Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 15 (4): 654-660, 2015
ISSN 1812-5654
© 2015 Asian Network for Scientific Information

ansinet
Asian Network for Scientific Information

RESEARCH ARTICLE OPEN ACCESS

DOI: 10.3923/jas.2015.654.660

Scheduling Algorithm for Multi-Component Digital Systems to Minimize Dynamic Power

Siwar Ben Haj Hassine, Bouraoui Ouni and Hatem Samaali
Laboratory Electronic and Microelectronic, Faculty of Sciences, University of Monastir, Monastir, 5019, Tunisia

ARTICLE INFO
Article History:
Received: October 29, 2014
Accepted: January 13, 2015

Corresponding Author:
Siwar Ben Haj Hassine,
National Engineering School of Sousse,
Sousse Erriadh, 4023, Tunisia
Tel: 0021673369500
Fax: 0021673369506

ABSTRACT

As embedded systems have become wide reaching technology the need for better performance has been persisting. For that researchers have made great effort to come up with new techniques in order to obtain a better performing system that consumes less power. From here, it comes the idea of developing a new scheduling algorithm that manages to achieve components, scheduling in order to reduce dynamic power consumption. The main purpose of that algorithm is to increase the latency of some components whenever it is possible without negatively influencing the dependency constraint. One of the main solutions to increase the latency of components is through decreasing frequency of their clocks. This decrease of the clock frequency brings about reduction in power consumption. To prove its efficiency, the new proposed algorithm has been tested at both levels: Simulation and physical one. In fact, design results have shown significant reduction in dynamic power.

Keywords: Scheduling algorithm, power consumption, electronic system, increase of latency, decrease of frequency

INTRODUCTION

As portable and embedded systems have become the worldwide leading technologies and the need for better performance has been persisting, the creation of a new efficient type of digital circuits has become a critical issue. Because low power dissipation is one of the most important parameters that improve the capabilities and the performance of these systems, the need of new ways to preserve power has become a vital necessity. Designers have to come up with many solutions that preserve power. In fact, some research groups have focused on the designing of batteries with extended life cycles (http://www.lbl.gov/) or the creation of new low power architectures in terms of materiel (i.e., silicon) (http://www.silabs.com/Pages/default.aspx). Unfortunately, several research groups don't have the necessary resources to both design batteries and lower power architecture. To overcome this dilemma, there exist other methods such as design technique that is based on scheduling between the components of a target system to minimize its dynamic power. From here comes the idea of developing a new algorithm that manages to bring us a less consuming system. That algorithm meant to increase the latency of some components without influencing the dependency constraint. There exist different ways to increase the latency of a component, one of them is the decrease of frequency value of its clock. It is widely acknowledged that a decrease in frequency is followed by a respective decrease in dynamic power. So, given a multi-component system, if any possibility of increasing component latency occurs then the proposed technique will definitely provide a gain in power. For instance, given a multi-component system with n components S (C1, C2...Ci...Cn) that are running with a main master clock; assuming that they use the same frequency F1 which is the frequency of the master clock, the dynamic power of a multi-component system before and after the scheduling will be as follows:

Knowing that, the dynamic power Pd of a system can be written as follows (Wang *et al.*, 2013):

$$\text{Pd}= \text{A}\times\text{C}\times\text{V}^2\times\text{F} \tag{1}$$

where, A concerns the percentage of active logic gates; that are charged dynamically, C represents the total capacitance load, V is the supply voltage and F refers to the execution frequency. Since all components are using the master clock, the dynamic power before scheduling is as follows:

$$Pd_{total\_before} = \Sigma A(i)\times C(i)\times V^2(i)\times F1 \quad (2)$$

For a simplification reason we presume that:

$$A(i)\times C(i)\times V^2(i) = \alpha \quad (3)$$

Using Eq. 2 and 3 we obtain:

$$Pd_{total\_before} = n\alpha F1 \quad (4)$$

Assuming that after scheduling at least one component latency can be increased, the dynamic power after scheduling has been calculated by the following way. As mentioned before one solution to increase the latency of a component is to decrease its clock frequency to be F2. F2 equals φ F1 where φ is a real <1

$$Pd_{total\_after} = (n-1)\ \alpha F1+\alpha F2 = (n-1)\ \alpha F1+\alpha\varphi F1 = \alpha\ F1\ ((n+\varphi)-1) \quad (5)$$

Based on Eq. 4 and 5, the gain in power:

$$G = 100\times\frac{Pd_{total\_before} - Pd_{total\_after}}{Pd_{total\_before}} = 100\times(1-\varphi)/n \quad (6)$$

Since φ<1, then G>0.

Thus, the purpose of this study is to propose a scheduling algorithm for multi-component system that aims to optimize its dynamic power through the increase of the latency of some components whenever it is possible.

MATERIALS AND METHODS

As mentioned previously, the aim of the study is to come up with a new scheduling technique that manages to optimize the dynamic power of a global circuit which contains many components, through the decrease of clock frequency of some components whenever it is possible while ensuring that such a decrease does not influence the dependency constraint. Figure 1 summarizes and clarifies the proposed approach.

Proposed algorithm: The input of the proposed algorithm is a system S = (C_1, C_2, C3, C4... Cn) where, C1, C2, C3, C4 and Cn present its components that run with a master clock having a frequency value equal to F1. Each block is characterized by its latency in terms of cycle numbers. The algorithm is divided into 4 steps:

Step 1: In this step the global system has been transformed into a graph with nodes represented by components and edges referring to the dependencies between them

Step 2: In this step, the as soon and as late scheduling has been applied on the graph above, respectively

Step 3: The start time $T_{startmin}$ and $T_{startmax}$ of each node has been determinate according to the as soon and to the as late scheduling, respectively

Step 4: In this step, for each node (ni) if $T_{startmax}(ni)>T_{startmin}(ni)$ then it will be attributed to (ni) a new clock running with a frequency Fi such as:

$$Fi = \frac{T_{startmin} + L(ni)}{T_{startmax} + L(ni)}\times F1 \quad (7)$$

where, L(ni) is the latency of the node (ni) in terms of number of cycle.

Pseudo-code:

```
Begin
•   Build a graph G(v,e) from S(c1, c2, ci....)
•   Apply the as soon and the as late scheduling on G(v, e)
•   Compute the start time $T_{startmin}$ (ni) of each node (ni) in G according to as
    soon as possible scheduling
•   Compute the start time $T_{startmax}$ (ni) of each node (ni) in G according to
    as late as possible scheduling
•   For all node n(i)
•   If ($T_{startmax}$ (ni)>$T_{startmin}$ (ni)) then

        $Fi = \frac{T_{startmin}(ni)+L(ni)}{T_{startmax}(ni)+L(ni)}\times F1//Fi$

    is the new frequency and
F1 is the previous one
L(ni) is the latency of the node
(ni) in term of number of cycle
End for
•   End begin
```

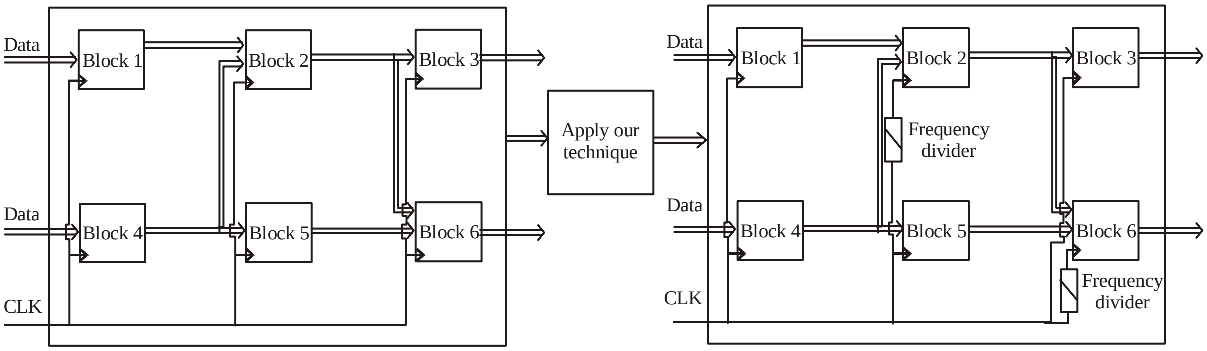


Fig. 1: Proposed approach

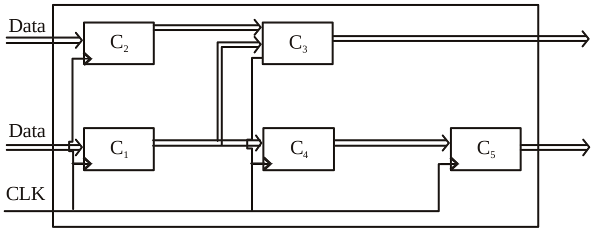


Fig. 2: Input system

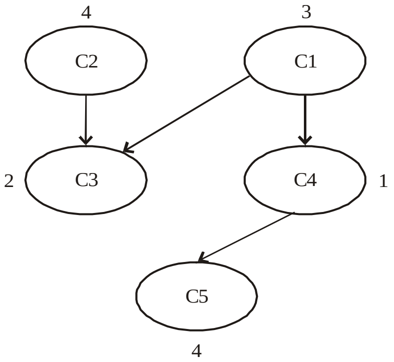


Fig. 3: Graph of input system

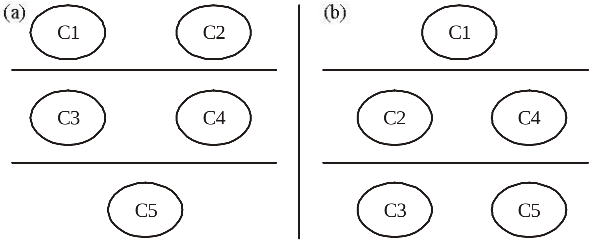


Fig. 4(a-b): Scheduling (a) As soon as possible and (b) As late as possible

Table 1: Components latencies

Components	Latency
C1	3
C2	4
C3	2
C4	1
C5	4

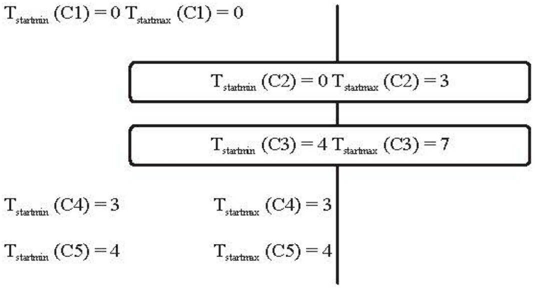
Illustrative example: To further clarify the algorithm, it has been applied on the below system shown in Fig. 2. Assuming that the master clock has a frequency equal to F1.

Table 1 represents the latencies in terms of number of cycles of each component.

First, the system, shows in Fig. 2 has been transformed to a graph as represents Fig. 3.

Second, the as soon and as late scheduling algorithms have been applied as shows Fig. 4.

Third, the $T_{startmin}$ and $T_{startmax}$ of each node have been calculated to obtain:


$T_{startmin}$ (C1) = 0 $T_{startmax}$ (C1) = 0
$T_{startmin}$ (C2) = 0 $T_{startmax}$ (C2) = 3
$T_{startmin}$ (C3) = 4 $T_{startmax}$ (C3) = 7
$T_{startmin}$ (C4) = 3
$T_{startmax}$ (C4) = 3
$T_{startmin}$ (C5) = 4
$T_{startmax}$ (C5) = 4

Fourth, $T_{startmax}$(C2)>$T_{startmin}$(C2) and $T_{startmax}$(C3)>$T_{startmin}$(C3), for that, the Eq. 7 will be applied to calculate the new clocks frequencies of both nodes C2 and C3, respectively:

$$F2= (0+4/3+4)\times F1 = (4/7)\ F1$$

$$F3= (4+2/7+2)\times F1 = (2/3)\ F1$$

According to the results above, the clock frequency of C2 and C3 have been decreased to (4/7)F1 and (2/3)F1, respectively. That decrease will be automatically followed by an increase in clock period which will lead to a simultaneous decrease in power consumption.

To prove the efficiency of the proposed technique concretely, the gain of dynamic power before and after the secluding algorithm has been calculated.

Dynamic power before secluding: Since all components are running with the master clock, dynamic power is as follows:

$$Pd_{total} = n\alpha F1 = 5\ \alpha F1 \quad (8)$$

Dynamic power after secluding: After scheduling only C1, C4 and C5 still run with the master clock. However, C2 and C3 use a new clocks frequencies which are F2 = (4/7) F1, F3 = (1/3) F1, respectively. Thus, the dynamic power is:

$$\begin{aligned} Pd_{total} &= (5\text{-}2)\ \alpha F1+2/3\ \alpha F1+(4/7)\ F1 \\ &=3\ \alpha\ F1+(4/7)\ \alpha\times F1+(2/3)\ \alpha\times F1 = 4.2\alpha F1 \end{aligned} \quad (9)$$

Based on Eq. 8 and 9 we deduce that the proposed approach managed to bring the consumed power into a lower level attaining a decrease of 16% compared to the classical technique. This interesting gain proves the effectiveness of the new technique compared to the other one in terms of dynamic power.

Experiment: In this part, the proposed algorithm has been applied on an application that converts RGB color to HMMD in order to carry out a comparative study about dynamic power dissipating by the system with applying the proposed technique and compared to a system in which we don’t use it. The study will cover three main fields which are power, response time and resources. It is meant to prove the efficacy of the presented technique and to illustrate the gain in dynamic power. For that, the following tools consisting of nexys 2 based on Spartan 3E the electronic board, the synthesis tool Xilinx ISE (Version 12.3) and the simulation tool ModelSim (Version 6.5) have been used. The application target is to convert an RGB to HMMD color. In fact, the color structure descriptor which is widely used in the area of image processing and compression, must be defined using the HMMD color space. Hence, the color pixels of any other color spaces like RGB must be converted to the HMMD color space. From here comes the necessity of the suggested application which convert the RGB color space to HMMD color space. Figure 5 represents the block diagram.

Table 2: Frequencies of each component after and before scheduling

	Frequency	
Parameters	Before scheduling	After scheduling
Comparator	F1	F1
Subtractor	F1	F1
Multiplier	F1	F1
Divider	F1	F1
Adder	F1	F1
Adder 2	F1	0.5F1
Divider 2	F1	0.2F1
Synchronous block	F1	0.8F1

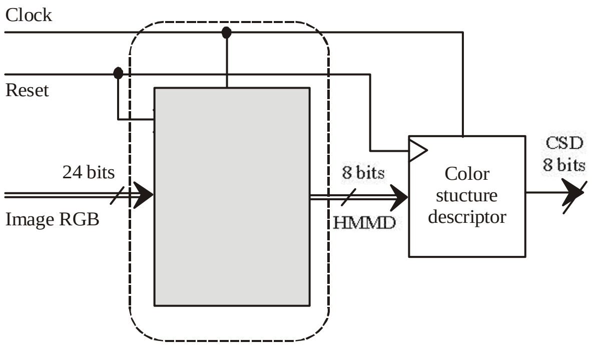


Fig. 5: Block diagram presenting conversion of RGB color space to HMMD color space

In this part, we are interested in the red block which is the converter block. That block consists of eight blocks as shown in Fig. 6. It consists of many inputs such as "St" which is meant to activate the application, when it is valid the application starts its treatment. The end of the treatment is presented when the "End" input is valid.

Frequencies before and after scheduling: After applying the new scheduling algorithm on the Fig. 6, the frequencies of Adder2, Devider2 and synchronous block have been changed while the other components have kept the same old frequency. Table 2 indicates the old and new frequencies of each component.

RESULTS

Design results at simulation level: In this part, the simulation results have been compared in terms of functionality, response time and power dissipating.

Comparison in terms of functionality: Figure 7 and 8 represents the simulation results of the system before and after scheduling, respectively.

According to the Fig. 7 and 8, the same functionality of the target system has been kept.

Comparison in terms of response time: The response time is the time elapsed from the moment where the application starts its execution until it gives final response. Practically speaking, from the moment where St is active until 'End' is valid, the Fig. 9 and 10 represents the response time of the application before and after scheduling, respectively.

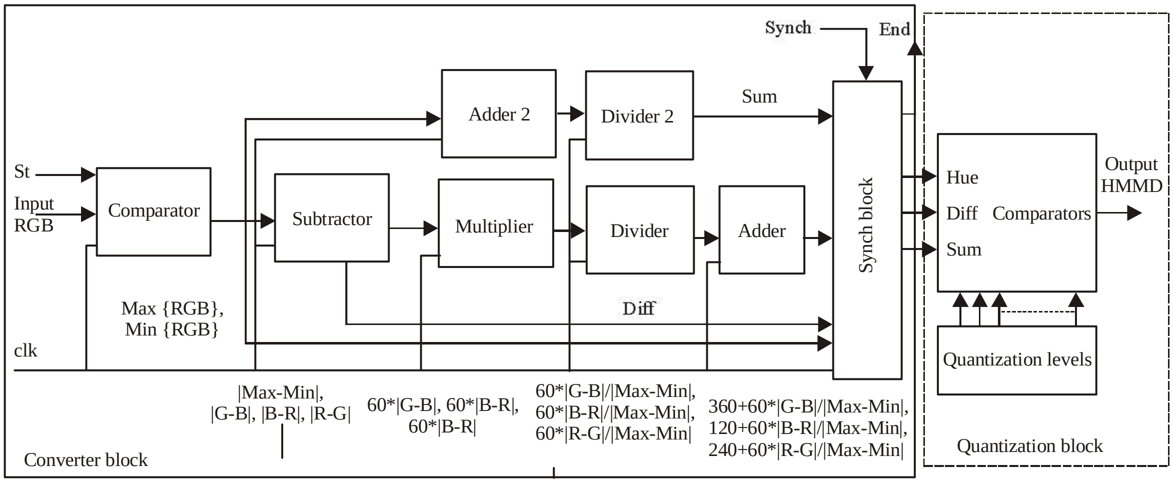


Fig. 6: Target application

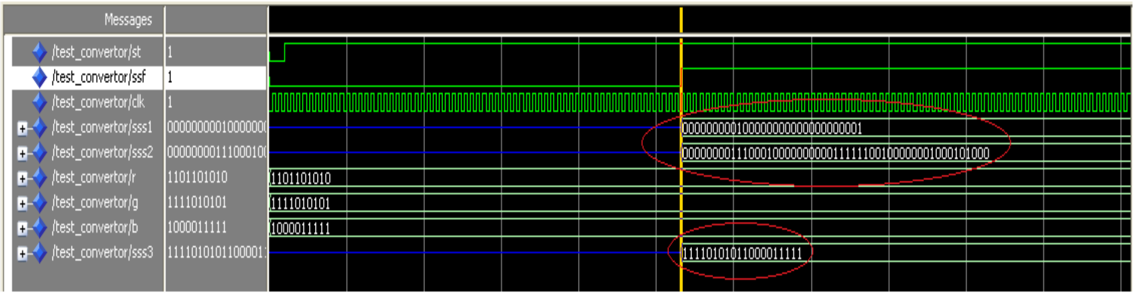


Fig. 7: Outputs of the system before scheduling

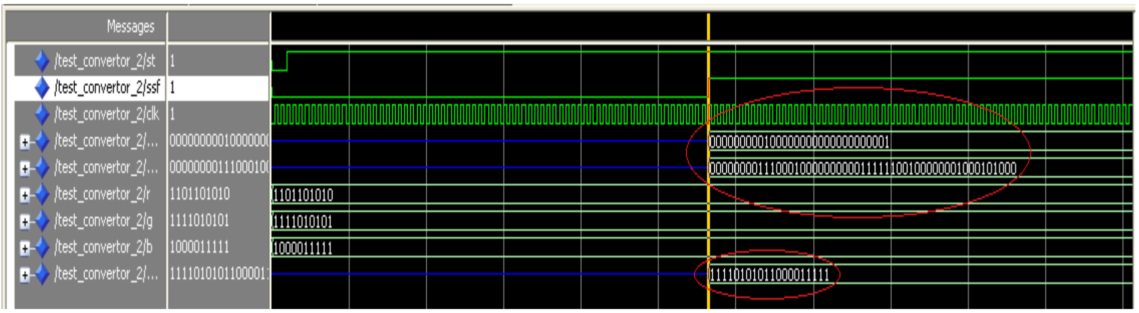


Fig. 8: Outputs of the system after scheduling

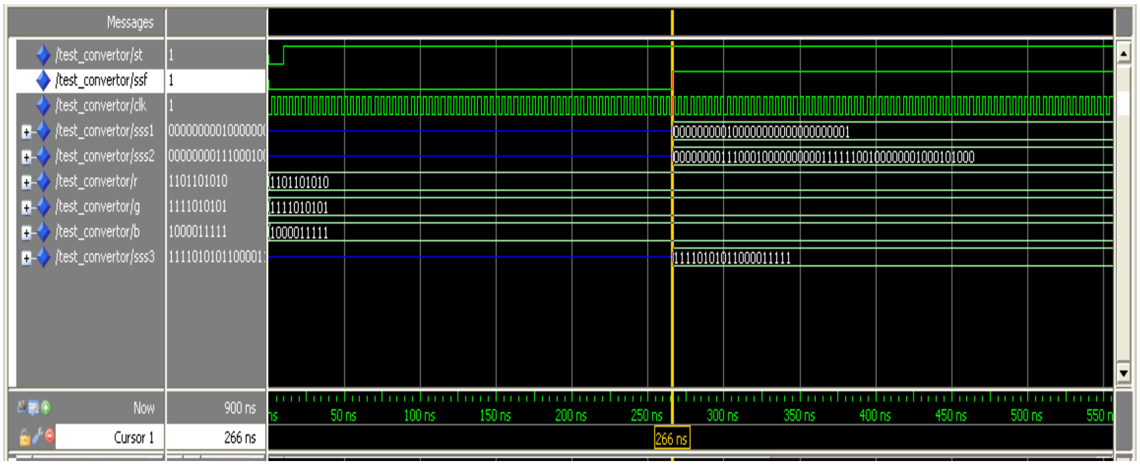


Fig. 9: Before scheduling

According to Fig. 9 and 10 we figure out that while the response time of the target application before scheduling is 266 nsec, the given response after scheduling is 282 nsec. Unfortunately, the proposed technique has produced a loss of 6% in response time but that loss may seem to be unimportant if it makes us to meet power optimization.

Comparison in terms of dynamic power: Dynamic power dissipated before scheduling:

$$Pd_{total\_before} = n\alpha F1 = 8\ \alpha F1 \qquad (10)$$

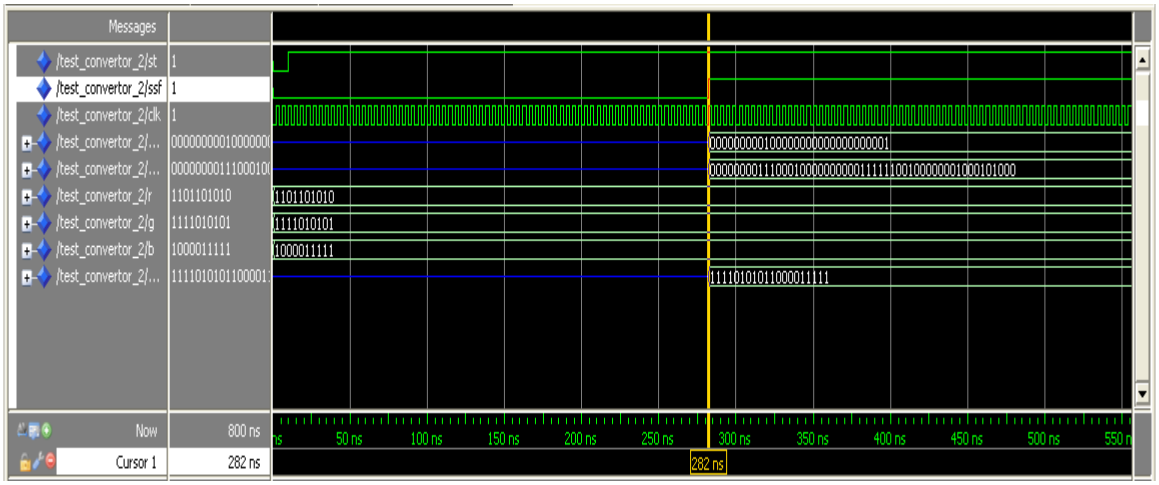
Dynamic power dissipated after scheduling:

$$Pd_{total\_after} = (8\text{-}3)\ \alpha F1 + 0.5\ \alpha F1 + 0.2\alpha F1 + 0.8\ \alpha F1 = 6.5\ \alpha F1 \qquad (11)$$

Gain in dynamic power:

$$G = \frac{Pd_{total\_before} - Pd_{total\_after}}{Pd_{total\_before}} \times 100$$

$$G = 18.75\% \qquad (12)$$



```
Messages
/test_convertor_2/st    1
/test_convertor_2/ssf   1
/test_convertor_2/clk   1
/test_convertor_2/...   00000000010000000   00000000010000000000000000001
/test_convertor_2/...   0000000011100010(   000000001110001000000000011111100100000001000101000
/test_convertor_2/r     1101101010          1101101010
/test_convertor_2/g     1111010101          1111010101
/test_convertor_2/b     1000011111          1000011111
/test_convertor_2/...   11110101011000011   11110101011000011111
Now                     800 ns
Cursor 1                282 ns              282 ns
```

Fig. 10: After scheduling

Table 3: Design result

Parameters	Response time	Power
Before scheduling	266 (nsec)	8 αF1
After scheduling	282 (nsec)	6.5 αF1
Loss (%)	6	-
Gain (%)	-	18.75

Table 4: Power design result

Parameters	Dynamic power (MW)
Before scheduling	4.13
After scheduling	3.14
Gain (%)	23.97

Table 5: Resources design results

Parameters	No. of slices	No. of slice flip flops
Before scheduling	629.00	563.00
After scheduling	725.00	660.00
Loss (%)	15.26	17.22

Based on Eq. 10-12, the described technique provides a low power consuming system. In fact it offers a gain of 18.75% in dynamic power compared to a system on which we have not applied our technique. To recapitalize Table 3 has been presented.

Design results at physical level: To further approve the efficiency of the proposed technique, a physical comparison in terms of power and resources between a classical systems on which we have not applied our technique and another one using our technique, has been applied.

Power design results: Table 4 summarizes the power consumed by the two systems, before and after scheduling, notably the gain obtained using the proposed algorithm. Noting that, in this application the execution frequency equal to 50 MHz.

A further indicator of the efficacy of the proposed technique is the fact that dissipated power before scheduling is as high as 4.13 MW whereas, in the system on which we have used the proposed technique there is a decrease with the dissipated power reaching a level of only 3.14 MW. In other word, our technique provides an interesting gain of dynamic power which is 23.79%. Therefore, researchers and designers looking for less power consumption should opt for this more efficient technique.

Resources design results: It is evident that to optimize important parameters like power, other parameters such as area will be influenced. Table 5 presents the obtained loss.

In Table 5, there is a loss of 15.26% in number of slices and 17.22% in number of slice Flip Flop. These lost values become insignificant compared to the gain obtained in power consumption.

DISCUSSION

To preserve power, there exist many solutions like designing batteries with longer life cycles or coming up with new architectures. As most of research groups don't have the necessary sources to do that they have chosen two ways in order to attain their aim which is power minimization. One of the most commonly used techniques is to use scheduling algorithms.

In fact, Fan *et al*. (2013) have proposed a dual-threshold voltage assignment algorithm in order to reduce the static power of the circuit. This algorithm that uses static timing analysis gets the timing information of each node by double traverse. In addition, it assigns the threshold voltage of all nodes through primary optimization and accurate one. In other words, this algorithm is meant to replace maximum number of high-Vt gates from low-Vt and reduce the static power as high as possible. As demands on cost and performance in modern digital system design have been hugely increasing, Cong *et al*.

(2011) have proposed a new algorithm that handles general array access beyond affine array references. That algorithm offers a technique of an automatic memory partitioning in order to improve throughput and minimize energy consumption of pipelined loop kernels for platform requirements and given throughput constraints. So, the scheduling algorithm is a widely used method that provides a gain in power. For that, many other authors have opted for that technique. For instance, Babu and Shunmugalatha (2013) have presented a self adaptive firefly algorithm in order to solve the economic load dispatch problem. This heuristic numeric optimization algorithm which is inspired by the behavior of fireflies has appeared to be a reliable and a robust technique. Additionally, Wang *et al.* (2010) have developed a scheduling heuristics and presented an application experience that manages to reduce power consumption of parallel tasks in a cluster with the Dynamic Voltage Frequency Scaling technique. Practically speaking, they have studied the slack time for non-critical jobs then extended their execution time to reduce energy consumption without influencing the whole task's execution time. Besides, Kuhn and Mashaly (2013) have used the self adapting algorithms as well to minimize power consumption. In fact, they have introduced a novel methodology that optimizes the system parameters all along with respecting the thresholds for the indicate delay. In addition, Choudhury and Pradhan (2012) who have used the genetic algorithm have proposed a probabilistic power model of power-gated design of FSM to solve the dilemma of encoding and bi-partitioning in order to optimize power consumption. So using scheduling algorithms, certain power minimization can be attained. One may, nevertheless, wonder about the advantage of our algorithm compared to other existing algorithms. Actually, each work has its own benefits; some works have focused only on power optimization while other works have focused on the optimization of the response time or the area constraint. Our algorithm is characterized by its simplicity and efficacy. In other word, it is unsophisticated and easy to use and guaranty a less consuming system.

CONCLUSION

Power optimization has become a high priority in electronic world. For that reason, researchers and designers have augmented their efforts to find new solution in order to reduce power. One of the most important ways is the use of scheduling algorithms. For that purpose, we have proposed a new scheduling algorithm that achieves components' scheduling in order to bring a reduction in power consumption. In fact, power design results have proved the efficiency of the novel technique which should be used by designers in order to obtain less consuming systems.

REFERENCES

Babu, B.S. and A. Shunmugalatha, 2013. Reducing power losses in power system by using self adaptive firefly algorithm. Proceedings of the 4th International Conference on Swarm, Evolutionary and Memetic Computing, December 19-21, 2013, Chennai, India, pp: 122-132.

Choudhury, P. and S.N. Pradhan, 2012. An approach for low power design of power gated finite state machines considering partitioning and state encoding together. J. Low Power Electron., 8: 452-463.

Cong, J., W. Jiang, B. Liu and Y. Zou, 2011. Automatic memory partitioning and scheduling for throughput and power optimization. ACM Trans. Des. Autom. Electron. Syst., Vol. 16. 10.1145/1929943.1929947.

Fan, R., Z. Dandan and Y. Xiaolang, 2013. An algorithm for reducing leakage power based on dual-threshold voltage technique. Proceedings of the 4th International Conference on Digital Manufacturing and Automation, June 29-30, 2013, Qingdao, pp: 132-134.

Kuhn, P.J. and M. Mashaly, 2013. Performance of self-adapting power-saving algorithms for ICT systems. Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, May 27-31, 2013, Ghent, pp: 720-723.

Wang, L., G. von Laszewski, J. Dayal and F. Wang, 2010. Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS. Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, May 17-20, 2010, Melbourne, Australia, pp: 368-377.

Wang, L., S.U. Khan, D. Chen, J. Kolodziej, R. Ranjan, C.Z. Xu and A. Zomaya, 2013. Energy-aware parallel task scheduling in a cluster. Future Gener. Comput. Syst., 29: 1661-1670.