IMPLEMENTATION OF HARDWARE SOFTWARE PARTITIONING IN EMBEDDED SYSTEM

MASYIRAH BINTI MOHD NOR

UNIVERSITI SAINS MALAYSIA

2018

IMPLEMENTATION OF HARDWARE SOFTWARE PARTITIONING IN EMBEDDED SYSTEM

By

MASYIRAH BINTI MOHD NOR

Thesis submitted in partial fulfillment of the

requirements for the degree of

Bachelor of Engineering (Electronic Engineering)

JUNE 2018

ACKNOWLEDGEMENTS

Alhamdulillah, all praise to Allah for the strengths and His blessing in completing this final year project and thesis.

First and foremost, I have to thank my final year project and research supervisor Assoc. Prof. Dr. Zaini Abdul Halim for her exemplary guidance, monitoring throughout time to time of doing my final year project. Without her assistance and dedicated involvement in every step throughout the process, this project would never been accomplish.

I would also like to show gratitude to my friend, who is study at University Sains Malaysia (Main Campus) that always willing to share his experience and knowledge to me and giving me advices about my final year project.

In addition, I want to thank to my course mate who are willing to spend a time with me, sitting together to helped me in completing this final year project through various stage.

Lastly, I thank almighty to my husband Mohd Azhar Bahari, my mother Meryam Ahmad and my daughter Nur Alayna Mikayla Mohd Azhar for providing me with unfailing support and continuous encouragement throughout my year of study and through the process of searching and writing this thesis. This accomplishment would not have been possible without them.

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURES

LIST OF ABBREVIATION

FPGA	Field Programmable Logic Array
ASIC	Application Specific Integrated Circuit
DSP	Digital Signal Processing
ASIP	Application Specific Instruction-Set Processors
PSO	Partical Swarm Optimization
GA	Genetic Algorithm
CLB	Configurable Logic Blocks
HDL	Hardware Description Language
CPU	Central Processing Unit
ALU	Arithmetic Logic Unit
SA	Simulated Annealing
ACO	Ant Colony Optimization
FCM	Fuzzy C-Means
RAM	Read and Write Memory
DAG	Direct Acyclic Graph
WSGA	Weighted Sum Genetic
NSGA-II	Nondominated Sorting Genetic

MOPSO-CD	Particle Swarm Optimization using Crowding Distance strategy
ER	Error Ratio
MFE	Maximum Pareto-Optimal Front Error
GD	Generational Distance

ABSTRAK

Pembahagian perkakasan/perisian adalah masalah penting dalam sistem tertanam. Ia terletak pada menentukan proses sistem tertanam yang perlu dilaksanakan pada perkakasan tertentu dan yang mana satu pada perisian. Untuk mencari satu bahagian yang optimum adalah sukar kerana bilangan besar dan ciri-ciri sistem yang berbeza perlu dipertimbangkan. Dalam kertas ini, kaedah heuristik digunakan untuk menilai prestasi algoritma PSO dan GA untuk mencapai kos terbaik, jumlah kawasan dan jumlah masa pelaksanaan. Algoritma PSO dan GA dipilih dalam projek ini untuk melaksanakan bahagian perisian perkakasan menggunakan Python 2.7.14. Kekangan algoritma ini digunakan untuk program ini disasarkan untuk jumlah kawasan (2500KB) dan jumlah masa pelaksanaan (2500μs). Tetapan parameter yang digunakan untuk kedua-dua algoritma adalah 15 bilangan tugas/nod, 500 bilangan lelaran maksimum, 100 bilangan zarah/saiz populasi dan kawasan perkakasan dan perisian yang telah ditetapkan dan masa pelaksanaan bagi setiap tugas sebagai masukkan awal algoritma . Kedua-dua algoritma ini mencapai kos yang sama iaitu 169.6022 dan penyelesaian optimum yang dicadangkan oleh kedua-dua algoritma adalah sembilan tugas yang perlu dijalankan dalam perkakasan dan enam tugas harus dijalankan dalam perisian. Hasil dari jumlah kawasan dan jumlah waktu pelaksanaan untuk PSO adalah 2495KB dan 2363μs. Sementara untuk GA, jumlah keseluruhannya ialah 1605KB dan jumlah masa pelaksanaannya adalah 2147μs. Sebagai kesimpulan, pembahagian perkakasan/perisian yang merupakan PSO dan GA telah berjaya dibangunkan dalam projek ini. Hasilnya menunjukkan bahawa, algoritma GA mempunyai jumlah kawasan yang lebih baik dan jumlah masa pelaksanaan tetapi kos yang sama berbanding dengan algoritma PSO. Walau bagaimanapun, untuk meningkatkan prestasi, penghibridan algoritma ini dicadangkan.

ABSTRACT

Hardware/Software partitioning is a crucial problem in embedded system. It resides on deciding which process of embedded system should be executed on specific hardware and which one on software. To find an optimal partition is hard due to large number and different characteristics of the system to be consider. In this paper, the heuristic method is used to evaluate the performance of PSO and GA algorithm in term of to achieve a best cost, total area and total execution time. PSO and GA algorithm are chosen in this project to perform hardware software partitioning using Python 2.7.14. The constraints of these algorithms applied to the program are targeted for total area (2500KB) and total execution time (2500μs). The parameter setting is used for both algorithms is 15 number of task/node, 500 maximum number of iteration, 100 no of particles/population size and the pre-defined hardware and software area and execution time for each task as the input of the algorithms. Both of the algorithms are achieve the same best cost which is 169.6022 and the optimum solution proposed by both algorithms are nine tasks should be run in hardware and six tasks should be run in software. The result of total area and total execution time for PSO is 2495KB and 2363μs. While for GA, the total area is 1605KB and the total execution time is 2147μs. As a conclusion, the hardware/software partitioning which is PSO and GA have been successfully develop in this project. The result shows that, the GA algorithm has better total area and total execution time but same cost compared to the PSO algorithm. However, in order to improve the performance, the hybridization of these algorithms is suggested.

CHAPTER 1

INTRODUCTION

1.1 Research Background

Embedded systems typically consist of application specific hardware parts, i.e. FPGA or ASICs, and programmable part, i.e., processors like DSPs or ASIPs. In comparisons to the hardware parts, the software parts are much easier and faster to develop and modify. Thus software is less expensive in term of development cost and time. Hardware, however, provides better performance [1].

Today, there are many algorithm that can be used to solve the Hardware/Software partitioning problem in embedded system such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search and etc.

In this project, algorithm that used to solve the Hardware/Software partitioning problem is Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Genetic algorithms are stochastic, non-linear optimization routines loosely based on theories of biological evolution, mechanics of natural selection and natural genetics [2]. Genetic algorithms were quickly adapted to a variety of contexts. In a simple genetic algorithm, research is regulated by three operators which is reproduction, crossover and mutation operator [3]. At each iteration, a set of candidate solutions is retained, and we'll generate better candidate solutions group with genetic operators and repeat until convergence occurs [4].

Particle Swarm Optimization (PSO) is a stochastic, iterative population-based evolutionary optimization algorithm. It uses the swarm intelligence that is based on social-psychological and biological social principle. By equivalent with the swarm intelligence, is swarm member (particle) takes advantage of private memory and has a degree of randomness in its movement as well as knowledge gained by the whole swarm to discover the best available food source. Thus the problem of food search can be solved by optimizing fitness function [5].

1.2 Problem Statement

Hardware/Software partitioning is one of the most crucial step in the co-design methodology. It can be presented as cooperative design of software and hardware components to decide the type of the algorithms that can be used to solve hardware software partitioning and to achieve the best embedded system performance. Traditionally, this step was carried out manually by system designer decision with small number of tasks. With increasing complexity of embedded application, automatic partitioning has emerged [5].

Thus algorithms to implement the hardware software partitioning can be divided into two method which known as Exact method and Heuristic method. Exact method tends to be quite slow for bigger dimensions of the problem. Therefore, Heuristic method which known as Evolutionary method is proposed for partitioning problems in this project in order to increase the overall performance of and at the same time reduce the cost of the system.

1.2 Objective of Research

i. To develop the Particle Swarm Optimization (PSO) algorithm and Genetic Algorithm (GA).

ii. To identify the best cost, total area and total execution time for Particle Swarm Optimization (PSO) algorithm and Genetic Algorithm (GA).

1.3 Scope Project

In this project, the experiment will perfoms using the Python 2.7.14 and the target of stopping criterion is made for total area (2500KB) and total execution time (2500μs) . The performance of Particle Swarm Optimization (PSO) algorithms and Genetic Algorithm (GA) will test seperately. Then, the cost, total area and total execution time will compare for both algorithms. The binary solution of hardware node and software node are assume with the value of 0 for software node and value of 1 for hardware node. After that, the result of the experiment will be analyze.

1.4 Thesis Outline

This thesis consists of 5 Chapters. Chapter 1 is introduction, which discusses about research background, problem statement, objective of the research and scope of project.

Chapter 2 is abaout literature review that reviews about related work done on Implementation of Hardware Software Partitioning in Embedded System and summary of related work.

Chapter 3 is methodology, it discuss about the flow of each algorithms used to be implement in the embedded system. Besides, the way to be implement the algorithms also being discussed.

Chapter 4 presents the result and analysis of the experiment designed in previous chapter. The performance of each algorithm in term of to achieve a best cost, total area and total execution time are compared. Then the analysis regarding the results will discussed.

Chapter 5 is conclusion that presents the summary of the work and results of the project. Future works of this project is also included.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

Recently, researchers have started to do research on hardware/software partitioning in embedded system. The Exact and Heuristics methods for hardware/software partitioning in embedded system have been studied for past few years. In this chapter, some fundamental knowledge related to project has been discussed. Besides, previous work done on implementation of hardware/software partitioning in embedded system also has been reviewed in this chapter also.

2.2 Field Programmable Gate Array (FPGA)

Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects [6]. The word Array is means of indicate a series of columns and rows of gates that can be programmed by the end user. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing process.

The programmable logic blocks that can be wired in different configurations. Thus, the different operations can be used, when blocks create a physical array of logic gates. This is because the gates are customizable, and any computing task of FPGAs can be optimized. Therefore, this gives FPGAs the potential to perform operations several times faster than a hard-wired processor [7].

Hardware description language, or HDL is used to specify the configuration of the logic block. HDL commands is used to configure the gate interconnects as well as

the gates themselves. A gate may be assigned a boolean operator and by linking several gates together, it is possible to perform advanced logic operations [7].

2.3 Processor (CPU)

Processor or Central Processing Unit (CPU) is the unit, which performs most of the processing inside a computer and carries out the instructions of a computer program. It performs the basic arithmetical, logical, and input/output operations of computer system [8]. The arithmetic logic unit (ALU) is the component of CPU that performs arithmetic and logic operations, register that store the result of ALU, and control unit that control the fetching and execution of instruction.

2.4 Related Works

I.Mhadhbi, et al. [5] had presented a comparative study of different hardware/software partitioning algorithms due to critical problem in the co-design methodology. This experiment are performed on Intel Celeron CPU having 2.16GHZ processor speed and 2GHZ RAM. After that, hardware/software optimization algorithm was implemented in Matlab environment by using Verilog language and executed in Window 7 Operating System. The researcher used general-purposed heuristic algorithms sush as Particle Swarm Optimization (PSO) algorithm, Simulated Annealing (SA) algorithm, Genetic Algorithm (GA), Ant Colony Optimization (ACO) algorithm and the Fuzzy C-Means (FCM) algorithm. The experiment carried out to decrease the overall implementation cost and increase system performance like execution time, power consumption, area and etc. Then, Imene MHADHBI compare PSO algorithm with SA, GA, ACO, FCM algorithms for both binary and extended partitioning approaches because it was characterized as the most attractive algorithm to solve hardware/software partitioning.

For the target of the experiment, it focus on hardware/software partitioning that can be applied for both binary and multi-way partitioning cases that starting from a graph specificaion. After simulation, it showed the PSO based algorithms outperforms SA, GA, ACO, FCM in both partitioning generated solution and processing time.

The research done by I.Mhadhbi, et al. [5] was very good because he compared each performance of the algorithms. Therefore, we can clearly seen the performance of each algorithms.

K.S. Mishra, Ashish, et al. [9] had presented the optimum solution of problem for hardware/software partitioning using Task Graph of Genetic Algorithm (GA) technique. The purposed method help to maintain the trade-off between cost and performance and get the optimized result. The task graph with eight node has been taken for partition optimization. Then, experimentation for various task graph for particular input value are made and the cost result and scheduling of task are tabulated. After that, to proof the correctness of the result, manual crosscheck has been made for all possible combination of partition.The design problem of embedded system has been promising approach to address, by employing strategy that can decides the hardware configuration to be employed and scheduling of the tasks. Next, the frame work had presented that can be used to investigate the tradeoff between cost and delay. This research is good, since Genetic Algorithm (GA) give an optimum result and corresponding partition. But it didn't compared with the other algorithm to show the performance of other algorithm can solve the problem of hardware/software partitioning in term of cost and delay.

E, Review, Guoshuai Li, et al. [10] had proposed a new partitioning algorithm based on Genetic Algorithm (GA). This experiment is performing in Pentium® Dual-

Core 2.5GHZ CPU, 2G Internal Storage and the coding was codded in Matlab R2007a. The execution process will be done in Window XP Operating System. The concept of hardware oriented is put forward and used in the process of producing initial colony and mutating. In this process, the reseacher not only avoided the blindness of creating initial colony through random method but the direction of mutation also controlled. Besides, an adaptive method was design for crossover/mutation probability. Then, for testing process, the several DAGs that have specefied number of node and average branches are randomly created. After that, it let every node asscociated with one function whose cost with 30, 60, 90, 120, 200, 400 nodes respectively. For verifying the effectiveness of the algorithms, ANGSA and GA is used to compare with proposed algorithm. To make it fair, all parameter of these algorithms is set on the same benchmark. To compare the proposed algorithm and the other two algorithms, the three algorithms is let to run 30 times respectively for the 6 DAGs and the average value of the results for each DAG are calculated. This research is good, since it compare with three algorithms. But, it didn't specify the detailed of the proposed algorithm used.

M.C. Bhuvaneswari,et al. [11] applied a Heuristic/evolutionary method for partitioning problem. The implementation of these experiment is aims at finding an optimal trade-off between conflicting parameter such as execution time and area. The design was focused on the application of three multi-objective optimization algorithm by using Weighted Sum Genetic Algorithm (WSGA), Nondominated Sorting Genetic Algorithm (NSGA-II), and Multi-objective Particle Swarm Optimization using Crowding Distance strategy (MOPSO-CD). The algorithmsm is programmed in C language and was run using 2.80 GHz, Pentium-IV processor with 1 GB RAM. Then, the pareto-optimal solutions for 10-node Directed Acyclic Graph (DAG) is obtained and analyzed for minimizing the area and task execution time problem. The true pareto-

optimal solutions found by determined the calculation of the performance metrics which is Error Ratio (ER), Maximum Pareto-Optimal Front Error (MFE), and Generational Distance (GD). In this research, M.C. Bhuvaneswari,et al [11], used exhaustive search to determine the true pareto-optimal solution. Then performance metrics ER, MFE and GD can be calculated.

M. Riabi, Y. Manai, and J. Haggege, et al. [12] had proposed a new method of Hardware/Software partitioning approach for embedded system design based on Genetic Algorithm (GA) to resolve the problem for AC drive application using Field Programmable Gate Array (FPGA). The objective of the experiment is to optimize a multi-objective problem to find optimal solution. Besides that, the binary of Genetic Algorithm (GA) used in this experiment as the purposed of Hw/Sw partitioning approach. To optimize the multi-objective function of partitioning problem, the Genetic Algorithm (GA) is used to tested the proper operation that depends on the characteristics of functional modules.

After that, the process of tested will be stop until reaching the objectives limit or maximum number of generation. Then, it will compare the results with those given by Non-Dominated Sorting Genetic Algorithm (NSGAII). This research is good because it make the comparison of the obtained result with the other algorithm. But it will be very good, if his try to test both algorithm without simply taking the result from other research. As a conclusion, Table 2.1 shows the summary of researches which is related to the hardware/software partitioning by using heuristic method.

Table 2.1 : Summary of Related Works

References	Year Published	Type of Algorithm	Measurement
[5] I. Mhadhbi	2017	PSO, GA, FCM, SA, ACO	Performance of time and cost
[9] K. S. Mishra, Ashish	2014	GA	Investigate the tradeoff between cost and delay.
[10] E, Review, Guoshuai Li	2014	Propose algorithm, GA, ANGSA	Simulate task cost
[11] M.C. Bhuvaneswari	2015	WSGA, NSGA-II, MOPSO-CD	Minimizing the area and task execution time
[12] M. Riabi, Y. Manai, and J. Haggege	2015	GA and NSGAII	To optimize multi-objective problem

2.5 Summary

In overall, many researchers did research on hardware software partitioning in embedded system by using Heuristic method. Researcher Imene MHADHBI [3] used general-proposed heuristic algorithms such as Particle Swarm Optimization (PSO) algorithm, Simulated Annealing (SA) algorithm, Genetic Algorithm (GA), Ant Colony Optimization (ACO) algorithm and the Fuzzy C-Means (FCM) algorithm to decrease the overall implementation cost and increase system performance like execution time, power consumption, area and etc. Then, most of the finding showed that, many researchers used Particle Swarm Optimization (PSO) as the one of the main algorithms to make a comparison. This is due to the powerful optimization that has been applied to wide range of solving many optimization problem. Besides, GA also one of the good algorithms in solving the problem because the operator can control the exploration and exploitation search projection: mutation, crossover and selection.

CHAPTER 3

METHODOLOGY

3.1 Overview

In this project, the algorithm of Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) will be implemented in embedded system using python 2.7.14. The constraints of these algorithm applied to the program are fixed for total area (2500KB) and execution time (2500μs). The performance analysis of PSO and GA is constructed seperately. These algorithm is design to overcome the problems of an embedded system by deciding the implementation in the field of hardware/software partitioning. The target of this experiment, is to achieve best cost, area and execution time. The result of each algorithms can be measured and analyze at python shell. With these collected and analyze data, the performance of each algorithms can be easily compared. Detailed data and experiment procedure will be explained in this chapter.

3.2 Implementation Platform

In this project, the algorithm of Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) will be used for implementation. Then, the performance analysis of proposed scheme with PSO and GA is constructed individually.

3.2.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO), which is an algorithm of intelligent optimization of swarm, was proposed by Kennedy and Eberhart in 1990s [13][14]. PSO is inspired by social behavior of bird flocking or fish schooling [15]. The bird-flocking is scenario of PSO learned and used it to solve the optimization problem. Each of single solution in PSO is a "bird" in the search space [1].

To design the PSO model, firstly PSO will be initialized with a random particles (solution) and evaluate the fitness of each solution. The solution (fitness) will updated during each of the iteration by the following two "best" value. The pbest is a position vector of the best solution (fitness) of the particle has achieved so far. While the current global best (gbest) is the another best position to tracked the best position obtained so far by any particle in the population.

After finding the best values, the velocity and position is updated according to equation (3.1) and equation (3.2) [16][17]. Then, the PSO flow chart is shown in Figure 3.1.

$$v[i] = (w*v[i]) + c1r1 (pbest[i] - x[i]) + c2r2(gbest - x[i]) \quad (3.1)$$

$$x[i] = x[i] + v[i] \quad (3.2)$$

where;

v[i] = velocity of particle number (i)

r1 = random number between 0 and 1

r2 = random number between 0 and 1

c1 = self-confidence (cognitive) factor

c2 = swarm confidence (social) factor

w = the inertia factor that takes values downward from1 to 0

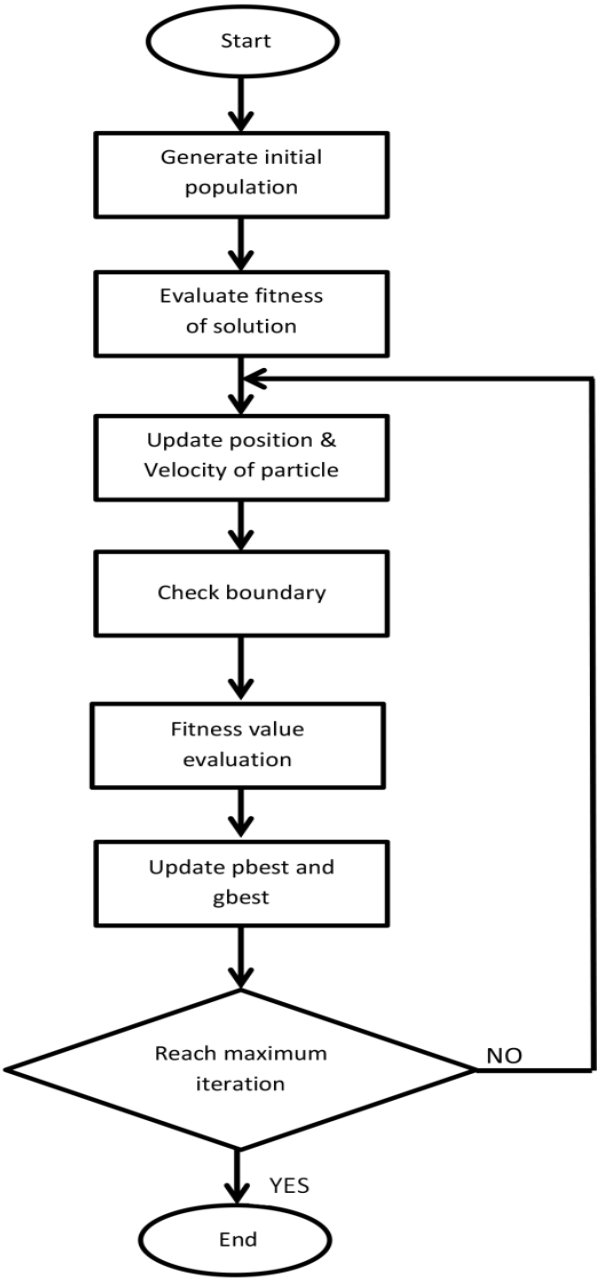


Figure 3.1 : PSO flow chart

3.2.2 Genetic Algorithm (GA)

Genetic algorithm have been developed by Holland [3][18]. It is a global optimization algorithm based on the theory of natural selection. In GA, individuals of a population having good phenotypic characteristics have greater survival and reproduction possibilities [19]. There are three sample of possible solution (individuals) and employs for optimization which is mutation, crossover and selection [1].

Figure 3.2 shown the flow chart of GA algorithm. Firstly, the initial population is generated randomly by GA and the fitness of each solution will evaluated. Once the fitness is done, the selection operation on the population is perform and mixing them in a crossover. The crossover operation requires that a comman node should exist between the two parents [20]. Next, the mutation operator will take and the extra process of solutions with lowest fitness will be eliminated after sorting all the solution according to fitness.

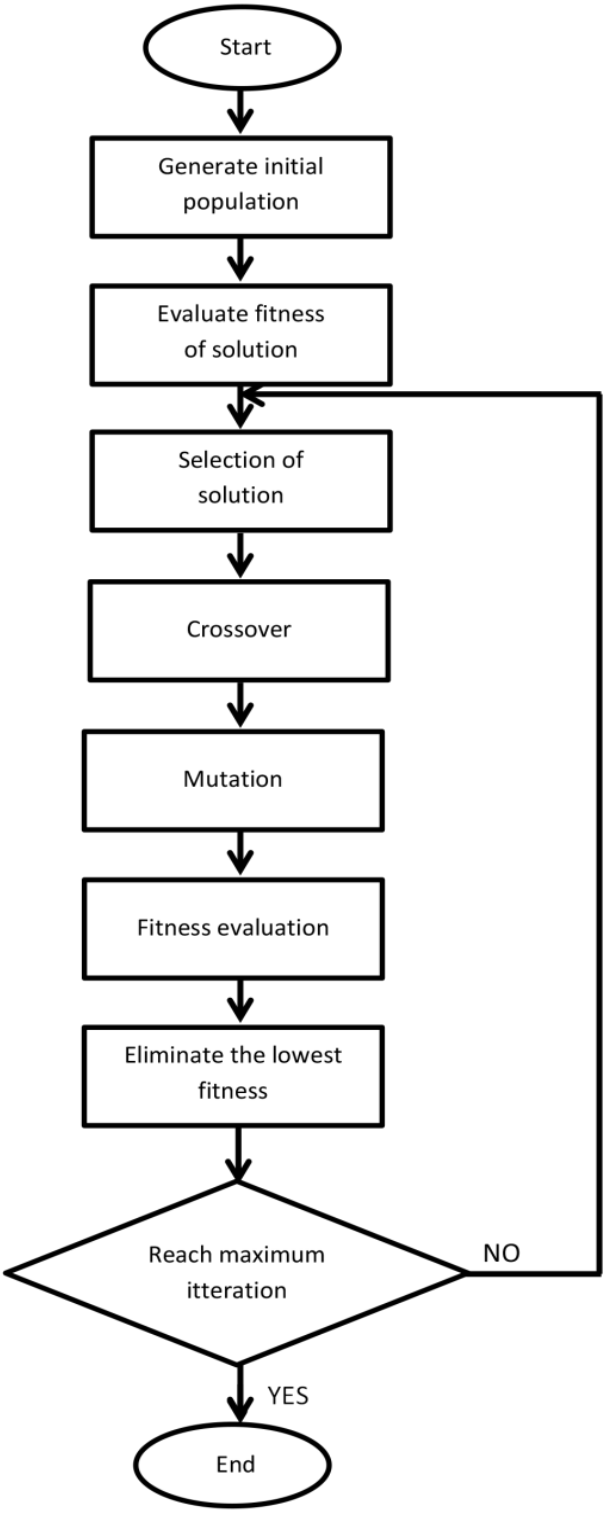


Figure 3.2 : GA algorithm flow chart

3.3 Parameter Setting

The input to the scripts is a design that consists of the number of nodes. Each node is associated with hardware and software cost parameter. The cost function is shown in equation (3.3) which is sum of the hardware area, hardware time, software area and software time. The "All AHW " and "All THW" are the maximum value of hardware area and hardware execution time when all node mapped to the hardware, while "All ASW" and "All TSW" are the maximum value of software area and software execution time when all node mapped to the software. The multiplication with 100 is only for readability.

$$Total\ Cost = [\sum \left(\frac{AHW}{All\ AHW}\right) + \left(\frac{ASW}{All\ ASW}\right) + (\frac{THW}{All\ TSW}) + \left(\frac{TSW}{All\ TSW}\right)] * 1\,00 \qquad (3.3)$$

For experimental purpose, the parameter are assigned for both algorithms as follows:

1. Maximum itteration/generation = 500
2. No of particles (population size) = 100
3. Number of nodes/tasks = 15
4. Pre-defined hardware and software area and execution time for each task

To get the best result for GA, the maximum iteration/generation is set as the suggested value in [21] followed with the PSO, because to make the fair comparison. For no of particles(population size), the more particles in the swarm used, the larger the initial diversity of the swarm. Thus it can provided a good uniform initialization scheme is used to initialize the particles [22]. Therefore 100 number of particles(population size) are set for both algorithm.

Table 3.1 shown the Pre-defined Hardware and Software Area and Execution Time for different task. The Pre-defined Hardware and Software Area and Execution Time is used as the input in the python PSO and GA algorithm.

Table 3.1 : Pre-defined Hardware and Software Area and Execution Time for different task

Node/Task	**Hardware Area (KB)**	**Hardware Time (µs)**	**Software Area (KB)**	**Software Time (µs)**
1	166	318	24	526
2	158	133	45	209
3	201	255	58	511
4	152	242	23	366
5	351	6	80	21
6	194	92	55	178
7	162	57	24	92
8	2342	8	237	204
9	239	117	67	277
10	261	122	59	317
11	178	45	6	158
12	92	58	92	201
13	204	23	57	152
14	277	80	8	351
15	317	55	117	194

A binary solution is expected by assume a “0” value (if node is implemented in software) or a “1” value (if the node is implemented in hardware). For a binary problem, the node value must be 0 or 1. Therefore, the particle is rounded by using hard decision rounding (HDR) where the node is mapped to hardware if the node value is lower than 0.5 and mapped to software if node value is greater than 0.5.

3.4 GA Operator Choice

Genetic Algorithm (GA) is a method of searching. It searches a result equal to or close to the answer of a given problem [23]. In GA algorithm, there are several ways to perform operators. Selection, crossover and mutation are the 3 of the operator that used in this project. Each of them has different methods to carry out and the choice of method will affect the output of the system as the GA algorithm solely depends on these three operators.

Selection is the process of selecting the chromosomes to apply Steady State Genetic Algorithm [24]. Thus, the method used in this project is Fitness Proportionate Selection. This method is used because of it is simple and fast for large number of particle.

There are the basic operation of selection:

I. The fitness for each particle is normalized.
II. The population is sorted by descending fitness value.
III. Accumulated normalized fitness value are computed (The accumulated fitness of the last individual should be 1).
IV. The random number R between 0 and 1 is chosen.

The selected individual is the last one whose accumulated normalized value is smaller than R.

Next operator is crossover. Crossover is an operator that allows the combination of the genetic material of two or more solutions [25]. This operator creates one child offspring from two parents, and only those offspring that are the best to fit stay, while other is discard [26]. The first step is the selection of a potential mate partner. The child gene can be obtained by using equation (3.4).

$$O1 = P1 + R(P2 - P1)O1 = P1 + R(P2 - P1) \quad (3.4)$$

Where,

O1 = The child gene

P1 and P2 = Parent genes

R = Random number between 0 and 1

After crossover the strings are subjected to mutation. Mutation prevents the algorithm to be trapped in a local minimum. The uniform mutation is applied into this algorithm for mutation operator. This operator replaces the original value of the chosen gene with a uniform random value generated between lower and upper boundary for gene.

3.5 Summary

This chapter discusses about the way to implementing Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) in embedded system using Python. The process flow of each algorithms has been explained briefly. The parameter setup has been stated and used for experimental purpose.Then, the result and analysis of the experiments will be discussed in chapter 4.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Overview

After the process of implementation of PSO and GA algorithm in Python2.7.14 by followed method that already explained in previous chapter. The experiment was carried out to evaluate the performance of algorithms in term of to achieve best cost, total area and total execution time. Then, the results was tabulated and comparisons of the performance for each algorithms have been discussed.

4.2 PSO algorithm

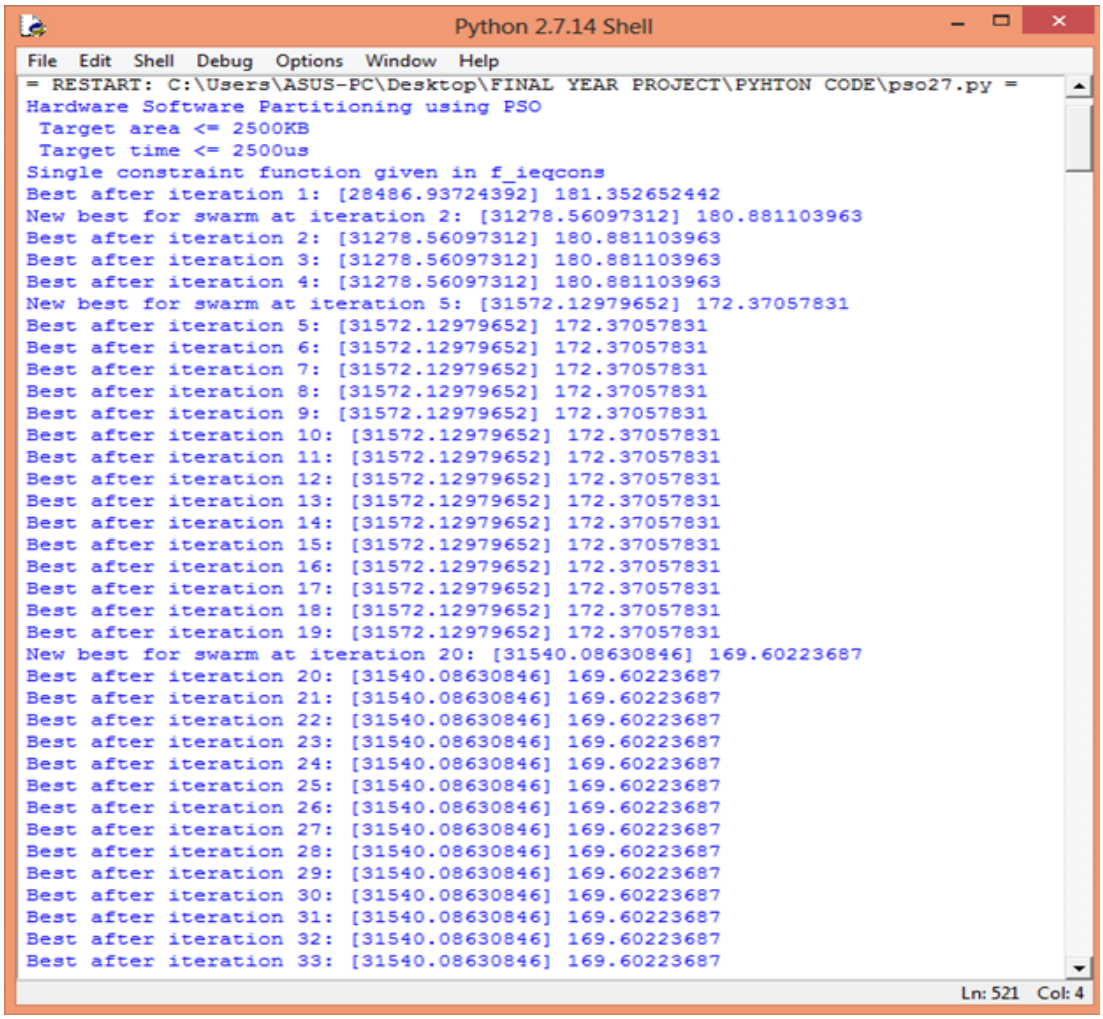
Based on the input provided in Table 3.1. The optimized solution of PSO algorithm are get and tabulated in Table 4.1.

Table 4.1 : The Optimized Solution of PSO algorithm

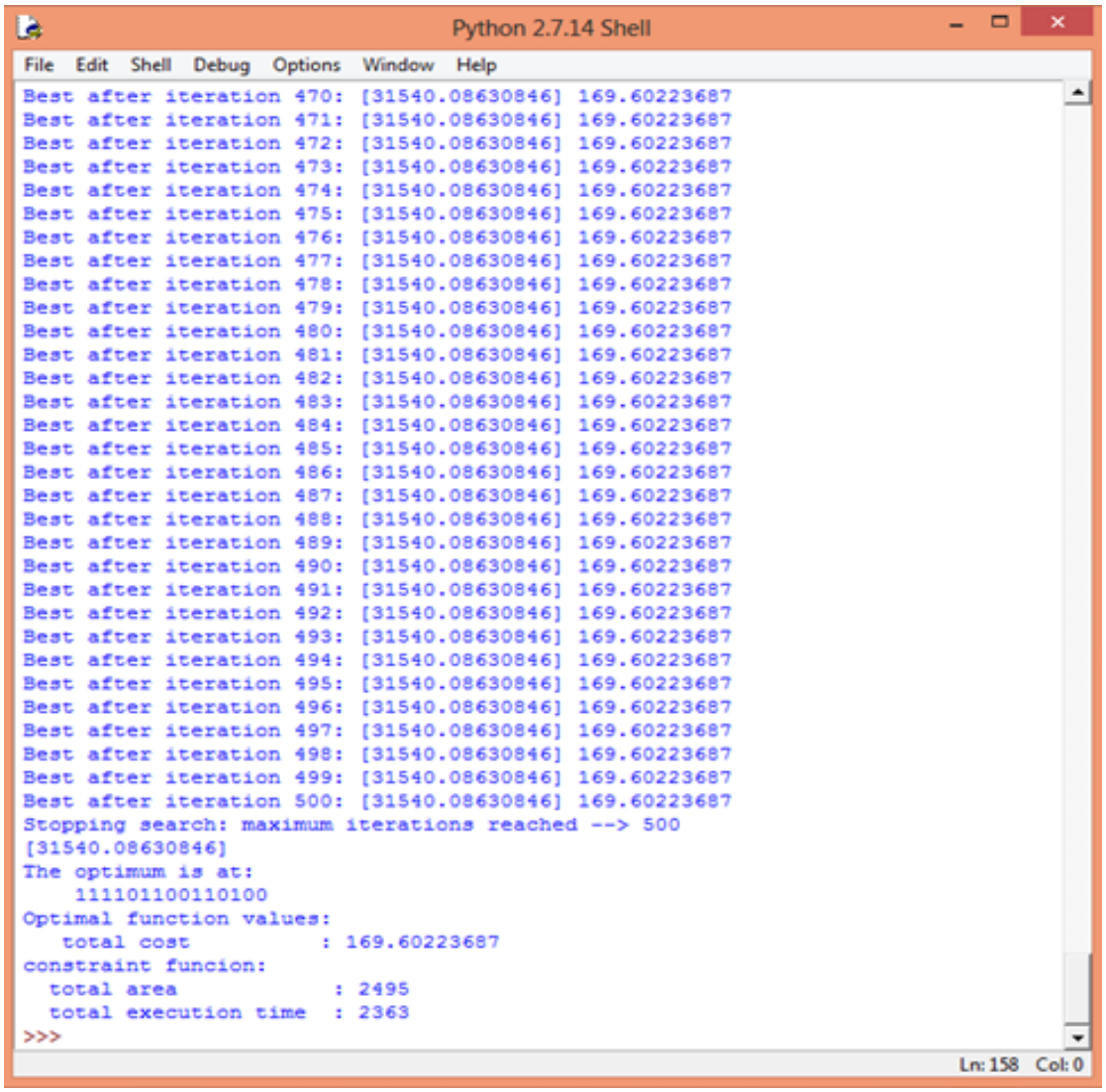
Number of itteration	**Node implementation**	**Total Area (KB)**	**Total Execution Time (µs)**	**Total Cost**
1	110111101000110	2472	2653	181.3527
2	111101000101110	2236	2269	180.8811
5	111101101010100	2277	2535	172.3706
20	111101100110100	2495	2363	169.6022

The most optimum show in Table 4.1 is at 20 itteration which is optimize the system that have six tasks should be run in software and nine tasks should be run in hardware. The total cost needed for this optimum solution is 169.6022, followed by the total area is 2495 (KB) and total time 2363 (µs). The optimize solution for 20 number of itteration are meet the target of total area and total time which is less than 2500 (KB) and 2500 (µs).

Figure 4.1 (a) and (b) is showed the output of the optimize solution of PSO algorithms for the first 33 iteration and the last 31 iteration that pop up at the Python Shell after run in the Python2.7.14.



(a)



(b)

Figure 4.1 (a) The output of PSO algorithm for the first 33 iteration,

(b) The output of PSO algorithm for the last 31 iteration

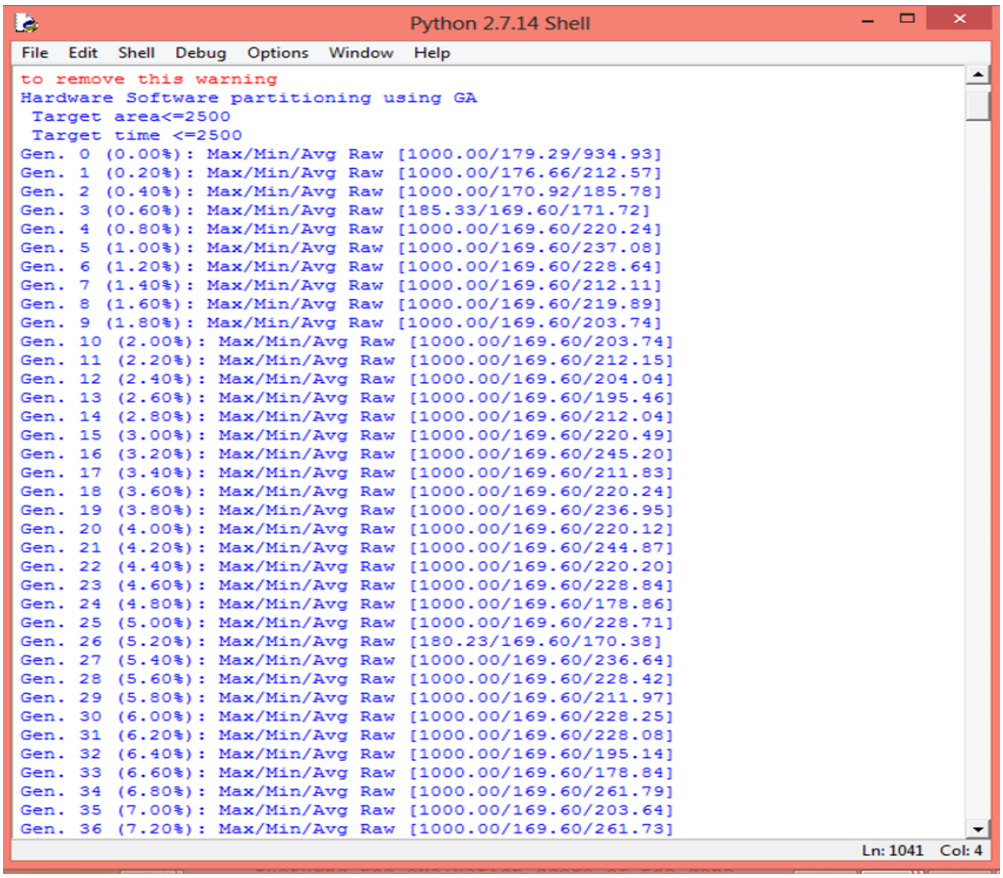
4.3 **GA algorithm**

Figure 4.2 (a) and (b) showed the output of 500 generation of GA algorithm that pop up in Python shell after run in the Python2.7.14 .

The optimized solution of GA algorithm that pop up in python shell was tabulated in Table 4.2. The optimum value of GA algorithm is the same with PSO algorithm which is 6 tasks to run in software and 9 tasks to run in hardware to optimize the solution. The total cost needed is also the same at 169.6022. The result of total area for GA algorithm is 1605(KB) and total time 2147(µs). In GA algorithm, the output also meet the target of total area and total time which is less than 2500 (KB) and 2500 (µs).

Table 4.2 : The Optimized Solution of GA algorithm

Node implementation	**Total Area (KB)**	**Total Execution Time (µs)**	**Total Cost**
111101100110100	1605	2147	169.6022



(a)