



Adaptable Task Scheduling Algorithm : A Review

Ashish B. Bhopale
Research Scholar
G. H. Rasoni University, Amravati
Anjangaon Bari Road, Amravati, India
profashishbhopale@gmail.com

Dr. Archana O. Vyas
*Dept. of Electronics and Telecommunication
Engineering*
G. H. Rasoni University, Amravati
Anjangaon Bari Road, Amravati, India
archana.vyas@ghru.edu.in

Abstract—

Reconfigurable architecture is the need of the time. It is the terminology which indicates existing hardware resources performing multiple tasks. The task which are to be executed in subsequent time slots are need to be arranged critically and process is diverted accordingly. This process is called as task scheduling. At hardware level this can be achieved through high computing Field Programmable Gate Arrays or by using ASIC architectures. Through the proposed research work we are about to disclose an innovative scheduling algorithm for task scheduling in reconfigurable architectures. Many researchers have developed various task scheduling algorithms, in the subsequent section of the paper we are about to study the diverse research work reported so far and identify the research gap as well.

2492

Keywords—ASIC, Cadence, FPGA, Task Scheduling, Xilinx Vivado

DOI Number: 10.14704/nq.2022.20.8.NQ22271

NeuroQuantology 2022; 20(8): 2492-2503

I. INTRODUCTION

In a case study of the reconfigurable architecture, the internal circuit of the reconfigurable architecture remains ready to change the netlist of the active components to execute the upcoming process smoothly. In traditional devices, the internal architecture and the circuit components remains fixed and hence their applications are restricted. To realize such reconfigurable architecture, the system is designed using General Purpose Processor (GPP) along with the reconfigurable architecture like FPGAs or CPLDs. Here GPP explores the instructions in the queue and compute the additional hardware required for execution of the instructions. Accordingly, netlist is prepared and the reconfigurable devices are configured to meet the need of the new hardware. In this case, it is possible that, while configuring the new hardware architecture, the previous process is under

execution and need more cycles for execution. In such situation, the processor may get engage with one or more netlists. Here, need of the smart scheduling algorithm arises to handle the situation. We are on a track of designing a novel scheduling algorithm for dynamically handling the multiple tasks and processes. Before doing this, a detailed literature survey is carried out as disclosed in the subsequent part of the paper.

II. PREVIOUSLY CITED TECHNOLOGY

Sudipa Mandal et. al. studied that higher chip density and tight optimization with highest possible operating speed made the integrated circuit to utilize more and more power. Optimization of power utilization hence becomes the primary job of the chip designers. Based on architecture and resource utilization, different segments of the chip create different thermal conditions, which in also affects the overall performance and reliability of the architecture.



To address this situation, authors [1] have designed an intelligent scheduling algorithm which helps to keep the thermal conditions within permissible conditions. To attain this, Dynamic Voltage and Frequency Scaling environment is targeted. Authors further use multi objective optimization technique to limit the power dissipation. They have also used Reinforcement Learning to create scheduling points based on reward function and energy optimization process.

Authors [2] have observed that, the OpenVX platform is the benchmark tool for development of computer vision-based applications, however, its algorithms do not provide proper mapping and scheduling of the tasks. To address this situation, first of all static scheduling algorithms are developed for OpenVX based on Heterogeneous Earliest Finish Time (HEFT) experiments, where 70% improvement in performance is observed. Secondly, authors have observed that, in multiple implementation environment, there is load imbalance and hence degraded performance is possible. After understanding all the situations and considering the constraints, the authors have proposed XEFT algorithm which overlap the single implementation primitives to balance the load. Authors have recorded 33% improved performance over HEFT and 82% over the OpenVX environment.

Laizhong Cui et. al. [3] has worked in the segment of edge computing in which computing is done near the source of data. This processing technique help to improve the response time and optimize the bandwidth utilization. For optimizing the resources in this area, Deep Learning as a Service is used to handle issues related to the scheduling. Further it is identified that, the important data are exposed to the unpredictable attacks and DLaaS training agents can be compromised to attack the overall system. To address this situation, authors have proposed a more secured and decentralized environment for edge computing that is SAPE. In this concept, edge clusters are created which are responsible for executing the task within optimum time. Further to protect the system from external attacks, Deep Reinforcement Learning method have also been developed. Further, authors have proposed use of grouped verification

scheme, which is methodology for verification of tasks against malevolent attacks, from tasks and clusters, to increase the reliability.

Big data frameworks are highly used for large data analysis. For analysis of tones of data, large resources are required, sometimes local resources are insufficient to handle the large data and public resources are hired. Since use of cloud-based analysis is costly affair, local and cloud resources are used jointly, this concept is called as the hybrid computing cluster. Now in such environment, scheduling tasks or events becomes challenging due to various service level agreements. To come out of this situation, authors [4] have proposed novel scheduling algorithm, suitable for hybrid cloud environment that not only explores various cost models to optimize the cost of virtual machine utilization cost but also maximizes the job deadline meet percentage, effectively.

VLSI architecture that is integrated circuits are very complex in nature and is very dense and posses' components which executes the process rapidly. Such architectures are employed in diverse area like Wireless Sensor Network, Biomedical Engineering Applications, Digital Signal Processing and Image Processing, Radar Signal Processing and many other countless applications. When devices and architectures are employed in such critical applications to increase reliability and fast & onsite or remote patch-up in built test circuits are described. When this test is executed, it creates huge temperature which may damage the integrated circuit. To address this issue, authors [5] have implemented 3D System on Chip test scheduling, based on adjacency exclusion principle, considering temperature rise and testing time as the critical parameter. The simulation is carried out using HotSpot-6.02 tool.

Juan Fang et. al. [6] has proposed a novel task scheduling approach to deal with the load balancing problem in heterogeneous computing system. For the single task in a queue, load aware task scheduling approach is adopted. In this approach the computing power of the CPU and GPU is computed and the computing task is allocated according to the perception ratio. In case of multi task system, genetic algorithm-based task scheduling approach is adopted.



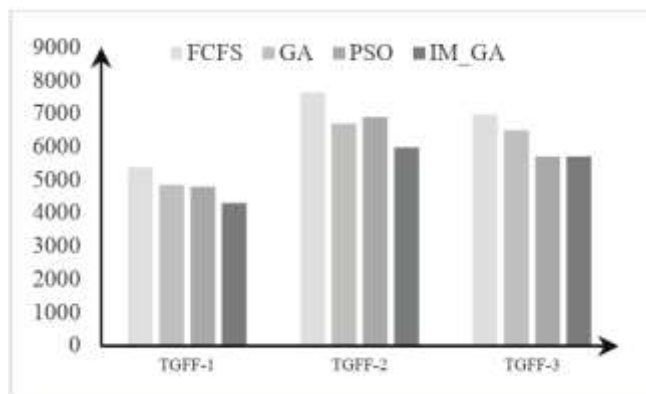


Fig. 1. Time Consumption Analysis

Need of high computation power led scientist to develop a multi-core embedded system architecture. In such architectures, the complex procedures are divided in to small number of functions or tasks. These tasks are properly arranged in a queue and are executed. For each of the task proper scheduling point is created, which indicates the starting point of that task. So, in the situation of multiple pending tasks to be executed without system collapse, task scheduling done. Xuemei Zhang et. al. [7] proposes a multi core scheduling technique for real time tasks with dependencies. The rate monolithic scheduling algorithm are implemented in LITMUSRT.

It is famous that, the dynamic scheduling algorithm like Earliest Deadline First (EDF) offers optimized use of processing capacity by giving high level of scheduling capability to the processing elements. But this is not the case when the processing element is not capable enough. Authors [8] calculated the overheads of the EDF and compared with the Rate Monotonic Scheduling and Fixed Priority Scheduling Algorithm. From the outcomes it is observed that, higher overheads of EDF-H may offer poor performance as compared to RMS.

2494

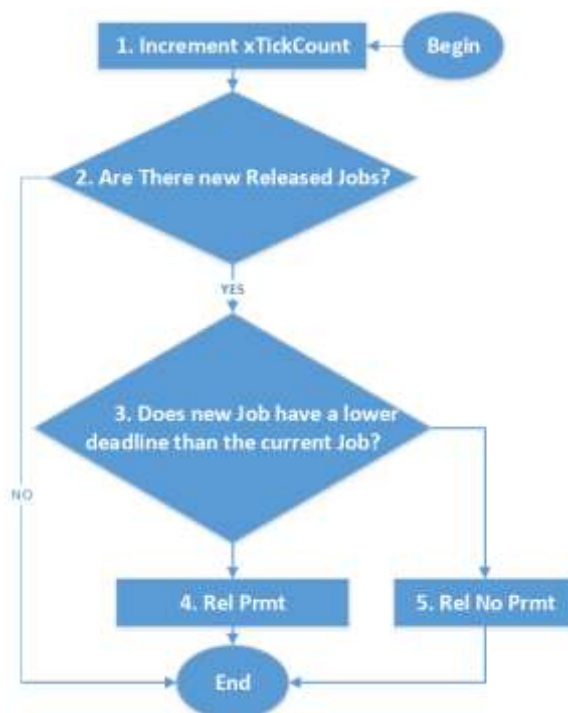


Fig. 2. Flowchart of xTaskIncrementTick Interrupt Routine

Yewon Jo, Suhyeon Yoo and Hyokyung Bahn[9] have disclosed a novel task scheduling algorithm to schedule real time tasks and interactive tasks. The proposed scheduling algorithm is claimed to be minimizes the power consumption of the system and

gives responses in reasonable amount of time and also gives deadline guarantees of real time tasks. Authors have further claimed that the proposed algorithm gives improved power consumption by 23%.



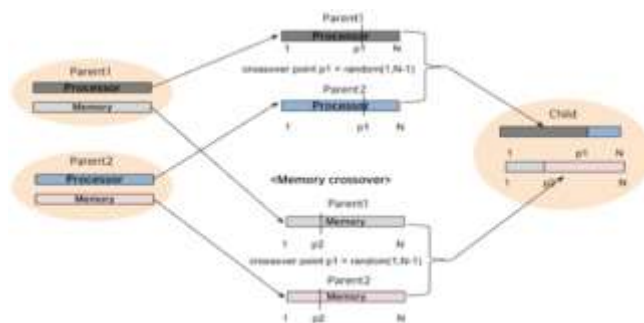


Fig. 3. Crossover Process

A scheduler considers many significant parameters while scheduling a task in embedded systems. Authors [10] have proposed a genetic algorithm based multi objective scheduling algorithm for efficient scheduling of task. The multiple objectives in these assumptions are reliability, power consumptions and time. For realizing the architecture, the architecture platform and the tasks are considered as the input to the algorithm. During the experiments, authors have observed better task scheduling as compared to the

other considered literatures and also the parameters like energy consumption, time and reliability is highly optimized when compared with the greedy method. The efficiency of the proposed algorithm, in task scheduling, as compared with the single objective heuristic scheduling algorithm, is 29.5% higher in terms of Scheduling Length Ratio (SLR) and 21% higher in terms of speed. The subsequent figure discloses initial population and final population using proposed algorithm.

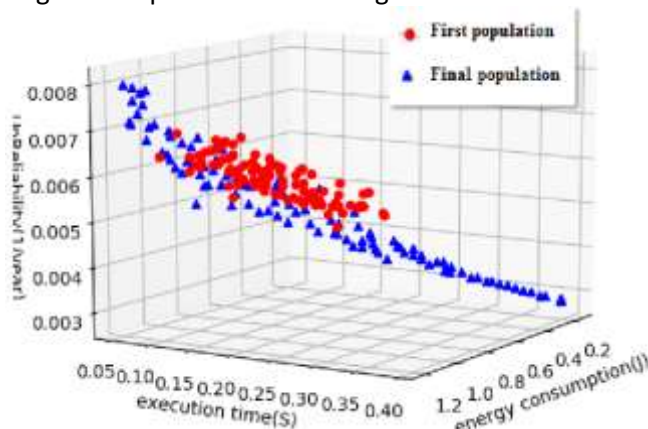


Fig. 4. Initial Population and Final Population using Proposed Algorithm

Hana Al-Momani and Kasim M. Al-Aubidv[11] have disclosed a novel real time intelligent scheduling algorithm for malleable manufacturing industries. In automated manufacturing environment different possible set considered are CNC machines, loading and unloading stations, multiple mobile robots and their charging stations. In this set, each CNC machine, loading and unloading station and mobile robots are considered as the node of the wireless sensor

network. The complete information of these nodes is exchanged with the main controller. The main controller is responsible for intelligently operating the different nodes that is different tasks. This intelligent operating or management is carried out using Fuzzy based task scheduling algorithm.

Dynamic Voltage Frequency Scaling (DVFS) is popular concept in the real-time embedded systems to conserve energy and increase battery life.



```

Input:
    • A task set  $\Gamma = \{\tau_1, \tau_2, \dots, \tau_n\}$ ;
Output:
    • Scheduling sequence on task set  $\Gamma$ .
2 Set  $t=0$ ,  $Ready\_List = \Phi$ ;
3 Call StaticF algorithm to calculate each task  $\tau_i$ 's static
  frequency  $s f_i$  ;
4 while (1) do
5   if (there's a task  $\tau_i$  is ready at time  $t$ ) then
6     | Insert  $\tau_i$  into  $Ready\_List = \Phi$  ;
7   end
8   if (there's a error on task  $\tau_i$  or  $\beta_i$  at time  $t$ ) then
9     | Insert  $\beta_i$  into  $Ready\_List = \Phi$  ;
10  end
11  if ( $Ready\_List$  is not empty) then
12    | Let  $running\_task$ =Select the task with the highest
13    | priority in  $Ready\_List$  according to RMS policy;
14    | Run  $running\_task$  at its static frequency;
15  end
16  if (there exists dynamic slack ) then
17    | Remove  $running\_task$  from  $Ready\_List$ ;
18    | Call AdjustF( $running\_task, Ready\_List$ ) algo-
19    | rithm to adjust the working frequency of the high-
20    | est priority task in  $Ready\_List$  ;
21  end
22   $t =$  next trigger time ;
23 end
    
```

Fig. 5. Proposed FTPART Algorithm

This motivates to design more sophisticated battery-operated applications like remote sensing using dust sensors. The systems which are designed using DVFS concepts are automatically adjust the frequency or speed of operation so as that power consumption is reduced and battery life is extended. But on the other side of the coin, this step increases the task execution time and does not guarantee the effective schedulability and reliability. To address this problem, authors [12] have proposed effective real time scheduling algorithm which is predictable to the fault and power. The proposed algorithm works around considering offline tasks assignment and online energy saving scheme by DVFS. Author guarantee that the real time tasks are completed before their deadlines and also reduces the energy consumption. Through simulation, further authors have also calculated that by using the proposed algorithm up to 22% and 16% energy can be

saved as compared to the reference algorithms. The figure 5 indicates the proposed FTPART algorithm.

Lukáš Kohútka et. al. [13] disclosed Application Specific Architecture in the form of Coprocessor for scheduling different process in mixed critical environment. Here author considered Earliest Deadline First algorithm for implementation at hardware level. The most eye-catchy feature we get through ASIC implementation is that it takes only two clock cycles for execution of the scheduling process. The implementation of the ASIC design is carried out using 28nm technology library of Taiwan Semiconductor Manufacturing Corporations (TSMC) fabrication laboratory. Authors further claimed that, the proposed architecture can handle diverse combinations of process cruciality, different timing parameters and deadlines along with higher CPU utilization. Further, it can also be used for scheduling non-, soft- and hard- real time processes.



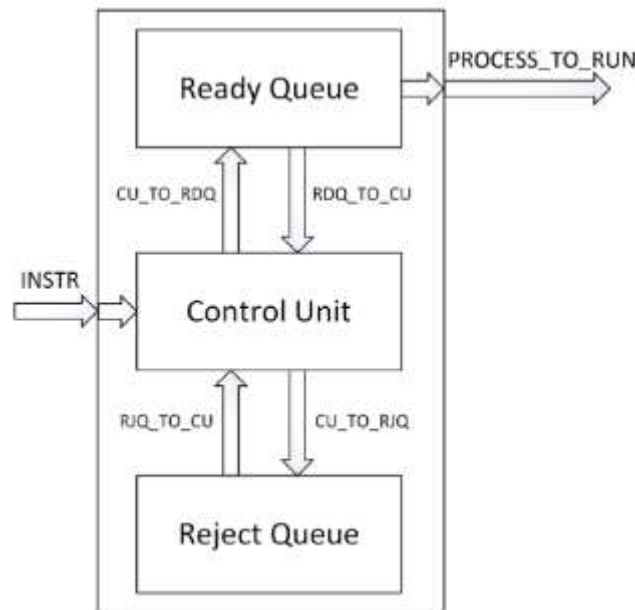


Fig. 6. RED Scheduler Top Module

In the distributed nature of the cloud computing, configuring, accessing, resource sharing and scheduling are the prime activities carried out. The authors [14] presents improved Round Robin scheduling algorithm for CPU activities with varying time quantum. The proposed algorithm found out to be better performing than the traditional Round Robin and IRRVQ algorithms, in terms of waiting time and turnaround time.

To address the different challenges like consideration of various possible configurations of CPU cores, challenges offered due to Dynamic Voltage Frequency Scaling (DVFS) for power consumption

management, balanced core utilization to avoid overheating hotspots and issues related to the hot-plugging, in heterogeneous processing environment, authors [15] have proposed use of Genetic Algorithm for running Deep Learning applications in heterogeneous environment of embedded system. To attain this, throughput of the single deep learning application is optimized then optimal response time for scheduling is identified and finally, overall energy consumption as per set constraints are identified. The performance evaluation is carried out on Galaxy S9 and HiKey 970.



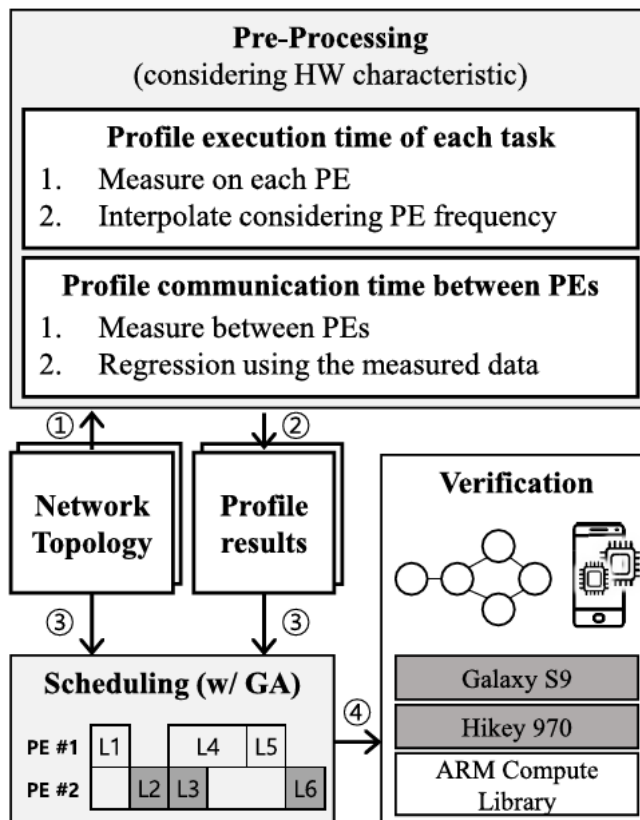


Fig. 7. Proposed Deep Learning Application Scheduling Flow

Scheduling is coming out to be real challenging assignment in real time systems. This is due to the complications of multiple tasks, their resource requirements, diverse constraints and functional characteristics of the tasks. Assuming these issues and to meet the real time requirements authors [16] have proposed Improved Earliest Deadline First (IEDF) algorithm designed with the queuing theory model. IEDF algorithm is designed with the foundation of EDF algorithm. In the modified version, the tasks

having high static priority are executed first and in the environment of ready queue, the deadline and execution time are considered. Through the simulation results it is noted that, total execution time of IEDF algorithm with large deadline is better than previously reported queuing algorithms. Further the number of errors is also optimized in improved Earliest Deadline First algorithm. The subsequent figure discloses times automata for access control software with time constraint.

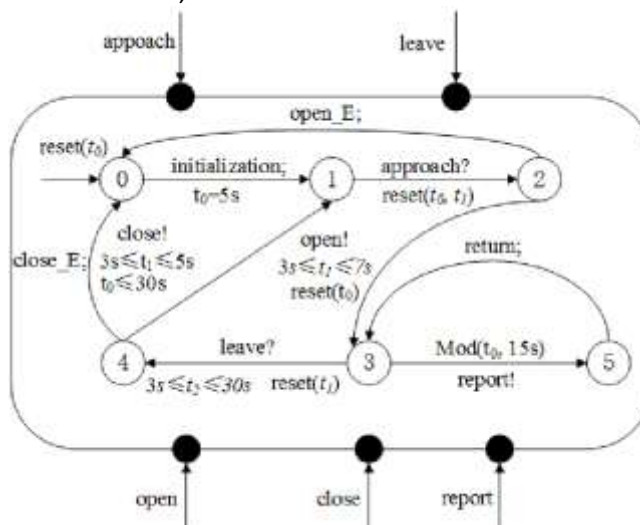


Fig. 8. Timed Automata for Access Control Software with Time Constraints.

Bashima Islam and Shahriar Nirjon[17] have proposed two scheduling algorithms

Celebi-Offline and Celebi-Online to address the challenges in the real time scheduling related to the



sporadic harvestable energy, exclusive computing and battery less systems. This newly proposed algorithm schedule computational and energy harvesting tasks through minimum energy harvesting and maximum schedulability of job. In the performance evaluation, it is found that, offline algorithm has 92% comparable

performance to the optimal scheduler and online scheduler schedules between 8% to 22% more tasks than Earliest Deadline First (EDF), Rate Monolithic (RM) and As Late As Possible (ALAP) scheduling algorithms.

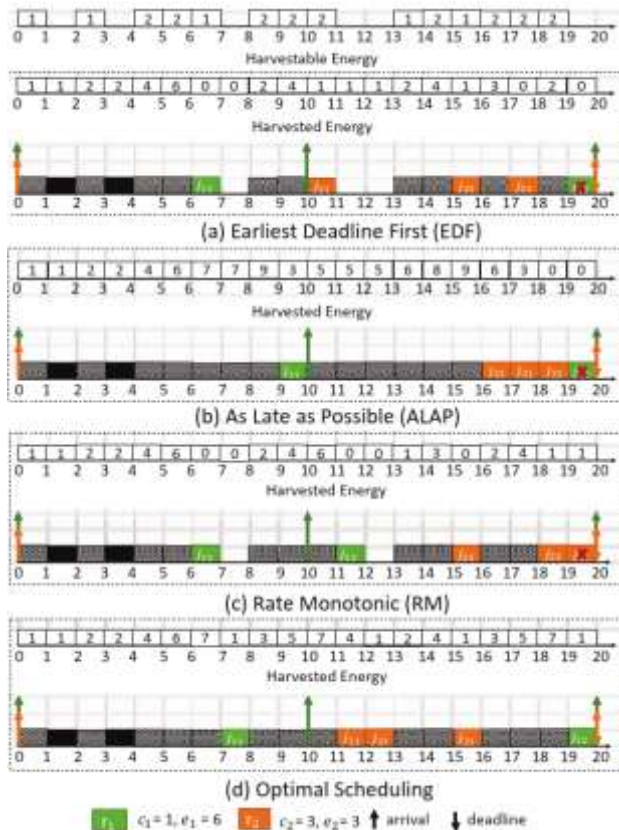


Fig. 9. Different scheduling Algorithms

2499

Authors [18] have found that, though the batteries are designed to have high energy density and are popularly used in the mobile embedded system applications, they suffer from rapid capacity degradation. As we observe in case of the new mobile handset, batter life is provided for 24 hours and as the

handset gets older, the life reduces to almost half day. To address this problem, authors observe the important factors which are responsible for capacity degradation of the batteries and can be controlled. Accordingly, battery scheduling algorithm is designed which minimizes the capacity degradation.

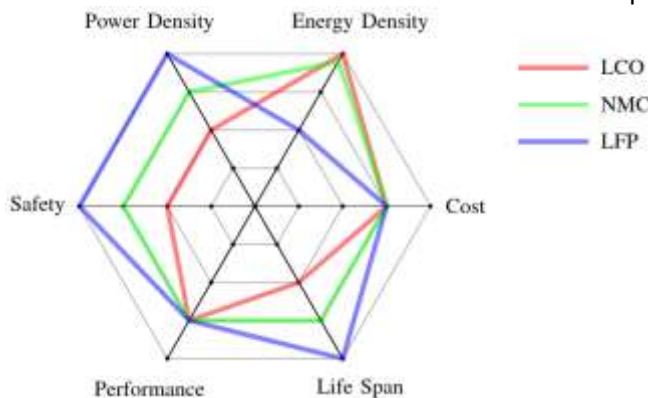


Fig. 10. Characteristics of Different Batteries

Authors [19] have proposed use of genetic algorithm to address the issues in the scheduling issues in the time triggered communication network

model along with the built-in compression. In the built-in compression model, correlated streams are combined and the size is also reduced. It means, the



highly correlated streams are compressed and their jobs are executed accordingly on the same systems. This helps to optimize the transmission time.

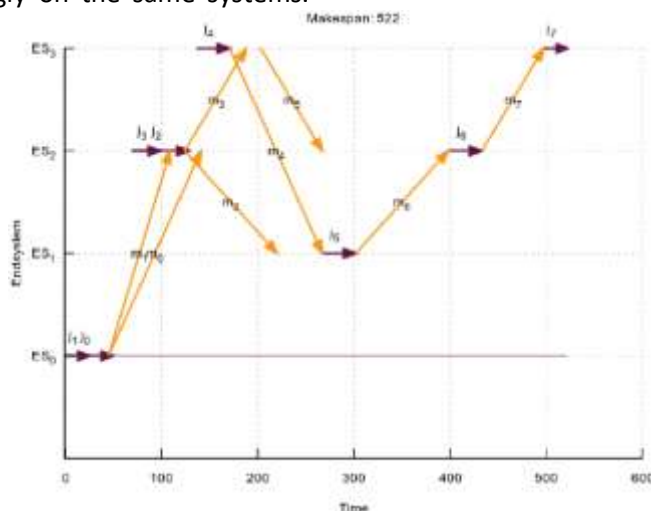


Fig. 11. Result for the Considered Model

Jianpeng Li et. al. [20] attempted to give solution to the issues arising in the scheduling of the tasks in heterogenous environments. In this solution, the processing stage and the urgency are considered as the key inputs in describing scheduling algorithm for heterogeneous environment. This scheduling is based on deadline constraints. In the proposed strategy, first of all free time and estimated execution time of each

node in the system is calculated and accordingly scheduling strategy is applied. This will also assure the priority of the emergency tasks. In the later section, different task switching strategies are employed conforming to the assigned tasks and relative deadlines of the tasks in a node. In the performance analysis, it is observed that, due to long waiting period no tasks are missing their deadlines.

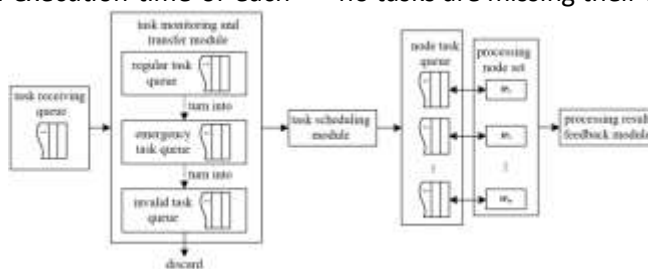


Fig. 12. Task Scheduling Model

In last few years, it is observed that automobile industries have incorporated smart features using modern electronics. These features required large amount of computation power for effective and reliable working. Along with this requirement, better scheduling algorithms also become need of today to effectively utilize the resource and manage the different tasks. Considering this requirement, authors [21] have proposed use of Dynamic Risk Allocation and Scheduling Algorithm for Electronics Control Unit is proposed. The proposed system is designed

considering the dynamic nature of the tasks appearing at the Engine Control Unit at different speed. In the Tri-Core scheduling environment, one of the cores is reserved for safety and critical task management and remaining two cores are left for routine tasks management. In the proposed scheduling algorithm, the hybrid architecture is proposed using Least Laxity First and Earliest Deadline First algorithms. Through the performance analysis it is observed that, there is increased core utilization and Less Context Switching.



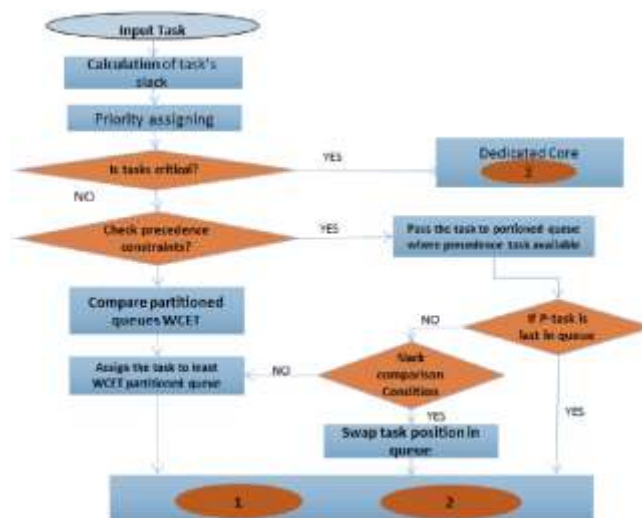


Fig. 13. Flowchart of Proposed Algorithm

Yanyue Yu and Yu Su[22] have disclosed a cloud-based scheduling algorithm based on Three Queues and dynamic priority management. In the proposed scheme, according to the priority of the tasks, different tasks are arranged in to the queue.

Subsequently, the tasks are divided according to the Data Input and Output Amount Phase, Total Number of Current Nodes, Completion Time of Map Tasks and Disk Input/Output Rate. With such bifurcation of the tasks, hardware resources are managed effectively.

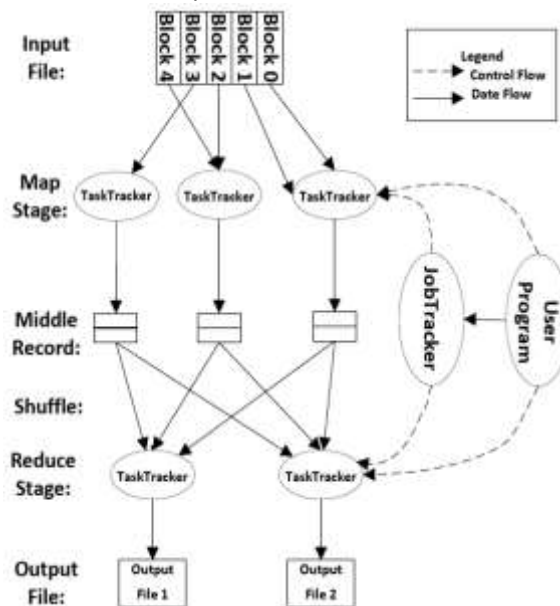


Fig. 14. Map Reduce Process Flow

III. CONCLUSION

Through the detailed literature survey, it is observed that, different authors have proposed scheduling algorithm for effective scheduling of the different tasks in VLSI designs, Real Time Embedded System applications and Computer Applications. The different constraints which are considered for effective management of tasks scheduling are scheduling points, task execution time, task deadline, feasible scheduling, power consumption and hot-spot management. Algorithms when implemented at hardware level gives rapid response, this is observed only in case of the 3D core management. Hence it is

identified that there is need to demonstrate the efficiency of the scheduling algorithm considering different case studies. Hence, in the subsequent version of the paper, novel hardware version of the scheduling algorithm will be disclosed.

REFERENCES

[1] S. Mandal, K. Gaurkar, P. Dasgupta and A. Hazra, "An RL based Approach for Thermal-Aware Energy Optimized Task Scheduling in Multi-core Processors," 2021 34th International Conference on VLSI Design and 2021 20th International Conference on Embedded Systems (VLSID), 2021,



- pp. 181-186, doi: 10.1109/VLSID51830.2021.00036.
- [2] F. Lumpp, S. Aldegheri, H. Patel and N. Bombieri, "Task Mapping and Scheduling for OpenVX Applications on Heterogeneous Multi/Many-core Architectures," in IEEE Transactions on Computers, doi: 10.1109/TC.2021.3059528.
- [3] L. Cui, Z. Chen, S. Yang, R. Chen and Z. Ming, "A Secure and Decentralized DLaaS Platform for Edge Resource Scheduling against Adversarial Attacks," in IEEE Transactions on Computers, doi: 10.1109/TC.2021.3074806.
- [4] M. T. Islam, H. Wu, S. Karunasekera and R. Buyya, "SLA-based Scheduling of Spark Jobs in Hybrid Cloud Computing Environments," in IEEE Transactions on Computers, doi: 10.1109/TC.2021.3075625.
- [5] I. Rawat and V. Singh, "Temperature efficient parallel test scheduling for higher order 3D stacked SoCs," 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2021, pp. 804-809, doi: 10.1109/ICCCIS51004.2021.9397211.
- [6] J. Fang, J. Zhang, S. Lu and H. Zhao, "Exploration on Task Scheduling Strategy for CPU-GPU Heterogeneous Computing System," 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2020, pp. 306-311, doi: 10.1109/ISVLSI49217.2020.00063.
- [7] X. Zhang, S. Huang and J. Li, "Implementation of Real-Time Scheduling Algorithm on Multi-Core Platform," 2020 International Conference on Computer Network, Electronic and Automation (ICCNEA), 2020, pp. 66-71, doi: 10.1109/ICCNEA50255.2020.00023.
- [8] G. Oliveira and G. Lima, "Evaluation of Scheduling Algorithms for Embedded FreeRTOS-based Systems," 2020 X Brazilian Symposium on Computing Systems Engineering (SBESC), 2020, pp. 1-8, doi: 10.1109/SBESC51047.2020.9277851.
- [9] Y. Jo, S. Yoo and H. Bahn, "Student Session: Power-Saving Integrated Task Scheduling in Multicore and Hybrid Memory Environment," 2020 IEEE 26th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), 2020, pp. 1-2, doi: 10.1109/RTCSA50079.2020.9203632.
- [10] M. Nikseresht and M. Raji, "A Multi-objective Task Scheduling Method for Embedded System Design," 2020 25th International Computer Conference, Computer Society of Iran (CSICC), 2020, pp. 1-7, doi: 10.1109/CSICC49403.2020.9050062.
- [11] H. Al-Momani and K. M. Al-Aubidv, "Fuzzy-Based Task Scheduling of Mobile Robots in Flexible Manufacturing Systems," 2020 17th International Multi-Conference on Systems, Signals & Devices (SSD), 2020, pp. 565-571, doi: 10.1109/SSD49366.2020.9364247.
- [12] P. Zhu, D. Luo and X. Chen, "Fault-tolerant and power-aware scheduling in embedded real-time systems," 2020 International Conference on Computer, Information and Telecommunication Systems (CITS), 2020, pp. 1-5, doi: 10.1109/CITS49457.2020.9232471.
- [13] L. Kohútka, L. Nagy and V. Stopjaková, "RED-based Scheduler on Chip for Mixed-Criticality Real-Time Systems," 2020 9th Mediterranean Conference on Embedded Computing (MECO), 2020, pp. 1-4, doi: 10.1109/MECO49872.2020.9134328.
- [14] F. Alaa, M. M. Zoulikha and B. Hayat, "Improved Round Robin Scheduling Algorithm With Varying Time Quantum," 2020 Second International Conference on Embedded & Distributed Systems (EDiS), 2020, pp. 33-37, doi: 10.1109/EDiS49545.2020.9296452.
- [15] D. Kang, J. Oh, J. Choi, Y. Yi and S. Ha, "Scheduling of Deep Learning Applications Onto Heterogeneous Processors in an Embedded Device," in IEEE Access, vol. 8, pp. 43980-43991, 2020, doi: 10.1109/ACCESS.2020.2977496.
- [16] Y. Wang et al., "Research on Real-Time Embedded Software Scheduling Model Based on EDF," in IEEE Access, vol. 8, pp. 20058-20066, 2020, doi: 10.1109/ACCESS.2020.2969229.
- [17] B. Islam and S. Nirjon, "Scheduling Computational and Energy Harvesting Tasks in Deadline-Aware Intermittent Systems," 2020 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2020, pp. 95-109, doi: 10.1109/RTAS48715.2020.00-14.
- [18] J. Kwak and J. Lee, "Minimizing Capacity Degradation of Heterogeneous Batteries in a Mobile Embedded System," in IEEE Embedded Systems Letters, vol. 12, no. 1, pp. 25-28, March 2020, doi: 10.1109/LES.2019.2929069.
- [19] S. Majidi and R. Obermaisser, "Genetic Algorithm for Scheduling Time-Triggered Communication Networks with Data Compression," 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), 2019, pp. 373-379, doi: 10.1109/ICPHYS.2019.8780192.



- [20]J. Li, G. Zheng, H. Zhang and G. Shi, "Task Scheduling Algorithm for Heterogeneous Real-time Systems Based on Deadline Constraints," 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), 2019, pp. 113-116, doi: 10.1109/ICEIEC.2019.8784641.
- [21]M. Faizan and A. S. Pillai, "Dynamic Task Allocation and Scheduling for Multicore Electronics Control Unit (ECU)," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 821-826, doi: 10.1109/ICECA.2019.8822086.
- [22]Y. Yu and Y. Su, "Cloud Task Scheduling Algorithm Based on Three Queues and Dynamic Priority," 2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), 2019, pp. 278-282, doi: 10.1109/ICPICS47731.2019.8942588.

