5. Shuffled Frog Leap Algorithm

The Shuffled Frog Leap algorithm is a memetic meta-heuristic that is designed to seek a global optimal solution by performing a heuristic search. It is based on the evolution of memes carried by individuals and a global exchange of information among the population. The SFL algorithm involves a population of possible solutions defined by a set of frogs that is partitioned into subsets referred to as memeplexes. Within each memeplexes the individual frogs hold ideas that can be influenced by the ideas of other frogs and evolve through a process of memetic evolution. After a number of memetic evolution steps, ideas are passed among memeplexes in a shuffling process. The local search and the shuffling continue until convergence criteria are satisfied.

An initial population of 'P' frogs are created randomly. For S-dimensional problems, each frog i is represented by S variables as $X_i = (x_{i1}, x_{i2} \ldots x_{is})$. The frogs sort in descending order according to their fitness. Then the entire population is divided into m memeplexes, each containing n frogs. Within each memeplexes, the frog with the best and the worst fitness are identified as X_b and X_w respectively. The frog with the global fitness is identified as X_g. Then the evolution process is applied to improve only the frog with the worst fitness in each cycle.

Change in frog position (D_i) = rand ( ). $(X_b - X_w)$.

New position X_w = Current position $X_w + D_i$

$$(D_{max} \geq D_i \geq D_{max})$$

where rand ( ) is a random number between 0 and 1 and D_{max} is the maximum change in frog's position. If this produces a better frog it replaces the worst frog. If no improvement becomes possible in the latter case then a new solution is randomly generated to replace the worst frog with another frog having any arbitrary fitness. The calculations will continue for specific numbers of evolutionary iterations within each memplexes

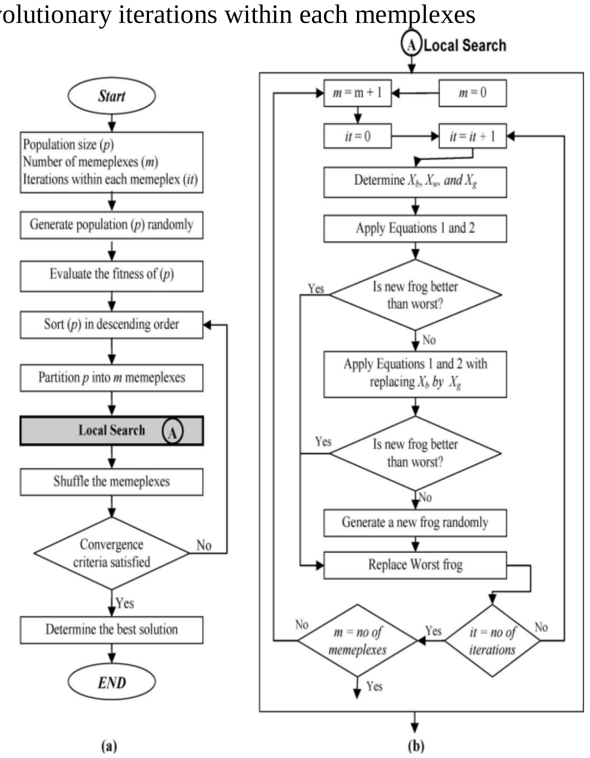

Figure 4: Flowchart of the Shuffled Frog Leap Algorithm

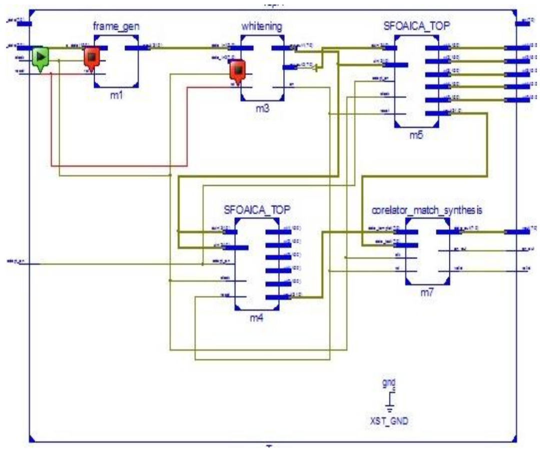

Figure 5: Top level RTL view of ICA Block

6. Results and Discussion

The concept of ICA can be used in various fields such as speech, image and Biomedical applications. By using advanced technologies such as VLSI the complexity parameters such as processing speed, power and the number of memory units can be drastically reduced. Various numbers of algorithms such as has been proposed based on ICA but by using the Frog Leap Algorithm we can optimize the complexity parameters to a greater extent. The simulations are done using Xilinx 14.7 and the tested using Spartan3A kit. The output weight vectors of the ICA and the RTL view are shown in Fig 3 and Fig 5 respectively.

7. Conclusion

ICA is a very general-purpose statistical technique in which observed random data are linearly transformed into components that are maximally independent from each other, and simultaneously have interesting distributions. ICA can be formulated as the estimation of a latent variable model. The intuitive notion of maximum nongaussianity can be used to derive different objective functions whose optimization enables the estimation of the ICA model. Alternatively, one may use more classical notions like maximum likelihood estimation or minimization of mutual information to estimate. A computationally very efficient method performing the actual estimation is given by the Frog Leap Optimization Algorithm. Applications of ICA can be found in many different areas such as audio processing, biomedical signal processing, image processing, telecommunications, and econometrics.

References

[1] H. Du, H. Qi and X. Wang, "Comparative Study of VLSI Solutions to Independent Component Analysis", IEEE Trans. Industrial Electronics, vol.54, no. 1, February, 2007.

[2] Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," NeuralComput., vol. 9, no. 7, pp. 1483–1492,Oct. 1997.

[3] Xia Li, Jianping Luo," An improved shuffled frog leaping algorithm with external optimisation for continuous optimization", 2010 Elsevier.

[4] Special issue: Swarm intelligence and its applications 2012 Published by Elsevier Inc.
[5] K. K. Shyu, M. H. Lee, Y. T. Wu and P. L. Lee, “Implementation of Pipelined FastICA on FPGA for Real-Time Blind Source Separation”, IEEE Trans. Neural Networks, vol. 19, no. 6, pp. 958-970, June,2008.
[6] Hyvarinen, “Fast and Robust Fixed-Point Algorithms for Independent Component Analysis”, IEEE Trans. Neural Networks, vol. 10, no. 3, May, 1999.
[7] S.Mishra and C.N.Bhende, “Bacterial ForagingTechnique-Based Optimized Active Power Filter for Load Compensation”, IEEE Trans. Power Del. vol.22, no.1, pp.457-465, Jan.2007.
[8] Y.Tan and J.Wang, “Nonlinear Blind SourceSeparation Using Higher Order Statistics and a Genetic Algorithm”, IEEE Trans. On Evolutionary Computation, vol.5, No.6, pp.600-611, Dec. 2001.
[9] T-W. Lee, M. Girolami, and T.J. Sejnowski,“Independent ComponentAnalysis using an Extended Infomax Algorithm for Mixed Sub-Gaussian and Super-Gaussian Sources,” Neural Computation,Vol.11,pp. 417-441, 1999.
[10] Xia Li, Jianping Luo,” An improved shuffled frog leaping algorithm with external optimisation for continuous optimization”, 2010 Elsevier.
[11] A. Papoulis. Probability, Random Variables, and Stochastic Processes. McGraw-Hill, 3rd edition,1991.
[12] A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. Neural Computation, 7:1129–1159, 1995.
[13] N. Delfosse and P. Loubaton. Adaptive blind separation of independent sources: a deflation approach. Signal Processing, 45:59–83, 1995.
[14] M.C. Jones and R. Sibson. What is projection pursuit ? J. of the Royal Statistical Society, Ser. A,150:1–36, 1987.
[15] R. Vigário, J. Särelä, and E. Oja. Independent component analysis in wave decomposition of auditoryevoked fields. In Proc. Int. Conf. on Artificial Neural Networks (ICANN’98), pages 287–292, Skövde,vSweden, 1998.