Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1191492, 11 pages
https://doi.org/10.1155/2022/1191492

Research Article

Local Binary Patterns Based on Neighbor-Center Difference Image for Color Texture Classification with Machine Learning Techniques

Himangi Verma,[1] Aditya Vidyarthi,[2] Abhijit V. Chitre,[3] Kirti H. Wanjale,[4] M. Anusha,[5] Ali Majrashi,[6] and Simon Karanja Hinga[7]

[1]*Department of Computer Science and Engineering, Guru Jambheswar University of Science and Technology, Hisar, Haryana, India*
[2]*Department of Information Technology, ITM Gwalior, Madhya Pradesh, India*
[3]*Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India*
[4]*Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India*
[5]*Associate Professor, Department of Computer Science & Engineering, Koneru Lakshmaiah, Education Foundation, Green Fields, Vaddeswaram, Andhra Pradesh, India*
[6]*Department of Biology, College of Science, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*
[7]*Department of Electrical and Electronic Engineering, Technical University of Mombasa, Mombasa, Kenya*

Correspondence should be addressed to Simon Karanja Hinga; kahinga@tum.ac.ke

Received 9 May 2022; Revised 23 June 2022; Accepted 29 June 2022; Published 8 August 2022

Academic Editor: Mohammad Farukh Hashmi

This is a topic that receives a lot of interest since many applications of computer vision focus on the detection of objects in visually appealing environments. Information about an object's appearance and information regarding the object's motion are both used as crucial signals in the process of identifying and recognising any given item. This information is used to characterise and recognise the item. The identification of objects based solely on their outward appearance has been the subject of a substantial amount of research. However, motion information in the recognition task has received only a marginal amount of attention, despite the fact that motion plays an essential role in the process of recognition. In order to analyze a moving picture in a way that is both fast and accurate, it is required to make use of motion information in conjunction with surface appearance in a strategy that has been designed. Dynamic texture is a kind of visual phenomenon that may be characterised as a type of visual phenomenon that shows spatially repeated features as well as some stationary properties during the course of time by using methodologies that are associated with machine learning. The design of modern VLSI systems takes into consideration a larger chip density, which results in a processor architecture with several cores that are capable of performing a wide range of functions (multicore processor architecture). It is becoming more challenging to run such complicated systems without the use of electric power. In order to increase the effectiveness of power optimization strategies while maintaining system performance for text data extraction, it has been developed and put into practice power optimization strategies that are based on scheduling algorithms. Over the last twenty years, texture analysis has been an increasingly busy and profitable field of study. Today, texture interpretation plays a vital role in various activities ranging from remote sensing to medical picture analysis. The absence of tools to newline analyze the many properties of texture pictures was the primary challenge faced by the texture analysis approach. Texture analysis may be roughly categorised as texture classification, texture segmentation, texture synthesis, and texture synthesis. Texture categorization is useful in numerous applications, such as the retrieval of picture databases, industrial agriculture applications, and biomedical applications. Texture categorization relies on three distinct methods, namely, statistical, spectral, and structural methods. Statistical methods are based on the statistical characteristics of the image's grey level. Features are collected using second order statistical order, autocorrelation function, and grey level co-occurrence matrix function.

1. Introduction

The categorization of textures is an essential part of the engineering and scientific study sectors. The retrieval of images from databases, as well as applications in industry, agriculture, and medicine, are all possible uses for it. There are two distinct processes that are involved in the categorization of texture images: the training phase and the testing phase or recognition phase. During the training phase, a collection of known texture pictures are trained using a feature extraction approach and saved in the library or database. This phase is also known as the learning phase. During the recognition phase, an unknown sample picture is evaluated by using the same feature extraction procedure as before, and the resulting values are compared to those of previously saved characteristics in the database. The unidentified sample may be accurately categorised using the classification method, but there is also a chance that it will be misclassified. The categorization of textures is a basic topic in computer vision that has many different applications. Despite the extensive research that has been done in the field, there are still no clear answers to the questions of how to define a reliable distance or similarity measure between textures and how to characterise textures by using derived features. Both of these questions are considered to be fundamental issues in the field of texture classification. Textural characteristics need to be invariant to picture fluctuations while yet being sensitive to the fundamental spatial patterns that characterise textures. This is necessary due to the fact that photos of the same underlying texture might differ greatly from one another. Texture features are often given based on assumptions for the sake of mathematical convenience. This is done since there is no evident characteristic that is shared by all texture pictures.

It is very important for the success of the texture classification that the appropriate technique of texture analysis be used for feature extraction. However, the measure that is used in the process of contrasting the feature vectors is undeniably crucial. Extracting texture characteristics may be done either directly from the picture statistics, such as via a cooccurrence matrix, or indirectly from the spatial frequency domain using a number of different approaches that have been presented.

Image pixel is taken as input parameters of Markov Random Fields, Gabor multichannel features, fractal-8-based features, and co-occurrence features. These wavelet transform representations included orthogonal and biorthogonal, tree-structured wavelet transform, and Gabor wavelet transform (GWT). The majority of these earlier research have concentrated their attention on the features, but have not paid much attention to the metric or attempted to simulate the noise distribution.

It was determined that the cooccurrence characteristics were the most effective for texture categorization. A research that was conducted by Conners and Harlow provides evidence for this assertion. Improved classification rates are the result of using wavelet decomposition to derive Haralick features, work. Haralick characteristics have been taken into consideration for use in texture classification who used wavelet packet decomposition. The characteristics of a surface's texture are determined by taking into account the intensity and contextual information collected from binary pictures. The intensity and binary pictures are used as inputs for the computation of the conditional cooccurrence histograms. Fixed thresholds have been applied throughout this process in order to produce binary pictures.

For the purpose of texture classification, Jen-Wei Lee et. al. [1] devised a feature extraction approach that makes use of wavelet-decomposed pictures of both an image and its complimentary image. The characterisation lays forth the parameters for the features that are created using the many permutations of subband pictures.

The findings of the experiments suggest that combining detail subbands with approximation subbands helps to enhance classification rates while also reducing the amount of processing effort required [2].

For the purpose of texture classification, they made use of spectral histogram as a feature statistic. The spectral histogram is made up of marginal distributions of responses from a bank of filters. It conveys indirectly, via the filtering step, the local structure of pictures and, through the histogram stage, the global appearance. This technique was based on local binary patterns and nonparametric discriminating of sample and prototype distributions [3].

Over 96 percent of a test set of 2806 photos collected from all 61 texture classes with unknown position and lighting were successfully classified using the texture classification method.

The inappropriate use of power is responsible for a considerable portion of the flaws that are present in technical systems. In this particular scenario, direct current field supply (DVFS), which is a very effective method for delivering the necessary amount of energy to the cores of the device, is used. However, when it is put into practice within the appropriate circuit constraint, it does not live up to expectations [4]. The C-MOS contributes to a reduction in the amount of power that is lost by the device while it is operating in the static mode. On the other hand, the C-MOS circuit poses a significant obstacle due to the presence of an inductor that acts as a filtering network in it. The power performance of the processor will be determined by whether you choose an LDO or a FIVR as the power regulator for your on-chip CPU for the power delivery unit. This choice will either improve or degrade the power performance of the processor [5].

In order to cut down on power consumption to an absolute minimum, multipliers provide a wide variety of different alternatives. When it comes to arithmetic circuits, multipliers are essential components since they determine the whole power range that may be used by the system. This strategy has many benefits, including a lower level of latency, a smaller physical footprint, and a lower level of power usage [6]. A number of academic academics have attempted and are now attempting to build a circuit that delivers regularity of architecture while taking up less space, or even a combination of the two in a single multiplier for compact VLSI implementation [7], among other things [5]. It is possible to do this by dividing the whole memory into smaller pieces in order to cut down on the amount of power that is used overall.

In order to address and ultimately triumph over these obstacles, it has been suggested that a technique for scheduling control-dominated design during high-level synthesis be developed [8]. This would allow for the proper handling of the situation. The decisions made by the user influence the control flow of the system, which may include conditional branching and control loops. This is done in response to the user's input [9]. The functioning of the FSM is exploited in order to build a scheduling schedule, which is then put into action. Aside from that, it is able to cope well with challenges involving a variety of difficult constraints [10]. The establishment of pipelines has an impact, in addition, on the use of electrical power [11]. As a result of this, depth may be modified in conjunction with a suitable fluctuation in supply voltage, which makes it possible for the amount of power used to remain the same while the performance of the CPU decreases, as seen in Figure 1.

In the most recent ten years, multiprocessor system-on-chips (also known as MPSOCs) have become an important kind of very large-scale integration (VLSI) (very large-scale integration). This technology is used to a significant degree in a variety of applications, including networking, communications, signal processing, and multimedia, to name just a few of the many [12]. This technology can be found just about everywhere. The multiprocessor system on a chip, often known as the MPSOC, is a kind of system on a chip that has several processors and is frequently used for embedded applications in personal computers. When implemented in a system, the use of a million-bit multiplier results in a significant cutback in the quantity of power that is consumed by the central processing unit of the system [13]. The technique is implemented in each individual core of the central processing unit (CPU) in order to obtain the lowest feasible power consumption per core. As a consequence of this, a queue is created for all of the presently active data, and it is sent to the core, which has an urgent need for power in order to operate. The amount of electricity that multicore computers need from their power supplies has been reduced in a variety of different ways. This category will include any and all data that was used in the process of selecting candidates [7]. On the other hand, the latency between logic gates and the power consumption of the CPU both rise in proportion to the number of cores that are present in the CPU. This internal improvement resulted in a large decrease in the power consumption of the circuit, as evidenced by the test results, which showed that the multiplier design that was recommended for low power and high-speed applications was put through its paces [5]. The power for the idle core, which is not being used at the moment, is provided by the active core. The lack of a filtering network, which is included in this system but was not included in previous systems, was an additional shortcoming of those systems. This system has a filtering network, which was not present in earlier systems but is present in this system. This is another benefit of this system that was not available in earlier systems [14].

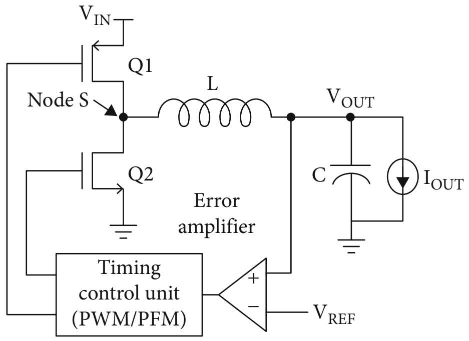


Figure 1: Simplified schematic of a step-down switching for text extraction.

2. Research on Power Optimization of MPSOCs

The theory of nonlinear image processing that is known as mathematical morphology is derived from the field of mathematical morphology. It was possible to examine the forms and structure of a picture by using structural components. This was accomplished by breaking the image matrix into smaller patterns and then analyzing the patterns that were produced as a consequence of this process [15]. The output picture of the morphological operation will have the same dimensions as the structuring element that was used in the morphological operation as a consequence of the morphological operation. As a result, the structural component has a significant impact on the morphological processes that are carried out by the organism. It is possible, with the help of morphological methods, to calculate the neighbourhood of each pixel in an input image and then to use this information to determine the value of each pixel in an output picture. This process may be carried out in two steps. The neighbourhood's size, shape, and composition are among the most important factors that have an effect on the outcome. Erosion and dilation are two of the most fundamental morphological processes that occur on the planet. Both of these processes are responsible for the formation of landforms. In particular, it is essential to emphasize the primary distinction between erosion and dilation, which is that whereas dilation adds a pixel to the image border, erosion removes that pixel from the picture border. This is the most important distinction between the two processes. In terms of pixels, the amount of pixels that are added to or subtracted from the final picture is determined by the kind of structural element that is utilised. The structural element is responsible for a variety of tasks, including determining the area around the pixel that is now being processed and recognising the properties of the pixel that is currently being processed. When analyzing binary and grayscale pictures, flat structuring components were employed rather than curved structuring parts since flat components are easier to manipulate. This value acts as the origin of the structuring element, and it is this value that decides which pixels are to be acted on as well as the calculation of the neighbourhood that surrounds the structuring element. The value may be found at the center of the structuring element. Each component of the architecture has a pair of pixel values associated with it, one of which is true and the other of which is false. Those pixels with genuine pixel values were used in the method for calculating, while those pixels with false pixel values were not utilised in the procedure for calculating. Those pixels with true pixel values were utilised in the procedure for calculating. If the technique's minimum and maximum parameters are set appropriately, it is feasible to produce a picture without the appearance of borders when using the dilation and erosion process. This is made possible by carefully selecting those values. When the border of an image does not seem to be consistent with the rest of the picture's area, this is an indication of border effects. The widening of the aperture of the pupillary membrane to have a complete understanding of the process of dilatation, there are a few fundamental ideas that need to be mastered. They are, in point of fact, right. Consider, for instance, the pixels that are located in the image's backdrop that was provided to us. In contrast to the pixels that make up the backdrop, the structuring element is overlaid on top of them in such a way that the point of origin of the structuring element is located above the pixel that is of interest. There is a possibility that one or more of the pixels in the structuring element corresponds to a pixel in the foreground. If this is the case, the value from the foreground is applied to the pixel that is located in the foreground. Because none of the pixels in the vicinity of the background pixel have been converted to the foreground state, the background pixel will continue to stay in the background until it is altered in a deliberate manner. As a result of this, the value of the output pixel in dilation is equal to one less than the total of the values of all of the pixels that are within one pixel's close vicinity of the input pixels. This is one of the consequences of the fact that this occurs. The final picture is duller than the original image, and the image's bright features have been muted to a lesser degree. The final image is darker than the original image. The surface of the planet is subject to the natural process of erosion. At each and every level of the algorithm, the pixels that make up the foreground are taken into account. If the pixel that is being input is set to the foreground and all of its eight neighbours are likewise set to the foreground; then, the pixel that is being input will stay in the forefront until it is either moved or updated in some other way. When a pixel is in the front but at least one of its neighbours is located in the background, it is considered to be part of the background region even if it has been allocated to the forefront. It is not feasible to alter the status of input pixels after they have been designated as belonging to the backdrop. This indicates that the value of an output pixel in erosion is equivalent to the sum of the values of all the pixels located in its immediate vicinity, which is equal to the value of the pixel that was used as input. The process of deleting pixels from the foreground that are no longer needed is referred to as erosion. The created picture is darker than the original image, but it contains more light parts than the original image had. The produced picture is brighter than the original image.

2.1. On-Chip Systems in the Case of Text Extractions. On-chip power delivery solutions have been offered in the form of both the fully integrated for text extraction (FIVR) power supply system and the low-dropout regulator (LDO) power delivery system. A number of studies have been carried out in order to ascertain the effect that various job scheduling strategies have on the power output of the whole system. While running at half-throughput in a hypothetical 256-core CPU with a DVFS assumption for each core, the FIVR-based power supply consumes 20 percent less power than the LDO-based power delivery. The difference in power consumption between FIVR-based and LDO-based power delivery systems, on the other hand, becomes less significant as the number of CPU cores on the system decreases. FIVR stands for field-programmable integrated voltage regulator, and LDO stands for linear direct current regulator. For instance, it was discovered that the FIVR-based design consumes about the same amount of power as the LDO-based design in the event of a 16-core CPU with per-core DVFS capabilities in the context of this scenario [5].

On the left side of the diagram is a generic step-down switching circuit for text extraction, which may be implemented as either an OFF-CHIP buck converter or an ON-CHIP FIVR. There are two MOSFET switches [Q1 and Q2], a filter network consisting of an inductor and a capacitor and a feedback control loop in the circuit. The comparator creates error voltage in order to give precise turn-on and turn-off timings for the upper and lower switching MOSFETs Q1 and Q2, respectively. A pulse width modulated (PWM) or a pulse frequency modulated (PFM) controller is then driven by the erroneous voltage generated. Power is supplied to the semiconductor via the power supply system, which is seen in Figure 2. The power supply system consists of a 16-phase buck regulator that is positioned outside of and on the motherboard, and it is visible in Figure 2. Furthermore, it absorbs energy from a 12 V supply and generates output voltage levels that may be used by subsequent converter stages inside the device in addition to the on-chip regulators (FIVR or LDO) and CPU cores [16], among other things. A single lumped resistor, Rext, is utilised to account for IR noise induced by external wire and package resistances [17] in order to account for IR noise.

By assumption, it made that due to *R*ext, the worst-case power loss is ~5%, which is typical of the current state-of-the-art power delivery networks [5].

According to Figure 3, the average system power with the LDO is less than the average system power with 16 FIVRs due to the per-core DVFS capabilities with the LDO (24 percent reduced power usage at normalized throughput of 0.5) [10].

2.2. The CAD Methodology. These systems, which make use of many on-chip cores, are capable of running a broad range of applications at the same time, resulting in a wide range of power-performance needs. It is necessary in such a setting to have a flexible multicore platform that has the capacity to exert fine grain control, as seen in Figure 4 in order to achieve the steep energy efficiency requirements.

The increased inefficiency of the DVFS may be attributed to two important factors:

The fundamental circuit parameters (such as gate sizes and threshold voltages) of the circuit, which are built for

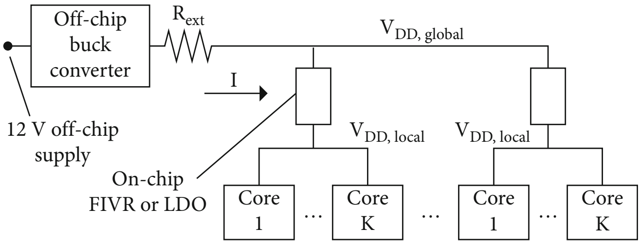


Figure 2: Block diagram of power delivery scheme under consideration.

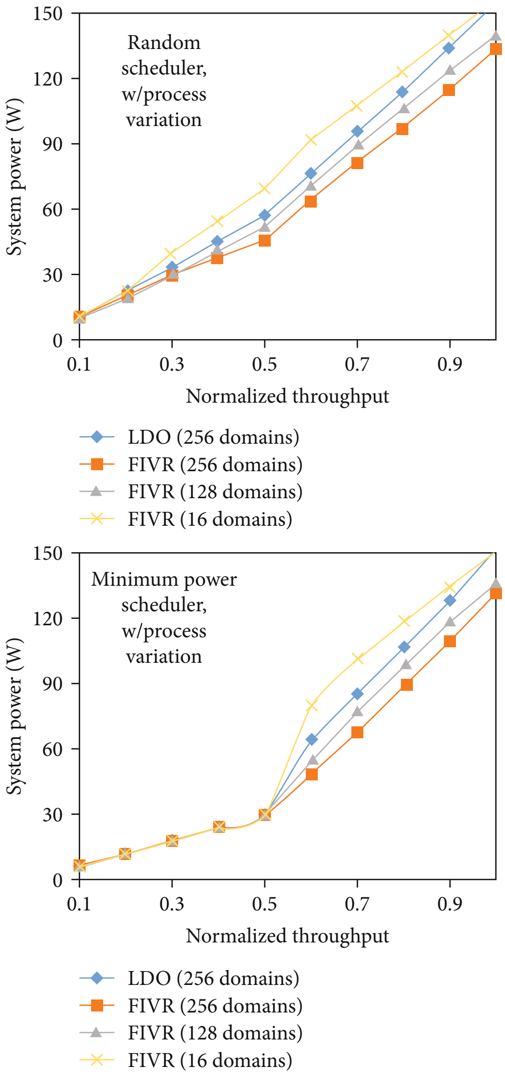


Figure 3: Comparison of FIVR and LDO.

the nominal frequency, are not altered by dynamic modifications, which results in the circuit being power hungry.

In the near future, new technology generations will constrain the supply voltage scaling margins, which is a critical component of the power reductions achieved by DVFS. In order to achieve energy efficiency, it is increasingly important to identify alternate, scalable methods of accomplishing this goal.

It has been proposed to use a system level computer-aided design (CAD) approach to design architecturally homogeneous, power-performance heterogeneous multicore systems (AHPH). This is in contrast to the conventional design style, in which all cores are designed to be power-performance optimal at the nominal frequency, which has been used previously [18].

Figures 4 and 5 present detailed internal circuit characteristics of ground optimal designs for 90 and 45 nm nodes at each of the six operating points were developed.

Voltage is for operation.

It can be observed that the internal circuit characteristics of a single functional unit may alter significantly depending on how the unit is optimised at various performance points. When a circuit is constructed for maximum performance, the circuit takes up the most space since the sizes of the various gates in the circuit are increased [16]. As predicted, we observe the greatest concentration of LVT devices at the highest performance level; nevertheless, at lower performance levels, the power consumption decreases dramatically due to the following factors:

(1) Smaller area (lower gate capacitance)

(2) Lower leakage owing to the decrease of LVT devices

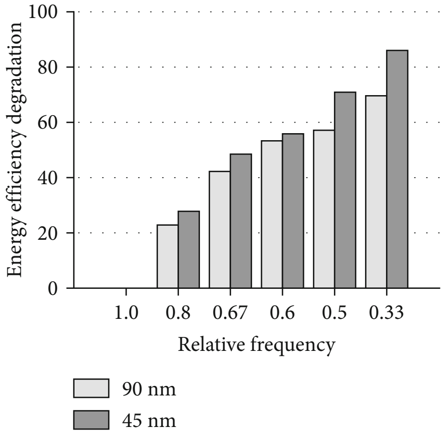
(3) Lower cost due to reduced LVT devices

(4) Reduce the voltage of the power source

In this study, we used a rigorous experimental approach that combined typical cell library-based CAD flows with architectural whole system simulation. We found that we could obtain an 11 percent–22 percent improvement over state-of-the-art DTM methodologies.

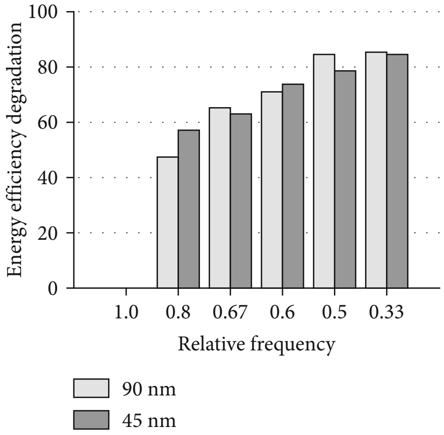
2.2.1. Architecture of the DIP Split. Through the use of NTT Ram technology, an efficient design that divides the multiplier bit into three FIFO architectures has been suggested. In addition to the synchronizer, state control module, and three-state FIFO buffer module, the architecture includes the following components: the synchronizer module is responsible for maintaining synchronization between the NTT RAM and the state controller arbiter. The control signal for the FIFO modules is sent by the state controller module. The FIFO module is used to divide a million bit multiplier into eight-bit multipliers, resulting in a total of 256 values. Now that we have three FIFO modules, we can reach 256∗256∗256 bits of value, which is about one million bits. On the basis of this technology, we must achieve high speed while also minimising power consumption and eventually memory efficient VLSI designs [7].

2.3. Multiplier with Three Split Technologies. The majority of prior efforts have concentrated on lowering the multiplication time while paying little attention to the area efficiency [18, 19]. Area efficiency is particularly significant since high-area-cost solutions often need the use of a high-end field-programmable gate array (FPGA) platform or an ASIC platform with a high gate count, both of which are prohibitively expensive for most practical applications [5, 7]. The



Rel. freq.	1.0	0.8	0.67	0.6	0.5	0.33
Tech. node	90 nm					
Area	286140	282621	280131	275695	260376	239775
HVT	31.2%	39.2%	54.6%	53.4%	72.1%	98.8%
RVT	35.1%	27.5%	32.8%	36.8%	23.2%	1.2%
LVT	33.7%	33.3%	12.7%	9.8%	4.6%	0.1%
Voltage V	1.0	1.0	1.0	0.97	0.97	0.97
Tech. node	45 nm					
Area	61530	59598	57542	52172	50626	48224
HVT	33.0%	42.8%	56.9%	65.3%	77.2%	94.5%
RVT	16.5%	19.4%	17.7%	15.2%	13.7%	4.9%
LVT	50.5%	37.8%	25.4%	19.5%	9.1%	0.6%
Voltage V	0.99	0.99	0.81	0.81	0.81	0.81

Figure 4: C7552 ground up designs.



Rel. freq.	1.0	0.8	0.67	0.6	0.5	0.33
Tech. node	90 nm					
Area	11237	8507	8338	6958	5256	4142
Voltage V	1.0	0.95	0.90	0.85	0.85	0.85
Tech. node	45 nm					
Area	2056	1269	1126	1039	980	955
Voltage V	0.99	0.84	0.84	0.81	0.81	0.81

Figure 5: C7552 ground up RVT transistor.

goal of this brief is to create a fast million-bit integer multiplier that is also space efficient in hardware NTT while yet maintaining high performance.

External RAMs are the other three RAMs in Figure 6, which are designated as NTT RAM A, NTT RAM B, and INTT RAM C, respectively. This is conceivable since the performance of the pipeline structure is dependent on the delay time of read or write operations in these three RAMs to a small extent.

2.3.1. Important Parameters. The outcomes of the implementation are shown in Figure 7. It takes up 7.9 k ALUTs, 3.6 K registers, 40 DSP blocks, and 5.3 Mb of block memory to implement our million-bit multiplier. The simulation result for a million-bit multiplier is compared against a software implementation using the NTL library to ensure that it is accurate.

Each of the six processes (data input, NTT, multiplication of NTT results, INTT, CRT, and data output) is represented by a clock cycle, which may be found in Figure 7. In sum, our multiplier is capable of performing a single 1024-bit multiplication in 1.6 million clock cycles, which is equivalent to 9.7 million clock cycles at the maximum frequency of 170 MHz. If we consider that the NTT and INTT in our system are two-stage pipelined [7], we can see that our design can calculate the product of two 1024 k-bit integers every 4.9 milliseconds.

The primary goal of the proposed study is to develop a VLSI architecture that is both power-efficient and memory-efficient for the construction of million-bit multiplier circuits. It is composed of three state-of-FIFO buffer modules, a synchronizer, a state control module, and a state-of-FIFO controller. The synchronizer module is responsible for keeping the RAM and state control arbitrator in sync. The FIFO module is used to divide the million bit multiplier into eight-bit multipliers [7], which are then divided again. We have completed a VLSI design using the double modulus approach that has been presented.

3. Proposed Work

3.1. Pipelined FSM-Based Power Scheduling Scheme. Figure 8 illustrates how the whole proposed block is a MPSOCs and how it is comprised by two different blocks of FIFO and synchronization that are combined to form the MPSOCs.

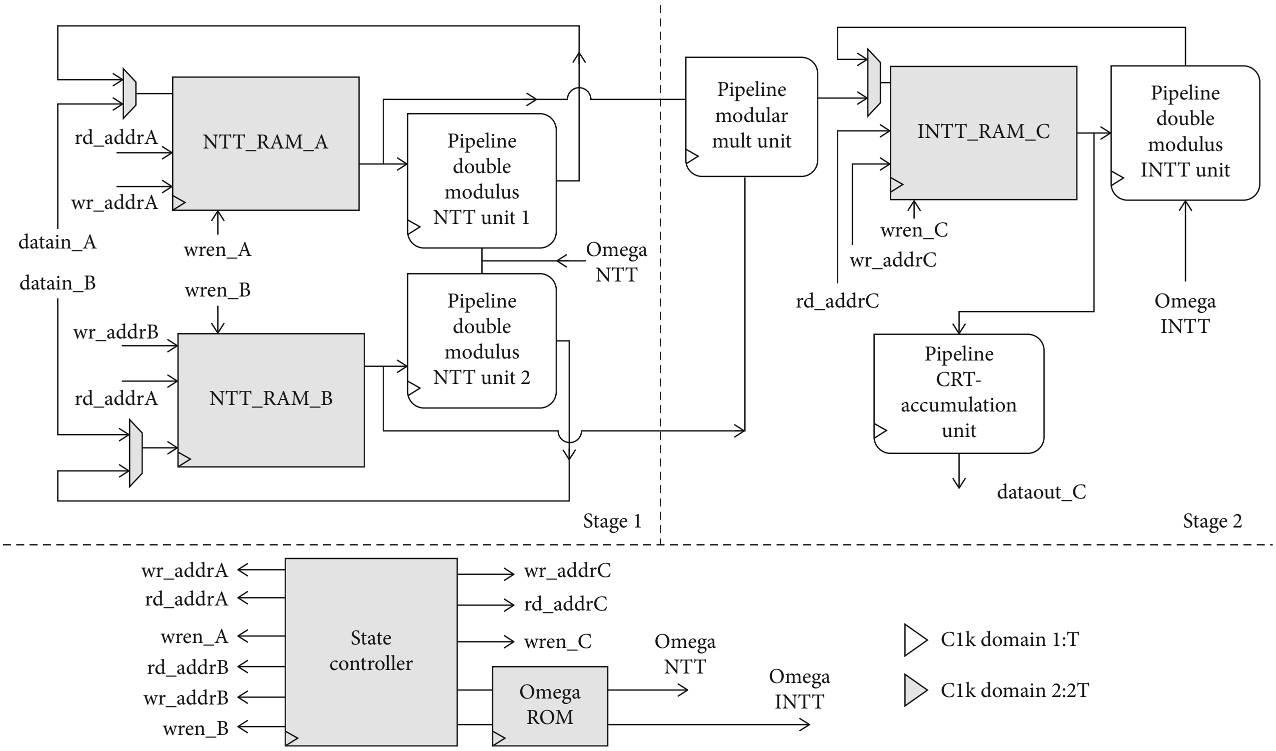
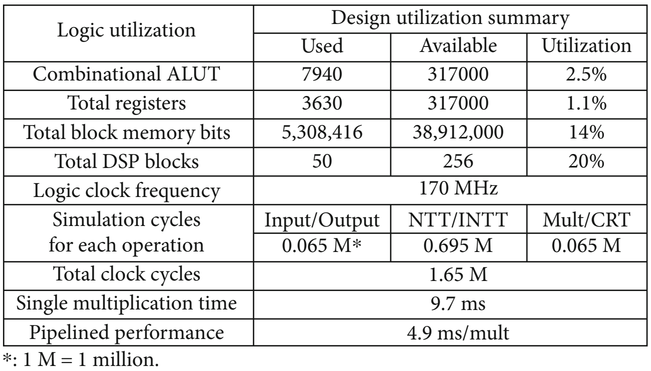


Figure 6: VLSI architecture of proposed integer multiplier.



Logic utilization	Design utilization summary		
	Used	Available	Utilization
Combinational ALUT	7940	317000	2.5%
Total registers	3630	317000	1.1%
Total block memory bits	5,308,416	38,912,000	14%
Total DSP blocks	50	256	20%
Logic clock frequency	170 MHz		
Simulation cycles for each operation	Input/Output	NTT/INTT	Mult/CRT
	0.065 M*	0.695 M	0.065 M
Total clock cycles	1.65 M		
Single multiplication time	9.7 ms		
Pipelined performance	4.9 ms/mult		

*: 1 M = 1 million.

Figure 7: Implementation results on Altera FPGA.

Specifically, it investigates approaches for optimizing the synchronization mechanisms for MPSOCs with complicated interconnect (network-on-chip), which are intended for use in future power-efficient systems. The suggested method is based on the concept of executing synchronization actions that need the use of the optimum power of a shared process, as opposed to using a separate process.

The suggested block is completed in its entirety by breaking it down into modules. The power is optimised from one of the other components, which has been divided. Figure 9 depicts the completed design, which includes the synchronizer, a state control module, and a three-state FIFO buffer module, among other things. We have completed a VLSI design of a million-bit integer multiplier using the double modulus approach that has been presented.

3.2. FIFO. The suggested work divided the multiplier bit into three FIFO architectures (first come, first served). The FIFO module is used to divide a million-bit multiplier into eight-bit multipliers, resulting in a total of 256 values. In this section, we will create a 3 FIFO module, which will allow us to get 256∗256∗256 bits of value, which is about one million bits. In both cases, write operations are disabled when the FIFO is full, and read operations are disabled when the FIFO is empty. When the FIFO is empty, the FIFO empty flag is set to high, and when the FIFO is filled, the FIFO full flag is set to high as well. Furthermore, we must include soft reset and lfd state in our code. This is the accurate FIFO unit, which is seen in Figure 4. It is made up of two components: RAM memory and FIFO control. When the read enb and write enb signals are activated, the FIFO receives requests to read and write data.

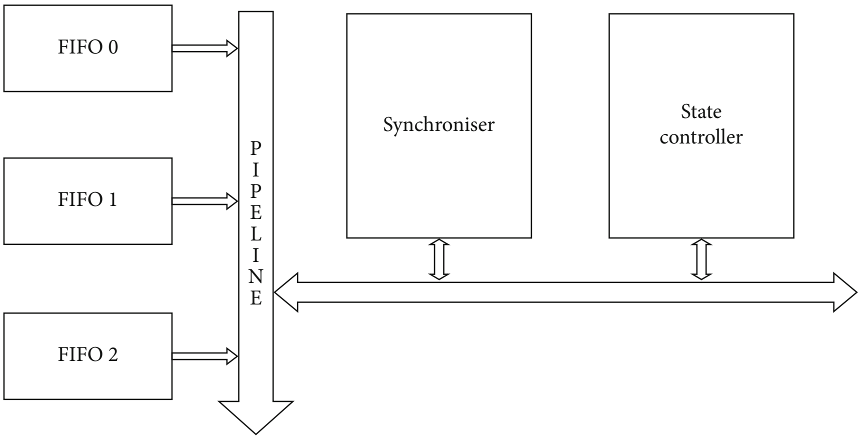
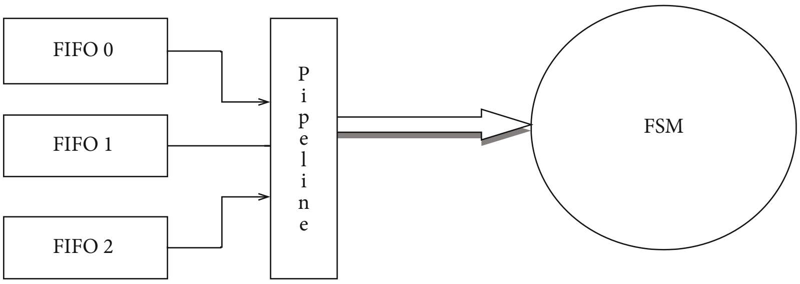


Figure 8: Proposed block of power optimization.



Supply source	Supply voltage	Quiescent current (mA)	Dynamic current (mA)	Total current (mA)
Vccint	1.140	62.07	0.10	62.16
Vccaux	2.125	45.00	0.03	45.03
Vcco25	2.125	3.00	0.52	3.52

Quiescent power (mW)	Dynamic power (mW)	Total power (mW)
173.52	1.40	174.93

Figure 9: Module of proposed block.

FIFO memories are commonly used for data buffering and flow control in many bio-signal sensing applications, and they take up a disproportionate amount of die space and use a disproportionate amount of power. A standard FIFO memory is made up of three basic components: storage elements, read/write pointers, and timing control circuitries, to name a few. Instead of using register-based memories, SRAM-based storage units are often used in applications requiring high density and low power consumption. In FIFO memory, each storage element may be operated separately in read, write, data-retention, and power-off modes with a variety of voltages, depending on the kind of storage element. The dynamic voltage scaling for FIFO memory may be applied at several levels of granularity, including row-level, bank-level, sub-array level, and array level, as shown in the following figure.

3.2.1. Pipelining and Synchronizer Synchronization. The synchronizer is a module that is used to link the FIFO and the controller, also known as the FSM, in order to ensure effective communications between them. The design of the multiplication may benefit from the use of the pipeline approach. To ensure that communications are as efficient as possible, the synchronizer is a module that links the FIFO and controller (i.e., FSM). The suggested design, which involves the sharing of buffers across surrounding input ports via the use of a synchronizer, is described below. In order to join the pipeline, successive activities or operations are launched one at a time, one per cycle. Once the pipeline is completely filled, one result emerges from the pipeline, which may then be used for an additional cycle. Following pipelining, a combinational circuit and the circuit's clock frequency are implemented. Through the use of pipelining, the performance of synchronous circuits is improved at the price of increased latency and area. Pipelining changes a combinational circuit into a sequential circuit by dividing the route into smaller delays, as shown in the diagram. The suggested design calls for the sharing of buffers between input ports that are close to one another. The input and output sections of the router architecture may be separated into two categories: the input and the output. The input section is in charge of allocating buffer space and receiving packets from nearby routers, among other things.

3.2.2. Scheduler (or) FSM. Power-gating is a technique that totally shuts off the power supply of a piece of a circuit, resulting in the entire elimination of power consumption for that area of the circuit. The charging of the subcircuit to be triggered, on the other hand, must take place for a period of time prior to the activation of the subcircuit. As a result, the decomposition of a finite state machine (FSM) for its power gated implementation might be influenced by this essential problem. FSM partitioning combined with state encoding provides a comprehensive solution to the issue of power scheduling in FSM synthesis, which was previously unsolvable. It achieves superior results in terms of dynamic and leaky power usage than the previous version. When not in use, processors' power may be turned off, but controllers are constantly in use and receive power from the system. This results in the controller using a significant amount of power from the rest of the system. The majority of controllers are built as finite state machines (FSMs). If you are designing an FSM, the most typical strategy to achieving low power consumption is to break the FSM into two or more sub-FSMs, with only one of these being active at a time. In this study, both the precomputation-based approach and the clock-gating technique were employed in conjunction with each other. The amount of power that may be saved by dividing the FSM is mostly governed on how well the partitioning has been performed.

3.2.3. FSM with a Multiprocessor Architecture. An FSM is a discrete dynamic system that converts a series of input vectors into a sequence of output vectors via discrete dynamic transformations. States in an FSM specification may be either symbolic or binary-encoded, depending on the implementation. State reduction and state assignment procedures are used to optimise the circuit at the sequential level, and they are both implemented. State reduction lowers the number of redundant states, while state assignment encodes the symbolic states into binary codes (as opposed to state assignment). Synthesizing the code produced for FSM with many processes is accomplished via the use of different speed and area limitations. LUTs are used in the technology mapping of this FSM, which has many processes of the state transition and hence requires a technology mapping. According to the results of this investigation, the FSM created with many processes has used less register by LUT than the FSM developed with a single process. The port via which data is being received in the current instance should be given the lowest priority for the next round of scheduling considerations. The algorithm causes all of the active data to flow into a queue and then to the core, which is in need of power. It has the ability to deal with many loops and conditional branching in the constraint system at the same time.

4. Result and Discussion

In the Spartan6 device family, the suggested system has been implemented. After performing a power analysis on the proposed MPSOCs, it has been discovered that their combined power consumption when operating at an ambient temperature of 250°C and a clock rate of 100 Hz is only 0.011 W, which is a reduction of 93.64 percent over the existing system, which consumes a total of 0.174 W. The power optimization of a million bit multiplier is dependent on the time between logic gates as well as the number of logic gates. As a result, the power consumption is represented by the gate delay of the multiplier. If the gate delay is minimized, it is more likely that the power optimization will be improved. In the current system of million-bit multiplier, there is just a single FIFO module. As a result, even for a modest input, the whole million-bit multiplier is used, which is superfluous. As a result, much electricity is needed to operate.

The power consumption of the MPSOCs may be determined from the data in the preceding table. The power study of the multiplier in the present system indicates that it produces around 0.174 W. It is anticipated that the planned MPSOCs would use less power than the present system due to the use of the Moore multiplier rather than an array multiplier.

4.1. Logic Gates and Look-Up Tables Are Two Types of Logic Gates. In CLBs, lookup tables are utilised to implement function generators, which are then used to generate code. Two function generators each get four distinct inputs, which are then combined to provide a single output (F1-F4 and G1-G4). These function generators may implement any arbitrarily specified Boolean functions with four inputs that have been defined in a previous step. It is composed of a block of S-RAM that is indexed by the inputs of the LUT. The value stored in the indexed position of the LUT's S-RAM is used as the output of the LUT. Because RAM is volatile, the contents of the chip must be initialized each time the chip is turned on. This is accomplished by the transfer of the contents of the configuration memory into the S-RAM storage area network. Lookup tables and flip-flops are used in numerous adjustable logic blocks, which may be customized. A great deal of attention has also been paid to lowering the number of look up tables and the number of gates. It provides information on the number of registers and flip-flops that have been utilised.

By changing the multiplier concept, the number of lookup table and number of gates have been reduced. The analysis of proposed gates and lookup table are as follows as Table 1.

As seen in Figure 10, it can be extrapolated that the total number of LUTs used is about 45 percent, the total number of logics used is approximately 45 percent, and the total number of slice registers utilised is approximately 11

Table 1: Analysis of logic gates and look up table.

Slice logic utilization	Used	Available	Utilization
Number of slice registers	528	4,800	11%
Number of slice LUTs	1,091	2,400	45%
Number used as logic	1,091	2,400	45%
Number of LUT flip flop pairs used	1,102		
Number with an unused flip flop	583	1,102	52%
Number with an unused LUT	11	1,102	1%
Number of fully used LUT-FF pairs	508	1,102	46%
Number used as flip flops	528		

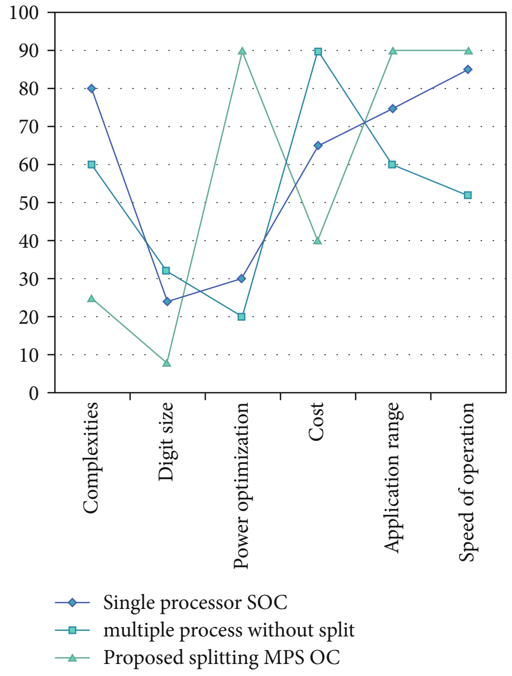


Figure 10: Comparison of processors.

percent. As a result, both the number of LUTs and the number of gates are lowered.

5. Conclusion and Future Enhancements

When planning the power consumption of a multicore CPU in the past, inefficient scheduling techniques were used, which was a limitation of the approaches that were in use at the time. In accordance with the power scheduling method, however, the suggested method of power scheduling algorithm based power optimization of MPSOCs is not based on a finite state machine scheduling strategy. Power optimization of MPSOCs is based on a finite state machine scheduling algorithm. In point of fact, the state controller, which is connected to the MPSOCs, is the one responsible for controlling this finite state machine from the very beginning. Several scheduling algorithms, including power aware scheduling, ALAP scheduling, ASAP scheduling, time restricted scheduling, resource-scheduling, and DVS, in addition to their impact on multiprocessors, as well as their benefits and drawbacks, are discussed in the chapter that came before this one. As was said before, each of these algorithms has already been predefined, and due to the fact that they are simply too intricate, we will not be able to apply them for the circuit that we choose since it would be impossible. DVS is quite different from traditional scheduling algorithms in a variety of significant respects. However, it contains an inductor that acts as a filtering network, and it is a tough process to remove the inductor from the DVS. The end result is that the scheduling method does not achieve its goal of maximising efficiency in terms of energy use. The approach that has been described is effective in overcoming the shortcomings of the scheduling technique that had been used before. The fact that the FIFO, pipeline, synchronizer, and state controller modules are all combined on a single chip helps to reduce the multiplier's overall power consumption. Due to the fact that it is a tiny circuit design, it can be readily included into bigger circuit designs, hence, minimising the manufacturing costs as well as the complexity of such designs. The term "multiprocessor system on chip" (MPSOC) is an acronym for "multiprocessor system on chip." Because of the many benefits associated with using a finite state machine to manage small amounts of data, MPSOCs make use of these machines. This not only makes the separation process simpler but it also makes it easier to get the necessary multiplier source and the appropriate quantity of power that is needed for the processing.

In this proposal, both the power efficiency and the area efficiency of a router for a long digit multiplier are put to the test and assessed. It was important to groom the system on chip in order to reach perfection in terms of power and area efficient network on chip, while also taking into consideration the total area and power of system on chip. This was a need in order to accomplish perfection. The network on chip, often known as NoC, was responsible for more than 41 percent of the total market share. The cost of each hop is determined by the microarchitecture of the router, which will be discussed in more detail later. This has an immediate and direct impact on the latency, throughput, and power consumption of the system. This kind of scheduling method, which makes use of the technology known as network on chip, has the ability to regulate the amount of power that is used by a particular CPU. As a direct result of this, the most common platforms on which this scheduling method is used nowadays are mobile phones, laptops, and computers with a large number of cores (up to 256).

Data Availability

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

The authors are thankful to the Taif University Researchers Supporting Project number (TURSP-2020/110), Taif University, Taif, Saudi Arabia, for providing the financial support and research facilities.

References

[1] J.-W. Lee, S.-C. Chung, H.-C. Chang, and C.-Y. Lee, "Efficient power-analysis-resistant dual-field elliptic curve cryptographic processor using heterogeneous dual-processing-element architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 1, pp. 49–61, 2014.

[2] C. Walravens and W. Dehaene, "Low-power digital signal processor architecture for wireless sensor nodes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 313–321, 2014.

[3] Z. Zhang, X. Kavousianos, K. Chakrabarty, and Y. Tsiatouhas, "Static power reduction using variation-tolerant and reconfigurable multi-mode power switches," *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 1, pp. 13–26, 2014.

[4] A. A. Sinkar, H. R. Ghasemi, M. J. Schulte, U. R. Karpuzcu, and N. S. Kim, "Low-cost per-core voltage domain support for power-constrained high-performance processors," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 4, pp. 747–758, 2014.

[5] A. Paul, S. P. Park, D. Somasekhar, Y. M. Kim, U. R. Karpuzcu, and C. H. Kim, "System-level power analysis of a multicore multipower domain processor with on-chip voltage regulators," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 12, pp. 3468–3476, 2016.

[6] C.-H. Huang, Y.-T. Ma, and W.-C. Liao, "Design of a low-voltage low-dropout regulator," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 6, pp. 1308–1313, 2014.

[7] X. Feng and S. Li, "Design of an area-effcient million-bit integer multiplier using double modulus NTT," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 14, 2013.

[8] S. Fekriershad and F. Tajeripour, "Color texture classification based on proposed impulse-noise resistant color local binary patterns and significant points selection algorithm," *Sensor Review*, vol. 37, no. 1, pp. 33–42, 2017.

[9] M. Verma and B. Raman, "Local neighborhood difference pattern: a new feature descriptor for natural and texture image retrieval," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 11843–11866, 2018.

[10] Y. Verma and R. Tiwari, "Low power approach of clock gating in synchronous system like FIFO: a novel clock gating approach and comparative analysis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 2, 2014.

[11] R. Sawhney and A. Kumar, "IEEE transactions on very large scale integration (VLSI) systems publication information," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 10, p. C2, 2013.

[12] D. M. Abdelkader and F. Omara, "Dynamic task scheduling algorithm with load balancing for heterogeneous computing system," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 4, 2015.

[13] M. Jasmin, "Optimization techniques for low power VLSI circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 16, 2014.

[14] Z. Baruch, "Scheduling algorithms for high-level synthesis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 10, 2015.

[15] T. Wu, Y. Liu, D. Zhang et al., "DVFS-based long-term task scheduling for dual-channel solar-powered sensor nodes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 11, pp. 2981–2994, 2017.

[16] K. Chakraborty and S. Roy, "Architecturally homogeneous power-performance heterogeneous multicore systems," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 21, no. 4, pp. 670–679, 2013.

[17] M. Gupta, K. K. Gupta, and P. K. Shukla, "Session key based fast, secure and lightweight image encryption algorithm," *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 10391–10416, 2021.

[18] P. Zhou, A. Paul, C. H. Kim, and S. S. Sapatnekar, "Distributed on-chip switched-capacitor DC–DC converters supporting DVFS in multicore systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 9, pp. 1954–1967, 2014.

[19] A. El-Maleh, S. M. Sait, and F. N. Khan, "Finite state machine state assignment for area and power minimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 6, 2013.