

HIGH-PERFORMANCE ADAPTIVE G.P.S ATTITUDE DETERMINATION VLSI ARCHITECTURE

Evangelos F. Stefatos and Tughrul Arslan

School of Engineering & Electronics
The University of Edinburgh, King's Buildings, Mayfield Rd, Edinburgh
EH9 3JL

{Evangelos.Stefatos,Tughrul.Arslan}@ee.ed.ac.uk

Abstract

This paper presents an adaptive VLSI hardware architecture of an Attitude Determination System (ADS), which is based on Global-Positioning-System (GPS) measurements. The system composes the digital core of a GPS receiver, which manipulates the input data in order to resolve the attitude of a vehicle. The adaptation of this architecture is achieved by using a fine-grained Parallel Genetic Algorithm (PGA) that is employed to compute more efficiently the attitude determination algorithm in terms of speed performance. For the purpose of this paper the PGA consists of 64 Processing Elements (PEs), which are connected in a formation of an 8x8 array. Moreover, the hardware block that computes the fitness function of the PGA employs Coordinate Rotation Digital Computer (CORDIC) algorithms in order to further increase the throughput rate of the ADS.

1. INTRODUCTION

This paper presents the hardware VLSI implementation of an Attitude Determination System (ADS), which composes the digital part of a GPS receiver. The attitude of a rigid body platform is determined by the orientation of the specified body frame coordinate system with respect to the reference coordinate system. The proposed ADS is an adaptive VLSI hardware architecture [1] that uses a PGA to manipulate the signals received from the GPS receiver. The PGA is a fine-grained model that aims at increasing the performance of the system in terms of speed and enhancing the quality of the resolved solutions. There are two key points associated with this architecture. Firstly, the overall system consists of 64 PEs that each is participating on the implementation of the PGA and secondly the fact that each PE employs efficient hardware algorithms for the computation of trigonometric functions

(sine, cosine), which are known as CORDIC algorithms [2].

Determination and control of attitude is a crucial issue relative to the navigation system design of suspended vehicles in space, like unmanned spacecrafts, telescopes, mobile satellites etc. Recently, there has been intensive research in exploring more efficient approaches in terms of accuracy, speed and complexity to determine the attitude of a vehicle. The most promising of them are based on Differential GPS (DGPS) that measures the phase difference of the GPS carrier signal observed from three antennas attached on the body of the vehicle. The most important concern in DGPS attitude determination is to resolve the correct carrier phase integer ambiguity values. In literature there have been several approaches that have been incorporated to solve this problem. A brief reference of these can be summarized in the following techniques: Ambiguity-Function-Method (AFM) [3], Least-Squares-Search (LSS) [4], Fast-Ambiguity-Search-Filter (FASF) [5] and Lambda-Decomposition [6].

The digital core, which is presented in this paper, employs the AFM algorithm that presents a significant advantage in respect to the other mentioned techniques. This is the fact that AFM is immune to cycle slips. However, AFM algorithm is more demanding in terms of the computational power that is needed to implement the algorithm. This drawback makes essential the employment of a PGA [7] that overcomes this problem by maximizing the instantaneous search area space of the AFM.

The rest of this paper is organized as follows. Section 2 gives a background reference to PGAs and describes the specific architecture that has been employed. In addition, Section 3 outlines the search method of the PGA that includes the analysis of the fitness function and the definition of the chromosome that represents a potential attitude value. Section 4 presents the main units that compose the hardware design of each PE of the PGA.

Section 5 discusses the simulation results associated with the overall implementation of the PGA. In Section 6, the throughput rate of the system is calculated by analyzing the timing results retrieved from the synthesis process. In addition the characteristics of the chip layout are presented. Finally, Section 7 presents the results of the post-layout power analysis of the chip.

2. DESCRIPTION OF THE PARALLEL GA

Genetic Algorithms (GAs) [8] are efficient search/optimization methods based on principles that have been adopted from biology and natural evolution. However, conventional GAs are not always suitable for complex, multi-objective problems because they usually waste a lot of time searching very large area spaces. Moreover, the achieved optimization is not always effective because they can be trapped in sub-optimal solutions. Hence, applications that deal with multi-objective problems and concurrently require real-time response, introduce bigger population that consequently implies higher computational cost in order to evaluate each one of the individuals that form the population. For the purpose of this paper, a Fine-Grained PGA [9] has been employed to cope with the attitude determination problem.

This model of parallelism partitions the population into a large number of sub-populations and moreover individuals can migrate to any other sub-population. The size of the grain defines the ratio of the time spent in computation over communication respectively. Therefore, an ideally fine-grained PGA implementation in hardware has a distinctive processing unit to compute the evaluation stage for each individual. Following this guideline, we implemented the synthesizable Verilog code of a fine-grained PGA that consists of 64 PEs that share a population of 64 individuals. Consequently, each member of the population performs *Crossover* with its immediate neighbours, where the neighbourhood is defined by the topology and some distance parameter [10] depicted in Figure 1.

This architecture presents significant advantages in respect to other parallel approaches (global parallelization, coarse-grained PGAs etc.) because the average time that is required by the GA to converge, is defined as $O(N \cdot \log(N))$ function evaluations [9], where N is denoted the size of the population used.

The Finite-State-Machine (FSM) that controls the operation of each PE is identical for each one of them. Figure 2 depicts a brief description of the states that compose the FSM. It can be deduced that in the third state (*Exchange Data*) of the FSM, each PE communicates with its four adjacent PEs that exist in the north, east, south and west direction.

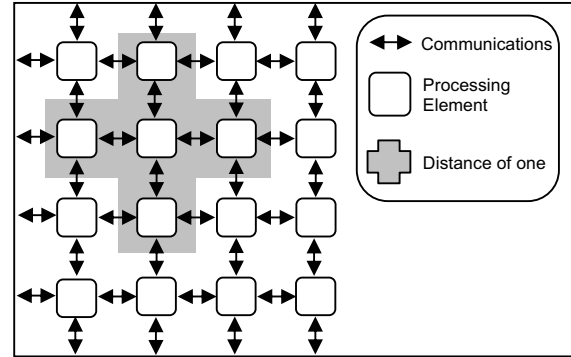


Figure 1: Topology of the PGA

Thus, it forwards to them its own best solution (chromosome) with its equivalent fitness score and respectively delivers the same information from each one of its neighbours. In addition, at the state of *Selection* the best delivered chromosome is selected, according to its fitness score, to participate with the unique chromosome that forms the population of each PE in a new generation.

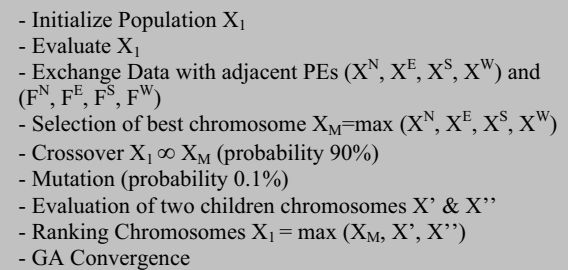


Figure 2: FSM of the PGA

3. DESCRIPTION OF PGA SEARCH METHOD

The fitness function of the proposed PGA comprises one of the most significant units within the evolution mechanism because apart from the fact that it guides the mechanism of the evolution, the mathematical formula that expresses it, affects significantly the performance of the system in terms of speed, area and power dissipation.

The detailed analysis and definition of the fitness function is given in [11]. The formula consists of multiplications, additions and subtractions of trigonometric functions (sine, cosine) that are quite demanding in terms of computational time. Therefore, in order to decrease the intensive computational processing that the *evaluation* procedure of the PGA, puts in each PE, efficient hardware algorithms have been employed, which are known as CORDIC algorithms [2]. According to these, the trigonometric functions are calculated with quite high precision, based on iterative methods, which

utilize only shifting and adding/subtracting operations. A significant advantage that this method presents in respect to conventional techniques is the real-time response they guarantee independently of the value of the angle they calculate. This superiority comes from the fact that these algorithms do not use huge amount of memories in order to store the potential results of the trigonometric functions.

Finally, the 64 PEs that form the PGA perform a parallel search to identify optimal values for the following parameters that define the chromosome of the PGA: *Azimuth Angle*, *Elevation Angle* and *Length* of the baseline. The first two parameters are used to determine the attitude of an object. The length of the baseline is defined as the distance between the two receiver antennas that are attached on the vehicle. The number of bits that are used in order to define these three parameters plays significant role in the resolution that it can be achieved by the final GPS receiver. For the purpose of this paper the search range of the azimuth angle is 360 degrees and because the system uses 14 bit to represent this value, the final resolution is 0.022 degrees. Similarly, the range step for the elevation angle (search range ± 15 degrees – 10 bit representation) and the length of the baseline (16.27 \pm 0.10m – 8 bit representation) is 0.030 degrees and 0.78 mm respectively.

4. SYSTEM ARCHITECTURE

The hardware implementation of each PE consists of three instantiated modules. Particularly, there is a module called GA that implements all the functionality relative to the operations that form a conventional GA, apart from the fitness evaluation operation. In addition there is the fitness-evaluation (FE) module that includes arithmetic units (cordic, adder/subtractor, multiplier), which compute the formula that expresses the fitness of the GA. Finally, there is the module of a Pseudo-Random-Binary-Sequence (PRBS) that generates random numbers, which are used for the purpose of the crossover and mutation operation. In Figure 3 is depicted the schematic of the top-level module of the PE, as it was captured from Cadence synthesis tool.

4.1. GA Module

This module is an amalgam of sequential and combinational logic. The sequential structure includes the implementation of a FSM that controls the operations of the GA. Each operation of the GA is expressed by a separate state of the FSM. Excluding the FE module, all the other operations have been implemented by using purely combinational logic. The interface that is established between the GA with the other two units is

depicted in Figure 3. As it can be seen the GA module controls the value of the reset signal of the FE unit. Hence, FE operates only when the FSM is at the corresponding state. This strategy positively affects the power dissipation of the system because FE is the most power demanding unit.

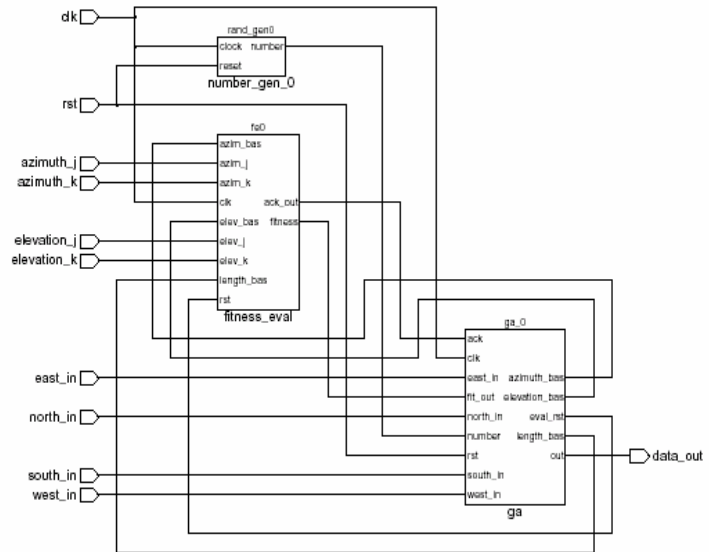


Figure 3: Schematic of the PE

4.2. FE Module

As it was mentioned in Section 4 the FE module has been assigned to compute the fitness-score of the GA. Therefore, its complexity is particularly increased with respect to the other modules. The most critical unit of this module is the CORDIC that presents a convergence method for evaluating the trigonometric functions that exist in the fitness function. The simple form of CORDIC is based on the observation that if a unit-length vector with end point $(x, y)=(1,0)$ is rotated by an angle Z , its new end point will be $(x,y)=(\cos(z), \sin(z))$. Thus, the values of $\cos(z)$ and $\sin(z)$ can be computed by finding the coordinates of the new end point after rotation by Z . After the analysis of the CORDIC theory the following equations derive that describe the overall functionality of the CORDIC unit.

$$\begin{aligned}
 x^{(i+1)} &= x^i - d_i y^i 2^{-i}, \\
 y^{(i+1)} &= y^i - d_i x^i 2^{-i}, \\
 z^{(i+1)} &= z^i - d_i \tan^{-1} 2^{-i}, \quad d_i \text{ can be either } -1 \text{ or } 1
 \end{aligned}$$

Figure 4: CORDIC Equations

The hardware computation of $x^{(i+1)}$ and $y^{(i+1)}$ can be performed by an i -bit right-shift and an addition/subtraction unit. However, to increase the performance of this unit all the computations are performed in parallel. Therefore, as it can also be seen in Figure 5 our architecture consists of 2 shifters and 3 addition/subtraction units, as the equations in Figure 4 imply. Moreover, there is a look-up-table (LUT), which stores sixteen pre-computed values of the function ($\tan^{-1}2^{-i}$) and three registers to store the values of the CORDIC equations.

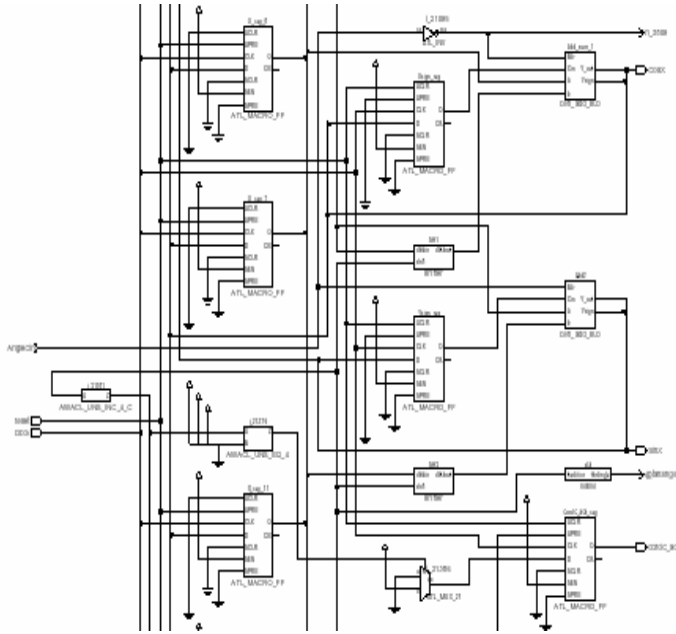


Figure 5: Schematic of the CORDIC unit

4.3. PRBS Module

The scope of this module is to generate random numbers, which are used during the evolution procedure. Initially, the generated numbers determine whether the new operation will perform crossover and mutation and then the next generated numbers define the point within the chromosome in which the two operations will be applied. The PRBS consists of a 9-bit parallel load register (8 DFFs), an EXNOR gate and a 9-bit counter. The length of the register implies that the range of the generated number is $2^9-1=511$. The EXNOR gate takes as inputs the middle and the last (MSb) bit of the register and guides its output in the Least-Significant-Bit (LSb) of the register. The purpose of the counter is to initialize each PRBS with a different seed so that the corresponding PE

to explore different search areas. Additionally, it prevents the PRBS from generating invalid numbers.

5. SIMULATION RESULTS

For the purpose of demonstrating the performance of the PGA in terms of the generations that each PE needs to converge, the ADS was simulated by using synthesizable-Verilog code. The results that have been obtained after the execution of almost 1253 generations for the whole array appear to be very promising in terms of the time that can be achieved to resolve the attitude determination of a vehicle. Obviously this speedup is mainly accomplished due to the parallelism that is achieved by the 64 PEs, which in conjunction run the PGA. Table 1 summarizes the obtained results that indicate the mean number of generations that the 64 PEs need in order for them to converge.

Table 1: Mean number of generations

Azimuth (satellite)		Elevation (satellite)		Generations
j-plain	k-plain	j-plain	k-plain	
98.875	78.9375	58.875	23.46875	3.93
98.125	78.4375	58.46875	23.21875	3.89
96.875	78.5625	56.625	19.46875	3.9
82.625	68.375	34.125	17.9375	4.16
31.75	20.125	6.8125	3.25	2.2
14.125	2.4375	2.46875	1.09375	1.06

At this point it is important to mention that during each generation two children chromosomes are produced and therefore there is the same number of evaluations associated with a new reproduction. Hence, considering the results of Table 1, it can be seen that the mean number of evaluations needed per PE is $3.19*2 \approx 7$. In addition, keeping in mind from Section II that statistically a single processor running solely a GA with population size of 64 needs approximately 116 evaluations ($64*\log(64)$), it can be estimated that the system becomes $116/8 \approx 14.5$ orders of magnitude faster by exploiting the features of the PGA because the 64 PEs explore concurrently different search areas. This estimation becomes more accurate as the size of the population increases. The calculation of the actual time (throughput) that is needed for the system to extract a new output value, it will be presented in Section V of this paper that discusses the results of the synthesis procedure of this design.

6. SYNTHESIS RESULTS

A version of a system that consists of 64 PEs was synthesized with Synopsys Design Compiler using UMC 0.18 micron standard cell technology library. The obtained results are associated with the clock frequency

(20MHz) of the system and the die area that a single PE occupies. Table 2 depicts the number of clocks that each module of the GA takes to be executed and how this number is translated in terms of actual time.

Table 2: Time schedule of the PGA

Module	Number of clock cycles	Execution time (nsec)
Exchange Phase	2	2*50 = 100
Selection	1	1*50 = 50
Crossover	1	1*50 = 50
Mutation	1	1*50 = 50
Initialization (1)	1	1*50 = 50
Evaluation (1)	116	116*50 = 5800
Initialization (2)	1	1*50 = 50
Evaluation (2)	116	116*50 = 5800
Chromosome Ranking	1	1*50 = 50
Convergence Decision	1	1*50 = 50
Generation	241	12050

In Section 4, it has been seen that the mean number of the generations is 4. From Table 2 it can be deduced that each generation of the GA takes 12050 nsec to be executed. Therefore, the throughput of the system is calculated to be approximately 48.2 μ sec. This number is indeed very promising for on-the-fly ambiguity resolution applications, considering that the update rate of GPS receivers is around 100Hz.

However, it is always very important to compare the performance results in terms of speed in comparison to the hardware resources that the specific architecture demands. The results associated with the layout generation of a single PE are summarized in Table 3.

Table 3: Layout Chip Results (mm²)

Genetic Algorithm Module	0.112
Fitness Evaluation Module	0.05
Pseudo Number Generator	0.003
Combinational Area	0.102
Non-Combinational Area	0.062
Net Area	0.060
Total Cell Area	0.165
Chip Area	0.225

7. POWER ANALYSIS

As it was mention in the introduction of this paper, the main disadvantage of the AFM algorithm, which is employed in this system, is the computational power that it consumes in comparison to the other attitude determination algorithms. In order to achieve more accurately results, the power calculation of the design was obtained with post-layout simulation considering the Standard-Delay-Format (SDF) information that was retrieved after the generation of the chip layout. In Figure

6 is depicted the hierarchy of the power that is dissipated by the internal cells and the switching activity of the interconnections, concerning a single PE. From the graph, it can be deduced that the cells consumes the 72% of the overall power that corresponds to almost 805 μ W.

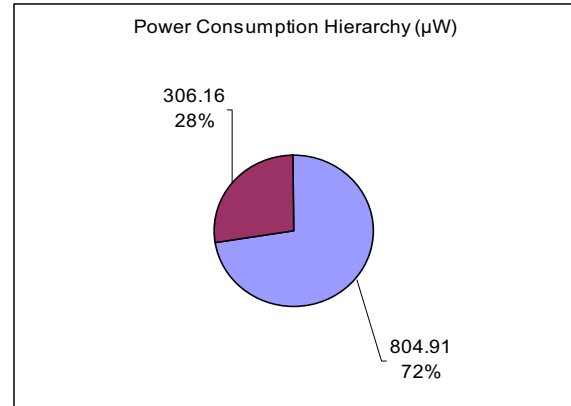


Figure 6: Post-Layout Power Consumption

Figure 7 illustrates the power hierarchy of the three modules that compose the whole design of a single PE and the switching activity within them. It can also be seen that the power consumed by the fitness evaluation module has been calculated separately from that dissipated by the rest GA modules. The huge difference verifies the complexity that the attitude determination algorithm introduces.

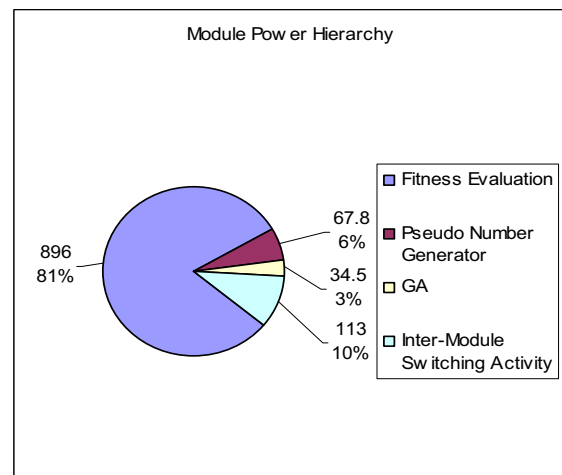


Figure 7: Power Hierarchy of the separate modules

8. CONCLUSIONS

In this paper the hardware implementation of a high-performance GPS based ADS was introduced. The analysis of the timing results that were obtained from this

research promises high-throughput rates, which are mainly achieved by employing real-time algorithms (CORDIC) and exploiting the features of an embedded fine-grained PGA. Future work can be done in order to optimize the design of the fitness evaluation module of the PGA and reduce the power that dissipates in order for the system to be viable for both time critical and low-power GPS navigation control applications.

9. REFERENCES

- [1] Stefatos, E. F., Arslan, T., "An efficient fault-tolerant VLSI architecture using parallel evolvable hardware technology", *Evolvable Hardware, 2004. Proceedings of 2004 NASA/DoD Conference on*, June 24-26, 2004, pp. 97-103.
- [2] Raymond J. Andraka, "Building a High Performance Bit Serial Processor in FPGA", *On-Chip System Design Conference* 1996.
- [3] C. C. Counselman and S. A. Gourevitch, "Miniature Interferometer Terminals for Earth Surveying: Ambiguity and Multipath with Global Positioning System", *IEEE Transactions on Geoscience and Remote Sensing*. GE-19, 4, October 1981.
- [4] R. Hatch, "Instantaneous Ambiguity Resolution", *Proceedings of KIS-90 Symposium*, Banff, Canada, August 1990.
- [5] D. Chen and G. Lachapelle, "A Comparison of the FASF and Least-Squares Search Algorithms for Ambiguity Resolution On-the-Fly", *Proceedings of KIS-94 Symposium*, Banff, Canada, August 1994.
- [6] P. J. G. Teunissen, "A New Method for Fast Carrier Phase Ambiguity Estimation", *Proceedings of IEEE PLANS*, Las Vegas, NV, April 1994.
- [7] J.Xu, T.Arslan, Q.Wang and D.Wan, "An EHW architecture for real-time GPS attitude determination based on parallel genetic algorithm", *NASA/DoD Conference on Evolvable Hardware*, pp. 133-141, 2002.
- [8] D. E. Goldberg, "Genetic Algorithms in Search, Optimisation & Machine Learning", *Addison-Wesley*, 1989.
- [9] E. Cantu-Paz, "A Summary of Research on Parallel Genetic Algorithms", *IlliGAL Report No. 95007. University of Illinois at Urbana-Champaign*, 1995.
- [10] B. C. H. Turton and T. Arslan, "A Parallel Genetic VLSI Architecture for Combinatorial Real-Time Applications", *1st IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp. 493-498, 12-14 September 1995.
- [11] J. Xu, T. Arslan, D. Wan and Q. Wang, "GPS Attitude Determination Using a Genetic Algorithm", *IEEE Congress on Evolutionary Computation (CEC'02), Volume 1*, pp. 998-1002, 2002.