Circuits and Systems, 2016, 7, 1734-1750
Published Online June 2016 in SciRes. http://www.scirp.org/journal/cs
http://dx.doi.org/10.4236/cs.2016.78150

FSM Based DFS Link for Network on Chip

Erulappan Sakthivel[1], Veluchamy Malathi[2], Muruganantham Arunraja[1], Govinndaraj Perumalvignesh[1]

[1]Electrical and Electronics Engineering, The Siliconharvest, Madurai, India
[2]Electrical and Electronics Engineering, Anna University Regional Campus, Madurai, India
Email: vlsisakthivel@gmail.com, vmeee@autmdu.ac.in, researcharunraja@gmail.com, vignesh@siliconharvest.net

Received 8 March 2016; accepted 5 April 2016; published 20 June 2016

Abstract

As low power consumption is the main design issue involved in a network on chip (NoC), researchers are concentrating more on both algorithms and architectural approaches. The conventional Dynamic Frequency Scaling (DFS) and history based Frequency Scaling (HDFS) algorithms are utilized to process the energy constrained data traffic. However, these conventional algorithms achieve higher energy efficiencies, and they result in performance degradation due to the auxiliary latency between clock domains. In this paper, we present a variable power optimization interface for NoC using a Finite State Machine (FSM) approach to attain better performance improvement. The parameters are estimated using 45 nm TSMC CMOS technology. In comparison with DFS system, the evaluation results show that FSM-DFS link achieves 81.55% dynamic power savings on the links in the on-chip network, and 37.5% leakage power savings of the link. Also, this proposed work is evaluated for various performance parameters and compared with conventional work. The simulation results are superior to conventional work.

Keywords

Network-on-Chip (NoC), Dynamic Frequency Scaling (DFS), Finite State Machines (FSM)

1. Introduction

The design complexity of a Network-on-Chip (NoC) is due to the requirement of number of steps involved in the design process, the time to market and design cost problems. Previous NoC researches have been dedicated to increasing the processing speed and analyzing the system-level performance [1]. NoC provides extremely high bandwidth by distributing the propagation delay across multiple switches that may cause a power disturbance in the circuit [2]. This NoC architecture consists of floating-point cores and packet-switched routers at 4 GHz. 15-F04 has mesochronous clocking and various techniques. The 65 nm 100 M transistor is designed to

How to cite this paper: Sakthivel, E., Malathi, V., Arunraja, M. and Perumalvignesh, G. (2016) FSM Based DFS Link for Network on Chip. *Circuits and Systems*, **7**, 1734-1750. http://dx.doi.org/10.4236/cs.2016.78150

achieve peak performance of 1.0TFLOPS at 1 V, while dissipating 98 w [3].

In [4], the Tile 64 processor design considerations such as arbitration, topology and length of physical links, width of physical links, buffer allocation, switching techniques, routing algorithms, and levels of service are addressed in the NoC core. This architectural challenge through "tiled" architecture can be connected by scalable and energy-efficient architecture.

In general, the NoC architecture provides performance degradation by means of more scalability and high power consumption [5]. To achieve lower power consumption and a high operating speed, designers prefer the Dynamic Voltage and Frequency Scaling algorithm (DVFS) [6]. To avoid this complication, History based Dynamic Frequency Scaling (H-DFS) is introduced by Lee *et al*. [7]. In [8], Dynamic Voltage Scaling algorithm (DVS) implemented in a scalable architecture is developed by external circuit combinations of system configurations. The existing low power algorithms are used to reduce power consumption by adjusting the power mode of the links to match the traffic flows. These algorithms fail to act on sudden traffic changes. To overcome this issue, a novel application-driven approach for predicting traffic described in [9]. Here, a novel data structure called Application Traffic Prediction Table (*ATPT*) is used to record the core's outgoing messages (traffic) and the DFS policy is applied to the link. Conventional traffic generation is a novel structural design template that yields a complicated system level design [9]. In modern research, reducing the data access latency and energy consumption in the NoC design plays a vital role. If the traffic is generated, the time constraint increases the energy consumption [10]. The generic traffic model is used to analyze the traffic characteristics of on-chip interconnection application specific network. This model is intended using statistical parameters to attain a high degree of accuracy [11]. Lee *et al.* [8] have presented a variable frequency link for a power-agile network using the clock boost mechanism, and an H-DFS policy is applied to the communication link. This H-DFS method proves maximum power reduction in a NoC link. Various traffic estimation methods are projected in [12] and [13] to estimate different application specific links. The end-to-end traffic prediction method [14] is able to expose the real traffic environment with increased computational time and design complexity [15]. Frequency Tuning approach discussed in [16].

Many researchers utilized Finite state machine (FSM) for system with discrete inputs and outputs. It is said to be the control unit that has a finite number of internal states. It can change its status in some definite manner. Novel FSM is implemented in various domains without any executable samples based on the automaton theory of the control structure with the estimated specifications [17]. FSM with fuzzy transitions are organized [18] to monitor the non-intrusive home appliance and to measure the electrical energy consumption in the domestic environment. The finite state lossless encoder is designed for compressing the individual sequence. It is operated in both invariable and fixed rates. To compress different sequences, the constructive coding theorem is used and it is said to be optimal for all sequences [19]. A comprehensive survey is found in Lee *et al*. [20], in which the authors summarize the principle and working of various finite state machines in automated compositional research. We introduce a FSM for variable frequency selection. The boosting of NoC router is recommended to increase the throughput and reduce latency of an adaptive wormhole router [21]. Cassel *et al.* have introduced New Active Learning Algorithm (NALA). This algorithm is designed for TCP and priority queues. NALA work is a combination of data flow and control behavior [22]. A finite state machine is one of the most used networking, image processing circuit level applications on today's electrical systems. Schiefer *et al.* have introduced the concept of self-configuring FSM, which has memory look-up tables [23] [24]. Zhan *et al.* have introduced a methodology to reduce the energy consumption of NoC without violating the latency deadlines of various applications. Here, feedback control strategy is used to enable dynamic frequency and voltage scaling on the network routers in conjunction with the energy optimization algorithm [25]. To design this feedback control strategy, the extra energy is required, which may affect the other performance of NoC. Wang *et al.* has introduced Power-efficient Network Calculus based (PNC) method to reduce the power consumption of NoC [26]. This method uses power-gating technique to reduce the active buffer size. It also utilizes both DFS and DVS techniques to reduce the voltage-frequency of each voltage-frequency island. Experimental results show that this method can save at most 50% of the total power consumption. Huang *et al.* has introduced power and latency aware DVFS links. The above conventional methods are focused only on power or latency. In order to achieve massive improvement in latency and power, we introduce an FSM based DFS concept. The boosting of NoC router is used to increase throughput and to reduce latency of an adaptive wormhole router [27].

We propose a novel FSM based strategy, which has extremely low latency and power for frequency scaling to provide traffic aware power reduction solution. This technique is composed of the following strategies: 1) FSM based observer; 2) frequency selection (**Table 3**); 3) FSM-DFS link performer; 4) Clock Distribution Network (CDN). In this paper, we present a control block that utilizes a dynamic frequency scaling (FSM-DFS) method along with adaptive strategies to avoid the process variations and reduce power consumption. CDN is the clock splitting mechanism which is used to validate clock and data to progress the clock to the router unit in NoC. The dynamic power consumption in CDN is reduced by the proposed adaptive clock gating scheme.

In this paper, we encourage the use of FSM based DFS link. Traffic estimator is used to estimate the traffic rate and according to the traffic id is passed to the router unit. Next, a FSM-DFS algorithm is proposed and applied to the NoC link. Finally, the power saving is achieved in on-chip interconnection network. To the best of our knowledge, this is the chief investigation of power reduction for on-chip interconnection network based on the clock boosting mechanism. It is used to predict the impact of DFS policy on system performance. This strategy is also used to reduce the complexity and improve the overall performance under various traffic scenarios. This work proposes a distributed network under various traffic scenarios, which can operate individual routers at different frequency levels effectively.

The rest of this paper is organized as follows. Related work is introduced in Section 2. The system model is introduced in Section 3. The problem constraints are discussed in Section 4. The proposed FSM based DFS link is discussed in Section 5. The performance measure analytical model is discussed in Section 6. The experimental results are presented in Sections 7. Finally, conclusions are drawn in Section 8.

2. Related Work

Low power network on chip design has become a vital paradigm in the CMOS technology. Since network on chips are likely to consume a considerable part of the total chip power, the design of low power on chip processor offers a general approach for overall performance improvement. NoC consumes a significant portion of total chip power in multicore systems. Some recent researches in a low power network on chip design [26]-[28] architectures are validated to be 10% - 36%. Therefore, necessities of latency and power-aware NoC lead to a serious issue in designing low power multicore systems. In order to provide those provisions, designers introduced several dynamic voltage and frequency scaling algorithms with application and traffic aware system. There is a single way to hold the power concern using conventional scaling algorithm. Conventional researchers have proposed DFS for general purpose [29] [30] and multimedia applications [31]. However, the results of conventional works focus on either processor or cache power reduction. Recently, frequency scaling algorithms on NoCs are proposed to further reduce the additional components in NoC like interconnect and core power dissipation. Some past work offers similar methods of NoCs [27] [28] [32] by scaling the voltage/frequency of individual routers, links, or the whole networks. However, these results still focus on general-purpose domain, whereas in multimedia application the traffic aware systems are the emerging research area. The conventional designs are examined under heavy traffic mode, the major problem in performance degradation. Those works are specifically focused on either power or latency, even both performances of NoC. In order to bring better solutions in terms of an end to end delay and other performance parameters, we introduce an FSM-DFS for NoC.

3. System Model

The key idea behind boosting of NoC router mechanism is to use frequency selection table. The functional diagram of FSM based DFS is shown in **Figure 1**. This FSM-DFS has the components such as an FSM based observer, frequency selection table and router. In this method, FSM-DFS is used to perform better than the conventional low power algorithm.

3.1. FSM Based Observer

The FSM based observer will collect the traffic information from the router. This will provide traffic ID to the frequency selection table.

3.2. Frequency Selection Table

Frequency selection table takes place with respect to traffic ID. The corresponding frequency is selected via frequency selection table to the router. The traffic estimator explicitly organizes the NoC elements by making the problem formulation more flexible and efficient. The output of the FSM is given in the frequency selection table, there it selects the corresponding frequency depending on FSM based DFS and corresponding frequency

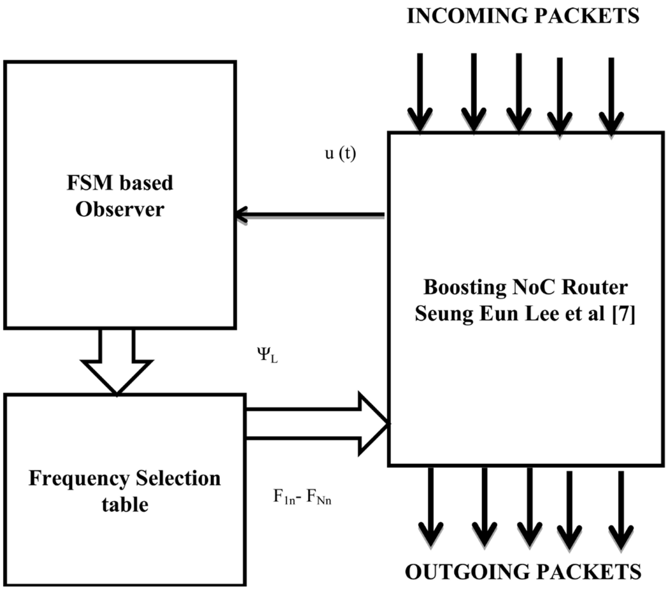

INCOMING PACKETS
FSM based Observer
u (t)
Boosting NoC Router Seung Eun Lee et al [7]
Ψ_L
Frequency Selection table
F_1n- F_Nn
OUTGOING PACKETS

Figure 1. Functional diagram of FSM based DFS.

is given to the router.

3.3. System Performance Model

Dynamic frequency scaling and history based dynamic frequency scaling are used to observe the power consumption, latency and energy consumption. For various traffic benchmarks, the traffic information, *tr* occupies an N_{tr}-tile region, where the frequencies of the tiles are $f_1, f_2, \cdots, f_{tr}$.

Due to the traffic information, the tile regions can be overlapped. Let the cache memory is used for many functional modules in the core. This traffic estimator is assessed to map only a single tile region of the core. The average traffic information reaches the traffic estimator and it is represented in the *T*. Thus, we have

$$\sum_{tr=1}^{T} N_{tr} = N , \tag{1}$$

where *N* is the total number of tiles in NoC core.

The performance of each core under various traffic benchmarks is observed. In execution cycles is modeled in terms of frequencies of its region/tiles, as follows.

The execution time is measured in cycles, which is a new form of regression model [33]. This execution time is modeled from Bishop *et al.* which is represented in terms of frequency of tiles in NoC core (see Equation (2)).

$$Cycle = T_{cycle}\left(f_1, \cdots, f_{tr} \right). \tag{2}$$

Using this execution time, we introduced a new model in the cycle (see Equation (3)). This model is refined from Bishop *et al.* according to various traffic benchmarks. Due to the traffic, threshold regression may occur. To evade this issue, the proposed model is introduced which satisfies the regression error.

$$Cycle = \sum_{tr=1}^{T} \beta eta_i \cdot \sqrt{f_i} \tag{3}$$

where βeta_i is the regression coefficient with respect to frequency of the region/tiles, f_i and *T* is the total traffic information.

3.4. Two Levels of Dynamic Power Model

Let us assume that NoC cores work with the same voltage level and the dynamic power of NoC core under various traffic benchmarks (N_{tr}) expressed as follows (see Equation (4)):

$$P_q = \sum_{j=1}^{N_{tr}} \beta_j \cdot CE_j \cdot f_j \cdot Volt^2 = \sum_{j=1}^{N_{tr}} X_j \cdot f_j \tag{4}$$

where β_j is the switching activity, CE_j is the effective capacitance, *Volt* is the voltage, $b_i \equiv CE_j \cdot f_j \cdot Volt^2$.

Let us assume that NoC cores work at the variable voltage level and variable frequency. The dynamic power of a NoC core under various traffic benchmarks (N_{tr}) is expressed as follows (see Equation (5)):

$$P_{tr} = \sum_{j=1}^{N_{tr}} \beta_j \cdot CE_j \cdot f_j^3 / K^2 = \sum_{j=1}^{N_{tr}} Y_{dest} \cdot f_j^3 \tag{5}$$

where K is a constant. Similarly, $Y_i \equiv \beta_i \cdot CE_i / K^2$.

4. Problem Constraints

With the previous models, the energy planning problem targets to reduce the peak latency in the input power budget. With T is various traffic scenarios by which all occupying an N_{tr}-tile region, we have

$$\sum_{tr=1}^{T} w_{tr} \cdot P_{tr} = Power \tag{6}$$

where each P_{tr} is the power budget for application *tr* at a given time *t*, and w_{tr} is user defined priority weight for various traffic scenarios.

In order to provide better solutions, the new objective function is optimized subject to the following constraints:

1) Traffic constraints: The distribution ratio between a given pair (source, destination) should be equal to 1 under average traffic (low to high) mode.

$$\sum_{num=1}^{L_{source,dest}} router(src, dest, num) = 1 \quad \forall (src, dest) \in C$$

$$router(src, dest, num) \geq 0 \quad \forall source, dest \in [1, N_{tile}],\ k \leq L_{src,dest} \tag{7}$$

where *src* is the source connected to the transmitter side of the router, *dest* is the destination connected to the receiver section of the router, *num* is the number of iterations, $L_{src,dest}$ is link of the *src*, *dest* connected to NoC router and C is the constraint.

2) Bandwidth constraints: The cumulative bandwidth used for a link should not surpass the link capacity.

$$Bandwidth = injection\ rate \times size \times frequency. \tag{8}$$

Assume that T_{cycle} is a cycle time and *src*, *dest* is a link in the NoC, a path $link(src, dest, num)$ will traverse this link if $link(src, dest, num) \in T_{source} \cap T_{dest}$. So, we have,

$$\sum_{\forall link(src,dest,num) \in T_i \cap T_j} \frac{router(src, dest, num) \times pi(src, dest) \times S_{bit}}{T_{cycle}} \leq C_{src,dest} \tag{9}$$

where *pi* is the packet injection rate.

3) End to end delay constraint: In order to examine the results of various benchmarks, we define QoS requirements in terms of speed and end-to-end (ETE) delay for each class of service [34]. ETE delay was measured in clock cycles of the link.

In order to solve the energy budget problem under various traffic scenarios, FSM based model is formed to regulate these problems with respect to various traffic information. The corresponding frequency is allotted to the corresponding N_{tr}-tile region**.**

5. Proposed System

The FSM-DFS is a traffic aware performance improvement solution to achieve both latency and power consumption. In this work, we model the procedure with four states, namely selection of processor and approximate frequency (same as HDFS), traffic observation, traffic ID departure, and desired frequency using a Mealy machine model in the router.

5.1. FSM

In FSM model, the output circuit is obtained in various sets of states (*i.e.* all output is defined as a state). A state register is used to hold the state of the machine. A next state logic decodes the next state and output register provides the output of the machine. The entire algorithm gives a detailed explanation in one process with the reduced hardware system in FSM.

5.2. FSM-DFS Link

The proposed FSM has a state diagram to construct Barn's benchmark with 16 particles, which split into four terms, namely t1, t2, t3, and t4. The selection process is considered as a selection of processor and appropriate frequency to obtain desired frequency. The traffic observation is used to examine the traffic during processing. Also, the traffic ID will be sent in order to place the desired frequency in the router. We have two input signals as clock and reset. When the positive edge of clock button is set, the machine will continue to work. When the reset button is pressed, the machine will come back to its initial state.

5.3. Design Methodology

The state diagram has four states as mentioned earlier. If the reset button is pressed initially, machine will be set to select the processor/frequency/traffic and it is considered as the initial state of the process. Then, the user should select the traffic to distribute. This is used to select any traffic within t1, t2, t3, and t4. The processor will verify the selected traffic information. If the traffic is selected as per the user need, the traffic ID will be sent. Finally, the exact frequency of the selected traffic is generated in the process. Hence, the particular frequency is placed in the router. If the traffic is not available in the processor, then the control unit will insist for the selection process, after getting reset. The complete methodology is explained in the flow diagram as shown in **Figure 2**.

5.4. Proposed FSM Model

Proposed FSM is based on State Assignment Process (SAP) which is targeting a low power and effective communication link for NoC. The two stage operation of the proposed FSM model is as follows: (a) The traffic information id assignment stage (b) The frequency boost performing stage.

Traffic information id assignment stage: In this stage, FSM based SAP assigns traffic information id to all possible pairs of states, which is an estimate of the similarity states to one another. This stage is used for the computation of traffic information id, which is represented in **Algorithm 1**.

To compute this id, all the state sets are examined first. For state$_{num}$, the edge traffic information (under various traffic modes) of state sets is checked from (1, num) to (num-1, num).

Let the num-1 states are distributed between all the nodes in the router. Thus, no two nodes get the similar sets without any conflicts. In a router, every node updates the traffic information independently.

Algorithm 1: The traffic information id assignment stage.

Result: Computation of the traffic information in all states:

```
1) For x = 1 to Ns do,
2) For y = 1 to num-1 do,
3) Compute the Traffic information weight ($s_{src}$, $s_{dest}$),
4) End,
5) End.
```

The Frequency scaling performing stage: The proposed frequency scaling stage involves assignment of unique frequency pattern to each state of the FSM model. This state is represented by simple counter and controller logic. Our proposed work implements FSM model using the split and performer modules as well as parallel operations.

The parallel concept is already done in many research for low power and high speed operation. We took the basic information of parallel operation from Samman *et al*. [35]. A common configuration is preserved in split and frequency scaling performer and parallel operations. The principle of parallel operations applies simultaneously in the frequency scaling operations.

On the other hand, the Split and Frequency scaling performed with respect to traffic ID lets routers make parallel operation. We use the same default Initial and Stopping Frequency boost in the router using of FSM model. At higher traffic rate, variable range frequency scaling is accepted with respect to traffic threshold. A history based dynamic frequency scaling is introduced with respect to traffic state, where the traffic ID sending and frequency scaling operation are performed with the router. At Lower Traffic rate, low range frequency scaling is accepted. A dynamic frequency boost is introduced with respect to traffic state.

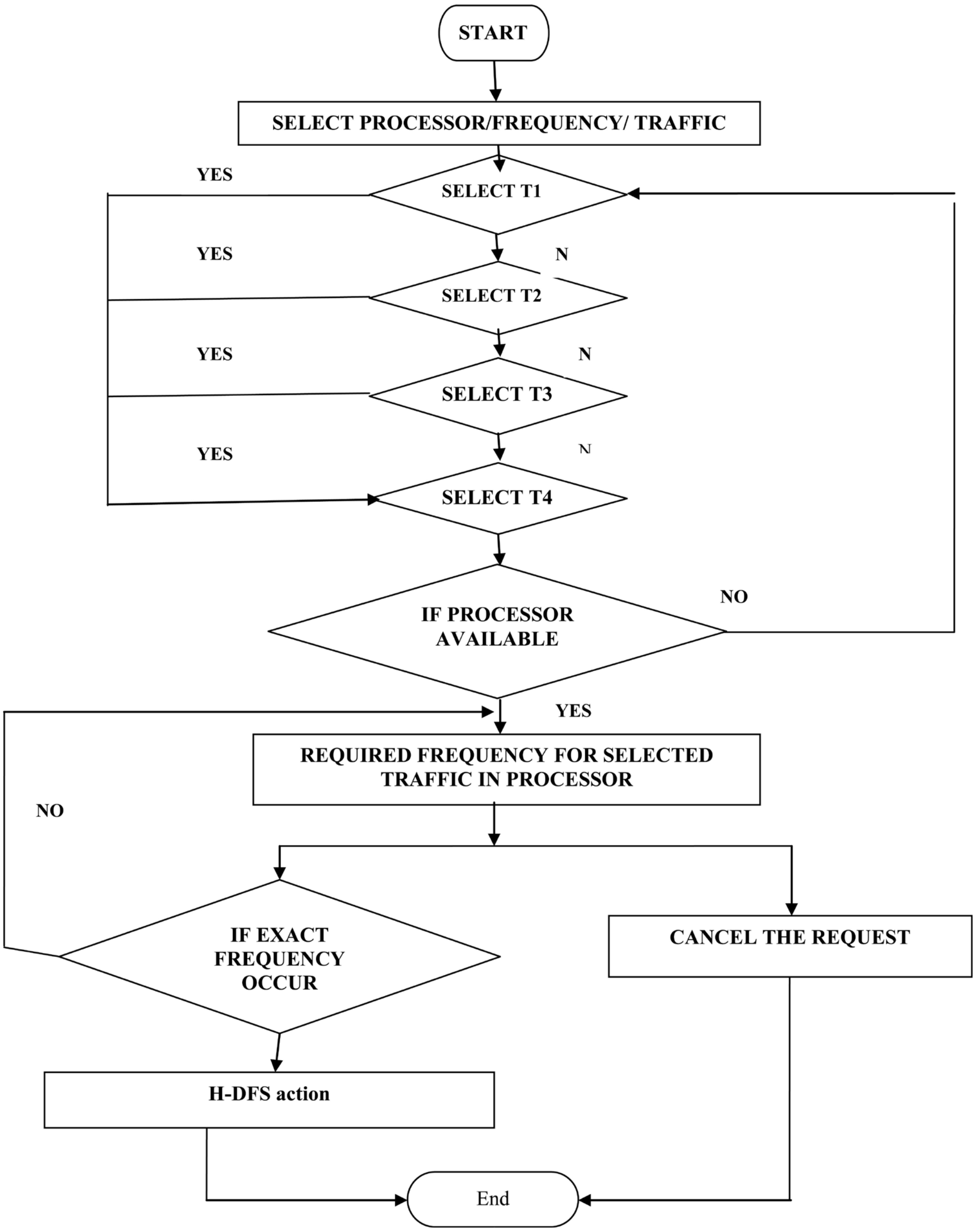


Figure 2. FSM-DFS flow chart.

```
Algorithm 2: The frequency boost performing algorithm by "split and frequency boost performer".
Output: The frequency boost performing by "split and frequency scaling performer".
1) Set the initial and stopping frequency boost process in the router,
2) Set the NoC Router parameter and initial traffic id,
3) Examine the initial no. of moves per router,
4) Split the state space into $S_{router}$ partitions,
5) While Current Traffic > Traffic_threshold do,
6) For $i$ = 1 to iteration do,
7) (Iteration is the max. no. of cores in router),
8) Calculate current traffic information to all the nodes,
9) Check input traffic,
10) Check whether the traffic information is already assigned,
```

11) If FSM states are allotted then,
12) Interchange the frequency of the two states,
13) Else,
14) Allocate new frequency to the states,
15) End,
16) Compute the current state of the local port in the router,
17) Admit or discard the interchange,
18) Updation,
19) End.

The general algorithm of the frequency boost performing algorithm is as follows:

1) Start with an initial and the stopping frequency boost process in router of states.
2) For a given traffic input, select two states at traffic threshold and assign frequency boost process or interchange their frequency of current state and the ideal state.
3) Compute the frequency change of each core.
4) These frequency scaling and Traffic estimation process are managed by the state of FSM model.
5) Admit the interchange for a lower traffic condition. Allow frequency boost process to be accepted, even if it higher traffic condition in the router.
6) Repeat steps 2-5, until a traffic id is getting into less than zero. Then lower traffic process is accepted and the corresponding frequency boost operations take place.
7) Stop, if the traffic id attains zero.

6. Performance Measure—Analytical Model

We examined the performance parameters such as delay, data rate, energy and static power consumption analysis in a network-on-chip. To have a better view, the performance parameter model is summarized here.

1) To estimate the latency flow, it is necessary to evaluate the waiting time of packets for routers.
2) Bandwidth estimation.
3) The power consumption and link power are calculated recursively for each communication path starting from the receiver section.
4) Given the energy delay product among the cores and routing algorithm, the energy consumption for each node in router is determined.
5) End to end delay and communication density are also modeled, with respect to each communication path starting from the receiver section.

6.1. Latency

The latency of a link is the addition of the latency to traverse the Frequency Boosting Mechanism (FBM) in the router and link latency. The latency of the link is defined by the frequency at which the link is operated [36]. Let router_distance denote the distance in mm a signal can traverse in 1 ns. This can be determined based on the design's technology core. Finally, the latency of a link is given by

$$N_{source,destination} = FBM\_latency + \frac{1}{F} \times \frac{lenght_{(source,destination)}}{router\_distance} \tag{10}$$

$$F = \begin{cases} frequency(source), \\ frequency(destination), \end{cases}$$

where F denotes the frequency of the link and it depends on the where the FBM is placed on the link, and $length_{source,destination}$ denotes the length of the link in mm.

6.2. Bandwidth

The bandwidth of a link is given by the product of the link width and frequency of operation of the link [36];

$$bw_{source,destination} = F \times link\_width_{source,destiation} \tag{11}$$

6.3. Power

6.3.1. Link Power

Link power is estimated from tool the standard link power estimation is followed in the recent simulator [37] for a NoC router. This power model considers the cross-coupling effect for N-wire interconnect, and also we can determine the total power for an N-wire link per unit length as follows: gate leak wire bias short.

$$P_{link} = \frac{1}{2} N_w V_{sv}^2 \left(C_{se}\alpha_{sa} + C_{Co}\alpha_{Co}\right) f + N\tau\alpha_s VI_{sh} \cdot f + N \cdot \left(VI_{bi,w} + VI_{le,g}\right) \tag{12}$$

where N_w is the total number of wires in the link, C_{se} and C_{co} are the self and coupling capacitance of a wire and neighboring nodes respectively, α_{sa} is the switching activity on a wire and α_{Co} is the switching activity with respect to the adjacent wires, τ is the short circuit period, V_{sv} is the supply voltage and I_{sh}, $I_{bi,w}$ and $I_{le,ga}$ are currents.

6.3.2. Static Power Consumption

Static power is the power dissipated by a gate or a wire when it is ideal or in an active state. The static power is mostly inclined by the structure of the architecture [37]. The static power dissipation can be more precise by the equation:

$$P_{static\ power} = VI_{b,w} + VI_{le,ga}. \tag{13}$$

6.4. Energy Consumption

We assume the energy consumption of each core of NoC *num* ($E_{NoC-num}$) is available after task mapping. In wormhole routing, each input information is distributed into several flits. For every input information, the head flits set up the way bearings for the body and the tail flits [38]. The representation of Parameters and Symbols are indicated in **Table 1**.

Total energy consumption for processing a single packet in router *i* is given by:

$$\Delta EC_{NoC-n} = \left(EC_{dispatch-router} + EC_{dispatch} + EC_{saw}\right) \times N_{Flit} + EC_{rp} + EC_{sel} + EC_{vca}. \tag{14}$$

6.5. End-to-End Delay Formulation

The End-to-End flow delay $L_{source,destination}$ of a specific flow $f_{source,destintination}$ is made up of three measures such as [39] 1) The time at the source s (T_{source}), 2) Information transfer time ($\beta_{source,destination}$) and 3) the Path Acquisition Time ($PAT_{s,d}$). It is expressed as

$$L_{source,destination} = T_{source} + \beta_{source,destination} + PAT_{source,destination}. \tag{15}$$

7. Result and Discussion

To evaluate and compare the performance of the proposed FSM-DFS interface with the conventional DFS and H-DFS interface, the components of interconnection networks are modeled considering by Shang *et al.* (2003) (including 7 × 7 NoC, 400 (1×) clock frequency). The NoC VHDL synthesized code is made to evaluate 45-nm

Table 1. Parameters and symbols.

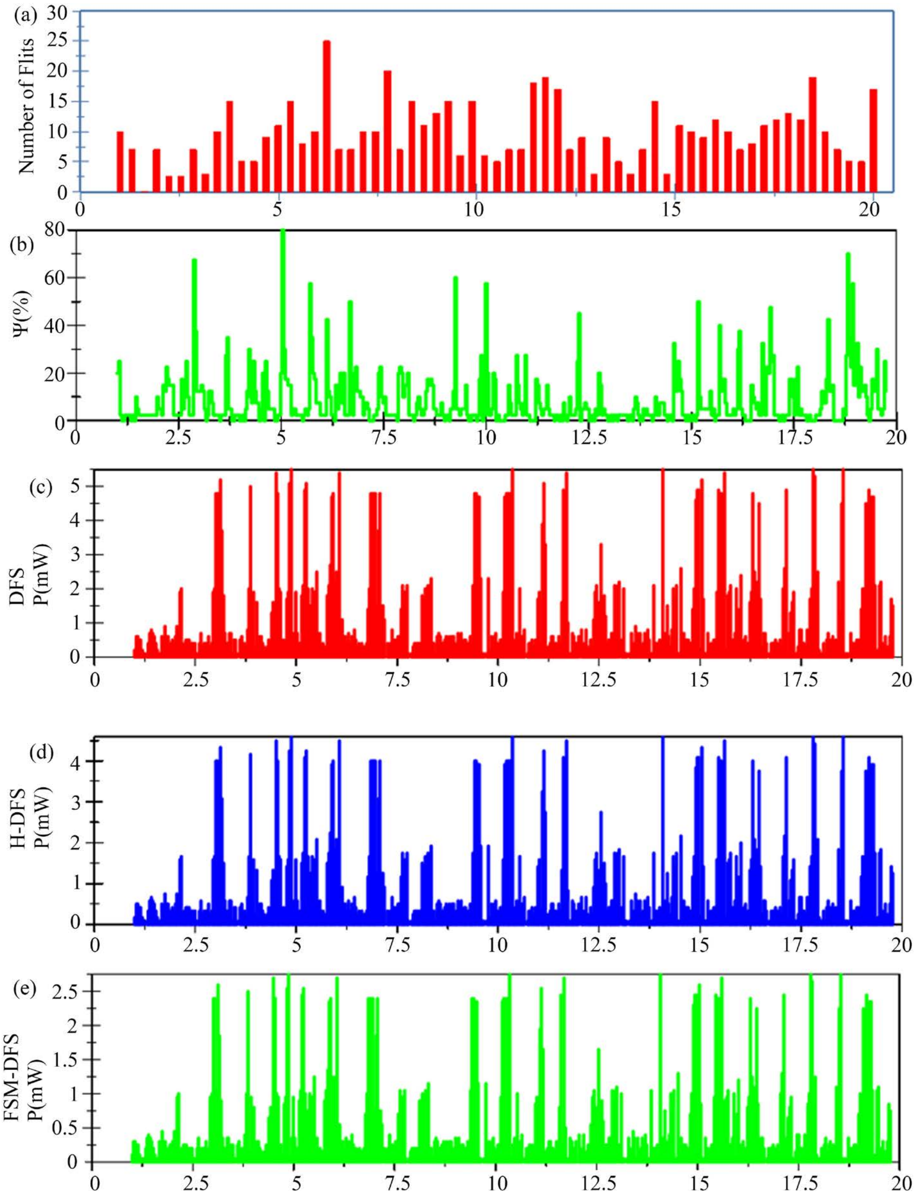
Symbols	Tile energy parameters
EC_{NoC-n}	Energy of NoC of the nth tile
$EC_{dispatch}$	Energy of dispatching a flit in the router
$EC_{buffer-router}$	Energy of buffer for a flit in the router
EC_{saw}	Energy of switch allocation energy
N_{flit}	Number of flits
EC_{rp}	Energy of routing process
EC_{op}	Energy of output port
EC_{vca}	Energy of virtual channel allocation energy

TSMC CMOS technology under 1-GHz operating frequency, a supply voltage of 1 V and a switching factor of 0.5. The RTL description is synthesized to the gate level net list with a Synopsys design compiler [40]. A power analysis is carried out using the Synopsys Prime Time PX tool [41].

The benchmark from the SPLASH-2 (Woo *et al*. 1995) suite is used to obtain the workload for the NoC interface system [41]. The experimental benchmark specification for this proposed work is as shown in **Table 2**. **Figure 3**

Table 2. SPLASH-2 specification [41].

Work load	Input parameter	Range & Delta	Description
Barnes	nobody	[16 K, 16 M], Δ = *2	The number of particles to generate under a plummer
Fmm	NP	[1 K, 4 M],],Δ = *2	Number of particles
Ocean	T	[14,400, 28,800], Δ = *2	Time steps in seconds
Radiosity	AE	[2000, 3000], Δ = +500	Area epsilon
Raytrace	File	{“teapot.env”}	Input image file
Volrend	Step	{4, 10, 20, 50, 100, 500, 100}	Rotate steps



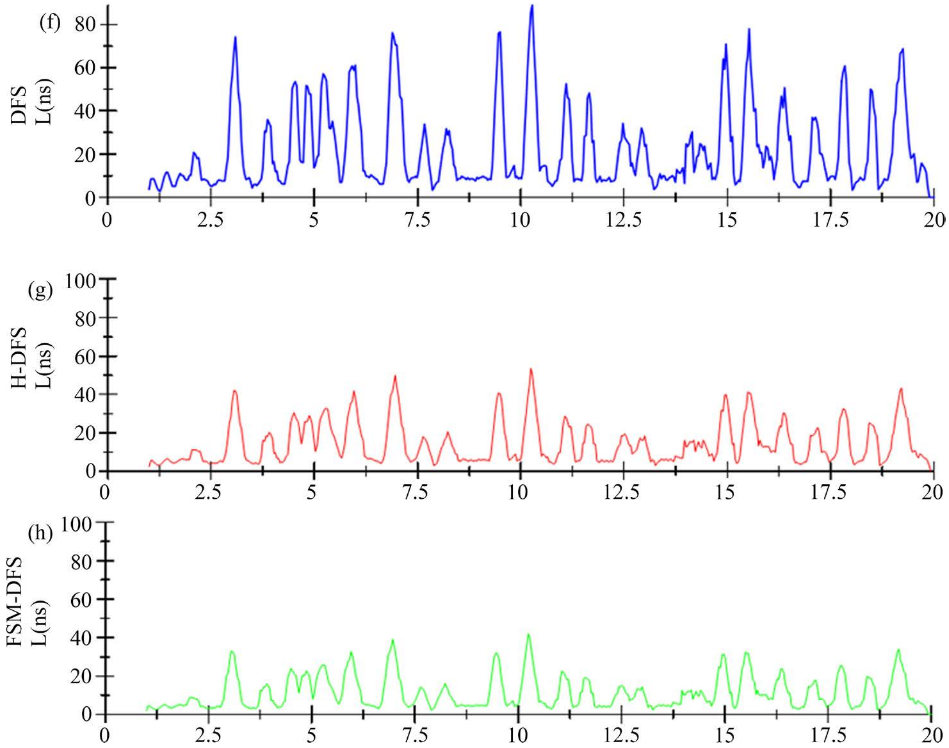


Figure 3. Simulation result of the various link policy when control period has eight cycles: 1) injected workload (**Figure 3(a)**); 2) link utilization estimation (**Figure 3(b)**); 3) DFS power consumption (**Figure 3(c)**); 4) HDFS power consumption (**Figure 3(d)**); 5) FSM-DFS power consumption (**Figure 3(e)**); 6) DFS latency (**Figure 3(f)**); 7) H-DFS latency (**Figure 3(g)**); 8) FSM-DFS latency (**Figure 3(h)**).

shows simulation results of various link policies with control periods of eight cycles: 1) injected workload (**Figure 3(a)**); 2) link utilization estimation (**Figure 3(b)**); 3) DFS power consumption (**Figure 3(c)**); 4) H-DFS power consumption (**Figure 3(d)**); 5) FSM-DFS power consumption (**Figure 3(e)**); 6) DFS latency (**Figure 3(f)**); 7) H-DFS latency (**Figure 3(g)**); and 8) FSM-DFS latency (**Figure 3(h)**).

When the router transmits data with specific traffic injection rate, interface link will dissipate static and dynamic power. The performance of conventional and proposed low power link with respect to dynamic and leakage power under different terminals such as the traffic generated, the traffic estimator, the router, the input buffer, the output buffer, and links is estimated at 45 nm technology and these results are plotted in **Figure 4**. Bandwidth sensitivity offers 14.84% system/instruction throughput improvement.

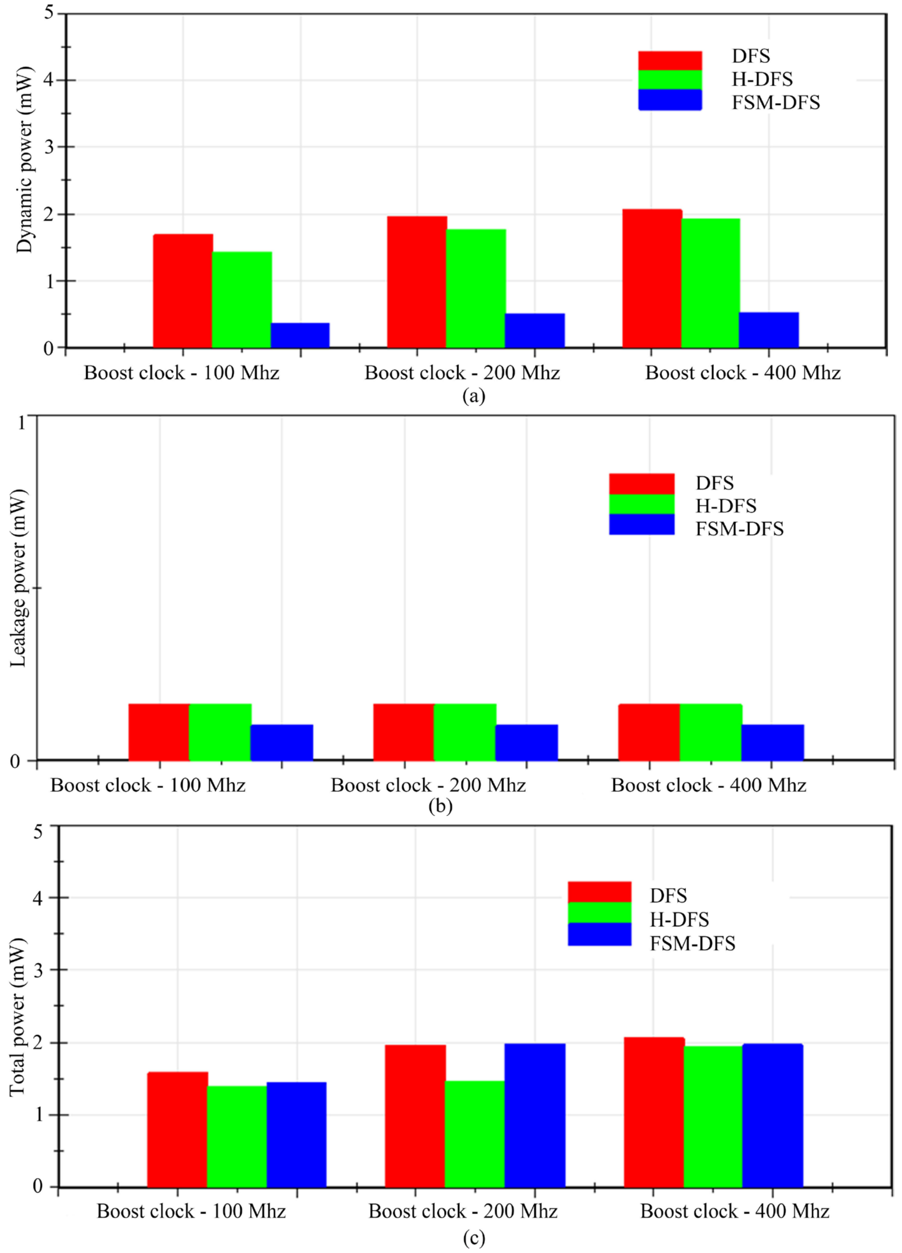
Latency of peak and average are observed. For a power optimized interface link, power-agile algorithm should offer very high throughput and low average latency for high flit rate data transmission. The FSM-DFS link characteristics for each boosting clock frequency are obtained by simulation and are summarized in **Table 3**.

The 1× boosting router finishes the entire packet transmission in 24:34 ms, spending more time than 2× and 4× boosting router. DFS method has the highest average and peak latency. When compared to FSM-DFS, it has 42.6 ns/flit for 1× boosting, 2× and 4× boosting have 8.9 ns/flit and 8.1 ns/flit. Similarly, 4× boosting router is much better compared to the 1x boosting router in terms of latency.

The overall power consumed by 1× boosting in DFS is 1.85 mw and the FSM-DFS system consumes 0.39 mw for the same 1× boosting. These experimental results demonstrate the feasibility of clock, boosting router in the FSM-DFS link for a power-aware of-chip interconnection network for a NoC platform.

Table 4 summarizes the experimental results of the DFS, FSM-DFS and history based DFS policy for varying the control period from 8 to 128 cycles of the 1× clock. Under varying control period, the physical parameters such as average latency, peak latency, end time, dynamic power, leakage power and total power are measured and these results are plotted in **Figure 5**. The FSM-DFS method is compared with previous DFS (Seung Eun Lee *et al*. 2009) and H-DFS.

The DFS has the highest average latency of 24.06 ms/flit for the 8 control period. Similarly, the FSM-DFS



(a)

(b)

(c)

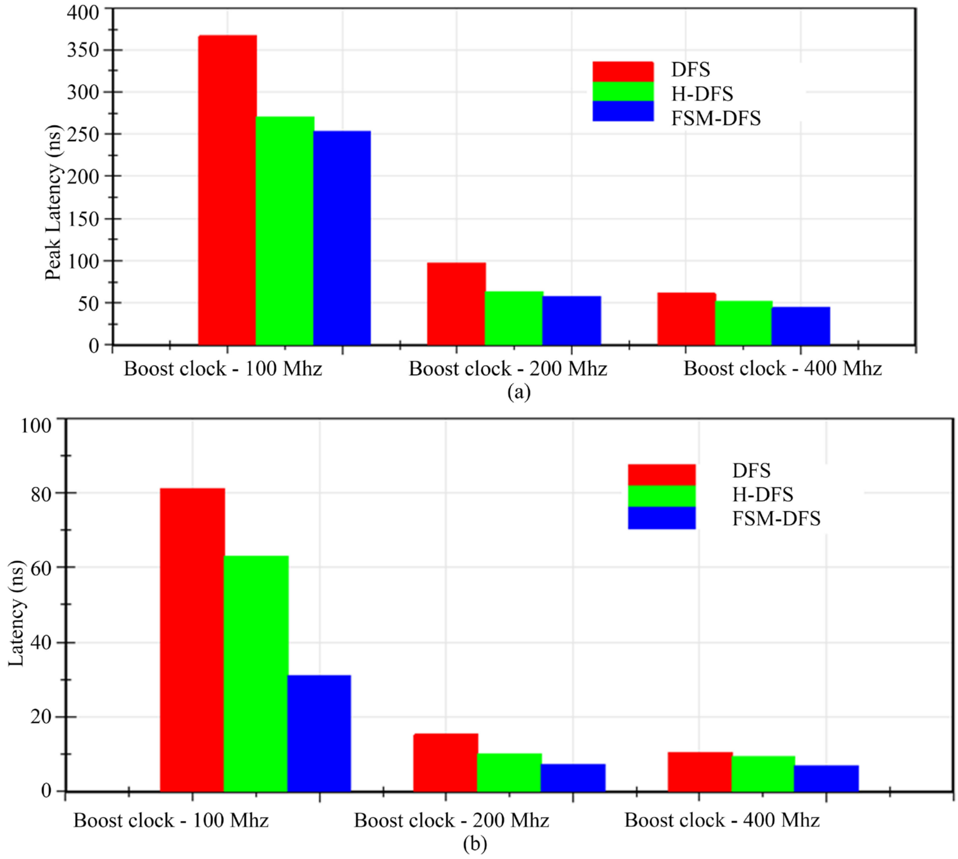
Figure 4. Dynamic, leakage and total power comparison.

Table 3. The comparison of FSM-DFS link characteristics.

	Boost clock (MHz)	Average latency (ns/flit)	Peak latency (ns/flit)	End time (ms)	Dynamic power (mw)	Leakage power (mw)	Total power (mw)
DFS	100 (1×)	81.2	367	24.34	1.69	0.16	1.85
H-DFS	100 (1×)	63.1	270	24.34	1.42	0.16	1.58
FSM-DFS	100 (1×)	42.6	253	24.34	0.31	0.16	0.37
DFS	200 (2×)	15.3	97	24.06	1.96	0.16	2.02
H-DFS	200 (2×)	10.2	63	24.06	1.77	0.16	1.93
FSM-DFS	200 (2×)	8.9	57	24.06	0.37	0.16	0.53
DFS	400 (4×)	10.4	61	24.05	2.06	0.16	2.18
H-DFS	400 (4×)	9.4	52	24.05	1.92	0.16	2.08
FSM-DFS	400 (4×)	8.1	44	24.05	0.41	0.16	0.57

Table 4. Summarizes the experimental results.

Methods	Control period	Average latency (ns/flit)	Peak latency (ns/flit)	End time (ms)	Dynamic power (mw)	Leakage power (mw)	Total power (mw)
DFS	8	24.06	63	24.05	2.06	0.16	2.22
H-DFS	8	24.3	116	24.05	1.93	0.16	2.09
FSM-DFS	8	19.7	106	20.05	0.38	0.10	0.48
DFS	16	24.07	122	24.05	2.09	0.16	2.15
H-DFS	16	25.2	146	24.05	1.91	0.16	2.07
FSM-DFS	16	20.4	123	24.05	0.39	0.10	0.49
DFS	32	28.2	130	24.23	2.01	0.16	2.17
H-DFS	32	27.6	187	24.23	1.90	0.16	2.06
FSM-DFS	32	26.3	173	24.23	0.41	0.10	0.51
DFS	64	34.7	213	24.34	1.98	0.16	2.14
H-DFS	64	33.2	242	24.34	1.88	0.16	2.04
FSM-DFS	64	31.5	236	24.34	0.43	0.10	0.53
DFS	128	49.98	245	24.05	1.97	0.16	2.03
H-DFS	128	48.21	265	24.05	1.91	0.16	2.07
FSM-DFS	128	46.62	253	24.05	0.44	0.10	0.54



(a)

(b)

Figure 5. Peak latency and average latency.

model obtains a value of 19.7 ms/flit. By varying the control period to 128, the DFS has higher latency of 49.98 ns/flit than FSM-DFS model. The FSM-DFS interface gives a superior result in terms of energy consumption and total power, when compared to the conventional links in the same experimental setup.

For the performance of area comparison (similar to Volos *et al.*), this proposed model is estimated in ORIAN 2.0. The observed result is tabulated in **Table 5**. For various modules such as Link, Buffer, Cross bar, this area comparison is already reported using Mesh based core and CCNoC in Volos *et al.* [28]. We examined conventional strategy model and proposed FSM based NoC. With the outcoming results, we prove the proposed system is giving better performance than conventional work. Energy-delay product comparison with conventional NoC is reported in **Table 6**. Also, we compared our new model (FSM-DFS-NoC) with the conventional architectures like Mesh, Homogeneous, Heterogeneous, CCNoC, HDFS-NoC. This proposed work contributes enhanced results. Also, we examined end to end delay with various flits using conventional and proposed strategy as tabulated in **Table 7**. Likewise, end to end delay and buffer size is also compared with conventional work as organized in **Table 8**.

The static power, overall dynamic power and energy of the three low power interface links are clearly estimated for NoC and listed in **Table 9**. The overall simulated results show that FSM-DFS interface attains 37.5% leakage power saving, 81.55% dynamic power saving and 61.8% energy savings in NoC. Finally, the static power, overall dynamic power and energy under various benchmark results are observed and listed in **Table 10**.

Table 5. Area comparison with conventional NoC.

Module	[28]			**Proposed work**
	Mesh	CCNoC	HDFS	FSM-DFS based NoC
Link	0.5	0.75	0.5	0.25
Buffer	1.0	0.75	0.66	0.50
Cross bar	0.5	0.25	0.20	0.15

Table 6. Energy-delay product comparison with conventional NoC.

Methods	**Energy delay product (%)**
Mesh [28]	137
Homogeneous [28]	116
Heterogeneous [28]	121
CCNoC [28]	100
HDFS based NoC	90
FSM-DFS based NoC	60

Table 7. Comparison of end to end delay for barnes.

Flit no	**25**	**50**	**75**	**100**
H-DFS	44	48	49	53
Proposed work	22	27	29	31

Table 8. End to end delay and buffer size.

Buffer Size	**2**	**4**	**8**	**16**	**32**	**64**	**128**
H-DFS	530	450	300	120	100	50	50
Proposed work	430	380	210	90	70	40	30

Table 9. Average power comparison of power and energy.

Work	**Leakage power (mw)**	**Dynamic power (mw)**	**Energy (pj)**
DFS	0.18	2.87	58
H-DFS	0.16	2.06	55
Proposed work	0.10	0.38	21

Table 10. Power comparison of power and energy for various benchmark.

Methods	Barnes			Fmm			Ocean			Radiosity			Raytrace			Volrend		
	Leakage power (mw)	**Dynamic power (mw)**	**Energy (pj)**	**Leakage power (mw)**	**Dynamic power (mw)**	**Energy (pj))**	**Leakage power (mw)**	**Dynamic power (mw)**	**Energy (pj)**	**Leakage power (mw)**	**Dynamic power (mw)**	**Energy (pj))**	**Leakage power (mw)**	**Dynamic power (mw)**	**Energy (pj)**	**Leakage power (mw)**	**Dynamic power (mw)**	**Energy (pj)**
DFS	0.18	2.87	63	0.18	2.88	64	0.18	2.89	63	0.18	2.88	63	0.19	2.93	65	0.18	2.94	65
H-DFS	0.16	2.06	55	0.16	2.08	56	0.16	2.09	55	0.16	2.10	54	0.17	2.12	56	0.16	2.15	57.2
FSM-DFS	0.10	0.38	21	0.10	0.38	21	0.111	0.37	22	0.12	0.37	22	0.10	0.35	20	22	0.34	20

8. Conclusions

The power optimization technique is achieved in NoC by successfully presenting the FSM based DFS link for NoC in algorithmic level. The proposed FSM based DFS interface is compared with the conventional low power interfaces such as DFS and H-DFS. Their performance metrics like dynamic power, leakage power, average throughput, average latency, and average energy per useful flits are evaluated using 45-nm technology. The experimental results reveal that the FSM-DFS is the finest power optimization interface for NoC platform.

In this paper, we proposed a FSM based DFS link to achieve low power in NoC. The traffic estimator is used to estimate the traffic rate of workload on the NoC. Based on the traffic, appropriate working frequency can be set to the link by DFS policy. The implementation of the proposed FSM-DFS policy is discussed in detail. An experimental result shows that the proposed policy attains 81.55% dynamic link power reduction, 37.5% leakage power reduction and 61.8% energy savings in NoC. In this way, the proposed work is examined using various benchmarks. All the simulation results of the FSM based DFS link for NoC contribute enhanced results, when associated with the conventional work.

References

[1] Benini, L. and De Micheli, G. (2002) Networks on Chips: A New SoC Paradigm. *Computer*, **35**, 70-78. http://dx.doi.org/10.1109/2.976921

[2] Martin, A.J. and Nyström, M. (2006) Asynchronous Techniques for System-on-Chip Design. *Proceedings of the IEEE*, **94**, 1089-1120. http://dx.doi.org/10.1109/JPROC.2006.875789

[3] Pande, P.P., Grecu, C., Jones, M., Ivanov, A. and Saleh, R. (2005) Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures. *IEEE Transactions on Computers*, **54**, 1025-1040.

[4] Vangal, S., Howard, J., Ruhl, G., Dighe, S., Wilson, H., Tschanz, J., Finan, D., Iyer, P., Singh, A., Jacob, T. and Jain, S. (2007) An 80-Tile 1.28 TFLOPS Network-on-Chip in 65 nm CMOS. *ISSCC* 2007 *IEEE International Solid-State Circuits Conference*, San Francisco, CA, USA, 1-15 February 2007, 98-99.

[5] Azimi, M., Cherukuri, N., Jayasimha, D.N., Kumar, A., Kundu, P., Park, S., Schoinas, I. and Vaidya, A.S. (2007) Integration Challenges and Tradeoffs for Tera-Scale Architectures. *Intel Technology Journal*, **11**, No. 3. http://dx.doi.org/10.1535/itj.1103.01

[6] Lai, C.Y., Lin, J.H. and Wang, Y.F. (2005) DVFS SoC Architecture and Implementation. *SoC Technology Journal*, **3**, 84-91.

[7] Lee, S.E. and Bagherzadeh, N. (2009) A Variable Frequency Link for a Power-Aware Network-on-Chip (NoC). *Integration*, *the VLSI Journal*, **42**, 479-485. http://dx.doi.org/10.1016/j.vlsi.2009.01.002

[8] Shang, L., Peh, L.S. and Jha, N.K. (2003) Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks. T*he Ninth International Symposium on High-Performance Computer Architecture*, Anaheim, CA, USA, 8-12 February 2003, 91-102.

[9] Huang, Y.S.C., Chou, K.K. and King, C.T. (2013) Application-Driven End-to-End Traffic Predictions for Low Power NoC Design. *IEEE Transactions on Very Large Scale Integration* (*VLSI*) *Systems*, **21**, 229-238.

[10] Mahadevan, S., Angiolini, F., Sparsø, J., Storgaard, M., Madsen, J. and Olsen, R.G. (2008) A Network Traffic Generator Model for Fast Network-on-Chip Simulation. In: Lauwereins, R. and Madsen, J., Eds., *Design*, *Automation*, *and*

Test in Europe, Springer, Netherlands, 173-184. http://dx.doi.org/10.1007/978-1-4020-6488-3_13

[11] Lahiri, K., Raghunathan, A. and Dey, S. (2001) Evaluation of the Traffic-Performance Characteristics of System-on-Chip Communication Architectures. *Fourteenth International Conference on VLSI Design*, Bangalore, 3-7 January 2001, 29-35. http://dx.doi.org/10.1109/icvd.2001.902636

[12] Sakthivel, E., Malathi, V. and Arunraja, M. (2014) MATHA: Multiple Sense Amplifiers with Transceiver for High performance Improvement in NoC Architecture. *Microprocessors and Microsystems*, **38**, 692-706. http://dx.doi.org/10.1016/j.micpro.2014.06.001

[13] Park, K. and Willinger, W. Eds. (2000) Self-Similar Network Traffic and Performance Evaluation. Wiley, New York, 94-95. http://dx.doi.org/10.1002/047120644X

[14] Huang, Y.S.C., Chou, K.C.K., King, C.T. and Tseng, S.Y. (2010) NTPT: On the End-to-End Traffic Prediction in the On-Chip Networks. 47*th ACM*/*IEEE Design Automation Conference*, Anaheim, CA, 13-18 June 2010, 449-452.

[15] Wiklund, D. and Liu, D. (2003) SoCBUS: Switched Network on Chip for Hard Real Time Embedded Systems. *International Parallel and Distributed Processing Symposium*, Nice, France, 22-26 April 2003, 1530-2075. http://dx.doi.org/10.1109/ipdps.2003.1213180

[16] Mishra, A.K., Das, R., Eachempati, S., Iyer, R., Vijaykrishnan, N. and Das, C.R. (2009) A Case for Dynamic Frequency Tuning in On-Chip Networks. *Proceedings of the* 42*nd Annual IEEE*/*ACM International Symposium on Microarchitecture*, New York, 12-16 December 2009, 292-303. http://dx.doi.org/10.1145/1669112.1669151

[17] Chow, T.S. (1978) Testing Software Design Modeled by Finite-State Machines. *IEEE Transactions on Software Engineering*, **4**, 178-187. http://dx.doi.org/10.1109/TSE.1978.231496

[18] Ducange, P., Marcelloni, F. and Antonelli, M. (2014) A Novel Approach Based on Finite-State Machines with Fuzzy Transitions for Nonintrusive Home Appliance Monitoring. *IEEE Transactions on Industrial Informatics*, **10**, 1185-1197.

[19] Ziv, J. and Lempel, A. (1978) Compression of Individual Sequences via Variable-Rate Coding. *IEEE Transactions on Information Theory*, **24**, 530-536.

[20] Lee, D. and Yannakakis, M. (1996) Principles and Methods of Testing Finite State Machines—A Survey. *Proceedings of the IEEE*, **84**, 1090-1123. http://dx.doi.org/10.1109/5.533956

[21] Dally, W.J. and Towles, B. (2001) Route Packets, Not Wires: On-Chip Interconnection Networks. *IEEE Proceedings of the Design Automation Conference*, Yokohama, Japan, 30 January-2 February 2001, 684-689. http://dx.doi.org/10.1109/dac.2001.935594

[22] Cassel, S., Howar, F., Jonsson, B. and Steffen, B. (2014) Learning Extended Finite State Machines. In: Giannakopoulou, D. and Salaün, G., Eds., *Software Engineering and Formal Methods*, Springer International Publishing, Gewerbestrasse, 250-264. http://dx.doi.org/10.1007/978-3-319-10431-7_18

[23] Schiefer, P., McWilliam, R. and Purvis, A. (2013) Creating a Self-Configuring Finite State Machine out of Memory Look-Up Tables. *Procedia CIRP*, **11**, 363-366. http://dx.doi.org/10.1016/j.procir.2013.07.030

[24] Gupta, S., Pareek, V., Jain, S. and Jain, D. (2014) Realization of Sequential Reversible Circuit from Finite State Machine. *International Computer Science and Engineering Conference* (*ICSEC*), Khon Kaen, 30 July-1 August 2014, 458-463.

[25] Wang, J., Qian, Y., Lu, J., Li, B., Zhu, M. and Dou, W. (2014) Designing Voltage-Frequency Island Aware Power-Efficient NoC through Slack Optimization. 2014 *International Conference on Information Science and Applications* (*ICISA*), Seoul, 6-9 May 2014, 1-4.

[26] Talwar, B., Kulkarni, S. and Amrutur, B. (2009) Latency, Power and Performance Trade-Offs in Network-on-Chips by Link Microarchitecture Exploration. 22*nd International Conference on VLSI Design*, New Delhi, 5-9 January 2009, 163-168.

[27] Jafarzadeh, N., Palesi, M., Khademzadeh, A. and Afzali-Kusha, A. (2014) Data Encoding Techniques for Reducing Energy Consumption in Network-on-Chip. *IEEE Transactions on Very Large Scale Integration* (*VLSI*) *Systems*, **22**, 675-685.

[28] Volos, S., Seiculescu, C., Grot, B., Pour, N.K., Falsafi, B. and De Micheli, G. (2012) CCNoC: Specializing On-Chip Interconnects for Energy Efficiency in Cache-Coherent Servers. 6*th IEEE*/*ACM International Symposium on Networks on Chip* (*NoCS*), Copenhagen, 9-11 May 2012, 67-74.

[29] Nowka, K.J., Carpenter, G.D., MacDonald, E.W., Ngo, H.C., Brock, B.C., Ishii, K.I., Nguyen, T.Y. and Burns, J.L. (2002) A 32-Bit PowerPC System-on-a-Chip with Support for Dynamic Voltage Scaling and Dynamic Frequency Scaling. *IEEE Journal of Solid-State Circuits*, **37**, 1441-1447.

[30] Brock, B. and Rajamani, K. (2003) Dynamic Power Management for Embedded Systems. *Proceedings of the IEEE International SOC Conference*, COEX ASEM Hall, Seoul Korea, 17-20 September 2003, 1-25.

[31] Dhiman, G. and Rosing, T.S. (2007) Dynamic Voltage Frequency Scaling for Multi-Tasking Systems Using Online Learning. *Proceedings of the* 2007 *International Symposium on Low Power Electronics and Design*, Portland, 27-29 August 2007, 207-212. http://dx.doi.org/10.1145/1283780.1283825

[32] Borkar, S. (2007) Thousand Core Chips: A Technology Perspective. *Proceedings of the* 44*th Annual Design Automation Conference*, San Diego, 4-8 June 2007, 746-749. http://dx.doi.org/10.1145/1278480.1278667

[33] Bishop, C.M. (2006) Pattern Recognition. Machine Learning. Springer, New York.

[34] Hemani, A., Jantsch, A., Kumar, S., Postula, A., Oberg, J., Millberg, M. and Lindqvist, D. (2000) Network on a Chip: An Architecture for Billion Transistor Era. *Proceeding of the IEEE NorChip Conference*, Vol. 31.

[35] Samman, F.A., Hollstein, T. and Glesner, M. (2008) Multicast Parallel Pipeline Router Architecture for Network-on-Chip. *Proceedings of the Conference on Design, Automation and Test in Europe*, Munich, 10-14 March 2008, 1396-1401.

[36] Kumar, A.S., Kumar, M.P., Murali, S., Kamakoti, V., Benini, L. and De Micheli, G. (2012) A Buffer-Sizing Algorithm for Network-on-Chips with Multiple Voltage-Frequency Islands. *Journal of Electrical and Computer Engineering*, **2012**, Article ID: 537286.

[37] Sakthivel, E., Malathi, V. and Arunraja, M. (2015) A New Simulator Based on Multi Core Processor with Improved Sense Amplifier. *Journal of Circuits*, *Systems and Computers*, **24**, 1550141. http://dx.doi.org/10.1142/S0218126615501418

[38] Palesi, M. and Daneshtalab, M. (2014) Routing Algorithms in Networks-on-Chip. Springer, Berlin. http://dx.doi.org/10.1007/978-1-4614-8274-1

[39] Coenen, M., Murali, S., Ruadulescu, A., Goossens, K. and De Micheli, G. (2006) A Buffer-Sizing Algorithm for Networks on Chip Using TDMA and Credit-Based End-to-End Flow Control. *Proceedings of the* 4*th International Conference on Hardware*/*Software Codesign and System Synthesis*, Seoul, 22-25 October 2006, 130-135. http://dx.doi.org/10.1145/1176254.1176287

[40] Synopsys, Inc. (1986) Mountain View, CA. http://www.synopsys.com

[41] Woo, S.C., Ohara, M., Torrie, E., Singh, J.P. and Gupta, A. (1995) The SPLASH-2 Programs: Characterization and Methodological Considerations. *ACM SIGARCH Computer Architecture News*, **23**, 24-36.