GENETIC ALGORITHM-AIDED FIXED-POINT DESIGN OF E-UTRA PRACH DETECTOR ON MULTI-CORE DSP

Rongrong Qian, Tao Peng, Yuan Qi, and Wenbo Wang

Key Lab. of Universal Wireless Communication, Ministry of Education,
Beijing Univ. of Posts and Telecommunications (BUPT)
10 XiTuCheng Road, Haidian District, Beijing, China, 100876
phone: +86 010 62285158 , fax: +86 010 62282921,
email: rongrongqian@bupt.edu.cn

ABSTRACT

This paper presents a new genetic algorithm (GA)-aided methodology of software implementation in Digital Signal Processors (DSPs) under both the computational accuracy and bus bandwidth constraints. The design issue is firstly stated as two classes of fixed-point design problems, one of which is then formulated to a constrained integer programming (CIP) problem. And the genetic algorithm is proposed to treat with such CIP problem for the sake of efficiency. Then the fixed-point evolved (E)-UTRA PRACH detector is presented, which further underlines the feasibility and convenience of applying this methodology to practice. Finally, the numeric results justify the proposed GA-aided approach and demonstrate that a speedup by a factor of 33 can be achieved compared to the exhaustive search for the solution of E-UTRA PRACH detector design problem.

1. INTRODUCTION

Realization of digital signal processing algorithms in the real-time applications requires the fixed-point arithmetic for the sake of cost and speed. Thanks to the former fruitful research works, many methodologies have been developed for settling the design issues of implementing fixed-point algorithms meeting requirement for the minimization of cost, power consumption and time to market of digital signal processing applications, e.g., the automatic floating-to-fixed point conversion has been systematically studied in [1] for software implementations, and different methods with different properties are presented to obtain the optimal wordlength design [2-5].

In the modern digital signal processing area, the computational pressure incurred by high data rates and complicated algorithms upon on the hardware is going to be extremely heavy. Hence the promising digital signal processor (DSP) architecture solutions, e.g., multi-core systems [12], emerge and help to mitigate such pressure, however, new challenges might come or existing issues still exist. One issue we care about is the conflict between the limited bus bandwidth of current hardware and the dramatically growing data throughput pressure. From the hardware perspective, the important way to handle such issue is using advanced interconnect topologies which include cross bar [6], 1-D linear array [7] and 2-D *mesh* [8], etc. And from the view of software implementation, one feasible method is making use of the efficient management and scheduling algorithms to treat with the bottleneck of bus bandwidth [9] [10]; and another proper method is to maximize the utilization of the bus resource, which is studied in this paper since there is few of literature addressing such utilization maximization problem.

In wireless communication community, evolved (E)-UTRA is currently being standardized as a long term evolution (LTE) of the 3GPP radio access technology [14]. The physical random access channel (PRACH) detector is a fundamental baseband unit of the E-UTRA base-station systems. Although many technical reports of 3GPP have presented the mathematic idea of implementing the PRACH detector [14], there has been rare of practical design scheme.

In this paper, we focus on the fixed-point bus-usage design issue under both the computational accuracy and bus bandwidth constraints which are common in modern DSP environment with real-time requirement. The design issue is stated as two classes of fixed-point design problems within which the 1st class problem can easily utilize the technology of optimal wordlength design hence is not extended in this study while the 2nd class problem is formulated to a constrained integer programming (CIP) problem. Then the genetic algorithm is applied to solve such CIP problem to avoid the traditionally exhaustive search for the solution, therefore, the whole design procedure can be accelerated. The fixed-point bus-usage design of E-UTRA PRACH detector which is based on the GA-aided methodology is provided finally.

2. FIXED-POINT BUS-USAGE DESIGN ON DSP

2.1 Design Problem Statement

In modern DSP architectures, bus is the basic component of the chipsets, which always has the fixed bandwidth. Consider the algorithm realizations on DSPs, the bus might become a bottleneck of the whole design due to the pressure from the high throughput requirements of applications [10], especially as in the advanced wireless communication systems. To authors knowledge, the literature which presents the systematic methodology of maximizing the utilization of the bus resource on the hardware with limited bus bandwidth is scarce. Therefore in this study, we address the issue of bus-usage design for the algorithm realizations on DSPs. The bus-usage design, exactly speaking, is determining the promising scheme of utilizing the bus for the given algorithm and DSP hardware, which should be able to greatly exploit the communication capability of the bus.

Figure 1 demonstrates the bus model we consider, a DSP processing core (unit) or a hardware accelerator always communicates other entities (e.g., other processing core, memory unit) via input and output bus. Assume that $\overline{L}$ bits are packed into a single bus transfer and the clock rate of the

bus is f (HZ). Hence, the bus bandwidth can be defined as $\overline{L} \cdot f$ (bits/s). Since the communication capability of bus is limited, the data transferred via bus can only have the finite wordlength which leads to the finite-precision numerical effect [11] and finally causes the performance degradation to the algorithm. So far based on former discussion, the bus-usage design we aim to manipulate can further be divided into two classes which are described as:

1. To minimize the used bus bandwidth by the algorithm realization while still fulfilling a given performance degradation constraint (in other words, the performance degradation must be considerable small). In this case, the unused bus bandwidth can be allocated to other algorithms;
2. To minimize the performance degradation when given by the maximum allowed bus bandwidth.

Furthermore, under the bus model of Figure 1, the data passing through the bus generally can be formatted into fixed-point or floating-point data type. Since in fixed-point arithmetic, there is no need to process the exponent and the mantissa [1], that leads to a definitively lower operation cost comparing with floating-point arithmetic, thus the bus-usage design discussed in this paper adopts the fixed-point data type.

2.2 Fixed-Point Bus-Usage Design

Between the above two classes of the bus-usage design problem with fixed-point data type, we announce that, the problem of 1st class can be settled similarly as the problem of wordlength optimization for fixed-point digital signal processing systems. Indeed, the optimal wordlength problem has already attracted many research interests, e.g., the simulation-based methods ([3] [4]) as well as the analytical approaches ([1] [2]) are proposed to evaluate the fixed-point accuracy. And for simulation-based methods, several search strategies (e.g., complete search, exhaustive search, sequential search and preplanned search) can be utilized to search for the optimum wordlengths [3] [5]. But for the 2nd class problem, there is no existing solution that can be directly utilized. Therefore, in this study, a novel simulation-based method will be proposed to perform the 2nd class fixed-point bus-usage design, which adopts the genetic algorithm (GA) based optimization strategy. Moreover, the signal-to-quantization-noise ratio (SQNR) which is the most commonly used metric to evaluate the fixed-point accuracy and equals to the ratio between the desired signal power and the quantization noise power, is chosen to assist on our fixed-point bus-usage design. The SQNR can be obtained from the simulation in which both the floating-point (wordlength is regarded to be sufficiently large) and fixed-point (wordlength is limited) algorithms are implemented to process the same input data and compute the difference between their output, which is regarded as the quantization noise [1]. Besides the fixed-point accuracy, the dynamic range that is the power level range in which the data passing through bus can be varied without causing too much performance degradation, is another design issue we should care about. In our study, the bus-usage design is obtained only according to the fixed-point accuracy. Then based on such design, we measure the dynamic range through inputting different power level of data. In future works, we aim to include the dynamic range factor into the design methodology, therefore make the methodology more systemic.

Before introducing the mathematic problem formulation,

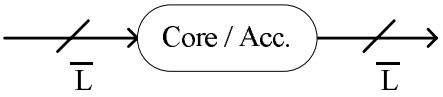

Figure 1: The bus model of DSP. A DSP core or a hardware accelerator (Acc. for short) always communicates other entities via input and output buses. $\overline{L}$ is the bus width per transfer.

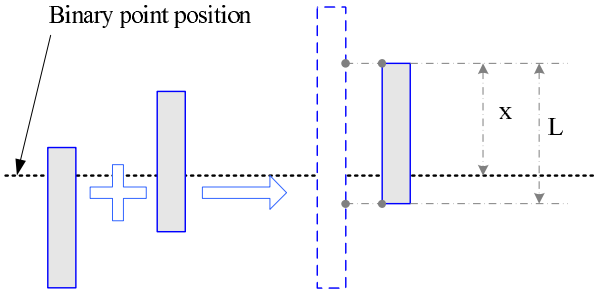

Figure 2: Data should be formatted before passing through the bus. Moreover, the data inner the processing core or accelerator can have different binary point position.

some definition and assumption need to be clarified. Let $< L,x >$ denote the fixed-point format of data passing through the given bus as explained in Figure 2, where L is fixed-point wordlength (WL), and x is the integer wordlength (IWL) that equals to the number of bits assigned to the integer representation [3]. The computation within the DSP processing core or accelerator is regarded to have the sufficiently long wordlength. Therefore, the bus-usage design can be seemed as a problem of selecting the proper integer pair $< L,x >$ for each bus use. Note that we name $< L,x >$ as the *bus-usage parameter pair*.

2.3 Mathematic Problem Formulation

The 2nd class fixed-point bus-usage design, exactly speaking, selecting the proper *bus-usage parameter pairs*, can be formulated as a constrained integer programming (CIP) problem which is given by

$$SQNR_{min} = \min_{<\mathbf{L},\,\mathbf{x}>} \{\max_{s} SQNR(\mathbf{L};\mathbf{x};s)\}, \quad (1)$$
$$\text{subject to: } L_n \le \overline{L} \cdot C_n, n = 1,\ldots,N,$$
$$s \in \mathscr{S}$$

where both $\mathbf{L} = [L_1, L_2, \ldots, L_N]$ and $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ are the integer vectors and each integer pair $< L_n, x_n >$ with $L_n \in \mathbb{N}$ and $x_n \in \mathbb{N}$ denoting the *bus-usage parameter pair* of the corresponding bus, and N denotes the number of above pairs to be determined within the whole design. And $\mathscr{S}$ is the set of all possible input signals to the fixed-point algorithm. Once it fulfils that $SQNR_{min} \le SQNR_0$, it means that, there exists a feasible solution of $\mathbf{L}$ and $\mathbf{x}$ which enables the fixed-point algorithm to reach the acceptable performance, where $SQNR_0$ is the maximum allowed performance loss caused by finite-precision numerical effect. Then we can obtain the *bus-usage parameter pairs* for the fixed-point design as

$$< \overline{\mathbf{L}}_{opt}, \overline{\mathbf{x}}_{opt} > = \arg \min_{<\mathbf{L},\,\mathbf{x}>} \{\max_{s} SQNR(\mathbf{L};\mathbf{x};s)\}. \quad (2)$$

$$\text{subject to: } L_n \le \overline{L} \cdot C_n, n = 1, \ldots, N, \quad s \in \mathscr{S}$$

Let us consider about the solution of (2), the searching for all the possible $\mathbf{L}$ and $\mathbf{x}$ is an oppressive work because the size of set $\mathscr{S}$ is always extremely large and even unpredictable. Since it is not necessary to find out $\overline{\mathbf{L}}_{opt}$ and $\overline{\mathbf{x}}_{opt}$ for the fixed-point design, the relaxation problem of (2) is feasible for us to achieve the final design as well, which is to find $\overline{\mathbf{L}}$ and $\overline{\mathbf{x}}$ that fulfils

$$< \overline{\mathbf{L}}, \overline{\mathbf{x}} > \in \{ < \mathbf{L}, \mathbf{x} > | \max_s SQNR(\mathbf{L}; \mathbf{x}; s) \le SQNR_0 \}. \quad (3)$$
$$\text{subject to: } L_n \le \overline{L} \cdot C_n, n = 1, \ldots, N, \quad s \in \mathscr{S}$$

Any $< \overline{\mathbf{L}}, \overline{\mathbf{x}} >$ is found, the solution is finished. Then it is no need to exhaustive search for all the possible $< \mathbf{L}, \mathbf{x} >$, and the execute time will be reduced. We will show in next section that the genetic algorithm can be applied to solve (3).

3. GENETIC ALGORITHM-AIDED FIXED-POINT DESIGN

3.1 Genetic Algorithm (GA)

Many optimization problems from the industrial engineering world are very complex in nature and quite hard to solve by conventional optimization techniques but not by genetic algorithm which was first described by Goldberg [13]. Genetic Algorithms are stochastic search techniques based on mechanism of natural selection and natural genetics.

In the procedure of solving optimization problems, the genetic algorithms start from an initial set of random solutions called *population*. Each individual in the population is called a *chromosome*, representing a solution to the problem at hand. A chromosome is a string of symbols; it is usually, but not necessarily, a binary bit string. The chromosome *evolve* through successive iterations, called *generations*. During each generation, the chromosome are evaluated, using some measures of fitness. To create the next generation, new chromosomes, called *offspring*, are formed by either (a) merging two chromosomes from current generation using a *crossover* operator or (b) modifying a chromosome using a *mutation* operator. A new generation is formed by (a) selecting, according to the *fitness* values, some of the parents and offsprings and (b) rejecting others so as to keep the population size constant. Fitter chromosomes have higher probability of being selected. After several generations, the algorithms converge to the best chromosome, with hopefully represents the optimum or suboptimal solution to the problem.

3.2 GA-Aided Parameter Pairs Selection

It is feasible to apply the GA to solve the problem defined by (3), i.e., let the integer vector $\mathbf{z} = [\mathbf{L} \quad \mathbf{x}]$ be the chromosome, and $f(\mathbf{z}) = -SQNR(\mathbf{L}; \mathbf{x}; s)$ is chosen as the *fitness* function. Recall the problem formulated as (3), let $f(\mathbf{z}) = +\infty$ for the chromosome $\mathbf{z}$ not fulfils the constraints that $L_n \le \overline{L} \cdot C_n, n = 1, \ldots, N$, therefore, these constraints have the effect on the GA-aided problem solution. Through the evolution generation by generation, once the *fitness* function reaches $-SQNR_0$, the solution $< \overline{\mathbf{L}}, \overline{\mathbf{x}} >$ is obtained. The GA-aided problem solution has the same time complexity as exhaustive search in worst case, nevertheless, GA can always converge quite fast in practice [13].

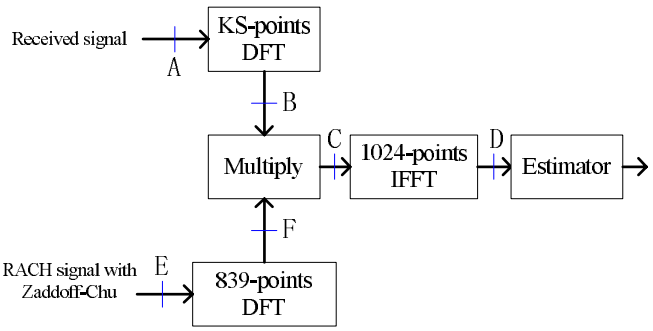

Figure 3: The structure of PRACH detector.

The GA-aided parameter pairs selection does not require too much efforts of designer, because Matlab already has provided us two useful tools. The first one is the quantization function *quantize*, which can quantize floating-point signals to the format $[L, I]$, see the help document of Matlab. The second tool is *gatool* that directly performs the GA to solve the optimization problem. Therefore, we argue that the proposed fixed-point design method is definitely convenient for practical applications.

4. FIXED-POINT DESIGN OF PRACH DETECTOR OF E-UTRA

In this section, the GA based fixed-point bus-usage design is presented. Through this study, we aim to provide a feasible software implementation scheme of the E-UTRA PRACH detector on picoArray which is a multi-core fixed-point DSP [16]. To realize the E-UTRA systems, one of the challenges is the extreme high data throughput that is required upon the hardware. Assume $L = 16$ is predetermined for all the bus uses to simplify the computation within process core, since the processing core of picoArray has 16bits processing arithmetic. Hence in this design case, $\mathbf{z} = [\mathbf{x}]$ recall from Section 3.2.

4.1 PRACH Detector of E-UTRA

Based on the definition of baseband signal generation in [14], the transmitted signal model of physical random access channel (PRACH) in E-UTRA system can be expressed as

$$s(p) = \sum_{m=0}^{KS-1} y(m) e^{j2\pi mp/KS} = KS \cdot \text{idft}(y(m), KS), \quad (4)$$

where

$$y(m) = \begin{cases} \bar{y}(m - m_0), & m_0 \le m < m_0 + N_{ZC} \\ 0, & \text{others} \end{cases} \quad (5)$$

where $m_0 = \varphi + K(k_0 + 1/2)$ and k_0 as well as the parameters N_{ZC}, φ, K and S are determined by the system configuration and their valid values are specified in [14], moreover,

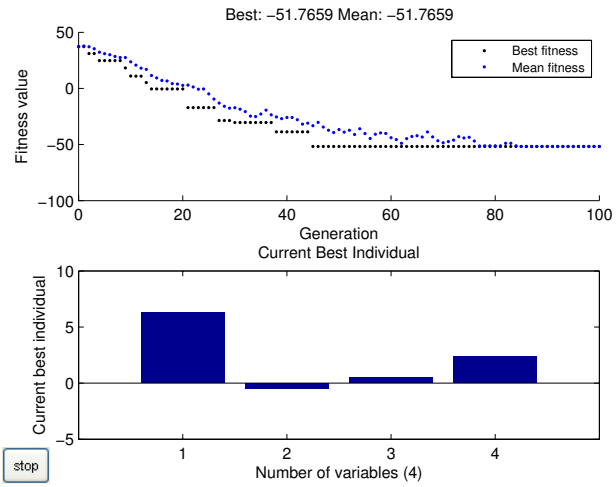

Figure 4: GA procedure of selecting the *bus-usage parameter* vector $\mathbf{z}$.

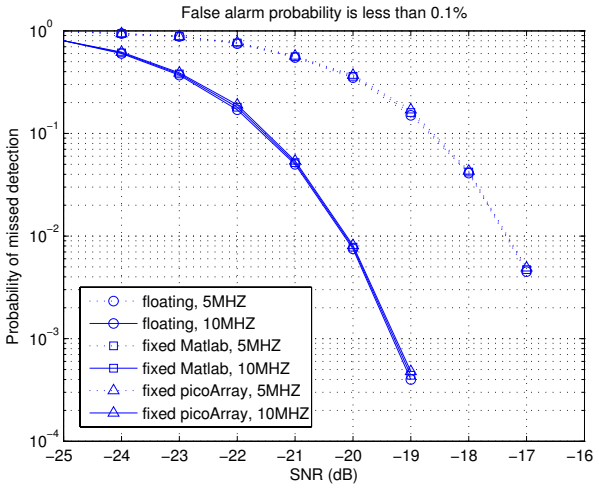

Figure 5: Performance comparisons in AWGN channel. 5/10MHZ denotes the system bandwidth.

$$\bar{y}(k) = \sum_{n=0}^{N_{ZC}-1} x_{u,v}(n) e^{-j2\pi nk/N_{ZC}} = \mathrm{dft}(x_{u,v}(n), N_{ZC}), \qquad (6)$$

where $x_{u,v}(n)$ is a cyclic shifted Zadoff-Chu sequence and drived from the uth root Zadoff-Chu sequence which is defined as $x_u(n) = exp(-j\pi un(n+1)/N_{ZC}), 1 \le n \le N_{ZC}$. And the details of PRACH signal generation can be found in [14]. Note that $\mathrm{dft}(x,N)$ and $\mathrm{idft}(x,N)$ denote the N-points discrete fourier transform (DFT) and inverse discrete fourier transform (IDFT) computations of x. In sequel, we focus on the PRACH detector of FDD system where $N_{ZC} = 839$.

Given the system model of PRACH, we design the detector as shown in Figure 3, in which each rectangle block denotes the computation realized in one DSP processor or hardware accelerator without using the bus and the data passing through the bus 'A', 'B',...,'F' are expressed as S_A, S_B,...S_F. S_A are the received data and S_E are the local PRACH data with Zadoff-Chu. Since the Zadoff-Chu sequence and its DFT output are constant amplitude zero autocorrelation (CAZAC) [15], the data S_E and S_F have the constant amplitude, i.e., 1 of S_E and $\sqrt{N_{ZC}}$ of S_F, this feature will in favor of the fixed-point design. In this design, KS-points DFT,1024 IFFT and 839-points DFT are assumed to be implemented in the DSP processing core or hardware accelerator without using the bus, a lot of literatures have presented the methods to implement them [11], and also the hardware accelerators of them have been available in many DSP platforms, e.g., picoArray [16].

4.2 Fixed-Point Bus-Usage Design of PRACH Detector

Consequently, we need to select all the proper *bus-usage parameter pairs* of the data from S_A to S_F for the PRACH detector. Given the assumption that $L = 16$ for all the bus uses in this case, we only need to choose the parameter vector $\mathbf{x} = [x_A, x_B, \ldots, x_F]$ in which every $x_A \in \mathbb{N}, x_B \in \mathbb{N}, \ldots$ is the *bus-usage parameter* x corresponding to the data $S_A, S_B, \ldots$ as presented in Figure 3. Note that, the data S_E and S_F are constant amplitude, therefore, their parameter pairs can be determined without any further analysis, as $x_E = 1$ and $x_F = 5$. In the following, the vector $\mathbf{z} = [z_1, z_2, z_3, z_4] = [x_A, x_B, x_C, x_D]$ represents the *bus-usage parameter* vector.

In the case of PRACH detector, the transmitted signals are generated from a set of basic sequences [14], which has not so larger size, we can enumerate all the possible transmitted signals in the simulation. Consider the above GA procedure, *SQNR* is measured in average manner that means the average signal power and the average quantization noise power of one detection are used.

5. NUMERICAL RESULTS

In this section, numerical results of fixed-point E-UTRA PRACH detector design are presented. The fixed-point bus-usage design is performed with the help of Matlab which provides the GUI version of *gatool*, therefore allows us to utilize the *gatool* in a visible way, which eventually reduces the design efforts and time. When using the *gatool*, we set the population size to 20 and stop the GA after run for $N_g = 100$ generations. The population type is set to double vector, therefore, we use the *fitness* function in the form of $f(\mathbf{z})$, where $\mathbf{z} = floor(\mathbf{y})$ is the function that rounds each element of the double vector $\mathbf{y}$ towards minus infinity to form an integer vector $\mathbf{z}$ that denotes the input variables vector of the GA.

During the GA simulation to select the proper *bus-usage parameter* vector $\mathbf{z}$, since we only care about the quantizing noise, the power of the received signal is set to 1/16384 (W), no outside noise is assumed to be included (but the noise effect will be verified in the following SNR simulations). Figure 4 demonstrate the GA procedure to select the *bus-usage parameter* vector $\mathbf{z}$. In the figure, the top subfigure shows that the *fitness* values of all the individuals improve as the evolution going on, i.e., the *fitness* value of the detector is -51.766 that means the final *SQNR* can reach 51.766 dB through the design; meanwhile the bottom subfigure shows the best fit individual which corresponds to the best *bus-usage parameter* vector to have the minimum *SQNR*, i.e., the variable 1, 2, 3 and 4 of the best individual in Figure 4 corresponds to y_1, y_2, y_3 and y_4. Since that $[y_1, y_2, y_3, y_4] = [6.26, -0.54, 0.54, 2.36]$, we can get the selected parameters for fixed-point PRACH detector according to the relation that

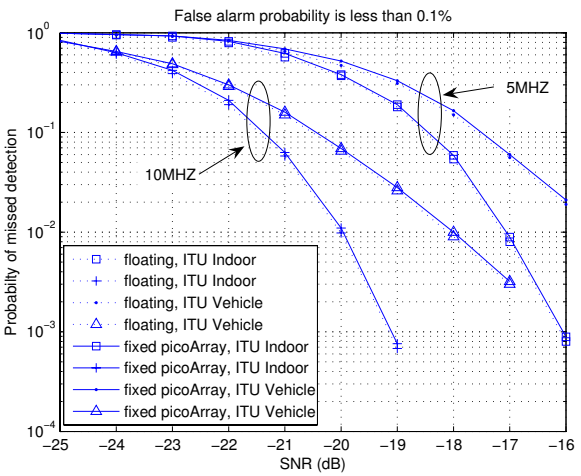

Figure 6: Performance comparisons in fading channel cases.

$\mathbf{z} = floor(\mathbf{y})$, which are $[z_1, z_2, z_3, z_4] = [6, -1, 0, 2]$. Then recall the pre-determined parameters $[x_E, x_F] = [1, 5]$, we successfully achieve the design.

The speedup brought by GA is a factor of 33 compared to the exhaustive search for the solution of design problem (note speedup is computed from the numbers of searched $\mathbf{z}$ candidates in above two search methods.). Based on the proposed fixed-point design of PRACH detector, we are able to realize it on the software platform of picoArray [16] that is a fixed-point DSP with multiple processing cores.

The performance comparisons in AWGN channel are shown as Figure 5, in which SNR is the signal-to-noise ratio of received signal and the floating-point performance is presented for the purpose of comparison. And the power of the received signal is still 1/16384 (W). And the fixed-point performance includes both the algorithms simulated in Matlab and picoArray's software platform. Furthermore, the performance comparisons in fading channel cases are illustrated in Figure 6. We can find in above two SNR simulations the fixed-point bus-usage design causes little performance loss that is not larger than 0.1dB. Therefore, the design is acceptable under the constraint of bus bandwidth.

Through further simulation, if 0.1dB is the maximum allowed performance loss caused by the fixed-point design, we find the dynamic range of the received signal power is 41dB, i.e., the power can vary from 6.21×10^{-7} (W) to 7.80×10^{-3} (W).

6. CONCLUSION

This paper addresses the problem of fixed-point bus-usage in the DSP environment with limited bus bandwidth. Our solution is adopting the genetic algorithm to handle the corresponding problem of constrainted optimization. Since Matlab has already provided us the tools of quantization and GA, the proposed design methodology is definitely convenient for practical applications. The fixed-point bus-usage design of PRACH detector under the aforementioned constraint in DSP environment is presented.

REFERENCES

[1] D. Menard, D. Chillet, and O. Sentieys, "Floating-to-Fixed-Point Conversion for Digital Signal Processors," *EURASIP Journal on Applied Signal Processing*, pp. 1–19, January 2006.

[2] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, "Wordlength optimization for linear digital signal processing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 10, pp. 1432–1442, 2003.

[3] K. Han and B. L. Evans, "Optimum Wordlength Search Using Sensitivity Information," *EURASIP Journal on Applied Signal Processing*, pp. 1–14, January 2006.

[4] W. Sung and K. I. Kum, "Simulation-based word-length optimization method for fixed-point digital signal processing systems," *IEEE Transactions on Signal Processing*, vol 43, no. 12, pp. 3087–3090, 1995.

[5] H. Choi and W. P. Burleson, "Search-based wordlength optimization for VLSI/DSP synthesis," in *IEEE Workshop on VLSI Signal Processing*, pp. 198–207, October 1994.

[6] A. Leon, J. Shin, K. Tam, W. Bryg, F. Schumacher, P. Kongetira, D.Weisner, and A. Strong, "A power-efficienct high-throughput 32-thread SPARC processor," in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 98–99, February 2006.

[7] H. Schmit, D. Whelihan, M. Moe, B. Levine, and R. R. Taylor, "PipeRench: A virtualized programmable datapath in 0.18 micron technology," in *IEEE Custom Integrated Circuits Conference (CICC)*, pp. 63–66, May 2002.

[8] A. M. Jones and M. Butts, "TeraOPS hardware: A new massively-parallel MIMD computing fabric IC," in *Hotchips*, August 2006.

[9] G. A. Wall, J. G. Hanko and J. D. Northcutt, "Bus bandwidth management in a high resolution video workstation," in *Proceedings of the Third International Workshop*, November 1992.

[10] C.D. Antonopoulos, D.S. Nikolopoulos and T.S. Papatheodorou, "Scheduling algorithms with bus bandwidth considerations for SMPs," in *ICPP*, October 2003.

[11] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time Signal Processing*, Second Edition. Pearson Education, 2005.

[12] Z. Y. Yu, *High Performance and Energy Efficient Multi-core Systems for DSP Applications*, PhD thesis, University of California, CA, USA, 2007.

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, First Edition. Addison-Wesley Longman, 1989.

[14] 3GPP Technical Specifications and Technical Reports. Web Site, http://www.3gpp.org.

[15] J. J. Benedetto, J. D. Donatelli, I. Konstantinidis, and C. Shaw, "A Doppler Statistic for Zero Autocorrelation Waveforms," in *the 40th Annual Conference on Information Sciences and Systems*, 2006, pp. 1403–1407.

[16] picoChip. Web Site, http://www.picochip.com.