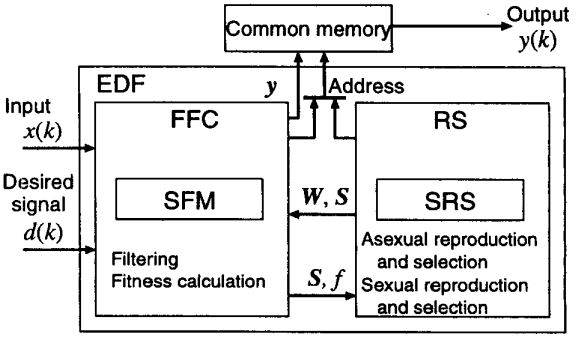

Figure 2: Block diagram of the hardware-based EDF.

each pair of parents creates one offspring, and they form a family $P_{sf,m}$ which contains themselves and their offspring, where $m = 1, 2, ..., N_{sp}/2$. We assume that the proposed mating method corresponds to calculating the middle point $\boldsymbol{W}_{sc,m}$ as the offspring feature of two coefficient vectors $\boldsymbol{W}_{sp,k(m)}$ and $\boldsymbol{W}_{sp,l(m)}$ as parent feature. Thus, this method updates the inner digital filter coefficients as individual feature according to

$$\boldsymbol{W}_{sc,m} = \frac{1}{2}(\boldsymbol{W}_{sp,k(m)} + \boldsymbol{W}_{sp,l(m)}) + s \cdot \boldsymbol{n}_m \tag{3}$$

where $k(m)$ and $l(m)$ are selected in $\{1, 2, ..., N_{sp}\}$ without duplicating, and $m = 1, 2, \cdots, N_{sp}/2$. The scalar s denotes the mating fluctuation, and $\boldsymbol{n}_m$ is a Gaussian random variable vector with zero mean and unit variance.

In this algorithm, the mating method corresponds to the global search and keeps various features of individuals. Therefore, this method is provided with the following strategy to select the candidate population for the next generation. In this method, one parent with higher fitness value in each family $P_{sf,m}$ is selected and the other parent dies out. In order to keep various features of individuals, the offspring in each family $P_{sf,m}$ is always selected regardless of their fitness values.

2.2. Computational Complexity

The EDF requires $(A - N_{ap} - \frac{1}{2}N_{sp})(N+M+1)/T_0$ multiplications per iteration for the adaptive process, where A is the total number of the evaluated individuals, that is, $A = N_{ap}(N_{ac}+1) + \frac{3}{2}N_{sp}$. In the adaptive algorithm of the EDF, the inner digital filter coefficients are updated every T_0 samples. Table 1 shows that the number of multiplications of the EDF is larger than that of the LMS-ADF.

Table 2: Specifications of the hardware-based EDF.

EDF	Fixed-point format	Q14
	Bit width of data	16 bits
	Bit width of instructions	16 bits
	Number of individuals	$N_{ap} + N_{sp} \leq 64$
	Order of filters	$N \leq 3, M \leq 3$
SFM	Number of instructions	45
	Program memory size	256×16 bits
	Data memory size	128×16 bits ×2
RS	Individual memory size	1,024×16 bits
-	Common memory size	11,040×16 bits

3. HARDWARE-BASED EVOLUTIONARY DIGITAL FILTERS

3.1. Hardware Structure of Evolutionary Digital Filters

In order to implement the EDF on parallel processors, we design and implement a hardware-based EDF.

Figure 2 shows the block diagram of the hardware-based EDF. The EDF module consists of two submodules, that is, a filtering and fitness calculation (FFC) module and a reproduction and selection (RS) module. This structure can perform parallel processing efficiently, since these modules work in parallel. Moreover, using the proposed structure, it is easy to design these modules and write HDL code for them.

The output of the EDF is the output of an inner filter for which fitness value is maximum. Therefore, the output of the EDF is selected after all fitness values of inner filters is evaluated. Thus, the EDF module has a common memory to keep output signals of all inner digital filters throughout T_0 samples every iteration as shown in Figure 2.

Table 2 shows specifications of the hardware-based EDF. Format of signals and coefficients on the hardware-based EDF is "Q14," that is, 16-bit fixed-point format with an integer part in the high-order 2bits and a fractional part in the low-order 14bits in consideration of the range of the coefficients. The minimum size of the common memory required to keep the output signals is $11,040 \times 16$ bits in the case of $A = 1104$ and $T_0 = 10$.

3.1.1. Filtering and Fitness Calculation Module

The FFC module has high computational requirement, since the FFC module performs filtering and fitness calculation of a large number of individuals. Thus, the FFC module has single filtering modules (SFMs) which are submodules and perform filtering and fitness calculation per individual. Figure 3 shows the block diagram of the FFC module. The FFC module proposed here has the high computational ability to calculate the output and the fitness value since the SFMs run in parallel.

3.1.2. Reproduction and Selection Module

Figure 4 shows the block diagram of the reproduction and selection module. This module consists of the following modules: a single reproduction and selection (SRS) module which perform a reproduction and selection opereation every individual, and an SRS control module.

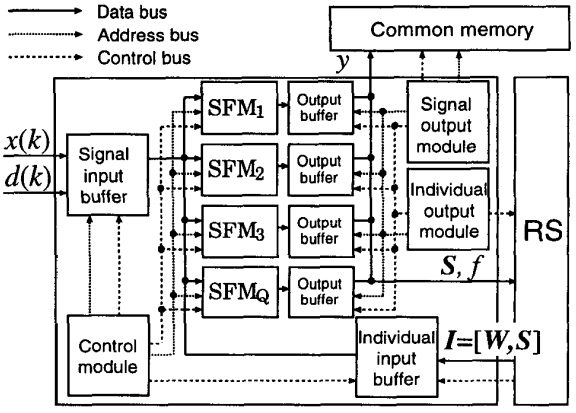

Figure 3: Block diagram of the FFC module.

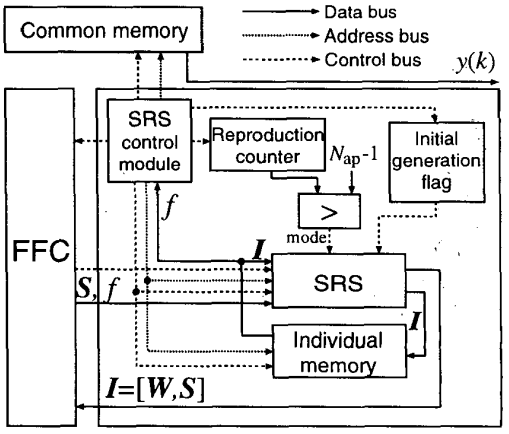

Figure 4: Block diagram of the RS module.

The RS module, first, repeats the following steps in parallel until fitness values of all individuals in population are evaluated.

Step 1. Reproduce an individual according to fitness value in population.

Step 2. Send an information of the individual to the FFC module.

Step 3. Receive a fitness value of the individual from the FFC module.

Second, the individual which has the maximum fitness value in population is selected. Finally, the output for the EDF is selected from the common memory.

3.2. Chip Implementation

A chip of the proposed structure of the EDF is implemented on a silicon area of 4.93×4.93-mm^2 in Rhom 0.35-μm CMOS process. Table 3 shows parameters for the EDF. The implemented chip has only one SFM since the chip is restricted in size. Synplicity and Avanti Apollo are used to synthesize and implement the proposed

Table 3: Parameters for the EDF.

N_{ap}	Number of parents using the cloning method	32
N_{ac}	Number of offsprings which are created by one parent using the cloning method	32
N_{sp}	Number of parents using the mating method	32
N	Order of the regressive average part	3
M	Order of the moving average part	2
T_0	Period of the evaluation	10

Table 4: Synthesis result of the EDF.

Clock frequency	Size
20.0 MHz	63,652 gates

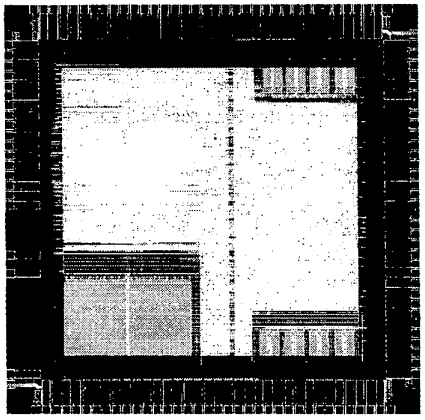

Figure 5: Chip layout of the EDF.

structure of the EDF written in Verilog HDL. The performance of the chip is analyzed by performing a post-layout simulation. Figure 5 shows the layout of the EDF.

Table 4 shows a synthesis result of the designed chip without the memories. Table 5 shows the memory size of the FFC module and the RS module.

Table 6 shows the performances of the FFC module and the RS module. They need 76.8 and 3.6 clocks per individual for processing one sample, respectively. Therefore, the maximum sampling rate of the implemented EDF chip with one SFM is 232.5Hz.

Moreover, Table 6 shows the number of clocks of the FFC module is 21.3 times that of the RS module. Thus, the FFC module needs the 21 SFMs in order that the number of clocks of the FFC module equals that of the RS module when the chip is not restricted in size. In that case, the maximum sampling rate of the EDF is 3.7kHz.

In order to evaluate the performance of the implemented chip, we compare the sampling rate of the hardware-based EDF with that of the software-based EDF. The software-based EDF is written in C and is compiled by gcc on Solaris 8. In that case, the maximum sampling rate of the software-based EDF is 1.7kHz which is executed on the Ultra SPARC III 900MHz. Therefore, the hardware-based EDF with 21 SFMs is 2.2 times faster than the software-based EDF.

Table 5: Memory cell utilization.

Module	Size	Number of cells
FFC	128× 8 bits	8
RS	2,048× 16 bits	1

Table 6: Clocks per individual for processing one sample.

Module	Number of clocks
FFC	76.8
RS	3.6

4. CONCLUDING REMARKS

In this paper, the hardware-based EDF has been designed and implemented. A synthesis result of the designed chip shows the clock frequency is 20.0MHz and the maximum sampling rate of the EDF is 3.7kHz. Moreover, the hardware-based EDF with 21 SFMs of the FFC is 2.2 times faster than the software-based EDF.

ACKNOWLEDGMENT

The authors would like to thank the master student N. Tsushima at Tohoku University for his contributions to the implementation of the proposed hardware-based EDF.

5. REFERENCES

[1] S. D. Stearns, R. A. David, and D. M. Etter, "A survey of IIR adaptive filtering algorithms," Proc. IEEE International Symposium on Circuits & Syst., pp. 709–711, May 1982.

[2] S. J. Flockton and M. S. White, "Pole-zero system identification using genetic algorithms," Proc. the Fifth International Conference on Genetic Algorithm, pp. 531–535, July 1993.

[3] Q. Ma and C. F. N. Cowan, "Genetic algorithms applied to the adaptation of IIR filters," Signal Processing, vol. 48, no. 2, pp. 155–163, Jan. 1996.

[4] S. C. Ng, S. H. Leung, C. Y. Chung, A. Luk, and W. H. Lau, "The genetic search approach — a new learning algorithm for adaptive IIR filtering —," IEEE Signal Processing Mag., vol. 13, no. 6, pp. 38–46, Nov. 1996.

[5] G. D. Cain, A. Yardim, J. Brun, and B. Summers, "Real-time IIR notch filtering using Darwinian adaption," Proc. IEEE International Symposium on Circuits & Syst., pp. 432–435, June 1991.

[6] M. Abe, M. Kawamata, and T. Higuchi, "Convergence behavior of evolutionary digital filters on a multiple-peak surface," Proc. IEEE International Symposium on Circuits & Syst., vol. 2, pp. 185–188, May 1996.

[7] M. Abe and M. Kawamata, "Evolutionary digital filtering for IIR adaptive digital filters based on the cloning and mating reproduction," IEICE Trans. Fundamentals, vol. E81–A, no. 3, pp. 398–406, March 1998.

[8] M. Abe and M. Kawamata, "Comparison of convergence behavior of distributed evolutionary digital filters," Proc. IEEE International Symposium on Circuits & Syst., vol. 2, pp. 729–732, May 2001.