Australian Journal of Basic and Applied Sciences, 4(9): 4184-4192, 2010
ISSN 1991-8178

Genetic Algorithm Optimization for Coefficient of FFT Processor

Pang Jia Hong, Nasri Sulaiman

Department of Electrical and Electronic Engineering Faculty of Engineering
Universiti Putra Malaysia 43400 Serdang, Selangor

Abstract: This paper describes the implementation of Single-objective Genetic Algorithm (SOGA) and Multi-objectives Genetic Algorithm (MOGA) to optimize the pipelined Fast Fourier Transform (FFT) coefficient in order to improve the performance of Signal to Noise Ratio (SNR) and also the Switching Activity (SA). The SA and SNR are optimized separately in a Radix-4 Single Path Delay Feedback (R4SDF) pipelined Fast Fourier Transform (FFT) processor using SOGA. The MOGA optimized both objectives using Weighted-Sum approach.

Key words: FFT processor, Signal to Noise Ratio, Switching Activity

INTRODUCTION

Fast Fourier Transform (FFT) processors are key building blocks in many multi-user mobile systems. The FFT algorithms is used to transform a function from time domain representation to frequency domain representation in various digital signal processing (DSP) applications such as telecommunications, speech processing, image processing, radar systems, medical electronics and seismic processing (Nasri Sulaiman, 2004).

The power consumption has become a very important issue when come to the design of mobile application systems. In multi-carrier code division multiple access (MC-CDMA) or orthogonal frequency division multiplexing (OFDM), the two most power consuming blocks in the receiver are FFT processor and Viterbi decoder (Zhang and Robert, 2000) due to the continuous input and output of data for computation.

When implementing the FFT algorithms in fixed-point number the word length is a very important factor to be considered since it will greatly affect both the performance and the complexity of FFT processor (Johannson *et al.,* 1999). The FFT processor with larger word length will provide better performance in term of accuracy. The accuracy of FFT processor is represent by Signal to Noise ratio (SNR), the processor design with higher SNR value is desirable.

On the other hand, the higher word length in FFT design will contribute to higher switching activities (SA) (Bright and Arslan, 1999). Thus in this paper a Single and Multi objectives genetic algorithm are developed to search for better SNR and SA value of the FFT processor. The Genetic algorithm programming is written in Verilog Hardware Description Language test bench in order to optimize both SNR and SA value of a 16-point R4SDF FFT processor which is designed in Verilog.

R4sdf Fft Processor:

The 16-points R4SDF FFT processor is chosen as it is the minimum size of FFT processor available in multi-carrier code division multiple accesses (MC-CDMA). The Radix 4 architecture is selected due to its less complexity in processing element and control. The operation of R4SDF FFT processor as illustrated in Figure 1. The output of the pipelined R4SDF FFT processor is in bit reversed order.

There are total 16 coefficients or twiddle factors (TF) in 16-point R4SDF processor. The maximum word length of the processor is 32 bits which 16 bits for real data and the other 16 bits for imaginary data. The TF of the FFT processor are stored in read-only memory (ROM) in FFT structure.

Genetic Algorithm Implementation:

Genetic algorithms (GA) is developed since 1960s when there has been increasing interest in imitating living beings for solving difficult optimization problems. GA is successfully applied in some of the Very-Large-Scale Integrated (VLSI) design like channel routing (Buttitta *et al.,* 1991), cell placement (Hedge and Ashmore, 1992) and also capable for optimizing fault model test generation for combinational circuits (Arslan and

Corresponding Author: Pang Jia Hong, Department of Electrical and Electronic Engineering Faculty of Engineering Universiti Putra Malaysia 43400 Serdang, Selangor

O'Dare, 1997). In this paper, the Single objective Genetic Algorithms is used to optimize two objectives SNR and SA separately while the Multi-objectives Genetic Algorithms optimized both objectives at the same time.

Since the GA programming is implemented in Verilog Hardware Description Language, the creation of the population is done using the *$random* function. The population size for implementing the GA optimization is 50. The flow chart in figure 2 shows the overall GA process.

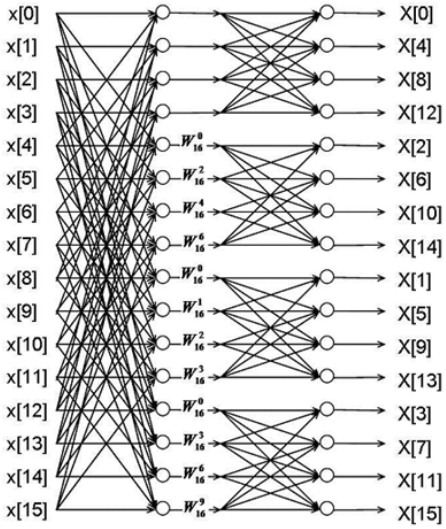

Fig. 1: Signal Flow Graph of 16-points Radix 4 FFT

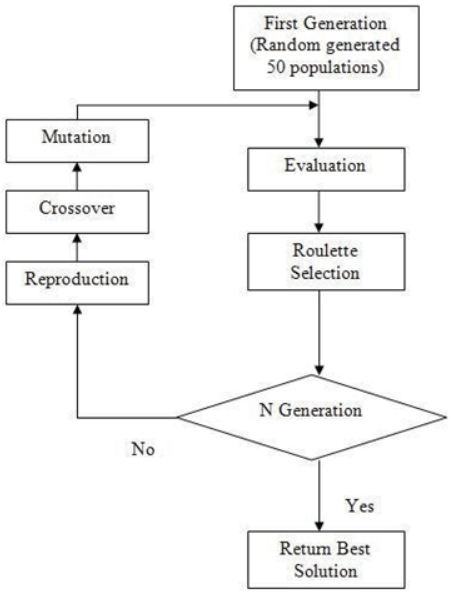

Fig. 2: Genetic Algorithm Flow Chart

Each population consists of 16 randomly generated FFT coefficients. All the randomly generated populations are stored in the ROM of FFT processor for the computation.

Chromosomes Representation:

Since the FFT coefficient is in hexadecimal value in verilog so it can be easily represented in binary strings. The 32 bits word length of FFT coefficients is divided into 16 bits for real data and the other 16 bits for imaginary data as in figure 3.

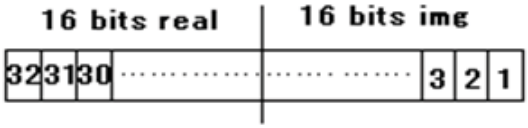

Fig. 3: Chromosome representation in Genetic Algorithm

Fitness Evaluation:

After random set of TF populations are generated, they are stored into ROM of FFT processor. A constant set of input data is used for the FFT computation, which means if there are total 50 randomly generated populations, there are total 50 computations needed with the same input data. The SNR fitness evaluation is done according (Yang-han *et al.,* 2006). Initially, with a sequence of input data, x(n) and 16 original FFT coefficients, the FFT processor calculates the outputs, $X_1(k)$. Next, with the same x(n) and with the randomly generated FFT coefficients, the FFT processor again calculates the outputs, $X_2(k)$. Both outputs are then compared for error calculations, e(k) as in figure 4.

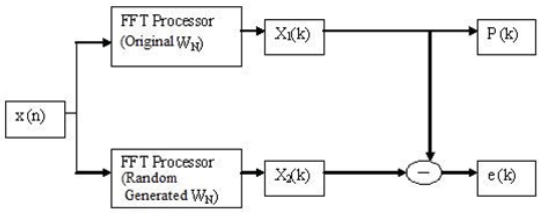

Fig. 4: Methodology to evaluate error fitness

The SNR value is calculated using equation 1 (Nasri Sulaiman, 1996). R_{16}, I_{16}, R_{wl}, and R_{wl} are the real and imaginary parts of the FFT output before and after computation respectively.

$$SNR(dB) = 10Log_{10}\left[\frac{\sum(R_{16})^2 + \sum(I_{16})^2}{\sum(R_{16} - R_{wl})^2 + \sum(I_{16} - I_{wl})^2}\right] \quad (1)$$

Besides the optimization of SNR, the single objective GA is also used to optimize the SA value. According to (Abu-Khater *et al.,* 1996), SA is the main source of power consumption in a typical CMOS logic gate.

$$P_{sw} = \frac{1}{2}kC_{load}V_{dd}^2 f \quad (2)$$

In equation (2), V_{dd} is the supply voltage, f is the clock frequency, C_{load} is the load capacitance of the gate, k is the switching activity factor which is defined as the average number of times the gate makes an active transition in a single clock cycle.

For the fitness evaluation SA, the method of Hamming Distance is used. The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. It is used to measure the minimum number of substitutions required to change one string into the other as in figure 5.

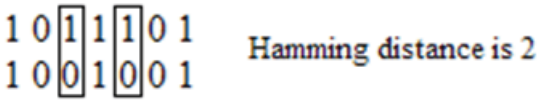

Fig. 5: Method of Hamming Distance Calculation

There are total 16 coefficients in R4SDF FFT processor, thus the SA is the sum of Hamming distances for the all 16 coefficient in sequence. The normal switching activities of original coefficient of 16-point FFT processor is 192 (Hasan *et al.,* 2003). During the optimization, the lower SA is desirable as lower power consumption is needed.

Selection:

After the fitness is assigned to all population, there is a need to select the population with better fitness value for the reproduction of next generation. The method used for the selection is called Roulette Selection or Fitness Proportionate selection. The roulette selection is done by applying the cumulative fitness of the population. The probability of being selected P_i as in equation (3)

$$P_i = \frac{f_i}{\sum_{j=1}^{N} f_i} \tag{3}$$

Where *fi* is the fitness of individual *i* in the population and *N* is the number of individuals in the population.

The idea of proportionate selection is shown in figure 6.

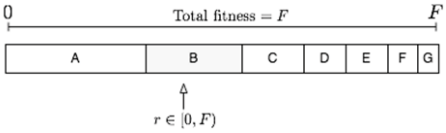

Fig. 6: Example of Fitness Proportionate Selection

In figure 6, r is a random number selected from fitness value 0 to F: in this example the individual B is selected. Since the GA programming is in verilog, the function *$random%[total cumulative fitness]* is used, which means a random fitness value will be generated and the value is equal or less than the total cumulative fitness of the whole population.

During the proportionate selection, to make sure the individual with better fitness have the higher chance to be selected for reproduction, some modifications are applied to the fitness evaluation for both SNR and SA optimization using the equation (4) and (5).

$$F_{SNR} = (F_i - F_{min})^2 \tag{4}$$

$$F_{SA} = (F_{max} - F_i)^2 \tag{5}$$

Which F_i is the original fitness calculated using equation (1) for SNR and value of 192 for SA. Since the higher SNR is desirable, thus a minimum of SNR value is assigned according to the parameter of GA optimization. For SA, the lower value is desirable and is assigned according to the parameter of GA optimization.

Crossover:

The process of crossover is carried out after the proportionate selection for the new generation. It is the most important operator in GA. In this process, two chromosomes called parents are combined to form two new chromosomes called offsprings. The information of two parents is exchanged in order to create offsprings with better fitness.

The crossover rate P_c is normally 0.6 to 1.0 used in many GA searches. The crossover rate of 0.9 is chosen in the crossover process. The single point crossover is applied; however the crossover point is randomly selected in the process.

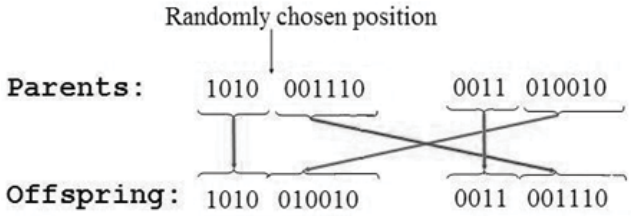

Fig. 7: Randomly Chosen Single Point Crossover

In a population, there are N=16 coefficients, beside the need to assign the crossover point of the bits string of coefficient, an effort has been done by select the crossover point of the set of coefficient as shown in figure 8.

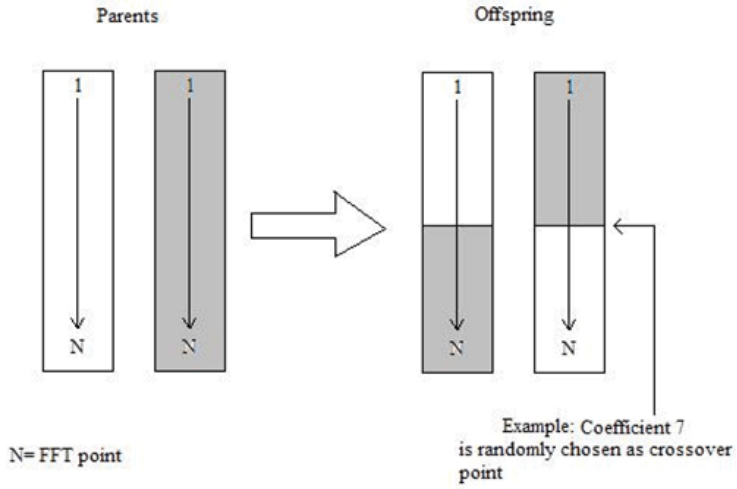

Fig. 8: Randomly Chosen Crossover point for set of Coefficient

Mutation:

Mutation is a process which is used to randomly alter the value in a string, the process is drastic. The mutation rate P_m of 0.01 is chosen. The high mutation rate is not desirable since it will greatly degrade quality of chromosome. The mutation process is illustrated in figure 9.Figure 9. Randomly Selected Point for MutationFor the mutation process, the limitation which has to be considered is that the mutation bits for the coefficient should always desirable near Less Significant Bit (LSB). This is because if the mutation happens in Most Significant Bit (MSB) it may alter the sign value of the FFT coefficient. If the mutation bit happens near MSB, the value of the coefficient will be greatly changed.

RESULTS AND DISCUSSION

Single Objectives GA:

The Single objective GA is used to optimize both SA and SNR separately. The parameter of the GA is shown in table below which is a common used GA parameter can be found in any GA reference books.

Table 1: Parameter of Single Objective GA for SNR and SA Optimization

Population Size	50
GA Generation	50
Crossover Rate Pc	0.9
Mutation Rate Pm	0.01

In this work, the Single Objective GA is used to search for better performance of SNR and SA. The results will be discussed are shown until 14 bits. The result is compared to the performance of identical bits of coefficient without GA optimization using truncation method in Table II.

SNR Optimization:

Since the GA is implemented in verilog programming, the SNR value in decibel (dB) is calculated using equation (6).

$$Snr(dB)=10\log_{10} SNR \quad (6)$$

The constant set of input data is used throughout the optimization. The figure 10 shows the average SNR value for 50 populations for 16 bits coefficient with GA optimization in an example search.

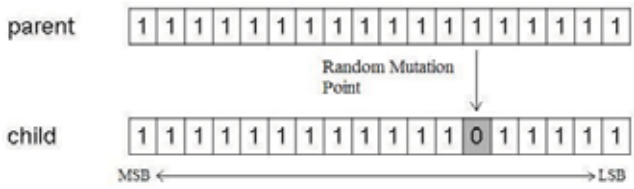

Fig. 9: Randomly Selected Point for Mutation

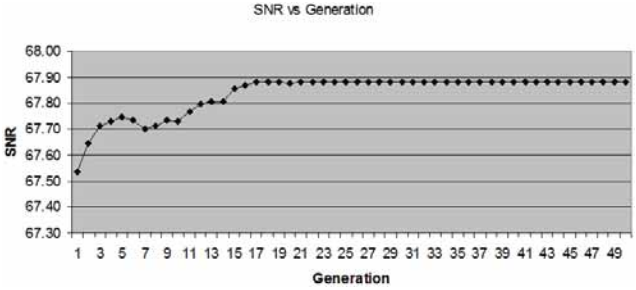

Fig. 10: Average SNR after GA Optimization for 16 bits word length

From figure 10, the maximum average SNR value obtained is 67.88 dB compare to SNR value without GA optimization which is 67.71 dB. Throughout the 50 generation the best chromosome obtained has the SNR value of 67.93 dB; however the SA value increases to 198 as a violation for better SNR value.

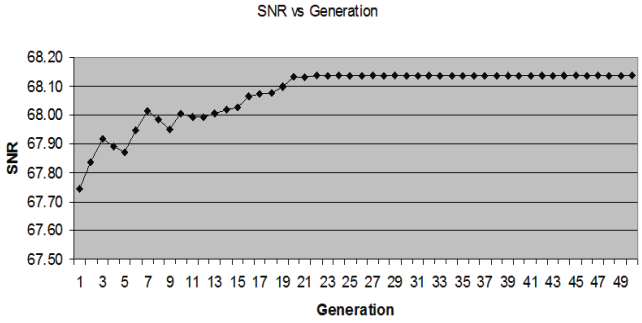

Fig. 11: Average SNR after GA Optimization for 15 bits word length

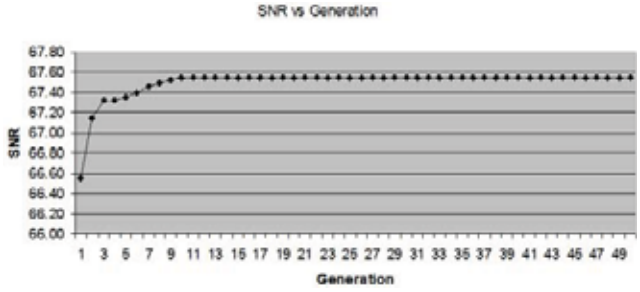

Fig. 12: Average SNR after GA Optimization for 14 bits word length

Figure 11 and 12 also show that the GA is capable to search for better SNR value although there is the reduction in bits. The maximum average SNR value obtained in these two searches are 68.14 dB for 15 bits and 67.55 dB for 14 bits SNR optimizations. The SNR value without GA optimization is 67.70 for 15 bits and 67.66 for 14 bits.

SA Optimization:

For the Switching activity optimization, the parameter of the GA used is the same as in Table 1. The reduction in FFT word length will greatly decrease the value of SA and contribute to lower power consumption as explained by equation (2). The graph in figure 13 shows the GA search for the chromosomes with lower SA value for 16 bits word length.

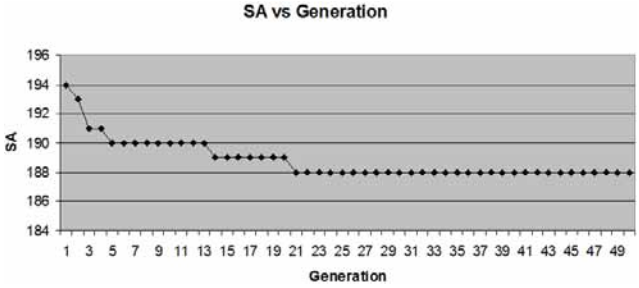

Fig. 13: Average SA after GA Optimization for 16 bits word length

The maximum SA for 16 bits word length is 192, the GA is able to reduce the SA value to 188. The reduction in SA however will sacrifice the SNR value. The SNR by default is 67.71 dB for 16 bit word length. After the GA optimization in SA, the SNR value is reduced to 67.64 dB.

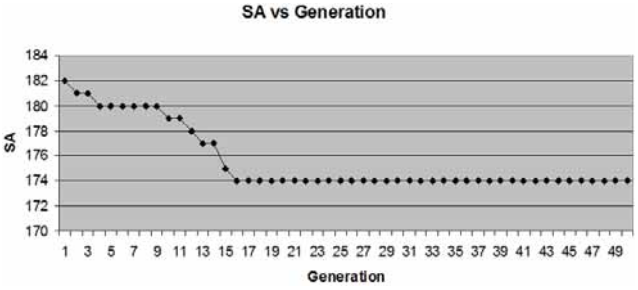

Fig. 14: Average SA after GA Optimization for 15 bits word length

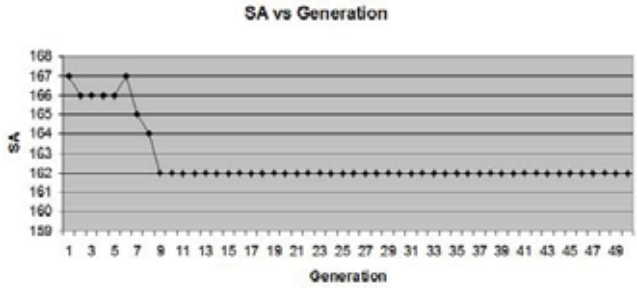

Fig. 15: Average SA after GA Optimization for 14 bits word length

The original SA value for 15 and 14 bits are 178 and 166 by default. From figure 13 and 14, the GA is able to optimize the SA value to 174 for 15 bits and 162 for 14 bits. However after the optimization, there are reductions in SNR value, the SNR reduced from 67.70 dB to 67.50 dB for 15 bits and 67.66 dB to 56.87 dB for 14 bits word length. Thus, there is a need to implement MOGA to optimize both SA and SNR value at the same time.

Multi-objectives GA:

In multi-objectives optimization, both objectives SNR and SA will conflict with each other. Larger word length will contribute to better SNR value but also increases the SA value. For low power and high accuracy design, the SNR value need to be large while the SA value need to be kept as small as possible.

In order to optimize both objectives, we implement the multi-objectives GA using Weighted-Sum approach. The weights are assigned to each objective function and both objectives are combined as a Single objective function. The power consumption factor SA and the Signal to Noise ratio SNR factor are defined to be equally important as in equation 7.

$$fitness = 0.5 \times \frac{SNR}{total\_SNR} + 0.5 \times \frac{invert\_switching\_activity}{total\_switching\_activity} \quad (7)$$

The fitness function in equation 7 is used to combine both objectives. The total SNR is 67.71 dB in the FFT design and the total SA is 512 which is 32 bits multiply by 16 coefficients. The inverted SA is calculated by subtract the SA with maximum SA which is 512. If the solution has lower SA value, then inverted SA value will be higher and provide higher fitness value.

Figure 15 and 16 show average value changes of SNR and SA for an example search of 16 bits word length optimization in 50 generations.

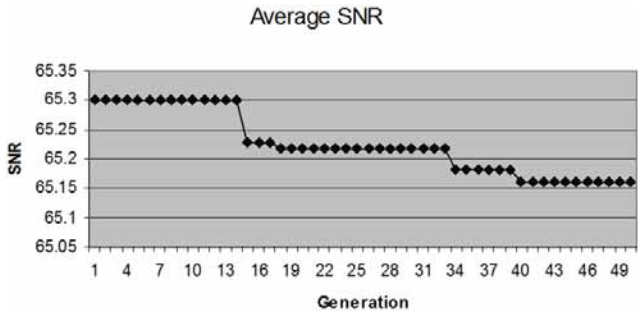

Fig. 16: Average SNR value for 16 bits in 50 generations

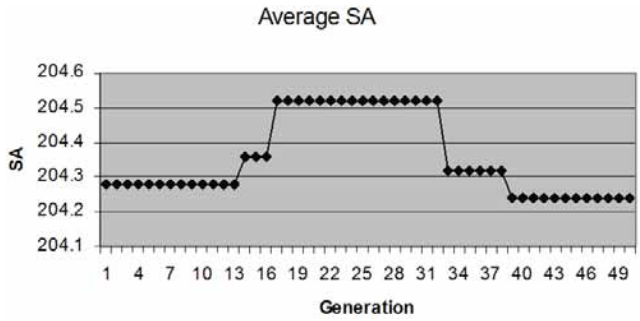

Fig. 17: Average SA value for 16 bits in 50 generations

In order to obtain lower SA value the average SNR is decreased as shown in figure 16. In the 16 bits word length optimization the best solution found has 68.38 dB for SNR and 174 for SA. The results in Table II and III show the comparison of the SNR and SA value using bit truncation method and GA optimization.

Table 2: Snr and Sa Values Using Truncation Method

	Bit Truncation Method	
Bits	SNR	SA
16	67.71	192
15	67.70	178
14	67.66	166
13	66.21	152
12	64.49	140
11	58.52	128
10	53.18	116

Table 3: Iiisnr and Sa Values Using Moga Optimization

	GA optimization	
Bits	SNR	SA
16	68.38	174
15	68.08	160
14	67.55	152
13	65.95	144
12	62.00	128
11	57.41	114
10	50.58	98

In MOGA optimization using Weighted Sum method, it is observed that the GA search is able to optimized both SNR and SA value until word length of 13 bit as in table II and III. After the bits reduction below 13 bits, although there is optimization in SA, however the SNR value is violated, which means the SNR value is degraded in order to obtain better SA value. The results show that the lower SA values are obtained with lower SNR values. The SNR values have larger drop in optimization of 12, 11 and 10 bits. The performance of MOGA using weighted sum method is very dependence on the fitness equation. Further improvement of fitness equation may provide better solutions.

Conclusion:

In this paper, the Single and Multi objectives GA are implemented to optimize both SA and SNR separately for the 16-point FFT processor. The results show that the GA is able to search for better chromosomes for both SA and SNR value. The MOGA using Weighted Sum method can search for solutions which meet both objectives until 13 bits. GA is useful in the FFT design when come to the preference to have a better accuracy processor or a better low power performance FFT processor. This will give a good contribution to many fields which required FFT processor in applications.

REFERENCES

Abu-Khater, I.S. , A. Bellaouar and M.I. Elmasry, 1996. "Circuit Techniques for CMOS Low-power High-performance Multipliers", IEEE Joumal of Solid-state Circuits, 31(10): 1535-1546.

Arslan, T. and M..J O'Dare, 1997. "A Genetic Algorithm for Multiple Fault Model Test Generation for Combinational VLSI Circuits". Genetic Algorithms in Engineering Systems: Inovation anf Applications, 2-4.

Bright, M.S., T. Arslan, 1999. "Multi-Objective Design Strategies for High-Level Low-Power Design of DSP System", in IEEE Int. Symposium on Circuits and Systems (ISCAC-99), Orlando, Florida, USA, 1: 80-83.

Buttitta, B., P. Orlando, F. Sorbello and G. Vassallo, Monreale 1991. "A New Genetic Algorithm for the Solution of The Channel Routing Problem" IEEE proceeding, CH3001-5, pp: 462-466.

Hasan, M., T. Arslan and J.S. Thompson, 2003. " A Novel Coefficient Ordering based Low Power Pipelined Radix-4 FFT Processor for Wireless LAN Applications", in IEEE Transactions on Consumer Electronics, 49(1): 128-134.

Hedge, U. and B. Ashmore, 1992. "A Feasibility Study of Genetic Placement", Texas Instruments Technical Journal, 9(6): 72-78.

Johannson, S., S. He and P. Nilsson, 1999. "Wordlength Optimization of a Pipelined FFT Processor", in 42nd Midwest Symposium on Circuits and Systems, pp: 501–503.

Nasri Sulaiman, 2004. "A Genetic Algorithm for the Optimisation of a reconfigurable Pipelined FFT processor", Proceedings of the NASA/DoD Conference on Evolution Hardware.

Nasri Sulaiman, 2006. "A Multi-objective Genetic Algorithm for On-chip Real-time Adaption of a Multi-Carrier based Telecommunications Receiver", Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems.

Yang-Han Lee, 2006. Yih-Guang Jan, Ming-Hsueh Chuang, Hsien-Wei Tseng, Liang-Lin Jau, Min-Ru Wen, Wei-Chen Li and Sheng-Kai Yu, "Co-Emulation Design for OFDM Baseband Transceiver", Department of Electrical Engineering, Tamkang University Tamsui, Taiwan 251, R.O.C, Department of Computer & Communication Engineering, St. John's University Tamsui, Taiwan 251, R.O.C. 9(1): 71-79.

Zhang, N. and W. Robert Broderson, 2000. "Architectural evaluation of flexible digital signal processing for wireless receivers" 34th Asilomar conference on signals, systems and computer, 1: 78-73.