Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 8 (2017) pp. 2395-2410
© Research India Publications
http://www.ripublication.com

Designing of Optimized Combinational Circuits Using Particle Swarm Optimization Algorithm

Ila Chaudhary*, Garima Grover and Prerna Kakkar*****

[1,3]Electronics & Communication Engineering Department, Faculty of Engineering & Technology, Manav Rachna International University, Faridabad, Haryana, India.

[2]Electronics & Communication Engineering Department, Manav Rachna University, Faridabad, Haryana, India.

Abstract

Particle Swarm Optimization (PSO) is an evolutionary computation technique which is used for optimization in several applications. In this paper, PSO is used for the optimization of digital combinational circuits. The result obtained after the implementation of the optimization algorithm is shown to have minimum number of gates as compared to human design method such as K-Map conventional technique. For the purpose VHDL code has been used to implement all the requisite blocks of the algorithm.

Index Terms Digital combinational circuit, VHDL, Particle Swarm Optimization, Human methods.

1. INTRODUCTION

Designing of optimized digital circuits is targeted with smaller area of Integrated Circuits. It directly leads to the requirement of reducing the hardware components needed to achieve the functionality of the system. At the synthesis level of digital circuits, gate count of the design is the most important parameter during its design phase. If the gate count is reduced, it ultimately helps in reducing the area[4]. Certain minimization techniques have been designed such as K-MAP and Boolean Algebra etc. These techniques cannot be utilized for the designing of larger systems. Hence, an automatic technique using Particle Swarm Optimization is being proposed in this paper to design the optimized combinational circuits. A code has been developed in VHDL which has successfully implemented PSO technique to optimize digital circuit's gate count.

2. THEORY OF PSO

As the name says, Particle Swarm Optimisation simulates the behaviour of swarm of birds. It is a computational method that optimizes a problem by running several iterations until the optimal solution is achieved. This evolutionary computation technique based on population was developed by Dr. Eberhart and Dr. Kennedy in 1995 [1,3,4]. To understand the simulation, let us suppose a large group of birds are searching for food in the field. There is only one piece of food in the area. All the birds do not know where the food is but they know the distance of food from their position in each movement. So the best strategy to find the food is to follow the bird which is nearest to the food [1].

PSO learned from the scenario and used it to solve the optimization problems [7]. In PSO, each single solution is a bird in the search space termed as particles. These particles are moved around in the search space according to simple mathematical formulae over position and velocity. Each particle's movement is influenced by two best values. Firstly the pbest i.e. particle's best position and secondly the gbest i.e. global best position among all the particles achieved until the current iteration. All the particles have their own fitness value in each iteration. Fitness value has been explained in the next section.

3. FITNESS VALUE OF THE PARTICLES

Fitness value helps us to achieve the desired circuit. A desired circuit is the one whose output matches with the output of the actual circuit by human design method, for all the possible input combinations. It is calculated by comparing the output of each circuit obtained in every iteration with the outputs of the actual circuit. For example if 4 output values are matched, then the fitness achieved is 4. Desired circuit is obtained if fitness value of 8 is obtained. Hence, the maximum fitness achieved in every iteration helps in finding the optimal solution.

4. IMPLEMENTATION OF MATRIX TO DEVELOP CIRCUITS

In our optimization application, we have taken the different combination of gates as the particles. Such circuits are developed with the help of matrix.7x3 matrix is used in our optimization application. In Fig. 4.1, it can be see that inputs to each gate are obtained from the outputs of previous column. In our circuits, we have taken A, B and

C as the primary inputs from the truth table of the human design method. R0, R1 and R2 are the outputs obtained from the gates in the first column. Similarly S0, S1 and S2 are the outputs obtained from the gates in the second column. F is the final output of the circuit obtained from PSO algorithm. This output is calculated for all the possible values of A, B and C. And then compared with the corresponding outputs of digital circuit and used for calculating fitness.

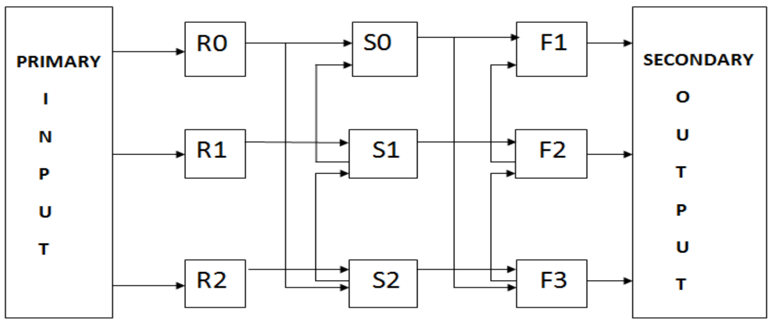


Fig.4.1 Structure of Random Matrix

In the 7 x 3 matrix, the values in the first and third column represent the inputs to the gates. For example, 1 is used to represent A, 2 is used to represent B, 4 is used to represent R0 etc. Similarly second column represents the type of gate. For example 0 is used to represent WIRE, 1 for AND etc. Other representations can be seen in Fig.4.2.

	WIRE	AND	OR	XOR	NOT
1(A)	0	1	2	3	4
2(B)					
3(C)					
4(R0)					
5(R1)					
6(R2)					
7(S0)					
8(S1)					
9(S2)					
10(F)					

Fig.4.2 Gate/Input Interconnectivity representation

$$X = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 1 & 3 \\ 0 & 0 & 0 \\ 4 & 2 & 5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 7 & 4 & 0 \end{bmatrix}$$

Fig.4.3 Initial Random Matrix

For the Full Adder's implementation, the initial circuits were presented in matrix form. The matrix in Figure 3 is one of the initial random matrix taken to represent the circuit. In this matrix (Figure 4.3), we can see that X (0, 0) = 1 indicates the input to the gate is A since 1 represents A. X (0, 2) = 3 indicates that the second input to the same gate is C since 3 represents C. X (0, 1) means that gate is AND since 1 in the second column represents AND gate. Similarly values of X(1,0) , X(1,1) and X(1,2) indicate that B and C are the inputs to the AND gate.

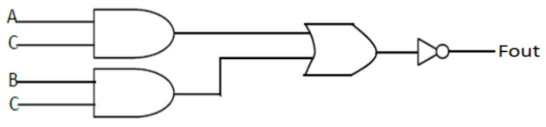


Fig.4.4 Circuit Evolution from Matrix Implementation

- To calculate the fitness, first convert the matrix above into circuit. Example conversion of X results in Fig.4.4.
- F calculated is compared with the output of the circuit obtained from human design method. The comparison can be seen in the table 1.

From the below table, we can see that after the comparison of actual output and matrix output, the Fitness calculated is 1

Table 1. Comparison of the Outputs of two Circuits

S.No.	A	B	C	Actual output of Carry	Matrix Output
1	0	0	0	0	1
2	0	0	1	0	1
3	0	1	0	0	1

4	**0**	**1**	**1**	**1**	**0**
5	**1**	**0**	**0**	**0**	**1**
6	**1**	**0**	**1**	**1**	**0**
7	**1**	**1**	**0**	**1**	**1**
8	**1**	**1**	**1**	**1**	**0**

5. USE OF PSO TO EVOLVE OPTIMIZED DIGITAL CIRCUITS

- A Population of particles i.e. in our optimization application, 5 random matrices are being used. They represent 5 initial position matrices. Similarly 5 Random velocity matrices have been taken.
- The fitness of each particle is used to obtain pbest and gbest.
- In each iteration, comparison of each particle's fitness with previous fitness is done. If the current value is better, then set the pbest equal to the current position of the particle. Else pbest remains the same as in previous iteration.
- Fitness of all particles is compared with each other. The gbest i.e global best of all the particles is updated with the pbest of the particle having highest fitness value.
- Change velocity and position of the particle according to the equations below.
- Repeat the above steps until the optimal solution is reached. Example, in 2 input digital circuits, the target is to achieve the maximum fitness value of 4.

  The equations for updating particle's velocity and position are [3,4]:

- $V_{i+1} = W^* V_i + C_1{}^* \text{rand}_1 * (\text{pbest} - X_1) + C_2 * \text{rand}_2 * (\text{gbest} - X_1)$
- $X_{i+1} = V_{i+1} + X_i$

Where V_i and X_i represent the particle's velocity and position of previous iteration; V_{i+1} and X_{i+1} represent the particle's velocity and position of current iteration. W is the weight that controls the exploration and exploitation of search space and i is the iteration number pbest and gbest have been explained before. C_1 and C_2 are acceleration constants that change the velocity of particle towards pbest and gbest.rand$_1$ and rand$_2$ are the two uniformly distributed random functions whose value lies in {0,1} [6]. The value of W has been taken as 1.50. Values of acceleration constants C_1 and C_2 have been taken as 1.5. Selection of parameters play a very important role in the optimization applications associated with PSO [2]. Parameters hold different values depending on the application for which Particle Swarm Optimization technique is being used.

6. PROPOSED FLOW MODEL

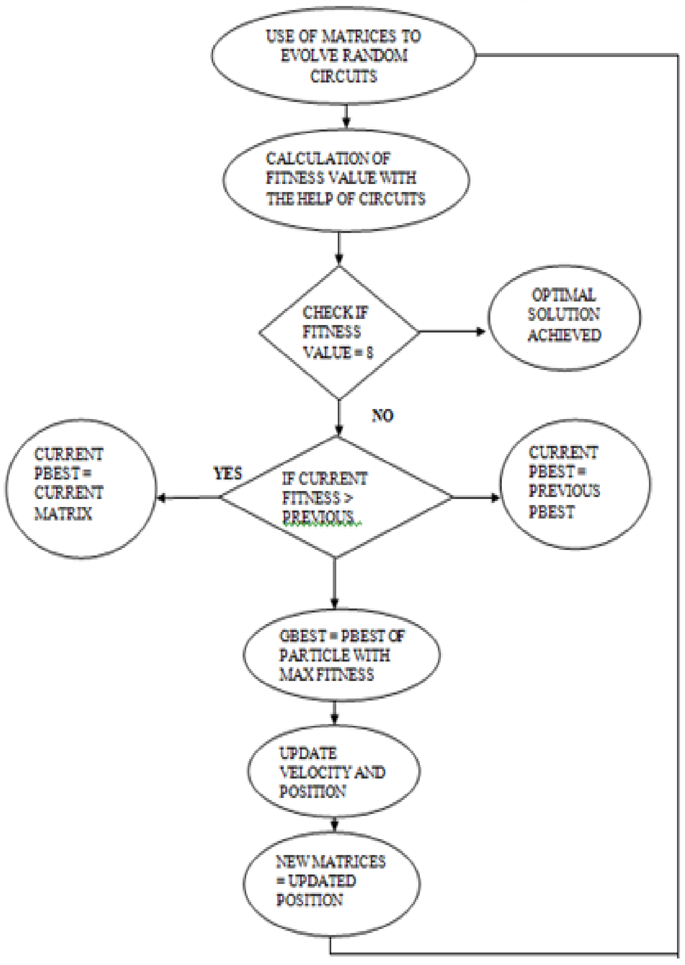
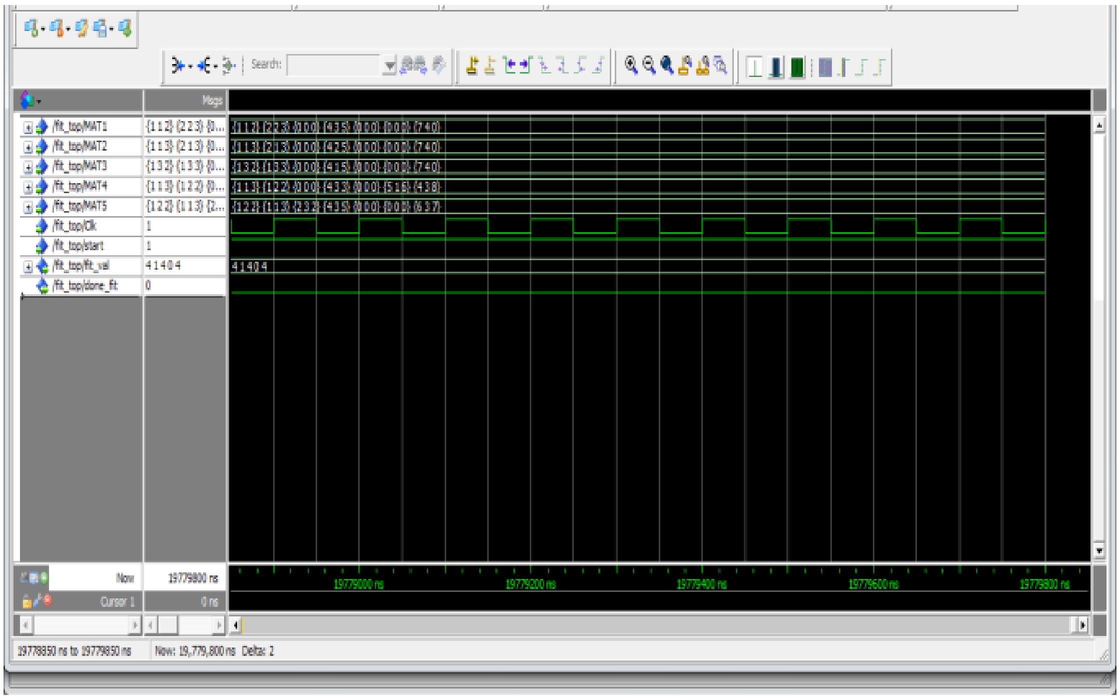

USE OF MATRICES TO EVOLVE RANDOM CIRCUITS
CALCULATION OF FITNESS VALUE WITH THE HELP OF CIRCUITS
CHECK IF FITNESS VALUE = 8
OPTIMAL SOLUTION ACHIEVED
NO
CURRENT PBEST = CURRENT MATRIX
YES
IF CURRENT FITNESS > PREVIOUS
CURRENT PBEST = PREVIOUS PBEST
GBEST = PBEST OF PARTICLE WITH MAX FITNESS
UPDATE VELOCITY AND POSITION
NEW MATRICES = UPDATED POSITION

Fig. 6.1 Flow Model to optimize digital circuits using PSO

The above flow model has been used to write the code in VHDL for the optimization of digital circuits [5]. The code has been divided into several parts, each part having its own functionality. Firstly, Fitness value is calculated for all five random circuits evolved from the random matrices taken. Secondly, once the fitness value is calculated, pbest of all the particles is calculated by comparing the previous and current fitness value. Also gbest is calculated by comparing the fitness values of all the particles in the current iteration. Thirdly random values are calculated using Linear Congruential Method. The random functions are taken in the form of 7x3 matrix. Fourthly, equations of velocity and position are used to update the velocity and position for the current iteration. All the parts of the block work in synchronization with each other, which result in successful optimization.

7. RESULTS AND CONCLUSION

FIRST ITERATION FOR CARRY



```
/fit_top/MAT1      {1 1 2} {2 2 3} {0...   {1 1 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
/fit_top/MAT2      {1 1 3} {2 1 3} {0...   {1 1 3} {2 1 3} {0 0 0} {4 2 5} {0 0 0} {0 0 0} {7 4 0}
/fit_top/MAT3      {1 3 2} {1 3 3} {0...   {1 3 2} {1 3 3} {0 0 0} {4 1 5} {0 0 0} {0 0 0} {7 4 0}
/fit_top/MAT4      {1 1 3} {1 2 2} {0...   {1 1 3} {1 2 2} {0 0 0} {4 3 3} {0 0 0} {5 1 6} {4 3 8}
/fit_top/MAT5      {1 2 2} {1 1 3} {2...   {1 2 2} {1 1 3} {2 3 2} {4 3 5} {0 0 0} {0 0 0} {6 3 7}
/fit_top/Clk       1
/fit_top/start     1
/fit_top/fit_val   4 1 4 0 4               4 1 4 0 4
/fit_top/done_fit  0

Now       19779800 ns     19779000 ns   19779200 ns   19779400 ns   19779600 ns   19779800 ns
Cursor 1  0 ns

19778850 ns to 19779850 ns     Now: 19,779,800 ns  Delta: 2
```

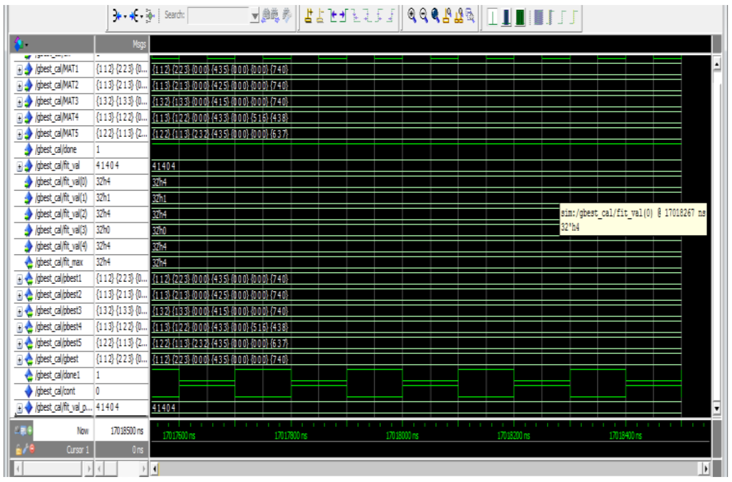
Fig.7.1. Fitness Value of Carry in First Iteration

The above waveform shows the fitness value for the five particles in the first iteration as shown in table 2.

Table 2. Fitness values of five particles

POSITION MATRIX	FITNESS VALUE
X1	4
X2	1
X3	4
X4	0
X5	4

These fitness values will be used for calculating maximum fitness, pbest of all particles and gbest in the current iteration. It can be seen in Fig.7.2.

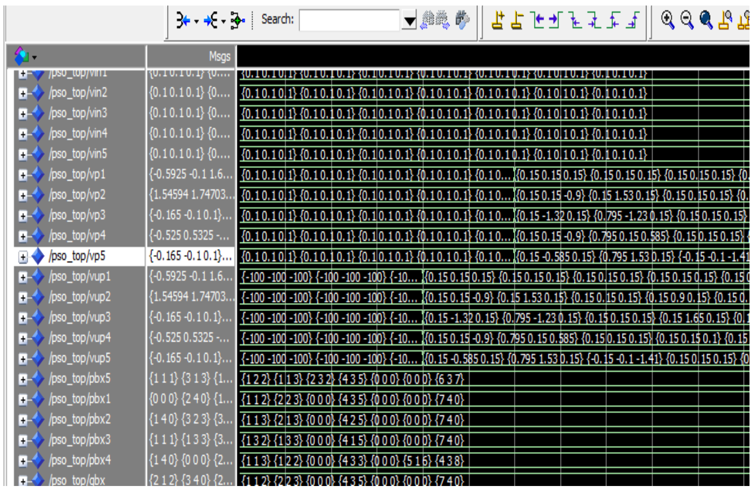


```
/gbest_cal/MAT1        {1 1 2} {2 2 3} {0...   {1 1 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
/gbest_cal/MAT2        {1 1 3} {2 1 3} {0...   {1 1 3} {2 1 3} {0 0 0} {4 2 5} {0 0 0} {0 0 0} {7 4 0}
/gbest_cal/MAT3        {1 3 2} {1 3 3} {0...   {1 3 2} {1 3 3} {0 0 0} {4 1 5} {0 0 0} {0 0 0} {7 4 0}
/gbest_cal/MAT4        {1 1 3} {1 2 2} {0...   {1 1 3} {1 2 2} {0 0 0} {4 3 3} {0 0 0} {5 1 6} {4 3 8}
/gbest_cal/MAT5        {1 2 2} {1 1 3} {2...   {1 2 2} {1 1 3} {2 3 2} {4 3 5} {0 0 0} {0 0 0} {6 3 7}
/gbest_cal/done        1
/gbest_cal/fit_val     41404                   41404
/gbest_cal/fit_val(0)  32'h4                   32'h4
/gbest_cal/fit_val(1)  32'h1                   32'h1
/gbest_cal/fit_val(2)  32'h4                   32'h4
/gbest_cal/fit_val(3)  32'h0                   32'h0
/gbest_cal/fit_val(4)  32'h4                   32'h4
/gbest_cal/fit_max     32'h4                   32'h4
/gbest_cal/pbest1      {1 1 2} {2 2 3} {0...   {1 1 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
/gbest_cal/pbest2      {1 1 3} {2 1 3} {0...   {1 1 3} {2 1 3} {0 0 0} {4 2 5} {0 0 0} {0 0 0} {7 4 0}
/gbest_cal/pbest3      {1 3 2} {1 3 3} {0...   {1 3 2} {1 3 3} {0 0 0} {4 1 5} {0 0 0} {0 0 0} {7 4 0}
/gbest_cal/pbest4      {1 1 3} {1 2 2} {0...   {1 1 3} {1 2 2} {0 0 0} {4 3 3} {0 0 0} {5 1 6} {4 3 8}
/gbest_cal/pbest5      {1 2 2} {1 1 3} {2...   {1 2 2} {1 1 3} {2 3 2} {4 3 5} {0 0 0} {0 0 0} {6 3 7}
/gbest_cal/gbest       {1 1 2} {2 2 3} {0...   {1 1 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
/gbest_cal/done1       1
/gbest_cal/cont        0
/gbest_cal/fit_val_p...  41404                 41404
Now      17018500 ns    17017600 ns   17017800 ns   17018000 ns   17018200 ns   17018400 ns
Cursor 1 0 ns
sim:/gbest_cal/fit_val(0) @ 17018267 ns
32'h4
```

Fig.7.2 Waveform of pbest and gbest in first iteration of carry

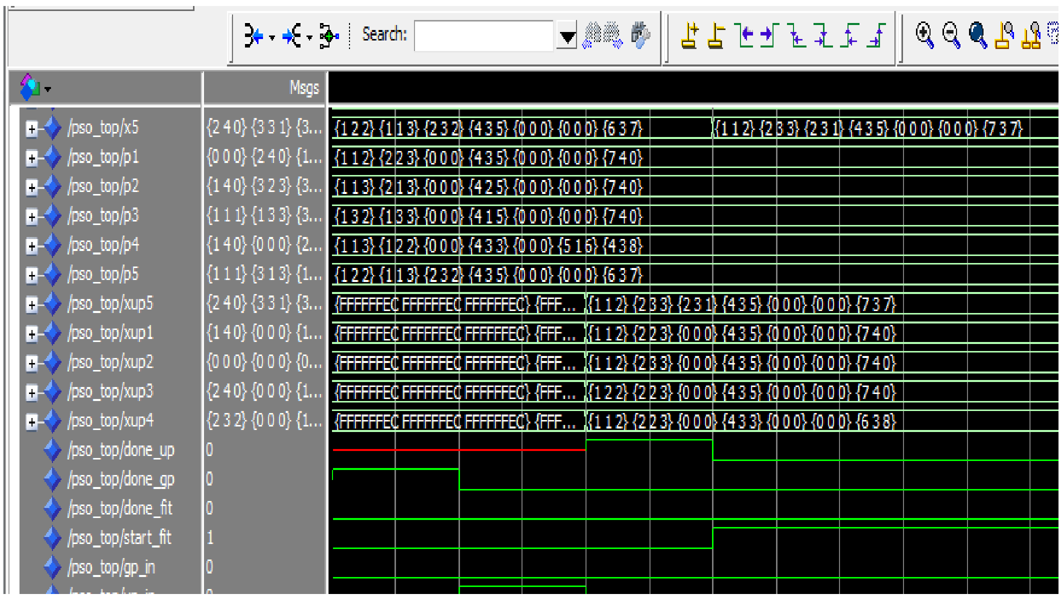
$$
\text{Gbest} = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 3 \\ 0 & 0 & 0 \\ 4 & 3 & 5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 7 & 4 & 0 \end{bmatrix}
$$

Fig.7.3. Gbest Matrix of first iteration of Carry Circuit



```
/pso_top/vin2   {0.1 0.1 0.1} {0....    {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1}
/pso_top/vin3   {0.1 0.1 0.1} {0....    {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1}
/pso_top/vin4   {0.1 0.1 0.1} {0....    {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1}
/pso_top/vin5   {0.1 0.1 0.1} {0....    {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1}
/pso_top/vp1    {-0.5925 -0.1 1.6...    {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0... {0.15 0.15 0.15} {0.15 0.15 0.15} {0.15 0.15 0.15} {0.
/pso_top/vp2    {1.54594 1.74703...     {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0... {0.15 0.15 -0.9} {0.15 1.53 0.15} {0.15 0.15 0.15} {0.
/pso_top/vp3    {-0.165 -0.1 0.1}...    {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0... {0.15 -1.32 0.15} {0.795 -1.23 0.15} {0.15 0.15 0.15}
/pso_top/vp4    {-0.525 0.5325 -...     {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0... {0.15 0.15 -0.9} {0.795 0.15 0.585} {0.15 0.15 0.15}
/pso_top/vp5    {-0.165 -0.1 0.1}...    {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0.1 0.1} {0.1 0... {0.15 -0.585 0.15} {0.795 1.53 0.15} {-0.15 -0.1 -1.41
/pso_top/vup1   {-0.5925 -0.1 1.6...    {-100 -100 -100} {-100 -100 -100} {-10... {0.15 0.15 0.15} {0.15 0.15 0.15} {0.15 0.15 0.15} {0.15 0.15 0.15} {0.15
/pso_top/vup2   {1.54594 1.74703...     {-100 -100 -100} {-100 -100 -100} {-10... {0.15 0.15 -0.9} {0.15 1.53 0.15} {0.15 0.15 0.15} {0.15 0.9 0.15} {0.15 0.
/pso_top/vup3   {-0.165 -0.1 0.1}...    {-100 -100 -100} {-100 -100 -100} {-10... {0.15 -1.32 0.15} {0.795 -1.23 0.15} {0.15 0.15 0.15} {0.15 1.65 0.15} {0.
/pso_top/vup4   {-0.525 0.5325 -...     {-100 -100 -100} {-100 -100 -100} {-10... {0.15 0.15 -0.9} {0.795 0.15 0.585} {0.15 0.15 0.15} {0.15 0.15 0.1} {0.15
/pso_top/vup5   {-0.165 -0.1 0.1}...    {-100 -100 -100} {-100 -100 -100} {-10... {0.15 -0.585 0.15} {0.795 1.53 0.15} {-0.15 -0.1 -1.41} {0.15 0.15 0.15} {0
/pso_top/pbx5   {1 1 1} {3 1 3} {1...   {1 2 2} {1 1 3} {2 3 2} {4 3 5} {0 0 0} {0 0 0} {6 3 7}
/pso_top/pbx1   {0 0 0} {2 4 0} {1...   {1 1 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
/pso_top/pbx2   {1 4 0} {3 2 3} {3...   {1 1 3} {2 1 3} {0 0 0} {4 2 5} {0 0 0} {0 0 0} {7 4 0}
/pso_top/pbx3   {1 1 1} {1 3 3} {3...   {1 3 2} {1 3 3} {0 0 0} {4 1 5} {0 0 0} {0 0 0} {7 4 0}
/pso_top/pbx4   {1 4 0} {0 0 0} {2...   {1 1 3} {1 2 2} {0 0 0} {4 3 3} {0 0 0} {5 1 6} {4 3 8}
/pso_top/gbx    {2 1 2} {3 4 0} {2...   {1 1 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
```

Fig.7.4 Updated velocity in first iteration of carry



```
/pso_top/x5        {2 4 0} {3 3 1} {3...   {1 2 2} {1 1 3} {2 3 2} {4 3 5} {0 0 0} {0 0 0} {6 3 7}   {1 1 2} {2 3 3} {2 3 1} {4 3 5} {0 0 0} {0 0 0} {7 3 7}
/pso_top/p1        {0 0 0} {2 4 0} {1...   {1 1 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
/pso_top/p2        {1 4 0} {3 2 3} {3...   {1 1 3} {2 1 3} {0 0 0} {4 2 5} {0 0 0} {0 0 0} {7 4 0}
/pso_top/p3        {1 1 1} {1 3 3} {3...   {1 3 2} {1 3 3} {0 0 0} {4 1 5} {0 0 0} {0 0 0} {7 4 0}
/pso_top/p4        {1 4 0} {0 0 0} {2...   {1 1 3} {1 2 2} {0 0 0} {4 3 3} {0 0 0} {5 1 6} {4 3 8}
/pso_top/p5        {1 1 1} {3 1 3} {1...   {1 2 2} {1 1 3} {2 3 2} {4 3 5} {0 0 0} {0 0 0} {6 3 7}
/pso_top/xup5      {2 4 0} {3 3 1} {3...   {FFFFFFEC FFFFFFEC FFFFFFEC} {FFF...   {1 1 2} {2 3 3} {2 3 1} {4 3 5} {0 0 0} {0 0 0} {7 3 7}
/pso_top/xup1      {1 4 0} {0 0 0} {1...   {FFFFFFEC FFFFFFEC FFFFFFEC} {FFF...   {1 1 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
/pso_top/xup2      {0 0 0} {0 0 0} {0...   {FFFFFFEC FFFFFFEC FFFFFFEC} {FFF...   {1 1 2} {2 3 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
/pso_top/xup3      {2 4 0} {0 0 0} {1...   {FFFFFFEC FFFFFFEC FFFFFFEC} {FFF...   {1 2 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
/pso_top/xup4      {2 3 2} {0 0 0} {1...   {FFFFFFEC FFFFFFEC FFFFFFEC} {FFF...   {1 1 2} {2 2 3} {0 0 0} {4 3 3} {0 0 0} {0 0 0} {6 3 8}
/pso_top/done_up   0
/pso_top/done_gp   0
/pso_top/done_fit  0
/pso_top/start_fit 1
/pso_top/gp_in     0
```

Fig.7.5 Updated position in first iteration of carry

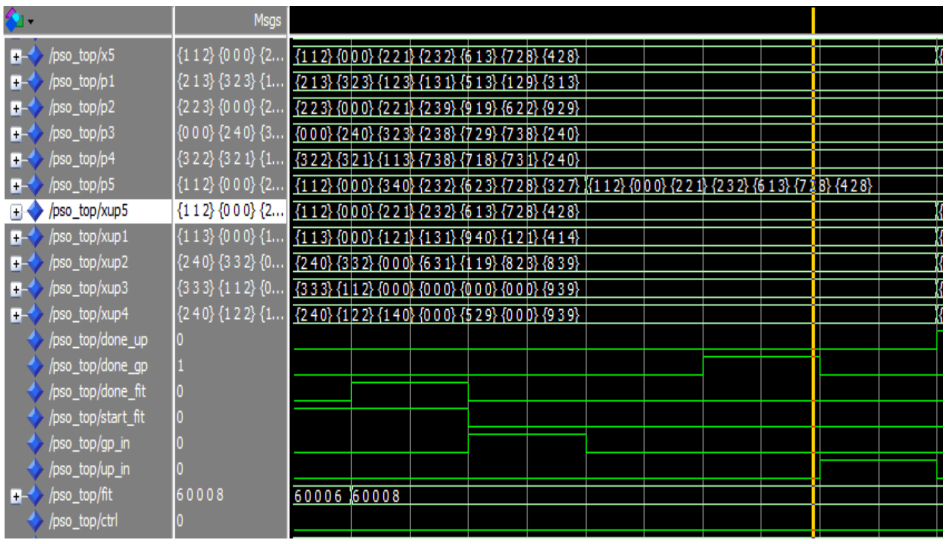
Once pbest and gbest are calculated, updated positions and velocities can be calculated. The above waveforms in Fig.7.4 and Fig.7.5 show the updated velocity and position in the first iteration. Updated position matrices of first iteration are shown in Fig.7.6

$$X1 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 3 & 3 \\ 2 & 3 & 1 \\ 4 & 3 & 5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 7 & 3 & 7 \end{bmatrix} \qquad X2 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 3 \\ 0 & 0 & 0 \\ 4 & 3 & 5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 7 & 4 & 0 \end{bmatrix} \qquad X3 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 3 & 3 \\ 0 & 0 & 0 \\ 4 & 3 & 5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 7 & 4 & 0 \end{bmatrix}$$

$$X4 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 3 \\ 0 & 0 & 0 \\ 4 & 3 & 5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 7 & 4 & 0 \end{bmatrix}$$

Fig.7.6 Updated Matrix of first iteration

LAST ITERATION FOR CARRY



```
                    Msgs
/pso_top/x5         {1 1 2} {0 0 0} {2...   {112} {000} {221} {232} {613} {728} {428}
/pso_top/p1         {2 1 3} {3 2 3} {1...   {213} {323} {123} {131} {513} {129} {313}
/pso_top/p2         {2 2 3} {0 0 0} {2...   {223} {000} {221} {239} {919} {622} {929}
/pso_top/p3         {0 0 0} {2 4 0} {3...   {000} {240} {323} {238} {729} {738} {240}
/pso_top/p4         {3 2 2} {3 2 1} {1...   {322} {321} {113} {738} {718} {731} {240}
/pso_top/p5         {1 1 2} {0 0 0} {2...   {112} {000} {340} {232} {623} {728} {327} {112} {000} {221} {232} {613} {728} {428}
/pso_top/xup5       {1 1 2} {0 0 0} {2...   {112} {000} {221} {232} {613} {728} {428}
/pso_top/xup1       {1 1 3} {0 0 0} {1...   {113} {000} {121} {131} {940} {121} {414}
/pso_top/xup2       {2 4 0} {3 3 2} {0...   {240} {332} {000} {631} {119} {823} {839}
/pso_top/xup3       {3 3 3} {1 1 2} {0...   {333} {112} {000} {000} {000} {000} {939}
/pso_top/xup4       {2 4 0} {1 2 2} {1...   {240} {122} {140} {000} {529} {000} {939}
/pso_top/done_up    0
/pso_top/done_gp    1
/pso_top/done_fit   0
/pso_top/start_fit  0
/pso_top/gp_in      0
/pso_top/up_in      0
/pso_top/fit        60008                   60006 60008
/pso_top/ctrl       0
```

Fig.7.7. Maximum Fitness of 8

Above waveform (Fig.7.6) show that fitness value of 8 is achieved, that gives its optimal solution. Final position matrices in the last iteration is shown in Fig.7.8.

$$X1 = \begin{bmatrix} 1 & 1 & 3 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \\ 1 & 3 & 1 \\ 9 & 4 & 0 \\ 1 & 2 & 1 \\ 4 & 1 & 4 \end{bmatrix} \quad X2 = \begin{bmatrix} 2 & 4 & 0 \\ 3 & 3 & 2 \\ 0 & 0 & 0 \\ 6 & 3 & 1 \\ 1 & 1 & 9 \\ 8 & 2 & 3 \\ 8 & 3 & 9 \end{bmatrix} \quad X3 = \begin{bmatrix} 3 & 3 & 3 \\ 1 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 9 & 3 & 9 \end{bmatrix}$$

$$X4 = \begin{bmatrix} 2 & 4 & 0 \\ 1 & 2 & 2 \\ 1 & 4 & 0 \\ 0 & 0 & 0 \\ 5 & 2 & 9 \\ 0 & 0 & 0 \\ 9 & 3 & 9 \end{bmatrix} \quad X5 = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 0 & 0 \\ 2 & 2 & 1 \\ 2 & 3 & 2 \\ 6 & 1 & 3 \\ 7 & 2 & 8 \\ 4 & 2 & 8 \end{bmatrix}$$

Fig, 7.8 Final position Matrices

Final Fitness values are shown in the following Table 3:

Table 3. Fitness Values of Last Iteration of Carry

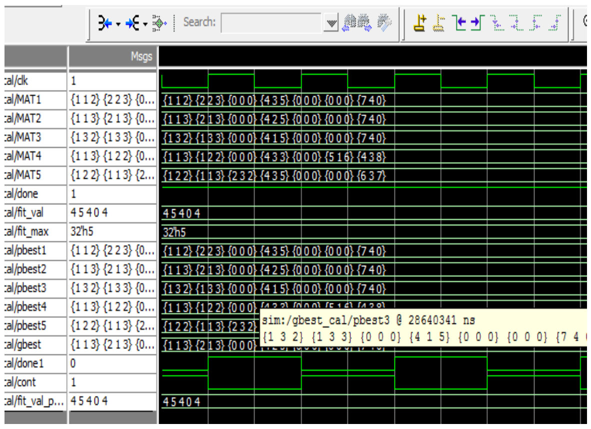
POSITION MATRICES	FITNESS VALUE
X1	6
X2	0
X3	0
X4	0
X5	8

$$\text{Gbest} = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 0 & 0 \\ 2 & 2 & 1 \\ 2 & 3 & 2 \\ 6 & 1 & 3 \\ 7 & 2 & 8 \\ 4 & 2 & 8 \end{bmatrix}$$

Fig.7.9. Final Gbest Matrix

Above is the gbest of the final iteration of carry is shown in Fig.7.9 since it led to the optimal solution i.e. fitness value of 8.

RESULTS OF SUM



```
Msgs
:al/clk        1
:al/MAT1       {1 1 2} {2 2 3} {0...   {1 1 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
:al/MAT2       {1 1 3} {2 1 3} {0...   {1 1 3} {2 1 3} {0 0 0} {4 2 5} {0 0 0} {0 0 0} {7 4 0}
:al/MAT3       {1 3 2} {1 3 3} {0...   {1 3 2} {1 3 3} {0 0 0} {4 1 5} {0 0 0} {0 0 0} {7 4 0}
:al/MAT4       {1 1 3} {1 2 2} {0...   {1 1 3} {1 2 2} {0 0 0} {4 3 3} {0 0 0} {5 1 6} {4 3 8}
:al/MAT5       {1 2 2} {1 1 3} {2...   {1 2 2} {1 1 3} {2 3 2} {4 3 5} {0 0 0} {0 0 0} {6 3 7}
:al/done       1
:al/fit_val    4 5 4 0 4               4 5 4 0 4
:al/fit_max    32'h5                   32'h5
:al/pbest1     {1 1 2} {2 2 3} {0...   {1 1 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
:al/pbest2     {1 1 3} {2 1 3} {0...   {1 1 3} {2 1 3} {0 0 0} {4 2 5} {0 0 0} {0 0 0} {7 4 0}
:al/pbest3     {1 3 2} {1 3 3} {0...   {1 3 2} {1 3 3} {0 0 0} {4 1 5} {0 0 0} {0 0 0} {7 4 0}
:al/pbest4     {1 1 3} {1 2 2} {0...   {1 1 3} {1 2 2} {0 0 0}
:al/pbest5     {1 2 2} {1 1 3} {2...   {1 2 2} {1 1 3} {2 3 2}
:al/gbest      {1 1 3} {2 1 3} {0...   {1 1 3} {2 1 3} {0 0 0}
:al/done1      0
:al/cont       1
:al/fit_val_p... 4 5 4 0 4             4 5 4 0 4

sim:/gbest_cal/pbest3 @ 28640341 ns
{1 3 2} {1 3 3} {0 0 0} {4 1 5} {0 0 0} {0 0 0} {7 4
```

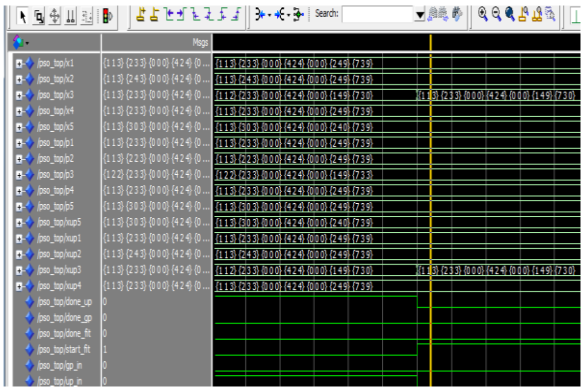
Fig.7.10. Fitness Value of First Iteration of Sum

During first iteration, fitness value is shown in the following table 4:

Table 4. Fitness Value of First Iteration of Sum

POSITION MATRIX	FITNESS VALUE
X1	4
X2	5
X3	4
X4	0
X5	4

Hence Maximum Fitness of 5 is achieved, as can be seen in the Fig.7.11. Iterations are run till the fitness value is equal to 8.



```
                 Msgs
/pso_top/x1      {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...
/pso_top/x2      {1 1 3} {2 4 3} {0 0 0} {4 2 4} {0 ...
/pso_top/x3      {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...
/pso_top/x4      {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...
/pso_top/x5      {1 1 3} {3 0 3} {0 0 0} {4 2 4} {0 ...
/pso_top/p1      {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...
/pso_top/p2      {1 1 3} {2 2 3} {0 0 0} {4 2 4} {0 ...
/pso_top/p3      {1 2 2} {2 3 3} {0 0 0} {4 2 4} {0 ...
/pso_top/p4      {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...
/pso_top/p5      {1 1 3} {3 0 3} {0 0 0} {4 2 4} {0 ...
/pso_top/xup5    {1 1 3} {3 0 3} {0 0 0} {4 2 4} {0 ...
/pso_top/xup1    {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...
/pso_top/xup2    {1 1 3} {2 4 3} {0 0 0} {4 2 4} {0 ...
/pso_top/xup3    {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...
/pso_top/xup4    {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...
/pso_top/done_up    0
/pso_top/done_gp    0
/pso_top/done_fit   0
/pso_top/start_fit  1
/pso_top/gp_in      0
/pso_top/up_in      0
```

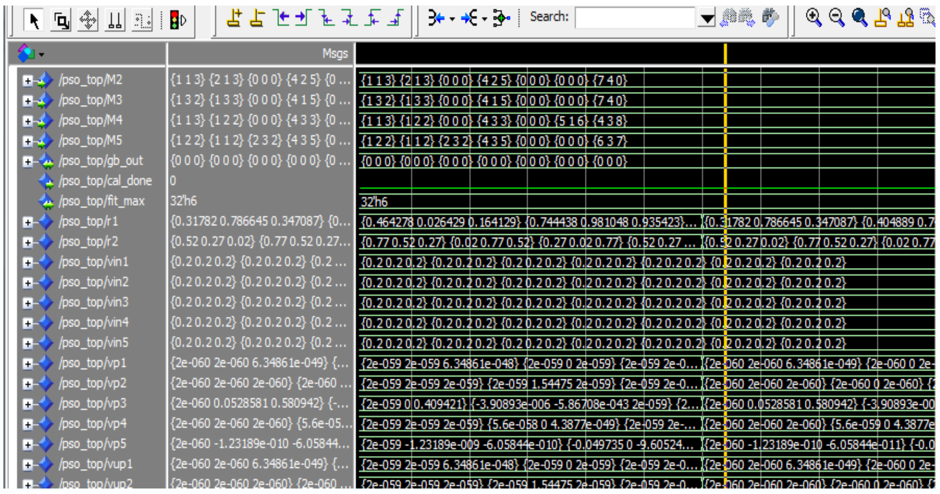
Fig.7.11. Updated positions of 30th iteration

During 30th iteration, updated position matrices are shown in Fig.7.12:

$$X1 = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 3 & 3 \\ 0 & 0 & 0 \\ 4 & 2 & 4 \\ 0 & 0 & 0 \\ 2 & 4 & 9 \\ 7 & 3 & 9 \end{bmatrix} \quad X2 = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 4 & 3 \\ 0 & 0 & 0 \\ 4 & 2 & 4 \\ 0 & 0 & 0 \\ 2 & 4 & 9 \\ 7 & 3 & 9 \end{bmatrix} \quad X3 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 3 \\ 0 & 0 & 0 \\ 4 & 2 & 4 \\ 0 & 0 & 0 \\ 7 & 4 & 9 \\ 7 & 3 & 0 \end{bmatrix}$$

$$X4 = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 3 & 3 \\ 0 & 0 & 0 \\ 4 & 2 & 4 \\ 0 & 0 & 0 \\ 2 & 4 & 0 \\ 7 & 3 & 9 \end{bmatrix} \qquad X5 = \begin{bmatrix} 1 & 1 & 3 \\ 3 & 0 & 3 \\ 0 & 0 & 0 \\ 4 & 2 & 4 \\ 0 & 0 & 0 \\ 2 & 4 & 0 \\ 7 & 3 & 9 \end{bmatrix}$$

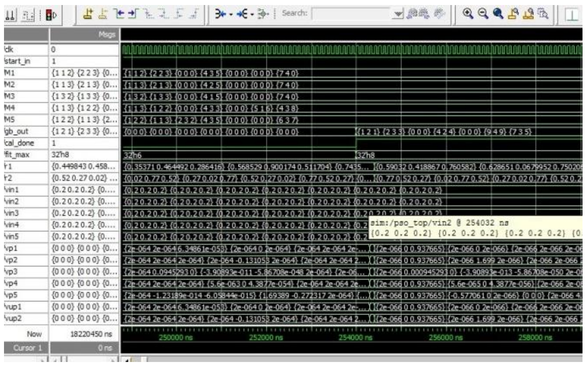
Fig. 7.12. 30th iteration updated position Matrix



```
                   Msgs
/pso_top/M2        {1 1 3} {2 1 3} {0 0 0} {4 2 5} {0 ...   {1 1 3} {2 1 3} {0 0 0} {4 2 5} {0 0 0} {0 0 0} {7 4 0}
/pso_top/M3        {1 3 2} {1 3 3} {0 0 0} {4 1 5} {0 ...   {1 3 2} {1 3 3} {0 0 0} {4 1 5} {0 0 0} {0 0 0} {7 4 0}
/pso_top/M4        {1 1 3} {1 2 2} {0 0 0} {4 3 3} {0 ...   {1 1 3} {1 2 2} {0 0 0} {4 3 3} {0 0 0} {5 1 6} {4 3 8}
/pso_top/M5        {1 2 2} {1 1 2} {2 3 2} {4 3 5} {0 ...   {1 2 2} {1 1 2} {2 3 2} {4 3 5} {0 0 0} {0 0 0} {6 3 7}
/pso_top/gb_out    {0 0 0} {0 0 0} {0 0 0} {0 0 0} {0 ...   {0 0 0} {0 0 0} {0 0 0} {0 0 0} {0 0 0} {0 0 0} {0 0 0}
/pso_top/cal_done  0
/pso_top/fit_max   32'h6                                    32'h6
/pso_top/r1        {0.31782 0.786645 0.347087} {0...        {0.464278 0.026429 0.164129} {0.744438 0.981048 0.935423}... {0.31782 0.786645 0.347087} {0.404889 0.7
/pso_top/r2        {0.52 0.27 0.02} {0.77 0.52 0.27...      {0.77 0.52 0.27} {0.02 0.77 0.52} {0.27 0.02 0.77} {0.52 0.27 ... {0.52 0.27 0.02} {0.77 0.52 0.27} {0.02 0.77
/pso_top/vin1      {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 ...     {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2}
/pso_top/vin2      {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 ...     {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2}
/pso_top/vin3      {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 ...     {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2}
/pso_top/vin4      {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 ...     {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2}
/pso_top/vin5      {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 ...     {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2}
/pso_top/vp1       {2e-060 2e-060 6.34861e-049} {...        {2e-059 2e-059 6.34861e-048} {2e-059 0 2e-059} {2e-059 2e-0... {2e-060 2e-060 6.34861e-049} {2e-060 0 2e-
/pso_top/vp2       {2e-060 2e-060 2e-060} {2e-060 ...       {2e-059 2e-059 2e-059} {2e-059 1.54475 2e-059} {2e-059 2e-0... {2e-060 2e-060 2e-060} {2e-060 0 2e-060} {
/pso_top/vp3       {2e-060 0.0528581 0.580942} {-...        {2e-059 0 0.409421} {-3.90893e-006 -5.86708e-043 2e-059} {2... {2e-060 0.0528581 0.580942} {-3.90893e-00
/pso_top/vp4       {2e-060 2e-060 2e-060} {5.6e-05...       {2e-059 2e-059 2e-059} {5.6e-058 0 4.3877e-049} {2e-059 2e-... {2e-060 2e-060 2e-060} {5.6e-059 0 4.3877e
/pso_top/vp5       {2e-060 -1.23189e-010 -6.05844...        {2e-059 -1.23189e-009 -6.05844e-010} {-0.049735 0 -9.60524... {2e-060 -1.23189e-010 -6.05844e-011} {-0.0
/pso_top/vup1      {2e-060 2e-060 6.34861e-049} {...        {2e-059 2e-059 6.34861e-048} {2e-059 0 2e-059} {2e-059 2e-0... {2e-060 2e-060 6.34861e-049} {2e-060 0 2e-
/pso_top/vup2      {2e-060 2e-060 2e-060} {2e-060 ...       {2e-059 2e-059 2e-059} {2e-059 1.54475 2e-059} {2e-059 2e-0... {2e-060 2e-060 2e-060} {2e-060 0 2e-060} {
```

Fig.7.13. Updated Velocities of 30th iteration of Sum

In 30th iteration, Maximum fitness of 6 is achieved as, can be seen in Fig.7.14.

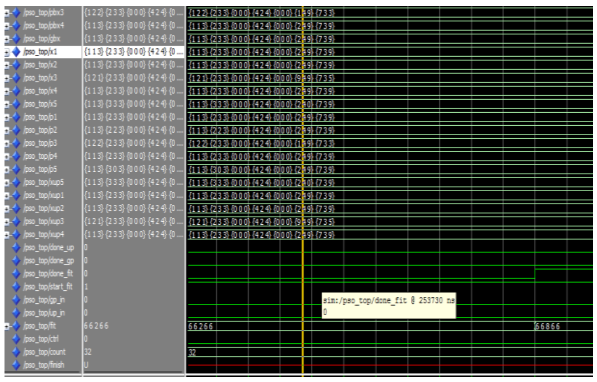


```
           Msgs
clk        0
start_in   1
M1         {1 1 2} {2 2 3} {0...     {1 1 2} {2 2 3} {0 0 0} {4 3 5} {0 0 0} {0 0 0} {7 4 0}
M2         {1 1 3} {2 1 3} {0...     {1 1 3} {2 1 3} {0 0 0} {4 2 5} {0 0 0} {0 0 0} {7 4 0}
M3         {1 3 2} {1 3 3} {0...     {1 3 2} {1 3 3} {0 0 0} {4 1 5} {0 0 0} {0 0 0} {7 4 0}
M4         {1 1 3} {1 2 2} {0...     {1 1 3} {1 2 2} {0 0 0} {4 3 3} {0 0 0} {5 1 6} {4 3 8}
M5         {1 2 2} {1 1 3} {2...     {1 2 2} {1 1 3} {2 3 2} {4 3 5} {0 0 0} {0 0 0} {6 3 7}
gb_out     {1 2 1} {2 3 3} {0...     {0 0 0} {0 0 0} {0 0 0} {0 0 0} {0 0 0} {0 0 0} {0 0 0}   {1 2 1} {2 3 3} {0 0 0} {4 2 4} {0 0 0} {9 4 9} {7 3 5}
cal_done   1
fit_max    32'h8                     32'h6   32'h8
r1         {0.449843 0.458...        {0.35371 0.464492 0.286416} {0.568529 0.900174 0.511704} {0.7435...  {0.59032 0.418867 0.760582} {0.628651 0.0679952 0.75020
r2         {0.52 0.27 0.02} ...      {0.02 0.77 0.52} {0.27 0.02 0.77} {0.52 0.27 0.02} {0.77 0.52 0.27} {0...  {0.77 0.52 0.27} {0.02 0.77 0.52} {0.27 0.02 0.77} {0.52 0.2
vin1       {0.2 0.2 0.2} {0....      {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2}
vin2       {0.2 0.2 0.2} {0....      {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2}
vin3       {0.2 0.2 0.2} {0....      {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2}
vin4       {0.2 0.2 0.2} {0....      {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2
vin5       {0.2 0.2 0.2} {0....      {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0.2
vp1        {0 0 0} {0 0 0} {0...     {2e-064 2e-064 6.34861e-053} {2e-064 0 2e-064} {2e-064 2e-064 2e-...  {2e-066 0 0.937665} {2e-066 0 2e-066} {2e-066 2e-066 2e-0
vp2        {0 0 0} {0 0 0} {0...     {2e-064 2e-064 2e-064} {2e-064 -0.131053 2e-064} {2e-064 2e-064 2...  {2e-066 0 0.937665} {2e-066 1.699 2e-066} {2e-066 2e-066
vp3        {0 0 0} {0 0 0} {0...     {2e-064 0.0945293 0} {-3.90893e-011 -5.86708e-048 2e-064} {2e-06...  {2e-066 0.000945293 0} {-3.90893e-013 -5.86708e-050 2e-0
vp4        {0 0 0} {0 0 0} {0...     {2e-064 2e-064 2e-064} {5.6e-063 0 4.3877e-054} {2e-064 2e-064 2e...  {2e-066 0 0.937665} {5.6e-065 0 4.3877e-056} {2e-066 2e-0
vp5        {0 0 0} {0 0 0} {0...     {2e-064 -1.23189e-014 -6.05844e-015} {1.69389 -0.272317 2e-064} {...}  {2e-066 0 0.937665} {-0.577061 0 2e-066} {0 0 0} {2e-066 4
vup1       {0 0 0} {0 0 0} {0...     {2e-064 2e-064 6.34861e-053} {2e-064 0 2e-064} {2e-064 2e-064 2e-...  {2e-066 0 0.937665} {2e-066 0 2e-066} {2e-066 2e-066 2e-0
vup2       {0 0 0} {0 0 0} {0...     {2e-064 2e-064 2e-064} {2e-064 -0.131053 2e-064} {2e-064 2e-064 2...  {2e-066 0 0.937665} {2e-066 1.699 2e-066} {2e-066 2e-066

Now       18220450 ns      250000 ns   252000 ns   254000 ns   256000 ns   258000 ns
Cursor 1  0 ns

sim:/pso_top/vin2 @ 254032 ns
{0.2 0.2 0.2} {0.2 0.2 0.2} {0.2 0.2 0.2} {0
```

Fig.7. 14. Maximum Fitness Value of 30th Iteration of Sum



```
/pso_top/pbx3      {1 2 2} {2 3 3} {0 0 0} {4 2 4} {0 ...   {122}{233}{000}{424}{000}{149}{733}
/pso_top/pbx4      {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...   {113}{233}{000}{424}{000}{249}{739}
/pso_top/gbx       {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...   {113}{233}{000}{424}{000}{249}{739}
/pso_top/x1        {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...   {113}{233}{000}{424}{000}{249}{739}
/pso_top/x2        {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...   {113}{233}{000}{424}{000}{249}{739}
/pso_top/x3        {1 2 1} {2 3 3} {0 0 0} {4 2 4} {0 ...   {121}{233}{000}{424}{000}{949}{735}
/pso_top/x4        {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...   {113}{233}{000}{424}{000}{249}{739}
/pso_top/x5        {1 1 3} {3 3 3} {0 0 0} {4 2 4} {0 ...   {113}{333}{000}{424}{000}{240}{739}
/pso_top/p1        {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...   {113}{233}{000}{424}{000}{249}{739}
/pso_top/p2        {1 1 3} {2 2 3} {0 0 0} {4 2 4} {0 ...   {113}{223}{000}{424}{000}{249}{739}
/pso_top/p3        {1 2 2} {2 3 3} {0 0 0} {4 2 4} {0 ...   {122}{233}{000}{424}{000}{149}{733}
/pso_top/p4        {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...   {113}{233}{000}{424}{000}{249}{739}
/pso_top/p5        {1 1 3} {3 0 3} {0 0 0} {4 2 4} {0 ...   {113}{303}{000}{424}{000}{249}{739}
/pso_top/xup5      {1 1 3} {3 3 3} {0 0 0} {4 2 4} {0 ...   {113}{333}{000}{424}{000}{240}{739}
/pso_top/xup1      {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...   {113}{233}{000}{424}{000}{249}{739}
/pso_top/xup2      {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...   {113}{233}{000}{424}{000}{249}{739}
/pso_top/xup3      {1 2 1} {2 3 3} {0 0 0} {4 2 4} {0 ...   {121}{233}{000}{424}{000}{949}{735}
/pso_top/xup4      {1 1 3} {2 3 3} {0 0 0} {4 2 4} {0 ...   {113}{233}{000}{424}{000}{249}{739}
/pso_top/done_up   0
/pso_top/done_gp   0
/pso_top/done_fit  0
/pso_top/start_fit 1
/pso_top/gp_in     0
/pso_top/up_in     0
/pso_top/fit       66266                                     66266                     66866
/pso_top/ctrl      0
/pso_top/count     32                                        32
/pso_top/finish    U

sim:/pso_top/done_fit @ 253730 ns
0
```

Fig.7.15. Maximum Fitness of 8 in Sum

In 32nd iteration, fitness value is equal to 8 (Fig.7.15). Hence optimal solution is achieved. Gbest of final iteration of sum, which led to the optimal solution of 8 is shown below.

$$\text{Gbest} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 3 \\ 0 & 0 & 0 \\ 4 & 2 & 4 \\ 0 & 0 & 0 \\ 9 & 4 & 9 \\ 7 & 3 & 5 \end{bmatrix}$$

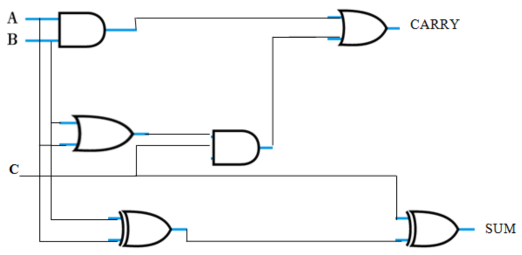


Fig.7.16. Full Adder Circuit using PSO technique

Final Gbest of sum and carry led to the evolution of above circuit in Fig.7.16. This is the optimized circuit consisting of 6 gates obtained using PSO technique.

Fig. 7.17 and Fig. 7.18 show the K-map for the full adder Carry and sum computation respectively. Fig. 7.19 shows the full adder circuit obtaiuned from K-map which is consisting of 7 gates.

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Fig.7.17. K-MAP of Carry

$$\text{Carry} = AB + BC + CA$$

A \ BC	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Fig.7.18 K-MAP of Sum

$$\text{Sum} = A\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C + ABC + \overline{A}\,B\,\overline{C}$$

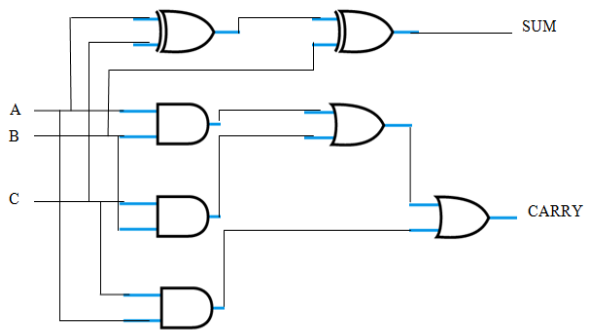


Fig.7.19. Full Adder Circuit obtained using K-MAP

Hence it can be concluded that the circuit developed using PSO has lesser number of gates as compared to the circuit developed from K-Map Technique. Hence full adder circuit has been designed and optimized using Particle Swarm Optimization. Similarly, the technique can be further generalised for optimizing various digital circuits. In a simple full adder circuit the technique has been able to produce a most optimized circuit which is helpful in reducing the gate count of such a simple circuit by 1. It can lead to the reduction of large number of gates when scaled for larger circuits.

REFERENCES

[1] James Kennedy, Russel Eberhart, “Particle Swarm Optimization”, Neural Networks, 1995, Proceedings IEEE conference vol. 4, page(s) 1942-1948.

[2] Yuhui Shi, Russell C. Eberhart “Parameter selection in particle swarm optimization”, Springer Link 1998, Evolutionary Programming VII, vol. 1447, page(s) 590-600

[3] Carlos A. Coello Coello, Arturo Hern´andez Aguirre, and Bill P. Buckles. Evolutionary Multiobjective Design of Combinational Logic Circuits. In Jason Lohn, Adrian Stoica, Didier Keymeulen, and Silvano Colombano, editors, *Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware*, pages 161–170, Los Alamitos, California, July 2000. IEEE Computer Society.

[4] Venus, G. G. and Ganesh, K. V., Evolving Digital Circuit Using Particle Swarm (2003), http://www.ieee+plore.ieee.org_12/12/2005.

[5] Feng Tian ; Sch. of Inf. Eng., Hebei Univ. of Technol., Tianjin, China; Zhenbin Gao ; YiCai Sun “VHDL Modeling of Particle Swarm Optimization Algorithm”, CISE 2009, Proceedings IEEE conference, page(s) 1-4.

[6] Magnus Erik Hvass Pedersen “Good Parameters for Particle Swarm Optimization” Hvass Laboratories Technical Report no. HL1001, 2010

[7] T. Niknam and B. Amiri. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. Applied Soft Computing, 10:183-197, 2010.