Interdiscip J Virtual Learn Med Sci. 2019 June; 10(2):e89300. doi: 10.5812/ijvlms.89300.

Published online 2019 June 5. **Research Article**

Energy-Efficient Algorithm for Mixed-Criticality Systems in E-Learning Environment

Seyed Hasan Sadeghzadeh [1, *]

[1]Department of Information and Communication Technology, Payame Noor University, Tehran, Iran

[*]*Corresponding author*: Department of Information and Communication Technology, Payame Noor University, Tehran, Iran. Email: sadeghzadeh1@gmail.com

Received 2019 January 15; **Revised** 2019 February 04; **Accepted** 2019 February 04.

Abstract

Background: Low-energy consumption is a vital concern in E-learning due to high-volume processing and the fact that mobile technologies are usually battery-operated devices.

Methods: The method is simulated by developing a discrete-event simulation in C#. The validation of the proposed method is performed on generated task sets as used in similar work. The characteristic of randomly produced tasks is similar to the well-known techniques of task generation in mixed-criticality (MC) systems.

Results: The simulation results show that energy consumption can be improved up to 23% in comparison to similar approaches. The most important factor for this satisfaction was the reservation times of critical tasks to further reduce the processor frequency.

Conclusions: The internet of thing (IoT) is poised to be one of the most disruptive technologies in E-learning environment. The IoT is a kind of MC system that integrates multiple things with different criticalities into the same platform. Mobile technologies provide education to people through mobile devices. These devices are usually battery-operated and owing to high-volume processing, Low-energy consumption becomes a vital concern in E-learning. Therefore, this paper was discussed about the MC system in general. Finally, the paper was proposed a scheduling technique to minimize the energy consumption of E-learning devices that use the IoT.

Keywords: E-Learning, IoT, Mixed-Criticality, Mobile Devices, Time Management, Energy Consumption, Scheduling

1. Background

The internet of thing (IoT) is a network of physical things (tasks) embedded with software, electronics, sensors, and network connectivity with the capability of sending and receiving information. The powerful possibilities opened with the IoT are being included in E-Learning in order to improve the quality of learning by facilitating the access to resources and educational services, and provides tools such as distance interaction and participation. The IoT is a real-time system that is used in applications such as management, automotive education, military, medical, and avionics applications (1-3). In the design phase of these systems, some vital requirements such as reducing the costs, weight and size have encouraged designers to integrate the tasks of varying importance (criticality levels) on a common hardware platform. Owing to the integration of critical (HI-criticality) tasks with non-critical (LO-criticality) tasks, these systems are called mixed-criticality (MC) systems (4-10). MC systems are considered to be the next generation of complex real-time systems (11-16). The IoT is a MC system that LO-criticality tasks only require validations issued by system designers; however, HI-criticality tasks not only require these validations but also require certifications issued by legal certification authorities (CAs) (4).

The concept of MC was first introduced by Vestal in 2007 (17). Vestal proved that traditional algorithms such as rate monotonic (RM) and deadline monotonic (DM) were not optimal for MC systems. Therefore, various scheduling algorithms have been proposed to schedule MC tasks based on two criticality levels (6-8). In addition to the schedulability, energy consumption in battery-operated devices becomes the main concern (18, 19). Dynamic voltage and frequency scaling (DVFS) is a well-known system-level energy management technique in real-time embedded systems (20). Indeed, this would be obtained at the expense of increasing the system response times, which can cause deadline violations in real-time systems. Therefore, there is a significant conflict between energy management techniques and real-time constraints (21). Many studies have been done on energy management in the domain of hard real-time systems (22, 23). However, in these studies, the criticality levels of tasks have been neglected and all tasks have been considered with the same criticality level.

Recently, several energy management techniques have been proposed for MC systems (24-28). In the study conducted by Huang et al.(24), virtual deadlines of HI-criticality tasks are changed based on the earliest deadline first with virtual deadline (EDF-VD) (6) to employ DVFS. In energy-constrained MC systems, a technique is provided (25), which permits to miss the deadlines of LO-criticality tasks. Moreover, it has been proposed a stretch technique to exploit service level degradation of LO-criticality tasks for energy management in MC systems (26). In fact, the aforementioned studies discuss energy management for single-core MC systems. Owing to high efficiency and high computing power of multi-core processors, they are increasingly employed in critical applications (29). In the study conducted by Legout et al. (27) an approach to trade-off between the number of missed deadlines for LO-criticality tasks and their energy minimization on multi-cores are provided, while deadlines of HI-criticality tasks are always guaranteed. A new energy-aware mapping technique is proposed (28) in which the frequency of each task is statically determined, according to the presented algorithm (24).

In MC systems, in order to guarantee system safety, it is necessary to reserve time budget $t = c_i^{high} - c_i^{low}$ for critical tasks, such that they can still meet their deadlines even if they overrun (24). However, static algorithms do not use the reservation times of critical tasks to minimize the energy consumption when the overrun does not occur.

In this paper, a scheduling technique is proposed to minimize the energy consumption of E-learning devices. The proposed technique focuses on the reservation times $t = c_i^{high} - c_i^{low}$ of critical tasks to further reduce the processor frequency. Reservation times are created due to reserve time budgets for task overrun. These reservation times are unused if critical tasks finish their c_i^{low} without any delay (overrun does not occur). Since task overrun is rare (24, 26), the proposed technique can greatly reduce the expected energy consumption for MC systems in LO-mode. After selecting a proper mapping technique, to minimize the energy consumption of tasks, the proposed technique has been applied to these tasks, which have been mapped to the processor's cores. Finally, the result of the proposed technique has been compared with the similar approach. This study was designed as a new dynamic Energy-aware scheduling technique to minimize the energy consumption of E-learning devices. The proposed technique focuses on the reservation times $t = c_i^{high} - c_i^{low}$ of critical tasks to further reduce the processor frequency.

2. Methods

The assumptions for Thing (task) model in this study, similar to MC thing model introduced (28, 30).

For notational convenience, following definitions will be used:

- Task τ_i L_i-mode utilization

$$U_{L_i}(\tau_i)\ :\ U_{L_i}(\tau_i) = \frac{c_i^{L_i}}{T_i} \tag{1}$$

where $L_i \in$ [LO, HI]

- LO-criticality task set L_i-mode utilization

$$U_{L_i}(\Gamma_L)\ :\ U_{L_i}(\Gamma_L) = \sum\nolimits_{\tau_i \in \Gamma_L} u_{L_i}(\tau_i) \tag{2}$$

where $L_i \in$ [LO, HI]

- HI-criticality task set L_i-mode utilization

$$U_{L_i}(\Gamma_H)\ :\ U_{L_i}(\Gamma_{LH}) = \sum\nolimits_{\tau_i \in \Gamma_H} u_{L_i}(\tau_i) \tag{3}$$

where $L_i \in$ [LO, HI]

Also, the model that has been employed for power consumption is the same as energy/power model suggested (30, 31). The power consumption of the system is calculated as follows (30, 31):

$$P(f(\tau_i)) = P_s + P_d \tag{4}$$

$$= P_s + (P_{ind} + P_{dep}) \tag{5}$$

$$= P_s + \left(P_{ind} + C_{ef} f(\tau_i)^{\theta}\right) \tag{6}$$

Where, P_s stands for the static power consumption when the system is in the standby state. P_d is dynamic power consumption when the system is in the working state. Parameters C_{ef} and θ are system dependent constants and $\theta \geq 2$ (30). If task τi executes on working frequency $f(\tau_i)$, its execution time is $C_i/f(\tau_i)$ (31).

2.1. MC Scheduling

EDF-VD is a widely used scheduling algorithm for MC task sets (6). VD for any HI-criticality task is calculated by multiplying its deadline in deadline shortening factor X ($0 < X \leq 1$) (6). In this paper, it is assumed that the partitioned EDF-VD algorithm is used to schedule the MC task sets (32).

2.2. Motivational Example

The task set with task parameters is presented in Table 1 (24) and DVFS scheduling of tasks are shown in Figure 1. As can be seen in Figure 1, there are still more times to further reduce the task frequency. Using a dynamic algorithm, the reservation times can be used for reducing energy consumption by applying DVFS. Therefore, the main aim of the proposed approach is providing a dynamic scheduling technique to reduce energy consumption by using reservation times, which are reserved for critical tasks. In this

Table 1. Example Task Set with Parameters in Units of ms

τ_i	c_i^{low}	c_i^{high}	P_i	l_i
τ_1	2	5	8	HI
τ_2	1	1	12	LO
τ_3	2	2	16	LO

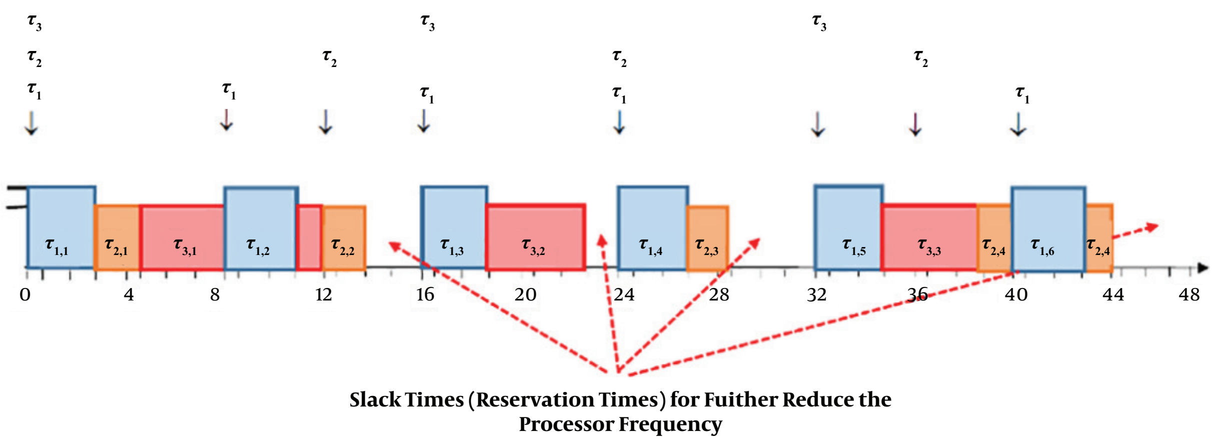

Figure 1. DVFS scheduling is shown (24)

paper, it is assumed that the EDF-VD algorithm is used to schedule the mixed-criticality task sets.

However, we aimed to minimize the expected IoT energy consumption by applying DVFS offline for tasks scheduled under EDF-VD. To describe the proposed scheduling technique, in the first step, representation of the problem is described and then the stages of the problem-solving will be introduced in section 3.3.

2.3. Representation of the Problem

It is required to represent the main problem factors. Therefore, the problem can be expressed using four factors as illustrated in the Equation 7.

$$Problem\ (T_s, M_{sys}, f_i, E_{sys}) \quad (7)$$

The constituting factors of the equation are as follows:

Task set (T_s): represents the tasks that will be executed.

- System mode (M_{sys}): represents the operational mode of the system. M_{sys} = low expresses the system in LO-mode and M_{sys} = high states the system in HI-mode.

- Task frequency (f_i): indicates the frequency that is assigned to each task in different system modes.

- System energy (E_{sys}): indicates the power consumption in elapsed time (time of task execution). Energy consumption of a task is represented in the Equation 8

$$E_{sys} = E_{LO} + E_{HI} \quad (8)$$

that E_{LO} represents energy consumption in M_{sys} = low and can be calculated by the Equation 9

$$E_{LO} = \sum_{T_i \in T_{LO}} \frac{c_i^{low}}{f_i^{LO}} \ (P_{Active}) \quad (9)$$

and E_{HI} represents energy consumption in M_{sys} = high and can be calculated by the Equation 10

$$E_{HI} = \sum_{T_i \in T_{HI}} \frac{c_i^{high}}{f_i^{HI}} \ (P_{Active}) \quad (10)$$

2.4. Stages of Problem-Solving

After the representation of the problem, solving the following stages is necessary for scheduling task:

- Finding an energy-efficient mapping technique.
- Calculating the frequency for LO-criticality tasks.
- Calculating the frequency for HI-criticality tasks.

2.5. Finding an Energy-Efficient Mapping Technique

There are several task mapping techniques in multi-core MC systems: Baruah et al.'s technique (32), Gu et al.'s technique (33), and Narayana et al.'s technique (EM3) (28). Therefore, in this study, all task mapping techniques in multi-core MC systems are employed to find an energy-efficient mapping technique.

2.6. Frequency Calculation

In scheduling, a reservation time is assigned to HI-criticality tasks. This time can be used for further reducing the frequency and improving energy consumption. In the proposed technique, to use reservation, the new frequency level of each task ($f_i^{'}$) dynamically is denoted by:

$$f_i^{'} = Max\,[f_{ee}, f_i \times U_i] \quad (11)$$

U_i is updated task utilization in case of completing HI-criticality tasks in c_i^{low}, and f_{ee} is energy-efficient frequency.

3. Results

The proposed technique has been evaluated on randomly generated task, similar to (24, 26, 28, 30). The method is simulated by developing a discrete-event simulation in C#. The characteristic of randomly produced tasks is similar to (28) which is one of the well-known techniques of task generation in MC systems (28). Task utilizations are chosen, according to the proposed UUniFast scheme (34).

3.1. Impact of Task Utilization

Figure 2 shows the average normalized energy consumption based on task utilization that varies from 1.1 to 3.7. It is obvious that as task set utilization increases, energy consumption in EM3 technique is less than other techniques. The reason is load balancing for cores, which leads to the optimal use of slack time to further reduce the frequency of tasks.

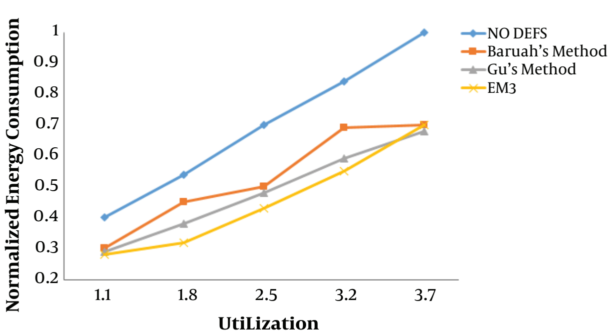

Figure 2. The impact of utilization on energy consumption is shown

3.2. Impact of μ

Consider U_L (Γ_L) and U_L (Γ_H) to be 0.5 and 0.4, respectively. Figure 3 shows the average normalized energy consumption based on task utilization that varies from 1.5 to 2.5. It is obvious that when μ increases, energy consumption in EM3 technique is less than other techniques.

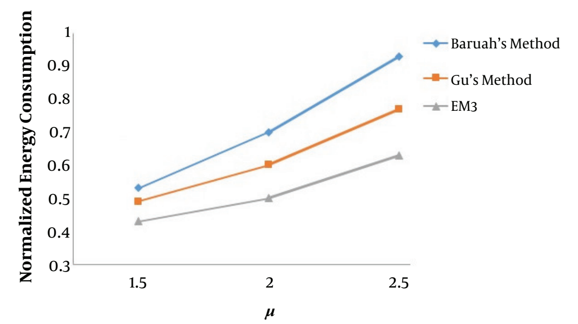

Figure 3. The impact of μ on energy consumption is indicated

3.3. Impact of the Number of Tasks

Figure 4 represents energy consumption based on numbers of tasks when U_L (Γ_L) = 0.6 and U_H (Γ_H) = 0.6. As can be seen the energy consumption of the system depends on the task utilization, not the number of the tasks.

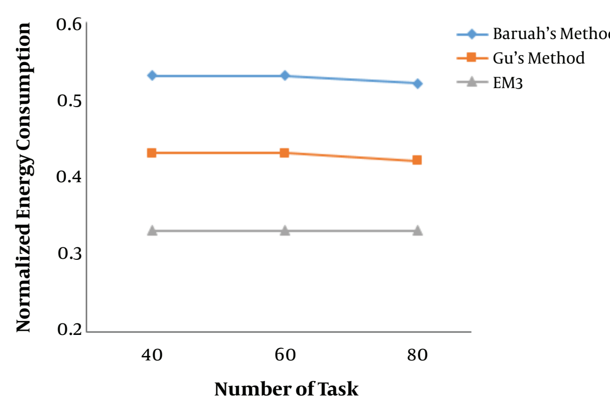

Figure 4. The impact of the number of tasks on energy consumption is indicated

3.4. Impact of Overrun

To investigate energy in different modes, we consider w_{LO} for M_{sys} = low and w_{HI} for M_{sys} = high, in which $w_{LO}, w_{HI} \in [0,1]$. In M_{sys} = low Λ w_{LO} = 1, w_{HI} = 0 and similarly in M_{sys} = high Λ w_{LO} = 1, w_{HI} = 0. According to Equation 10, energy consumption is calculated as follows:

$$E = (w_{LO} \times E_{LO}) + (w_{HI} \times E_{HI}) \quad (12)$$

As can be seen in Figure 5 as w_{LO} increases, energy consumption decreases. The main reason is that when w_{LO} increases from 0 to 1, the system is more active in M_{sys} = low and owing to the use of DVFS in M_{sys} = low, energy consumption decreases.

Finding an energy-efficient mapping technique

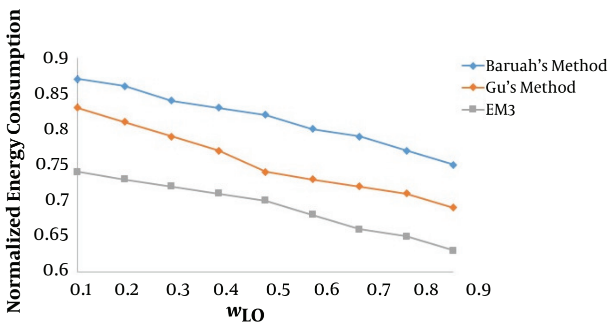

Figure 5. The impact of weight factors on energy consumption is indicated

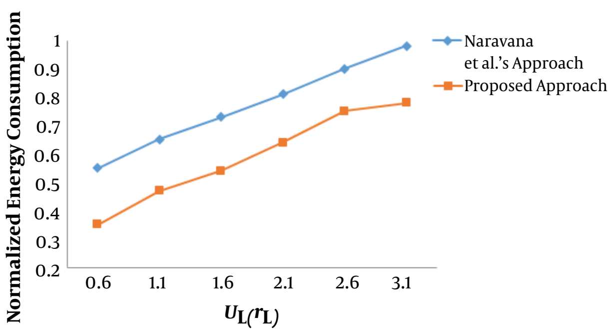

Figure 6. Comparing energy consumption of the proposed technique with Narayana et al. approach (28)

In the previous sub-sections, energy consumption has been evaluated using different task mapping techniques. Therefore, EM3 technique can be an energy-efficient mapping technique, which leaves more slack time to employ DVFS.

3.5. Comparison with Narayana et al.'s approach

In Narayana et al.'s approach (28), a technique has been presented for energy consumption in a multi-core system. In this paper after task mapping, task frequency has been statically determined, according to the presented algorithm (24). However, since this algorithm is static, created slack time during task execution cannot be used in order to decrease task frequency. Figure 6 shows energy consumption when U_H $(\Gamma_H) = 0.5$ and U_L (Γ_L) varies from 0.6 to 3.3. It is obvious that energy consumption in proposed technique is less than Narayana et al.'s approach. The main reason is that in the proposed technique, created slack time during task execution is used in order to decrease task frequency, which leads to a decrease in energy consumption.

4. Discussion

E-learning is a concept that integrates learning and information technology in teaching. Internet of Things (IoT) is changing everything, and E-Learning is not an exception. The IoT is reducing the difference between on-campus education and distance education. The IoT is a mixed-criticality (MC) system that integrates multiple things (tasks) with different criticalities into a same platform. The state-of-the-art studies have focused on providing timely management for tasks having different criticality levels. This is achieved by dropping non-critical tasks when the system changes to high-critical behavior due to overrun of critical tasks. In fact, E-learning carried out through mobile technologies such as mobile phones, personal digital assistants (PDAs), audio players, electronic books, etc. However, Low-energy consumption becomes a vital concern in E-learning due to high-volume processing and the fact that mobile technologies are usually battery-operated devices. This paper discusses the MC system in general. Finally, the paper proposes a scheduling technique to minimize the energy consumption of E-learning devices using the IoT. This technique uses the reservation times ($t = c_i^{high} - c_i^{low}$) of critical tasks for further reducing the processor frequency. Reservation times are created due to reserve time budgets for task overrun. These reservation times are unused if critical tasks finish their c_i^{low} without any delay. Since task overrun is rare, the proposed technique can greatly reduce the expected energy consumption of E-learning devices. The simulation results show that energy consumption of the proposed technique can be improved up to 23% in comparison to similar approaches.

Acknowledgments

I thank my colleagues from DDEmS Lab in Department of Computer Engineering from the Ferdowsi University of Mashhad who provided insight and expertise that greatly assisted the research, although they may not agree with all of the conclusions of this paper.

Footnotes

Conflict of Interests: There is no conflict of interests in this study.

Funding/Support: No financial support or funding were received for this research article.

References

1. Xu H, Li R, Zeng L, Li K, Pan C. Energy-efficient scheduling with reliability guarantee in embedded real-time systems. *Sustain Comput Infor Syst*. 2018;**18**:137–48. doi: 10.1016/j.suscom.2018.01.005.

2. Kopetz H. *Real-time systems: design principles for distributed embedded applications*. Springer Science & Business Media; 2011.
3. Baruah SK, Cucu-Grosjean L, Davis RI, Maiza C. Mixed criticality on multicore/manycore platforms (dagstuhl seminar 15121). *Dagstuhl Seminar 15121*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik; 2015.
4. Baruah SK, Bonifaci V, D'Angelo G, Li H, Marchetti-Spaccamela A, Megow N, et al. Scheduling real-time mixed-criticality jobs. *IEEE Tran Comput*. 2012;**61**(8):1140–52. doi: 10.1109/tc.2011.142.
5. Santy F, George L, Thierry P, Goossens J. Relaxing mixed-criticality scheduling strictness for task sets scheduled with FP. *Real-Time Systems (ECRTS), 24th Euromicro Conference on 2012 Jul 11*. IEEE; 2012. p. 155–65.
6. Baruah S, Bonifaci V, Dangelo G, Li H, Marchetti-Spaccamela A, van der Ster S, et al. The preemptive uniprocessor scheduling of mixed-criticality implicit-deadline sporadic task systems. *Real-Time Systems (ECRTS), 24th Euromicro Conference on 2012 Jul 11*. IEEE; 2012. p. 145–54.
7. Park T, Kim S. Dynamic scheduling algorithm and its schedulability analysis for certifiable dual-criticality systems. *Embedded Software (EMSOFT), Proceedings of the International Conference on 2011 Oct 9*. IEEE; 2011. 253 p.
8. Baruah S, Li H, Stougie L. Towards the design of certifiable mixed-criticality systems. *Real-Time and Embedded Technology and Applications Symposium (RTAS), 16th IEEE 2010 Apr 12*. IEEE; 2010. p. 13–22.
9. Niz D, Lakshmanan K, Rajkumar R. On the scheduling of mixed-criticality real-time task sets. *Real-Time Systems Symposium, RTSS. 30th IEEE 2009 Dec 1*. IEEE; 2009. p. 291–300.
10. Baruah S, Vestal S. Schedulability analysis of sporadic tasks with multiple criticality specifications. *Real-Time Systems. ECRTS'08. Euromicro Conference on 2008 Jul 2*. IEEE; 2008. p. 147–55.
11. Su H, Zhu D, Brandt S. An elastic mixed-criticality task model and early-release EDF scheduling algorithms. *ACM Trans Desig Autom Electron Syst*. 2016;**22**(2):1–25. doi: 10.1145/2984633.
12. Su H, Zhu D. An elastic mixed-criticality task model and its scheduling algorithm. *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE; 2013. p. 147–52.
13. Huang P, Yang H, Thiele L. On the scheduling of fault-tolerant mixed-criticality systems. *Design Automation Conference (DAC), 51st ACM/EDAC/IEEE*. IEEE; 2014. p. 1–6.
14. Ekberg P, Yi W. Bounding and shaping the demand of generalized mixed-criticality sporadic task systems. *Real-Time Syst*. 2013;**50**(1):48–86. doi: 10.1007/s11241-013-9187-z.
15. Su H, Guan N, Zhu D. Service guarantee exploration for mixed-criticality systems. *2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*. IEEE; 2014. p. 1–10.
16. Schreiner S, Gruttner K, Rosinger S, Rettberg A. Autonomous flight control meets custom payload processing: A mixed-critical avionics architecture approach for civilian UAVs. *Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), IEEE 17th International Symposium on 2014 Jun 10*. 2014. p. 348–57.
17. Vestal S. Preemptive Scheduling of Multi-criticality Systems with Varying Degrees of Execution Time Assurance. *Real-Time Systems Symposium. RTSS 2007. 28th IEEE International 2007 Dec 3*. IEEE; 2007. p. 239–43.
18. Legout V, Jan M, Pautet L. Scheduling algorithms to reduce the static energy consumption of real-time systems. *Real-Time Syst*. 2015;**51**(2):153–91. doi: 10.1007/s11241-014-9207-7.
19. Marwedel P. *Embedded system design*. **1**. New York: Springer; 2006.
20. Cheng D, Zhou X, Lama P, Ji M, Jiang C. Energy Efficiency Aware Task Assignment with DVFS in Heterogeneous Hadoop Clusters. *IEEE Trans Parallel Distrib Systems*. 2018;**29**(1):70–82. doi: 10.1109/tpds.2017.2745571.
21. Weste NH, Eshraghian K. *Principles of CMOS VLSI design: A systems perspective*. California: Addision-Wesley Publishing; 1994.
22. Bambagini M, Marinoni M, Aydin H, Buttazzo G. Energy-Aware Scheduling for Real-Time Systems. *ACM Tran Embed Comput Syst*. 2016;**15**(1):1–34. doi: 10.1145/2808231.
23. Xie G, Zeng G, Xiao X, Li R, Li K. Energy-efficient scheduling algorithms for real-time parallel applications on heterogeneous distributed embedded systems. *IEEE Trans Parallel Distrib Systems*. 2017;**28**(12):3426–42. doi: 10.1109/tpds.2017.2730876.
24. Huang P, Kumar P, Giannopoulou G, Thiele L. Energy efficient DVFS scheduling for mixed-criticality systems. *Proceedings of the 14th International Conference on Embedded Software*. ACM; 2014. p. 1–10.
25. Volp M, Hahnel M, Lackorzynski A. Has energy surpassed timeliness? Scheduling energy-constrained mixed-criticality systems. *Real-Time and Embedded Technology and Applications Symposium (RTAS), IEEE 20th*. IEEE; 2014. p. 275–84.
26. Taherin A, Salehi M, Ejlali A. Reliability-aware energy management in mixed-criticality systems. *IEEE Tran Sustainable Comput*. 2018;**3**(3):195–208. doi: 10.1109/tsusc.2018.2801123.
27. Legout V, Jan M, Pautet L. Mixed-criticality multiprocessor real-time systems: Energy consumption vs deadline misses. *First Workshop on Real-Time Mixed Criticality Systems (ReTiMiCS)*. 2013. p. 1–6.
28. Narayana S, Huang P, Giannopoulou G, Thiele L, Prasad RV. Exploring energy saving for mixed-criticality systems on multi-cores. *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE; 2016. p. 1–12.
29. Han JJ, Tao X, Zhu D, Aydin H, Shao Z, Yang LT. Multicore mixed-criticality systems: Partitioned scheduling and utilization bound. *IEEE Tran Comput Aided Des Integr Circ Syst*. 2018;**37**(1):21–34. doi: 10.1109/tcad.2017.2697955.
30. Li Z, Guo C, Hua X, Ren S. Reliability guaranteed energy minimization on mixed-criticality systems. *J Syst Softw*. 2016;**112**:1–10. doi: 10.1016/j.jss.2015.10.029.
31. Moghaddas V, Fazeli M, Patooghy A. Reliability-oriented scheduling for static-priority real-time tasks in standby-sparing systems. *Microprocess Microsy*. 2016;**45**:208–15. doi: 10.1016/j.micpro.2016.05.005.
32. Baruah S, Chattopadhyay B, Li H, Shin I. Mixed-criticality scheduling on multiprocessors. *Real-Time Syst*. 2013;**50**(1):142–77. doi: 10.1007/s11241-013-9184-2.
33. Gu C, Guan N, Deng Q, Yi W. Partitioned mixed-criticality scheduling on multiprocessor platforms. *Proceedings of the conference on Design, Automation & Test in Europe*. European Design and Automation Association; 2014. 292 p.
34. Bini E, Buttazzo GC. Biasing effects in schedulability measures. *16th Euromicro Conference on Real-Time Systems,*. 2004. p. 196–203.