International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016 [ICSSCCET 2016]

ISBN	978-81-929866-6-1	**VOL**	02
Website	icssccet.org	**eMail**	icssccet@asdf.res.in
Received	25 – February – 2016	**Accepted**	10 - March – 2016
Article ID	ICSSCCET074	**eAID**	ICSSCCET.2016.074

An Enhanced Meta Heuristic Model for Reduction of Energy Consumption in Embedded Systems

S Lavanya[1], K S Bhuvaneshwari[2], Dr D Bhanu[3]
[1]Assistant Professor, Department of IT, [2]Assistant Professor, Department of CSE, [3]Professor, Department of IT,
Karpagam Institute of Technology, Coimbatore, TamilNadu, India,

Abstract: *In the multiprocessing embedded framework the productive utilization of the on-chip and off-chip memory code repositioning is done. For the purpose of improvement in the embedded system the SPM and cache is used for the code processing. The code layout is developed to place the code in memory for preventing the cache conflicts and misses. Even though many researchers have illustrated the use of SPM and cache to improve the efficiency, hybrid approach of combining these two methods was not done. In this study the comparison of energy consumption is done with the help of three models namely ILP model, Heuristic model and two stage meta-heuristic models. Two stages Meta heuristic model is the proposed model in which along with the SPM and Cache, code layout is developed to place the code in it. The result reveals that compared to other two models the two stage meta-heuristic model yields more efficiency and consume less energy than other two models. As much as approximately 55% of additional energy can be saved by applying both code repositioning and SPM code selection techniques using the proposed model.*

Keywords: *Optimal Code layout, embedded systems, Scratch Pad Memory, Heuristic Model, ILP.*

1. INTRODUCTION

An embedded system is one that has computer-hardware with software embedded in it. This is one of its most important components. Embedded System is pre-programmed to do a specific function while a general purpose system could be used to run any program of your own choice. Today's embedded system also support the use of multimedia applications, image reorganization, advanced signal processing etc., In the embedded system the use of on-chip and off-chip memory plays important role in both storage and code execution. Other than the hardware devices in the embedded systems, the memory device consumes more energy while processing the code. To reduce the energy consumption of the memories in embedded system many techniques have been proposed. There is always a trade-off between the energy consumption of on-chip and off-chip memories. Embedded systems typically use reduced instruction set computer (RISC) processors which have higher memory requirements than complex instruction set computer (CISC) processors due to the low code density and load-store architecture of RISC processors.

For the reduction of energy consumption the SPM and cache is used for the code processing in which caches consist of instructions and SPM consist of object code in it. The instruction caches (I-Caches) do the job of filling the gap between the processor and memory in the system. In general placement of the code in the caches is done in runtime and there will be no rearrangement of code, so the cache is also called as transparent compilers. As the result, the code with the temporal proximity can produce the cache misses and conflict misses, which is also called as cache thrashing. The code placement in SPM is done by means of either user or by the compiler by means of algorithm given by user. Therefore the reduction of energy consumption by on-chip memories can be done by using of SPM along with code repositioning in both SPM, cacheable and non-cacheable regions. The basic idea is to improve the efficiency without increasing the cache size is the optimal code technique, so that the cache misses and conflict misses is reduced.

Cite this article as: S Lavanya, K S Bhuvaneshwari, Dr D Bhanu. "An Enhanced Meta Heuristic Model for Reduction of Energy Consumption in Embedded Systems". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 388-394. Print.

Embedded processor such as Cyrix 6x86, PS2 Emotion Engine, ARM, and CELL uses on-chip scratch pad memory. All the running software of the embedded systems is already known to it, and so it has the pre determined instruction access in it. Compilers and programmers can move frequent accesses code or data blocks to the SPM to further improve the performance and energy efficiency of embedded systems. This paper proposes a metaheuristic solution based on partial enumeration technique and Tabu search for the code placement in the memory.

2. Related Work

Many researchers have encountered many memory related issues in embedded system and handles via many techniques in recent years. Huang et al, [1] illustrated the result that revels with allocating a dedicated portion of the on- chip SPM is always better than allocating the code to cache only configuration. This approach compares the three scenarios of cache only configuration, SPM only configuration and combining the SPM and Cache. The limitation of this paper are, It did not delve further into design space exploration issue and did not optimize SRAM partitions and code object sizes for a specific target application. Panda et al. [7] proposed the advantage of using the SPM and cache for code processing for the improvement of efficiency in embedded system. This approach uses the value density approximation algorithm to place the code in SPM and Cache. The limitations of this work are data size is bigger than the 1 K SRAM, which cause the compulsory cache misses leads to slight degradation of performance. Baiocchi and Childers [8] illustrated some policies for storing and replacing the code objects in SPM by means of instruction caches. In this paper the comparison of DBT model and HCC model is done to prove the performance. The limitation of this work is that even though they have combined the use of Cache and SPM, they did not deal about the placement of code in it.

Ishitobi et al. [9] demonstrated the benefit of embedded system by properly reordering the code in the cache, so that the off-chip memory access can be reduced. It also uses the ILP model for the development of code layout for SPM, Cacheable and non-Cacheable regions in memory hierarchy. Even though they have mentioned the benefits of developing the code layout, the remaining code in the main memory are placed in continues manner which are allocated by the compiler. Thus this leads to the constraint as the processor cannot find that two code objects compete for same memory block in system. Thus, the general approach is to allow gaps in between the program objects in the main memory which can further ease cache conflicts.

Verma et al. [14] proposed the CASA algorithm to minimize the conflicts occurring in memory processing and to locate the memory objects in SPM based on the traces got from the instruction tracer. However, in this method the ILP model is used which is difficult to solve and sometime it will become a NP hard problem. Tanja Van et al [16] illustrated the repositioning of the code according to the past histories available. This is done by vector/matrix abstraction technique to find the frequently accessed code and placing in the cache for the improvement in the performance of the system. The limitation of this work is finding the history for the long code is difficult and time consuming process.

M Kandemir [17] proposed the technique that the SPM is managed by user or the compiler by means of algorithm by user. In this the optimization suite is developed to place the code in SPM and manage it. The limitation of this approach is that the code replacement can be done only to SPM and not to the Caches in the embedded system. Hence, there are two major differences between previous studies and the proposed method: 1) this study do the code processing by means of three different models and energy is calculated accordingly and 2) this study compares those three models by means of specified benchmarks and revels the result that the proposed method uses the less energy compared to other two models.

3. Methodology

This study uses the Harvard architecture shown in Fig 1 for the experimental purpose in which the placement of SRAM and Cache resides in the same level of memory hierarchy. The experiment deals about the placement of code objects in the SPM, cacheable and non cacheable regions of the memory based on the optimal code layout design.

Cite this article as: S Lavanya, K S Bhuvaneshwari, Dr D Bhanu. "An Enhanced Meta Heuristic Model for Reduction of Energy Consumption in Embedded Systems". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 388-394. Print.

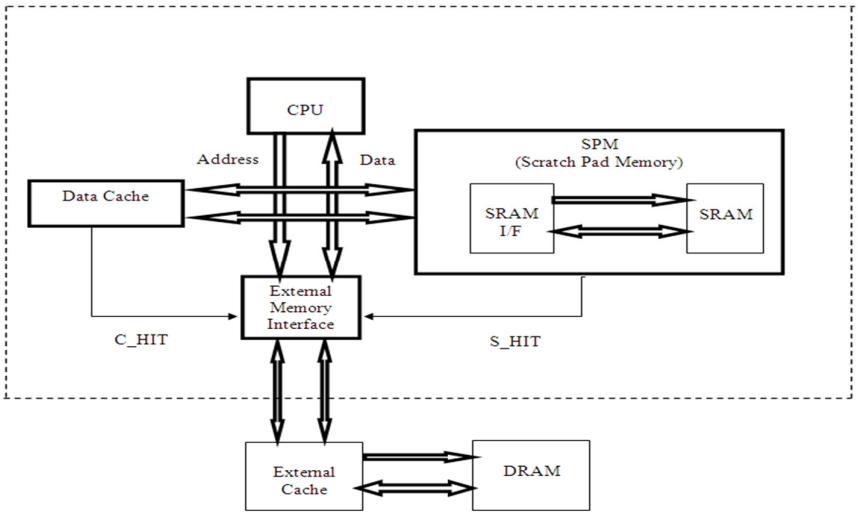

Fig-1 Architecture Diagram

The primary advantage of the usage of Harvard architecture is the simultaneous access of more than one memory for the code processing. Generally when the code objects is loaded into the system memory; it is placed in the main memory in contiguous manner. It is also noted that the size of SPM is always larger than the L1 caches and the main memory.
The proposed architectural design consists of on chip SPM and cache along with the main memory. The code placement layout is formulated so that the placement of code according to that layout will reduce the consumption of energy and improve the performance.

3.1. Scratch Pad Memory

SPM is similar to that of cache used for the fast and rapid recovery of code objects for processing of data. Software managed memory such as scratch pad memories are important particularly in case of image and video processing application where heavy use of multi dimensional array is implemented. As these applications need large area for its storage, scratch pad architectures are used, which results in large saving in energy and improvement in the efficiency of the system. Scratch pad memory based system uses, the programmer or the compiler who are responsible for scheduling the data transfer between the SPM and the off chip memory. Effective scheduling of memory will improve both the efficiency and performance of the application.

Scratchpad memories, consisting of data memory and address decoding circuitry it require less on chip area. However unlike caches, scratchpads require complex program analysis and explicit support from the compiler. The difference between the cache and the SPM is it does not have the copy of data available in the main memory. Scratchpad is used for simplifying the logic of caching and does the work without the contention of the main memory. Generally the energy consumption of the SPM code processing is always less than the cache code processing. The fig 2 represents the presence of SPM and Cache in system architecture.

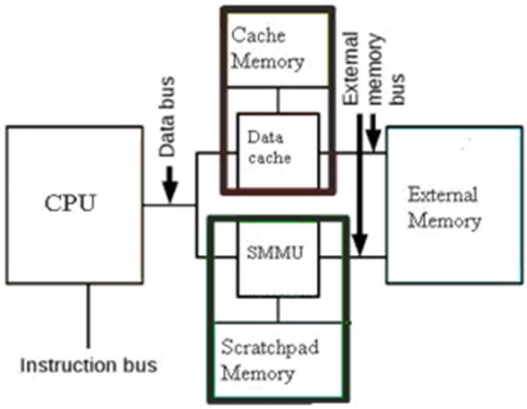

Fig-2 Scratch Pad Memory and Cache memory

Cite this article as: S Lavanya, K S Bhuvaneshwari, Dr D Bhanu. "An Enhanced Meta Heuristic Model for Reduction of Energy Consumption in Embedded Systems". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 388-394. Print.

3.2. Cache Memory

Caches usually reside in the system memory and used for the storing the code objects for frequent access of processing for the decrease in latency. Caches are mainly used to avoid the spatial and temporal locality of memory access. If one ignores the SPM then cache is the name given to the highest level of memory hierarchy once the address leaves the processor. The word cache is generally used where buffering or reusing of data is employed such as file cache, name cache etc. Cache is used to reduce the average time to access memory. Three different types of caches used are

- Instruction cache-It is mainly used to speed up the instruction fetch.
- Data fetch- It is mainly used to speed up the LOAD and STORE operation.
- Translation Look Up Header-Used to speed up the virtual to physical address translation.

3.3. ILP Model

ILP model is one of the methods to process the code objects in the embedded processor. ILP model is already proposed by some researchers but the methodology of combining the SPM with the Cache and formulating the optimal code layout for these memory hierarchies is executed by this model. ILP model function is to linearize the constraints according to the functions and equations. This ILP model is used as one of the optimizer in the overall system architecture to attain the expected result. ILP model includes the following constraints:

- Early start.
- SPM capacity.
- Non-overlapping in main memory.
- Spatial conflict.
- Temporal conflict misses.

3.4. Heuristic Model

The heuristic model is the method for processing the code objects in the embedded system. This heuristic model is used as the memory hierarchy simulator to place the code in SPM, cacheable and non-cacheable regions of embedded system. The heuristic model consist of a main routine and three sub routines in it and they are

- Explore Scratchpad region
- Swap location of memory objects
- Explore non-cacheable region

In main function of this routine is to calculate the total execution for the object code. The main routine calls the subroutine and search the optimal location of each memory object. Explore scratchpad region does the function of checking whether or not the memory object can be placed in SPM. Swap location of memory object does the function of the location of the code object is exchanged among the SPM, and Cache. Explore non-cacheable regions tentatively place the code in the main memory.

3.5. Two Stage Meta-Heuristic Model

The two stage meta-heuristic model is the method for processing the code objects in the embedded system. This model is used as the memory hierarchy simulator to place the code in memory of the embedded system. The two stage meta-heuristic model consist of two methods in it they are,

- Partial enumeration Technique.
- Tabu Search

In the above mentioned heuristic the partial enumeration technique is used to select the code to be placed in the SPM. The code is arranged in the non increasing order of access density and by means of partial enumeration technique it is divided into hotter and colder subsets. The hotter subset is placed in SPM and the remaining object code is placed in the main memory.
The Tabu search is similar to the travelling salesman problem, in which the optimal path is found out to reduce the travelling cost and increase the efficiency. Likewise tabu search will find out the optimal way to the placing of the object code in SPM, Cacheable region and Non-Cacheable region.

Cite this article as: S Lavanya, K S Bhuvaneshwari, Dr D Bhanu. "An Enhanced Meta Heuristic Model for Reduction of Energy Consumption in Embedded Systems". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 388-394. Print.

3.6. Objective Function

The goal of this study is to compare the consumption of energy between three memory hierarchy simulators namely,

- ILP model
- Heuristic model
- Meta heuristic model.

In this the comparison is made by means of usage of benchmarks available. The consumption of energy of the models is calculated by means of three parameters they are,

EC_Hit	Energy consumed when the code is available in cache
EC_SPM	Energy consumed when the code is available in SPM
EC_Miss	Energy consumed when the code is not available in both SPM and Cache and present in main memory

Table-1 Energy calculation Parameters

In the above three parameters EC-Hit and EC_SPM consumes less energy as the code will be available in the Cache and SPM correspondingly. The energy parameter EC_Miss will be comparatively large as the code will not be available in the cacheable region, and the code has to searched first in SPM and then cache and then in main memory. The total energy consumption can be calculated by means of instruction fetches rather than the combining the power over time. Therefore the instruction fetch for processing the code has to be minimized. In the third case only the misses occur which will consume more energy than other two cases. The two types of misses available they are,

1. Cold Misses: Also called as compulsory misses and it will occur at the first time of processing of the code. As in the first time no code will be placed in the cache and SPM. But using the proposed model the cold miss also can be avoided as before staring the processing itself the code is placed in the SPM.

2. Conflict Misses: conflict misses generally occur due to spatial conflict and the temporal conflict. These misses can also be avoided by means of proposed model of two stage Meta heuristic model.

4. Proposed System

The Fig-3 illustrates the experimental flow of the proposed system. To get the real benefit of the code it includes all the library files and initializations in it. Even though set associativity mapping can be applicable to the proposed system we use the direct mapped associative system for the execution. In the Fig-3 the source files are the user input files and the optimizer is the described three models namely ILP, heuristic and Meta heuristic model. The work of the optimizer is to formulate the optimal code layout for the SPM, cacheable and non-cacheable regions.

The instruction trace can be got by giving the input as user input data and program codes to the ARMulator [4] which will acts as the simulator and find the instruction traces of the program code. This instruction trace should be fed to both optimizer and memory hierarchy simulator for the further processing of the code. The optimizer will get the instruction trace, executable code and the SRAM and Cache budget and develop the link script which consists of details about the placement of code in SPM, Cacheable and non-Cacheable regions of the memory hierarchy. According to the link script the compiler will recompile the code and place in those memory areas. After the processing of the code using the optimizer models the energy consumption should be calculated. The CACTI tool can be used for the calculation of the energy consumption. Then after calculating the energy the energy parameters of those three models are compared by means of the bench marks available like Qsort, Basicmath, FFT, Dyrstone etc..,

Cite this article as: S Lavanya, K S Bhuvaneshwari, Dr D Bhanu. "An Enhanced Meta Heuristic Model for Reduction of Energy Consumption in Embedded Systems". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 388-394. Print.

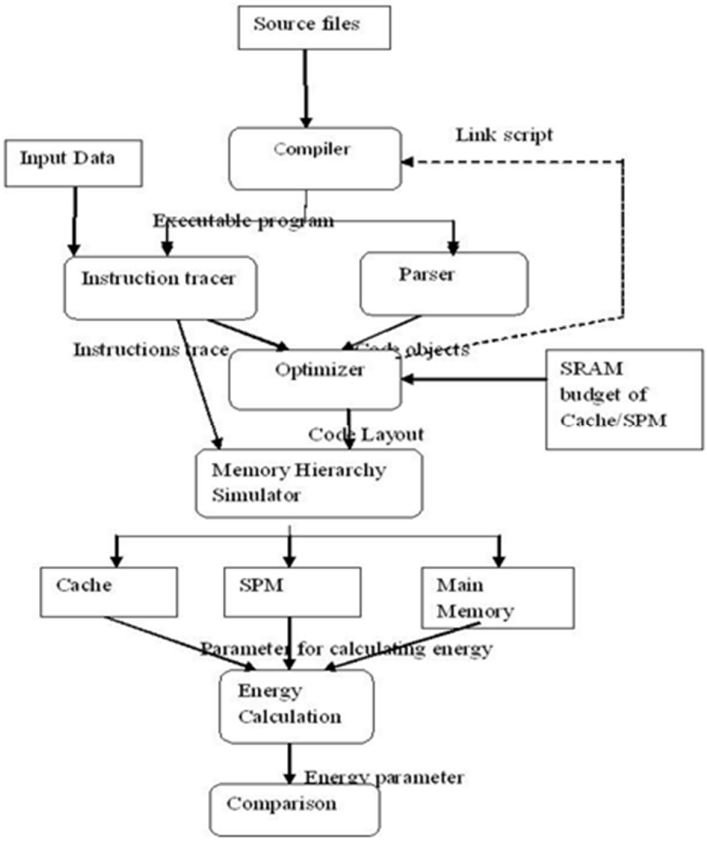

Fig-3 Experimental flow

5. Experimental Results

This section evaluates the effectiveness of the combined cache, SPM and main memory for the instruction fetch along with the optimal code repositioning. This section compares the energy consumption by instruction fetches by means of the three methods of ILP, Heuristic and two stage Meta heuristic algorithms. The experimental result of first module of ILP is shown in fig-4 in which the energy consumed by SPM, cache and main memory is shown.

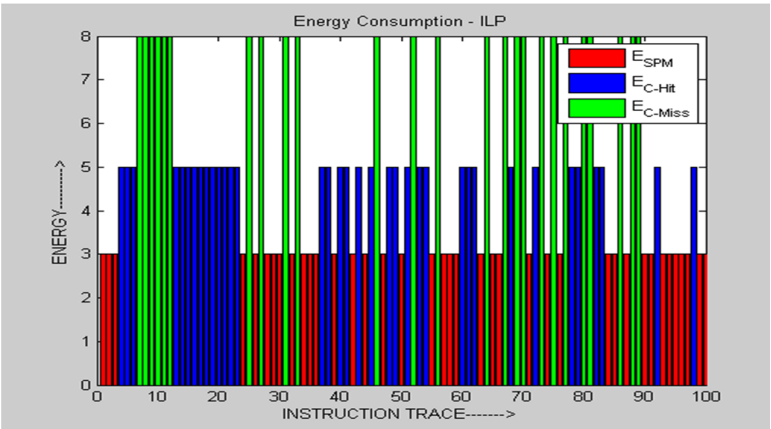

Fig-4 Energy consumption by Cache SPM and Main memory in case of ILP model

The experimental result of next module heuristic module is shown in fig-5 in which the energy consumed by SPM, cache and main memory is shown.

Cite this article as: S Lavanya, K S Bhuvaneshwari, Dr D Bhanu. "An Enhanced Meta Heuristic Model for Reduction of Energy Consumption in Embedded Systems". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 388-394. Print.

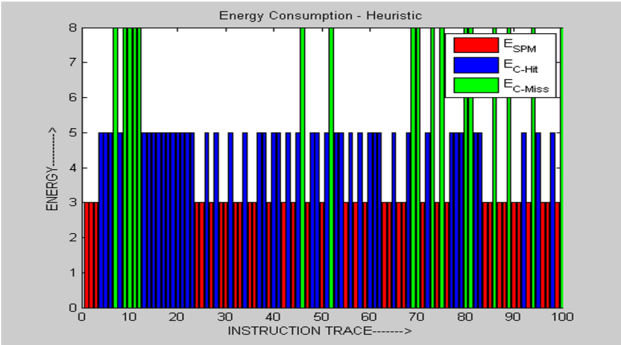

Fig-5: Energy consumption by Cache SPM and Main memory in heuristic model.

6. Conclusion

This paper presents the model of optimizer which works on two stages Meta heuristic model in which the combining of SPM, Cache and developing the optimal code layout for the placing of the code in these memories is done. The proposed models address layout problems under different models as optimizer for formulating the code layout for memory hierarchies. The result reveals that the proposed two stage Meta heuristic algorithm yields the best performance compared to the other two models and consumption of energy is less in this. The benefit of energy consumption is got by repositioning of the code among the SPM, Cacheable and non-Cacheable regions of the embedded system. The proposed system also reduces the energy consumption by handling and resolving the cache misses and conflict misses.

References

1. Minimizing Energy Consumption of Embedded Systems via Optimal Code Layout by Chen-Wei Huang and Shiao-Li tsao IEEE transactions on computers, vol. 61, no. 8, august 2012
2. An Optimization Technique to Improve Power Consumption of Embedded System by B Anuradha, Sandhya Nair LJ, Dr C Vivekanandan. International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 10, April 2013
3. Advanced Memory Optimization Techniques for Low-Power Embedded Processors by Manish Verma .Altera European Technology Center, High Wycombe, UK and Peter Marwedel University of Dortmund, Germany.
4. ARM, http://www.arm.com/products/processors/classic/arm11/ index.php, 2011.
5. IBM, http://www.research.ibm.com/cell/, 2011.
6. Freescale, http://www.freescale.com/webapp/sps/site/homepage.jsp
7. P.R. Panda, N.D. Dutt, and A. Nicolau, "On-Chip vs. Off-Chip Memory: The Data Partitioning Problem in Embedded Processor Based Systems," ACM Trans. Design Automation of Electronic Systems, vol. 5, no. 3, pp. 682-704, 2000.
8. Hyungmin Cho Bernhard Egger Jaejin Lee Heonshik Shin "Dynamic Data Scratchpad Memory Management for a Memory Subsystem with an MMU"pp 541-547.
9. Y. Ishitobi, T. Ishihara and H. Yasuura, "Code and Data Placement for Embedded Processors with Scratchpad and Cache Memories," J. Signal Processing Systems, vol. 60, pp. 221-224, Aug.2010.
10. S Wilton and Norm Jouppi: Cacti: An enhanced access and cycle time model, IEEE Joural of Solid State Circuit, May 1996.
11. R. Banakar, S. Steinke, B.S. Lee, M. Balakrishnan, and P.Marwedel, "Scratchpad Memory: A Design Alternative for Cache On-Chip Memory in Embedded Systems," Proc. 10th Int'l Symp. Hardware/Software Codesign (CODES '02), pp. 73-78, 2002.
12. J.L. Hennessy and D.A. Patterson, Computer Architecture: A Quantitative Approach, fourth ed. Morgan Kaufmann, 2006.
13. M kandemir, associate member, IEEE, J.Ramanujam, Member IEEE, Mary Jane Irwin ,Fellow ,IEEE "A Complier Based Approach for Dynamically Managing Scratch Pad Memories in Embedded System".
14. M. Verma, L. Wehmeyer, and P. Marwedel, "Cache-Aware Scratchpad Allocation Algorithms for Energy- Constrained Embedded Systems," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 10, pp. 2035-2051, Oct. 2006.
15. Mahumut Kandemir, Associate Member ,IEEE,J Ramanujam, Member, IEEE,Mary Jane Irwin, Fellow, IEEE, N Vijaykrishna, Associate Member, IEEE, Ismail Kadayif, and Amisha Parikh" Compiler directed Scratch pad Memory Hierarchy Design and Management" .
16. Tanja Van Achteren, Geert Deconinck, K.U.Leuven ESAT/ACCA, Belgium and F Catthoor, Rudy Lauwereins "Data Reuse Exploration Technique for Loop dominated Application".
17. Preethi ranjan panda and Nikil D Dutt "On chip Vs Off chip Memory: Data Partitioning Problem in Embedded Processor-Based Systems.
18. D. Keitel-Schulz and N. Wehn, "Embedded DRAM Development: Technology, Physical Design, and Application Issues," IEEE Design and Test of Computers, vol. 18, no. 3, pp. 7-15, May 2001.

Cite this article as: S Lavanya, K S Bhuvaneshwari, Dr D Bhanu. "An Enhanced Meta Heuristic Model for Reduction of Energy Consumption in Embedded Systems". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 388-394. Print.