Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 1368 – 1382
© TÜBİTAK
doi:10.3906/elk-2002-150

Efficient power macromodeling approach for an IP-based SoC system using discrete water cycle algorithm

Faisal SIDDIQ[1,*], **Yaseer Arafat DURRANI**[2]
[1]Department of Electrical Engineering, Faculty of Electronics & Electrical Engineering,
University of Engineering & Technology, Taxila, Pakistan
[2]Department of Electronics Engineering, Faculty of Electronics & Electrical Engineering,
University of Engineering & Technology, Taxila, Pakistan

Received: 24.02.2020 • **Accepted/Published Online:** 07.08.2020 • **Final Version:** 31.05.2021

Abstract: Low-power consumption is becoming a crucial concern that cannot be neglected in system-on-chip (SoC) designs. Low-power solutions help designers to provide a powerful methodology to analyze, estimate, and optimize today's power concerns. Early estimation of power at a high level reduces the redesign cycle and turn-around time. Power dissipation is a function of input patterns and their characteristics. This paper describes a power estimation technique by using predefined statistical characteristics-based input patterns, which gives the average power dissipation of individual intellectual property (IP) blocks and interconnects/buses in an SoC system design. During the power estimation phase, the discrete water cycle algorithm is implemented to generate optimized input patterns. Then Monte-Carlo zero-delay simulation approach is used for each individual IP block and buses at a high level. Total system power is the simple addition of average powers of IP blocks and buses. In experiments of entire IP-based SoC system with buses, our modified macromodel gives an average error of 7.91%. Accuracy of the proposed model is improved as compared to our previous model.

Key words: System-on-chip, intellectual property, discrete water cycle algorithm, Monte-Carlo simulation, power estimation

1. Introduction

In the modern era, consumers demand low-power, emerging, and high-reliability portable systems. Therefore, we must reduce the power consumption due to the availability of limited resources. In high-performance embedded electronic systems, power dissipation is an important and top-priority issue. Integrated circuit (IC) design flow is a very powerful tool for analysis, estimation, and optimization of an increasing demand for power. Conventional design approaches are not adequate to deal with power estimation and optimization of modern multicore systems. It is difficult to maintain the balance between speed and efficiency. Therefore, it is a challenging task to approach a low-power successful system-on-chip (SoC) design [1]. In the initial stages of system design, electronic design automation (EDA) tools are helpful for designers for power estimation. These EDA tools improve the performance metrics in IC design. Earlier design analysis has ignored the redesigning procedure and it also manages the power requirement during the initial design phase [2]. Due to the evolutionary growth of a system's complexity, verification of system is a very time-consuming procedure; therefore, power and performance analysis are very essential for reducing the turnaround time. Design time and cost of the

[*] Correspondence: faisal.siddiq@uettaxila.edu.pk

digital electronic system can be minimized by reusing the predesigned blocks. Intellectual property (IP) blocks are a solution for the problems that are associated with the SoC design. Designers are at ease by using prevalidated components and IPs in system design. Reuse of IP is just like a plug-and-play and includes interconnect buses and hierarchical infrastructures. In the initial design phase, designers are very keen to find the design space exploration, technology libraries, bus designs, memory sizes, and IP power constraints [3–5]. Such challenges can be handled by power estimation models used at various abstraction levels with variable power and speed requirements. Among various high-level power estimation approaches, power macromodeling is a promising technique for handling the difficulties. Power macromodeling is applied on different IP-cores of an SoC-based system; input patterns with predefined statistical characteristics are required. For better estimation and accuracy, designers must perform different functional simulations [6]. Therefore, an efficient power macromodel for IP-based digital systems based on improved and optimized input characteristics at higher level of abstraction is proposed in this research. It motivates the designers for an early estimate of power consumption and it also reduces the redesign cycle and turnaround time. The proposed model is also time-efficient because it reduces the input variables of a macromodel function; the number of functional simulations and percentage of error are also reduced. This is proved by simulating IP modules and analysis of experimental results.

2. Literature review

In complementary metal-oxide-semiconductor circuits, total power consumption is a summation of dynamic and static powers. Dynamic power dominates when there is switching at the circuit inputs and it also propagates through the circuit. There are two major power estimation techniques: static and dynamic. Dynamic techniques are efficient because they simulate the circuits with typical input sequences [7, 8]. A fast and scalable method is presented in [9] for finding a toggle rate for field programmable gate array (FPGA)-based circuits. This approach uses a novel method for finding spatial correlation by using the XOR-based decomposition method. The power estimation of the IP-based SoC system at a high level is presented in [6]. It includes the power models for local and global bus interconnects. Various input patterns are generated by genetic algorithm (GA) and these patterns have special statistical characteristics. The look-up table (LUT)-based macromodel is used to estimate the power of the system. Estimated power is independent of the system and it is dependent on input pattern characteristics. In high-level designs, the bottom-up approaches are the most efficient because of the reuse of IP cores/blocks. In these models, internal details of IPs are given, and the construction of the macromodel is easy with the help of power characterizations [10]. The water cycle algorithm is applied for the harmonic estimation of time-varying noisy signals and for solving the nonconvex problem of hydro-thermal coordination in power systems [11–13]. Discrete water cycle algorithm (DWCA) is used for optimal detection of power system stabilizers for power system applications. It finds the optimal parameters with minimum values of overshoot and settling time within minimum time and also increases the stability of power systems [14]. DWCA is used for generation of 4-bit optimal coding sequence matrix for wide-band radar cross-section reduction [15]. DWCA is used for constraint handling for generation expansion planning with the least cost and reduced time compared to other optimization algorithms [16].

At register-transfer-level (RTL), there are several approaches discussed in the literature to find the switching activities inde the circuit [17]. These approaches are quite accurate, but due to slowness and expensiveness, they limit the length of the input stream. For an accurate estimate of power dissipation, we need several lengths of the sample size of input streams. To handle the sample size of the input stream, the vector compaction technique reduces the sample size of input streams in the stimulus. Among various high-level power estimation approaches, the power-macromodeling- and LUT-based approach is a promising technique to find the average power dissipation [6]. Power macromodeling is applied on different IP-cores of an SoC-based system; input patterns with predefined statistical characteristics are required. Recently, many researchers proposed power estimation models to model the exact behavior of the circuit with minimum simulation time and better accuracy. In [18], a copower estimation technique is proposed for the power estimation of different parts of SoC. Different speed-up techniques are applied in order to reduce the workload of different simulators. An efficient and accurate dynamic power estimator is proposed in [19] for embedded multipliers in FPGAs. The efficiency of this technique is better than other low-level commercial tools. This model is also available for integration with high-level power optimization techniques. In [20], a power estimation methodology is presented for complex systems through FPGA accelerators. This technique improves both speed and accuracy and can be integrated into high-performance FPGA-based simulators.

2.1. Paper summary and contributions

In this paper, we use the predefined-statistical-properties-based input pattern generator to estimate the average power consumption of different IP blocks and interconnects in an SoC-based design. Previously, we used a GA for generation of input patterns for estimation of IP-based SoC digital system in [3]. In this work, we propose a discrete water cycle algorithm (DWCA) for the generation of optimized input pattern generation for better accuracy of power estimate of any IP-based SoC system. Moreover, there is no precedent study on the adaption of DWCA for input pattern generation for power estimation of any digital system. The contributions of the proposed work are as follows:

1. Predefined statistical-properties-based optimized input patterns are generated using DWCA
2. Improved version of IP-based power macromodel for SoC digital system is formulated
3. Full span coverage of input signal statistical properties is tested to model IP-based SoC system design
4. Monte-Carlo simulation is used for the convergence analysis of power model from functional simulations
5. The proposed model is evaluated for IP blocks and interconnects
6. Regression and convergence analysis is used for evaluation of the proposed model

The paper is organized as follows. Section 2 presents the literature review. Section 3 describes the power macromodeling and analysis of different components of the system. DWCA and its modeling is presented in Section 4. Section 5 describes the Monte-Carlo simulation approach. The experimental results and discussion are presented in Section 6. Finally, Section 7 gives the summary, conclusion, and future recommendations of the research work.

2.2. Problem formulation

The statistical power macromodel is used for the estimation of each IP block and interconnects/buses for any SoC-based digital system. This model helps designers for an early estimate of the system at a higher level. DWCA is used for pattern generations with predefined statistical characteristics for the power estimation of different IP blocks in the SoC system. Power estimation for IP-based SoC system under assumption of zero-delay model is stated as: "Given the IP-based SoC system having N number of IP blocks and interconnects/buses, $(IP_1, IP_2, ...IP_N, ...Bus_1, Bus_2, ...Bus_N)$ under the assumption of zero-delay model provide the early estimate of the average power for individual IPs and buses by applying statistical-properties-based input patterns as shown in Figure 1". The Input signal statistical parameters are explained in the next section.

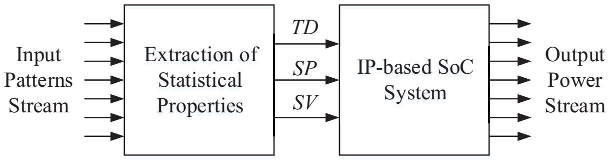

Figure 1. Early power estimation of IP-based SoC system.

3. Power macromodeling of IP-based SoC system

In any system designing, the choice of model's parameters and their characteristics are the most important concerns for designers. Selected characteristics have the ability to model the exact features for power consumption and they also help us for better estimation of power dissipation with higher accuracy.

3.1. Power modeling for IP cores

Previously, we used a GA for generation of input test patterns with characterization of TD, SP, S_C, T_C discussed in [3, 6, 21]. TD and SP are the transition density and signal probability of input patterns. The S_C and T_C are spatial and temporal correlations. Now we improved our previous work by modifying the input parameters and by using DWCA for the generation of optimized input patterns. This macromodel has a linear relationship with input variables and gives the average power dissipation in (1).

$$P_{IPBlock-avg} = f(TD, SP, SV) \tag{1}$$

$$TD = \frac{\sum_{j=1}^{r} \sum_{i=1}^{s-1} q_{ij} \bigoplus q_{i+1,j}}{r * (s-1)} \tag{2}$$

$$SP = \frac{\sum_{i=1}^{r} \sum_{j=1}^{s} (q_{ij})}{(r * s)} \tag{3}$$

$$SV = \frac{\sum_{i=1}^{r} \sum_{j=1}^{s} (q_{ij} - \ SP)^2}{(r * s) - 1} \tag{4}$$

Here $P_{IPBlock-avg}$ is the average power of IP block. For each of IP block, the number of inputs are r and binary input stream q has a length of s. SV is the signal variance of input patterns. TD, SP, and SV cover all the aspects of the proposed model and its accuracy will also be proved by experimental results. These parameters also define the correlation factor of the input patterns. They affect the power dissipation of any IP-based SoC system. This model also reduces the computational complexity. Therefore, accurate modeling of parameters is required for better accuracy and the exact behavior of the digital system.

3.2. Power modeling for buses

Various power modeling techniques for interconnects/buses were discussed in [22, 23]. Previously, we introduced a new statistical power model for buses given in (5):

$$P_{Bus-avg} = f(S_{TD_{Bus}}, C_{TD_{Bus}}) \tag{5}$$

$P_{Bus-avg}$ is the average power of buses and it is a function of $C_{TD_{Bus}}$ and $S_{TD_{Bus}}$. The $C_{TD_{Bus}}$ is a coupling transition density of buses and is dependent on switching activity factor and communication between two adjacent buses. Similarly, $S_{TD_{Bus}}$ is self-transition density and is dependent on local bus components and covers local components.

3.3. Power modeling for IP-based SoC system

The IP-block–based SoC digital system is shown in Figure 2. The function described in (1) and (5) is a LUT-based method that is used in the simulation and characterization phase of input patterns. To find the accurate function and their dependency, we must generate several number of input pattern streams for each individual IP block and bus. Power model of the IP-based SoC digital system with interconnect/buses is defined in equation (6):

$$P_{SoC-system} = \sum_{i=1}^{n} P_{i_{IPblock-avg}} + \sum_{i=1}^{k} P_{i_{Bus-avg}} \tag{6}$$

$P_{SoC-system}$ is the average power of the SoC system having n number of IP blocks and k number of buses. Our aim is to find the average total power dissipation of IP blocks and buses by applying user-specified input patterns at single-core level. Power estimation is done by using the Monte-Carlo simulation approach so that a high-level power estimate satisfies the user-specified confidence interval. We sum up the estimates of all IP blocks and buses for finding the total power consumption of a single core.

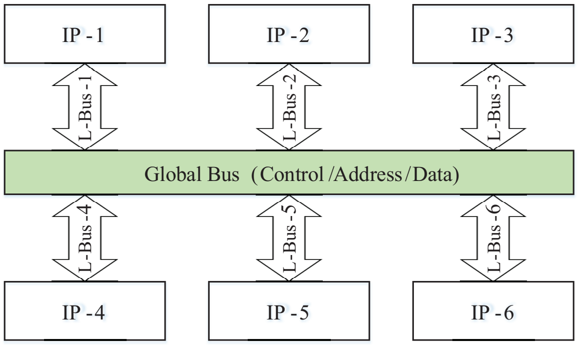

Figure 2. IP-based SoC system.

4. Discrete water cycle algorithm (DWCA)

4.1. Modeling of DWCA

DWCA was established by Eskandar et al. [24] and is a concept based on water flow in the form of streams, rivers, and sea. When the rain falls and snow melts from ice glaciers, water flows into streams. Flow of streams generates rivers and these rivers eventually fall into the sea. Water from streams is evaporated and the trees also transpire the water during the photosynthesis process into the atmosphere to generate clouds. The proposed algorithm starts with raindrops as the initial population. The best raindrop which is chosen is the sea and then the river and the remaining is considered as streams. The modeling of the algorithm is given in the following relations: For the solution of the Nj dimensional problem, the array of raindrops R_k is defined as:

$$R_k = \begin{bmatrix} N_1^k & N_2^k & N_3^k & \cdots & N_j^k & \cdots & N_J^k \end{bmatrix}, \qquad \forall k, \tag{7}$$

where (N_1 , N_2 , N_3 , ...N_j) are the decision values with a set of predefined values for discrete and continuous problems. Population matrix of raindrops is selected as initial population and is given by:

$$Population = \begin{bmatrix} N_1^1 & N_2^1 & \cdots & N_J^1 \\ N_1^2 & N_2^2 & \cdots & N_J^2 \\ \vdots & \vdots & N_j^k & \vdots \\ N_1^K & N_2^K & \cdots & N_J^K \end{bmatrix}, \tag{8}$$

The raindrop's cost is evaluated using the cost function $C_W\left(k\right)$:

$$\begin{aligned} C_W\left(k\right) &= f_{obj}\left(R_k\right), \qquad \forall k, \\ &= f_{obj}\left(N_1^k,\ N_2^k,\ N_3^k,\ \cdots N_J^k\right), \end{aligned} \tag{9}$$

At first, K raindrops are generated and the best individuals with minimum values of rivers and 1 sea are selected.

$$N_{sr} = N_r + 1, \tag{10}$$

where N_{sr} is the total number of rivers and sea. The remaining population of raindrops which forms the streams N_d which flow into the river or flows directly into the sea is calculated as:

$$N_d = K - N_{sr}. \tag{11}$$

Three steps are involved in the movement of water from one place to another. In the first crossover step, streams move into the sea. Two decision variables are generated from movement of the sea and streams and are given by:

$$\begin{gathered} \begin{bmatrix} R_{c1} & R_{c2} \end{bmatrix} = \text{crossover} \begin{bmatrix} R_s(b) & R_{sea} \end{bmatrix}, \\ b = \left\{1, 2, 3, \cdots N_d\right\}, \end{gathered} \tag{12}$$

where R_{c1} and R_{c2} are the child decision variables of crossover operations.

$$R_{sea} = \begin{cases} R_{c1} & \text{if} \quad C_W(R_{c1}) < C_W(R_{c2}) \\ R_{c2} & \text{if} \quad C_W(R_{c1}) > C_W(R_{c2}) \end{cases}, \tag{13}$$

where R_{sea} is designated as the best child selected as sea variable.

In the second crossover step, two-child decision variables are generated by the movement of the river and sea. The best cost of one child is designated as sea and movement of steams into rivers and is given by:

$$\begin{gathered} \begin{bmatrix} R_{c1} & R_{c2} \end{bmatrix} = \text{crossover} \begin{bmatrix} R_s\left(b\right) & R_r\left(c\right) \end{bmatrix}, \\ b = \left\{1, 2, 3, \cdots N_d\right\}, \\ c = \left\{1, 2, 3, \cdots N_r\right\}, \end{gathered} \tag{14}$$

where R_r is the best river in the population.

$$R_r = \begin{cases} R_{c1} & \text{if} \quad C_W(R_{c1}) < C_W(R_{c2}) \\ R_{c2} & \text{if} \quad C_W(R_{c1}) > C_W(R_{c2}) \end{cases}. \tag{15}$$

In the third crossover step, two-child decision variables are generated from the movement of rivers into the sea. The best cost of one child is designated as sea and movement of rivers into sea is given by:

$$\begin{gathered} \begin{bmatrix} R_{c1} & R_{c2} \end{bmatrix} = \text{crossover} \begin{bmatrix} R_r\left(c\right) & R_{sea} \end{bmatrix}, \\ c = \left\{1, 2, 3, \cdots N_r\right\}., \end{gathered} \tag{16}$$

$$R_{sea} = \begin{cases} R_{c1} & \text{if} \quad C_W(R_{c1}) < C_W(R_{c2}) \\ R_{c2} & \text{if} \quad C_W(R_{c1}) > C_W(R_{c2}) \end{cases}. \tag{17}$$

After the completion of the above three steps, evaporation makes the new population. The mutation process is used for evaporation and is applied to both rivers and streams which will be compared to the sea. The μ_m defines search range for optimum solution and the best value is set to 0.1. Mutation operator performed on rivers and streams is given by

$$\begin{gathered} R_{mu} = \text{mutation}\left[\text{if}\left(\mu_m \times R_s\left(b\right) \cong R_{sea}\right)\right], \\ b = \left\{1, 2, 3, \cdots N_d\right\}, \end{gathered} \tag{18}$$

where R_{mu} is the mutation operator of river and streams.

$$\begin{gathered} R_{mu} = \text{mutation}\left[\text{if}\left(\mu_m \times R_r\left(c\right) \cong R_{sea}\right)\right], \\ c = \left\{1, 2, 3, \cdots N_r\right\}. \end{gathered} \tag{19}$$

4.2. Crossover operators

Three types of crossover operators are mostly used, which include single-point, double-point, and uniform. These operators are used to generate child agents from the parent population [25]. In the case of single-point crossover, samples are cut from a single point to make child variables. Two different sets are cut to make child variables in double-point crossover. Similarly, samples are cut uniformly from parents in order to make child variables. The crossover of these types is explained in Figure 3.

4.3. Mutation operators

The mutation operator helps us to find a near-optimal solution. It also avoids local search algorithms from trapping in local optimum solutions. Four different mutation techniques are used, which are swap, insert, reverse, and random. This operator is applied to the sea, river, and stream. In the swap mutation technique, two bits are exchanged randomly to form a mutated agent. Some of the bits of the mutated agents are inserted from one child to another in insert mutation. In reverse mutation, bits are reversed in mutated agents to make new mutants. Finally, in random mutation, random bits are replaced in the mutated agent [25]. Mutation techniques are explained in Figure 4. The simulation parameters of the DWCA are shown in Table 1. Similarly, the flowchart of the DWCA is shown in Figure 5. The steps of the DWCA for input patterns generation are described in the following pseudocode:

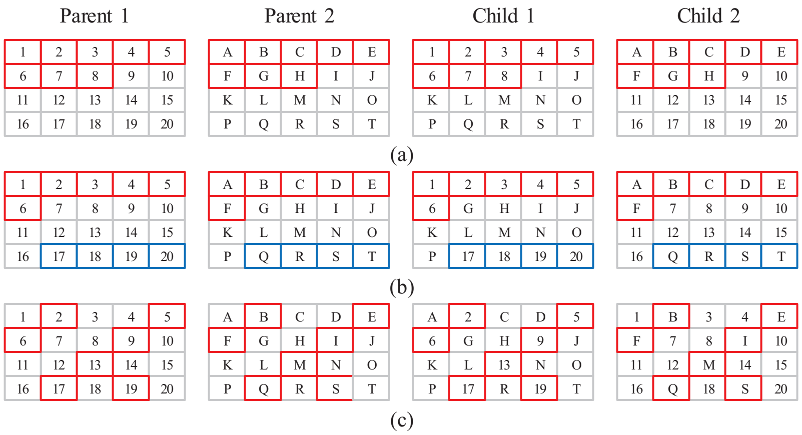

Figure 3. Crossover strategies: (a) single-point, (b) double-point, (c) uniform.

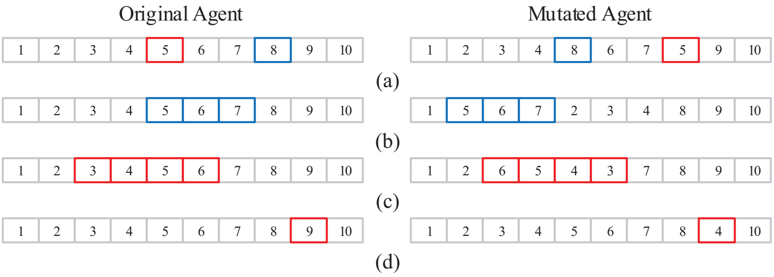

Figure 4. Mutation strategies: (a) swap, (b) insert, (c) reverse, (d) random

5. Monte-Carlo simulation

The Monte-Carlo zero-delay simulation approach is used for power estimation with a user-defined confidence interval [26, 27]. This method uses the original samples of input vectors and a simulation method is used for deriving the average value of samples which is used for finding the average value of the power estimation. From the central limit theorem, the sample value of power estimates assumes the normal distribution with length l approaches infinity. If the random variable x is far away from the normal distribution, the basis of this simulation-based method fails and it may a have larger percentage of error than expected. In this article, we use the IP-based statistical sampling technique that is based on the Monte-Carlo simulation approach. Each IP module is described at RTL level and power estimation technique is also applied at the same level of abstraction. The aim of the proposed approach is to estimate the average power consumption of each IP module under the user-specified input stream. It is estimated under the user-defined confidence interval and level of error percentage. It is very important to find the accurate input patterns and also the number of simulations required for the convergence of power samples. The smaller sample granularity increases the overall efficiency of this approach. There are two points that should be considered for selection and numbers of simulations for convergence are (i) reduced sample variance and (ii) small sample granularity for achieving near-normal distribution of samples. The circuit under test is simulated by applying the input vector stream of predefined statistical characteristics. The power results are monitored, and sample mean and variance are calculated. The process terminates if it meets the predefined stopping criteria for simulation. The response of power behavior is recorded on a power simulator. By using convergence analysis, simulated power converges to its average or

Algorithm 1: Pseudocode of the DWCA

```
Initialize the random population of $K$ raindrops
Compute fitness of each raindrop
while $iter < maximum\ iterations$ do
    /* Movement of water raindrops - Crossover Operators */
    for $b = 1 : N_d$ do
        Crossover operator is performed between the sea and selected stream // Flow of streams to sea
    end
    for $b = 1 : N_d$ do
        for $c = 1 : N_r$ do
            Crossover operator is performed between the selected stream and river // Flow of streams to rivers
        end
    end
    for $c = 1 : N_r$ do
        Crossover operator is performed between the selected river and sea // Flow of rivers to sea
    end
    /* Evaporation Process - Mutation Operator */
    for $b = 1 : N_d$ do
        if selected stream is similar to the sea with probability $\mu_m$ then
            Mutation operator is performed between the stream and sea // Evaporation from the streams
        end
    end
    for $c = 1 : N_r$ do
        if selected river is similar to the sea with probability $\mu_m$ then
            Mutation operator is performed between the river and sea // Evaporation from the rivers
        end
    end
    Find the best solution among the raindrops
end
```

Table 1. Simulations parameters for the DWCA.

Parameters	Values
Probabilities of crossover operators (single-double-uniform)	0.1, 0.3, 0.6
Probabilities of mutation operators (swap-insert-reverse-random)	0.1, 0.2, 0.1, 0.6
Population size	30
Number of iterations	50
Number of rivers and sea	4

true power value.

$$P_{MC} = \sum_{i=1}^{N} P(i). \tag{20}$$

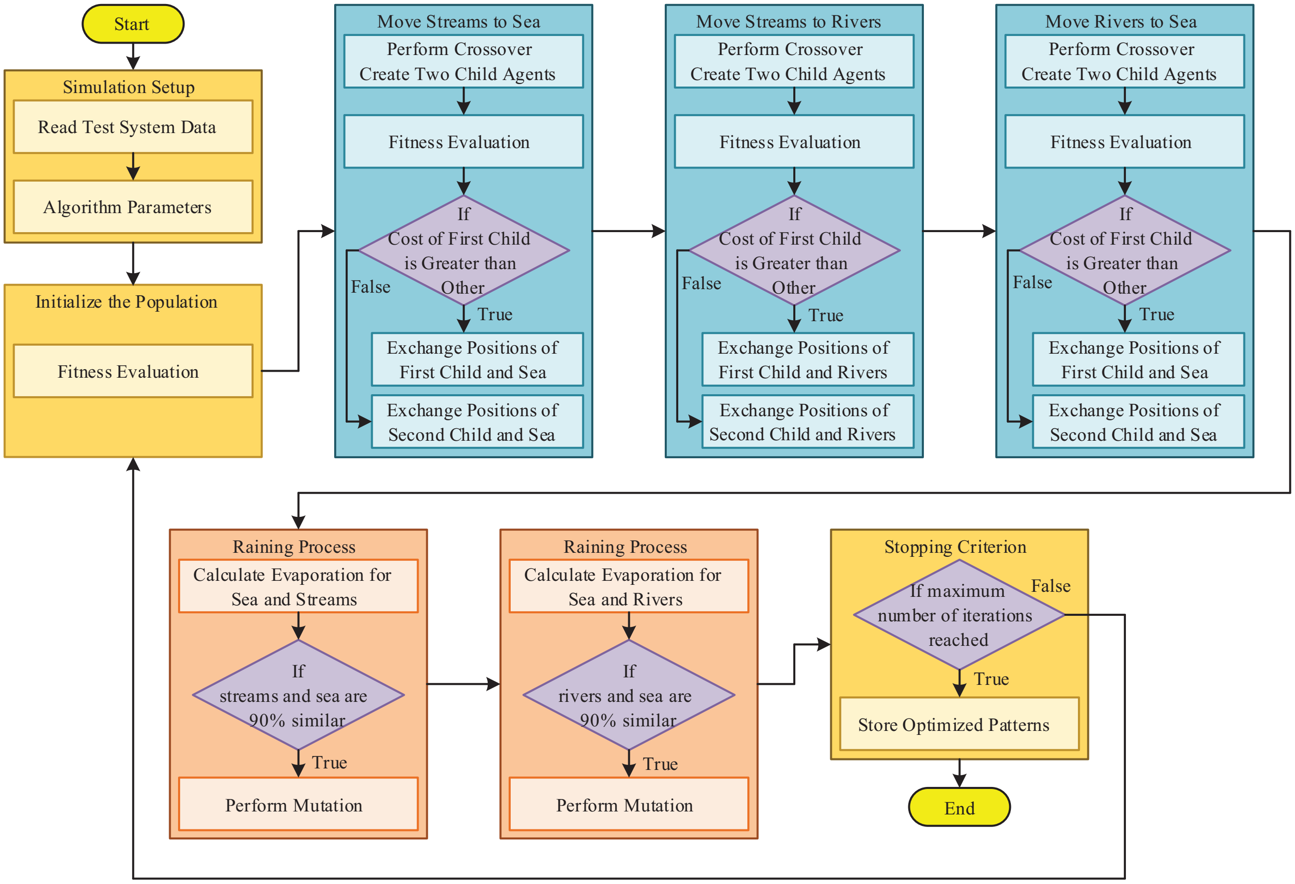

Figure 5. Flow chart of DWCA

P_{MC} is the Monte-Carlo power, N is the maximum number of simulations, and P(i) is the power of individual IP blocks and buses/interconnects.

The Monte-Carlo simulation method is used for computation of average values of power by combining the response with its previous values, and it also tells whether the average value converges to its true value or the model requires more simulations for convergence of predefined criteria [28]. Better results are obtained by using the convergence analysis and central limit theorem by fitting the normal distribution. The confidence interval of a normal distribution is set to 95% with the value of α is set to 0.05. The number of simulations required for each IP-block may be different. It depends on the components in each IP block, and when power values converge to its true and steady value. For this purpose, full-span coverage and uniformly distributed input patterns stream are applied for finding all possible real-time responses of the test system.

6. Experimental results and discussion

Previously, we introduced a power macromodel for individual IP blocks and systems in [3]. In this section, we are presenting an extended and improved version of high-level power macromodeling. Experimental work is implemented on the IP-based SoC digital system shown in Figure 2. Our macromodel function in equation (6) gives the average power dissipation of the system. The modified macromodel in (1) gives accurate results and also consumes less time compared to our previous model. Our previous model has 6250 simulations based on sets of input pattern characteristics to be performed, but in the proposed model it is now reduced to 1250 simulations for each IP block. The GA needs 374 s to generate 6250 sets of input patterns, whereas DWCA needs just 61 s to generate 1250 sets of input patterns. On the other hand, our proposed model also outperforms the previous model by an average of 4.1 h to 20.8 h of Vivado simulations for each IP block. This work is done by using multiple platforms by multitasking in order to save time. The comparison is shown in Table 2.

Table 2. Comparison of the proposed model with the existing model.

Sr. No.	Parameter	Proposed model using DWCA	Previous model using a GA
1	Number of simulations	1250	6250
2	Simulation time (s)	61	374
3	Percentage of error (%)	7.91	11.42

During the power estimation phase, several input patterns are generated with defined characteristics of TD, SP, SV and $S_{TD_{Bus}}$, $C_{TD_{Bus}}$ for individual IPs and buses. By using the proposed model, computational complexity is reduced, because we reduce the input pattern variables. Therefore, extensive amount of simulations are also reduced in order to minimize the computational time with better accuracy of results. By using the LUT approach, functional simulations at the RTL level are performed with power simulators in order to find the average power dissipation of each IP block or bus. During the characterization phase, optimized input patterns with characteristic parameter range of [0–1] are generated by using DWCA. During the estimation phase, a Monte-Carlo zero-delay simulation is performed with several input pattern characteristics for accurate power estimation. Convergence analysis is performed to find the accurate power estimates so that it satisfies the defined level of confidence interval [29]. Linear regression analysis is performed to find the quality of the model's fit and accuracy. The accuracy of the proposed macromodel is tested by running RTL simulations. Power macromodeling results are compared with Vivado HLS. In the end, the average power errors and their fitting are computed. Correlation between estimated and simulated power is shown in Figure 6. Random 25 sets are taken to validate the correlation and accuracy between simulated and estimated powers for IP-2. Different IP blocks are connected with local and global buses to construct an IP-based SoC system. The proposed macromodel makes the power estimation an easy task for designers by simply adding the power estimates of IPs and buses. It also helps designers to reuse and explore different complex IP blocks in real time in a plug-and-play fashion. By using the power macromodeling approach, a power estimate for any IP-based SoC system can be found easily by the simple addition of power estimates of individual IP-blocks. The estimated power $P_{SoC-system}$ in equation (6) is compared with simulated power $P_{Simulated}$ (Vivado HLS Simulator) to evaluate the accuracy of power estimation.

Due to randomly generated input patterns, power estimates of the macromodel have certain uncertainties. To give valid conclusions, error analysis should be performed based on the experimental results. In the experiments, the accuracy of our power estimates for individual IP blocks and buses are summarized in Tables 3 and 4, respectively. For individual IP blocks/buses, 125 distinct sequences are generated with TD, SP, SV and $S_{TD_{Bus}}$, $C_{TD_{Bus}}$ over the entire space of [0–1]. An extensive number of simulations for each block helps us to reveal the accurate relationship of power estimates with input characteristics. The full span of input characteristics is tested to model the real-time scenarios of the system. It is observed that our macromodel is more accurate for the span of [0.25–0.75] and less accurate for the span of [0–0.25] and [0.75–1]. For the verification of randomly generated sequences, $P_{Simulated}$ and $P_{Estimated}$ are compared and a correlation factor

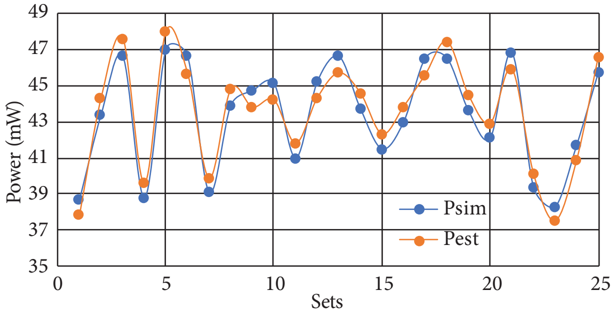

Figure 6. Correlation between the simulated and estimated powers for IP-2.

is found with a range of 95–99% for each individual IP block as shown in Table 3. Twenty-five different sets are selected at random for the validity and accuracy of our power macromodel. Several input sequences of 4, 8, 16, 32 bits are generated. A low-power microprocessor/SoC system is implemented and verified @100 MHz operating frequency. The simulations for test system are performed on a desktop PC: Make HP, Intel Core i7 @ 3.4 GHz processor, 8 GB RAM and 64 bit operating system (Windows 10). This SoC system can be used for low-power commercial applications. All the sequences are uniformly generated with a 95% confidence interval over the entire space of [0–1]. In Table 3, the 2nd column represents the type of IP Block, 3rd column represents the average power, while errors are demonstrated in columns 4–6. The correlation factor between simulated and estimated power is presented in column 7. Similarly, Table 4 represents the power estimates for individual buses/interconnects. Columns 2, 3, and 4 represent the type, size, and power estimates of busses. Column 5 represents the errors between simulated and estimated powers. For IP block-2, by using the Monte-Carlo simulation approach, as shown in Figure 7, the first 400 iterations are called the warm-up region. Then from 500 to 700, this region is called the steady-state region and we take the average of steady-state power values. Convergence analysis for IP-2 using the Monte-Carlo simulation is also presented in Figure 7. Results presented below demonstrate that our power macromodel gives designers an early and accurate estimate of power for an IP-based SoC system. We have used DWCA for the generation of improved and optimized input patterns with predefined statistical characteristics having full span coverage [0–1] to improve the accuracy of our previous macromodel. Further improvement in the macromodel can be made by finding another input-dependent characteristic, which also reduces the error with better accuracy.

Table 3. Power estimates and accuracy for IP blocks.

Sr. No.	IP block	Avg.power (mW)	Avg. error (%)	Min. error (%)	Max. error (%)	Correlation
1	IP-1	87.27	4.59	0.11	12.32	96.8
2	IP-2	47.65	2.04	0.15	7.18	95.6
3	IP-3	108.95	10.06	0.073	17.23	96.7
4	IP-4	26.45	3.91	0.13	11.17	98
5	IP-5	6.35	8.19	0.00006	16.93	98.2
6	IP-6	15.1	15.65	0.22	19.51	98.9
Average			7.4	0.11	14.06	97.3

In this paper, an average error of the entire IP-based SoC system is measured 7.91%. This error can

Table 4. Power estimates and accuracy for buses.

Sr. No.	Bus type	Bus size (Width)	Avg. power (mW)	Avg. error (%)
1	G-Bus	32	64.84	0.9
2	L-Bus-1	32	62.3	0.84
3	L-Bus-2	16	31.7	0.44
4	L-Bus-3	5	11.2	0.4
5	L-Bus-4	5	12.4	0.39
6	L-Bus-5	5	11.5	0.41
7	L-Bus-6	4	8.5	0.35
Average				0.53

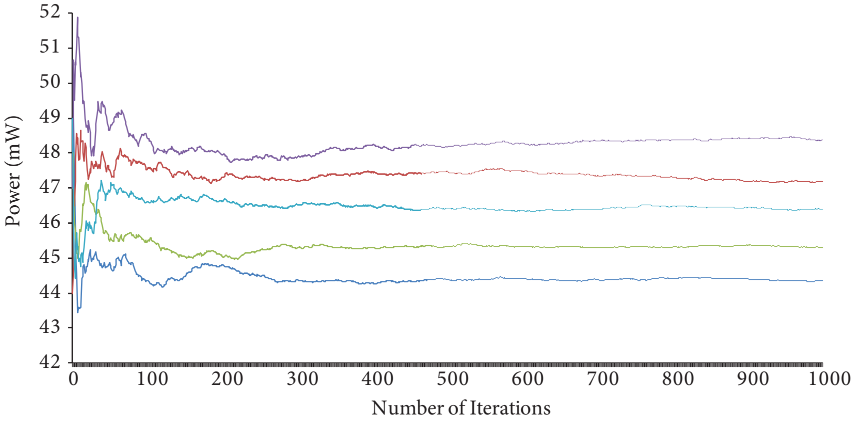

Figure 7. Convergence analysis of IP-2 using the Monte-Carlo simulation.

be further improved by model for propagation delay, glitches, and jitters, etc. For simplicity, we assume our macromodel with a zero-delay approach.

7. Conclusion and future recommendations

A power macromodeling approach with modified input parameters for the IP-based SoC system is presented. For better accuracy, the DWCA is used for optimized input patterns with specified characterizations generated. Two individual power macromodels are used for power estimation of IPs and buses/interconnects. For an entire system, it is simply an addition of powers of all IP blocks and buses/interconnects. In experiments of entire IP-based SoC system with buses, our modified macromodel gives an average error of 7.91%. Accuracy of the proposed model is improved compared to our previous model. One of the important sources of error is delay elements like glitches and jitters. For simplicity, we perform the zero-delay Monte-Carlo simulation approach and ignore delay elements, glitches, and jitters. The proposed model can also be used for low-power SoC commercial applications and it also saves time and budget by giving an early estimate of power consumption.

The proposed model can be used for power estimation of homogeneous/heterogeneous integration of 3D ICs. The power models for 3D integration includes through-silicon-via and their characteristics. A thermal model can also be proposed for thermal analysis of system-on-chip for future recommendation. The error percentage can be further improved by adding models of glitches and jitters.

References

[1] Bazzaz M, Salehi M, Ejlali A. An accurate instruction-level energy estimation model and tool for embedded systems. IEEE Transactions on Instrumentation and Measurement 2013; 62: 1927-1934. doi: 10.1109/TIM.2013.2248288

[2] Durrani YA, Riesgo T. High-level power analysis for IP-based digital systems. ASP Journal of Low Power Electronics 2013; 9: 435-444. doi: 10.1166/jolpe.2013.1283

[3] Durrani YA, Riesgo T. Power macromodeling technique and its application to SoC-based design. International Journal of Numerical Modelling: Electronic Networks, Devices and Fields 2017; 30: e2207. doi: 10.1002/jnm.2207

[4] Durrani YA, Riesgo T, Khan MI, Mahmood T. Power analysis approach and its application to IP-based SoC design. COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering 2016; 35: 1218-1236. doi: 10.1108/COMPEL-08-2015-0283

[5] Trabelsi C, Atitallah RB, Meftali S, Dekeyser J-L, Jemai A. A model-driven approach for hybrid power estimation in embedded systems design. EURASIP Journal on Embedded Systems 2011; 2011: 569031. doi: 10.1155/2011/569031

[6] Durrani YA, Riesgo T. Efficient power analysis approach and its application to system-on-chip design. Microprocessors and Microsystems 2016; 46: 11-20. doi: 10.1016/j.micpro.2016.09.003

[7] Ahmad RW, Gani A, Hamid SHA, Shojafar M, Ahmed A IA et al. A survey on energy estimation and power modeling schemes for smartphone applications. International Journal of Communication Systems 2017; 30: e3234. doi: 10.1002/dac.3234

[8] Vellingiri G, Jayabalan R. Adaptive neuro fuzzy inference system-based power estimation method for CMOS VLSI circuits. International Journal of Electronics 2018; 105: 398-411. doi: 10.1080/00207217.2017.1357763

[9] Czajkowski TS, Brown SD. Fast toggle rate computation for FPGA circuits. In: IEEE 2008 International Conference on Field Programmable Logic and Applications; Heidelberg, Germany; 2008. pp. 65-70. doi: 10.1109/FPL.2008.4629909

[10] Zhang F, Sultania AK. Estimation of power consumed by combinatorial circuitry. U.S. Patent No. 10,302,698. 28 May 2019.

[11] Ashraf M, Ullah MO, Malik T, Waqas A, Iqbal M et al. A hybrid water cycle algorithm-least square based framework for robust estimation of harmonics. The Nucleus 2018; 55: 47-60.

[12] Haroon SS, Malik TN. Evaporation rate based water cycle algorithm for the environmental economic scheduling of hydrothermal energy systems. Journal of Renewable and Sustainable Energy 2016; 8: 044501. doi: 10.1063/1.4958995

[13] Osaba E, Del Ser J, Sadollah A, Bilbao MN, Camacho D. A discrete water cycle algorithm for solving the symmetric and asymmetric traveling salesman problem. Applied Soft Computing 2018; 71: 277-290. doi: 10.1016/j.asoc.2018.06.047

[14] Ghaffarzadeh N. Water cycle algorithm based power system stabilizer robust design for power systems. Journal of Electrical Engineering 2015; 66: 91-96. doi: 10.1515/jee-2015-0014

[15] Saifullah Y, Waqas AB, Yang GM, Zhang F, Xu F. 4-Bit optimized coding metasurface for Wideband RCS Reduction. IEEE Access 2019; 7: 122378-122386. doi: 10.1109/ACCESS.2019.2937849

[16] Ashraf MM, Malik TN. Least cost generation expansion planning in the presence of renewable energy sources using correction matrix method with indicators-based discrete water cycle algorithm. Journal of Renewable and Sustainable Energy 2019; 11: 056301. doi: 10.1063/1.5094540

[17] Schuermans S, Leupers R. Power Estimation on Electronic System Level using Linear Power Models. Switzerland: Springer, 2019.

[18] Lajolo M, Raghunathan A, Dey S, Lavagno L. Efficient power co-estimation techniques for system-on-chip design. In: IEEE 2000 Proceedings Design, Automation and Test in Europe Conference and Exhibition;(Cat. No. PR00537);Paris, France; 2000. pp. 27-34. doi: 10.1109/DATE.2000.840011

[19] Jevtic R, Carreras C. Power estimation of embedded multiplier blocks in FPGAs. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2010; 18: 835-839. doi: 10.1109/TVLSI.2009.2015326

[20] Sunwoo D, Wu GY, Patil NA, Chiou D. PrEsto: An FPGA-accelerated power estimation methodology for complex systems. In: IEEE 2010 International Conference on Field Programmable Logic and Applications; Milano, Italy; 2010. pp. 310-317. doi: 10.1109/FPL.2010.69

[21] Durrani YA, Riesgo T. Power estimation for intellectual property-based digital systems at the architectural level. Journal of King Saud University-Computer and Information Sciences 2014; 26: 287-295. doi: 10.1016/j.jksuci.2014.03.005

[22] Petrov P, Orailoglu A. Low-power instruction bus encoding for embedded processors. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2004; 12: 812-826. doi: 10.1109/TVLSI.2004.831468

[23] Macchiarulo L, Macii E, Poncino M. Wire placement for crosstalk energy minimization in address buses. In: IEEE 2002 Proceedings of the conference on Design, automation and test in Europe; Paris, France; 2002. pp. 158. doi: 10.1109/DATE.2002.998264

[24] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. Computers & Structures 2012; 110: 151-166. doi: 10.1016/j.compstruc.2012.07.010

[25] Goldberg DE. Genetic Algorithms. USA: Pearson Education India, 2006.

[26] Burch R, Najm FN, Yang P, Trick TN. A Monte Carlo approach for power estimation. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 1993; 1: 63-71. doi: 10.1109/92.219908

[27] de Micheaux PL, Tran VA. PoweR: A reproducible research tool to ease Monte Carlo power simulation studies for goodness-of-fit tests in R. Journal of Statistical Software 2016; 69: 1-42. doi: 10.18637/jss.v069.i03

[28] Rice JA. Mathematical Statistics And Data Analysis. USA: Cengage Learning, 2006.

[29] Gong L, Flegal JM. A practical sequential stopping rule for high-dimensional Markov chain Monte Carlo. Journal of Computational and Graphical Statistics 2016; 25: 684-700.