All Sciences Proceedings

All Sciences Proceedings

http://as-proceeding.com/

2nd International Conference on Engineering, Natural and Social Sciences

April 4-6, 2023 : Konya, Turkey

© 2023 Published by All Sciences Proceedings

ICENSOS

https://www.icensos.com/

A nature inspired optimized application mapping technique for On-Chip Networks

Yusra Zaheer Lodhi*, Muhammad Iram Baig [2]

[1]Department of Electrical Engineering, University of Engineering and Technology Taxila, Pakistan
[2]Department of Electrical Engineering, University of Engineering and Technology Taxila, Pakistan

*(yusra.zaheer.lodhi@gmail.com, iram.baig@uettaxila.edu.pk)

Abstract **–** Network-on-Chip (NoC) is a promising communication infrastructure that packs multiple cores into one chip for efficient data exchange. In such NoC architecture, application mapping is the process of assigning tasks to processing cores. Optimized application mapping technique improves chip performance and reduces overall chip power consumption. Optimizing application mapping is essential to NoC design. Mapping the task graph of the application onto a Network-on-Chip (NoC) Intellectual Property (IP) core is an NP-hard problem. The performance of the NoC network depends primarily on an effective and efficient mapping technique and optimization of performance and cost metrics. These metrics primarily include communication cost, energy, throughput, latency, power dissipation, and simulation time. A state-of-the-art nature-inspired mapping technique for NoC's called Sparrow Search Algorithm (SSA) has been introduced in this work which has never been applied with NoC. The proposed algorithm minimizes NoC power consumption by a cognitive base utilizing shared K-nearest neighbor clustering method based on six standard available benchmarks including VOPD, MPEG-4, MWD, MP3enc MP3dec, 263enc MP3dec and 263dec MP3dec. The analysis of the experimental results demonstrate that the proposed technique outperforms as compared to other existing nature-inspired meta-heuristic application mapping approaches in terms of various performance metrics, such as energy consumption, communication cost, and average packet latency.

Keywords – Application Mapping, Communication Cost, Intellectual Property, Network-On-Chip, Sparrow Search Algorithm,

I. Introduction

As the number of Intellectual Property (IP) cores embedded in a System-on-Chip (SoC) increases, the SoC's overall performance and scalability decrease. A new promising solution called Network-on-Chip (NoC) has been proposed to improve the overall performance and flexibility of SoC’s. NoC came as a solution to these issues by providing a modular and scalable solution to the needs of current applications with the bandwidth efficiency and scalability provided by its regular structures. A simple 4×4 NoC architecture is shown in figure 1. Router-based packet communication is used in NoC for IP cores interaction on SoC [1]. Every IP core has its router and performs tasks allocated to them. Routers work as switches and are connected in a certain topology, for IP cores communication among them. NoC contributed about 40% to the overall power consumption of the system [2]. Different topologies are present for NoC interconnection networks including torus, mesh, butterfly, etc. Mesh topology is vastly used in NoC architecture by offering a short communication path between IP cores and equivalent link sizes with the same regular structure [3].

Applications running on NoC are represented by the application task graph. An application task graph is a set of tasks with communication amongst them. The desired core is selected and tasks are allocated to them resulting in a core graph. The core graph is then mapped to the topological graph utilizing a mapping technique. To generate synthesized NoC, a mapped topology graph is passed through routing and then the scheduling phase [4]. The application mapping technique gives the improved arrangements of cores on NoC tiles by optimizing NoC performance parameters including power, energy, latency, throughput, communication cost, and simulation time [5]. Various mapping approaches have been suggested for the search-based and exact method. Application mapping is divided into three main branches dynamic mapping, static mapping, and hybrid application mapping. In dynamic mapping, the communication bandwidth of cores and the load on them are analyzed during the runtime and then tasks are mapped on the cores. The assignment of tasks can be changed during the runtime. Time consumption during the task allocation is quite critical as it influences the application's overall execution time. On the other side in static mapping tasks are allocated to the cores during their design time and cannot be altered during runtime and application task resources are also defined offline and cannot be altered. Design-stage and run-time issues are covered in hybrid mapping. The hybrid approach combines the characteristics of the design stage in real-time control and minimal computational time to get an effective mapping solution.

Search-based application mapping techniques result in the best solution through NoC performance metrics. Thus the efficiency of hard problem solutions is dependent on certain meta-heuristic algorithms. Meta-heuristic calculation is an all-inclusive arrangement technique that works with cooperation between nearby improvement systems and high-level methodologies [6]. It should be acceptable in exploitation and exploration. Exploration is defined as a global search known as diversification and exploitation is a local search known as intensification. An initial solution is not required for global search and its objective is to find global optimization of the cost function.

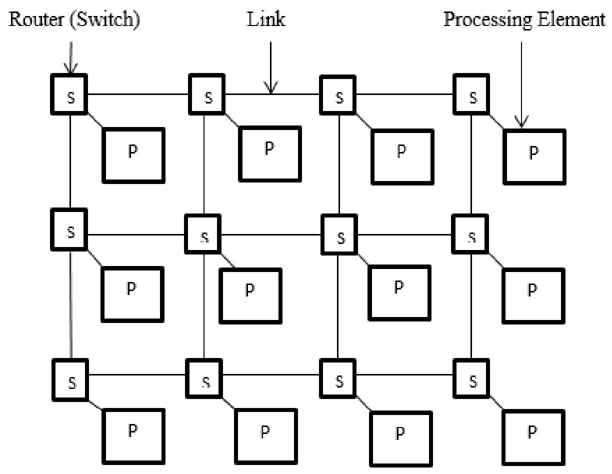

Fig. 1 4×4 NoC platform

To explore NoC performance metrics, this paper used a new meta-heuristic optimization algorithm, SSA. SSA provides a good balance between exploitation and exploration to avoid early convergence. SSA is a static search-based mapping technique, modeled after a group of sparrows who works together for finding food from different sources and preventing themselves from their predators. Searching for food and a new location is collective behavior they learned from the environment. The efficiency of SSA is demonstrated by checking the mapping of the NoC benchmark in a mesh-based two-dimensional (2D) topology. If food is available at a certain location whole group flew to that and if there is danger from a predator they move to a new location.

The rest of the paper is organized as follows. The literature review is given in Section II. Section III shows the inspiration for the Sparrow Search Algorithm. In section IV the mapping using SSA algorithm is given. In section V model used for metric analysis and section VI summarizes the proposed mapping algorithm. The experimental setup for this research while results are given in section VII and section VIII conclude the paper.

II. LITERATURE REVIEW

Numerous mapping algorithms are already proposed to map the application to different topologies. In [7], G. Chen presented a complier-based mapping technique, which performs packet routing, scheduling, data mapping, and process mapping. This method needs exceptionally high compilation time which might corrupt the performance of the system. In [8], Best Neighbor

(BN) mapping technique is proposed that helps in reducing the execution time of the system. In [2], static mapping based clustered Integer Linear Programming (ILP) technique for mapping is presented, in which the whole mesh network is divided into small mesh clusters and then they are mapped to smaller mesh. For the final result, all meshes are combined to form one mesh. ILP results in improved execution time but communication cost is increased. In [9], a dynamic mapping technique named Lower Energy Consumption based on Dependencies -Neighbourhood (LEC-DN) is proposed. Hop distance, communication volume, and proximity in hops numbers are used as cost functions. The Nearest Neighbor (NN) technique is used in a spiral fashion. A low-complexity mapping algorithm known as CastNet is proposed in [10], highest priority task is selected first. Symmetric groups are created to select the initial core for mapping the task. After the first task is mapped, the remaining are mapped by calculating the communication time with the already mapped task and the remaining cores are selected by calculating the communication cost.

In [3], M. Fattah presented an approach based on online and offline applications within the same network. If the online applications are critical then they are placed close to each other, otherwise, they are spread on any available node. This technique helps in minimizing the average latency. In [4], authors proposed a proactive region selection strategy (MapPro) based on first node selection. Its activates the free time of two successive mapping requests of applications to find the candidate mapping region and give priority to the node which gives low dispersion and congestion adjacent mapping node is selected by the ripple effect. MapPro gives a reduction in execution time, average latency, and low congestion. In [5], Weighted-Neighbourhood-Allocation (WeNA) mapping algorithm is presented that tries to enhance non-contiguous mapping. The task mapping of the same applications to the available cores is decided on their communication volume. Improvements in average network latency in comparison with other algorithms are observed. A Dynamic Task mapping with Congestion Speculation (DTMCS) is presented in [6], and tasks are compared with the number of available nodes. Source task is placed on priority and is based on the Manhattan distance. It reduces the overall congestion of the system and communication latency. In [11], author's proposed Hierarchical and Dependency Aware task mapping technique (HAD). The bottom left core of architecture is selected as a Global Manager (GM) which runs the algorithm. In the first step, the number of available cores for mapping is compared with the task transition and then cores placement is decided through the Mean occupied Position (MP). This algorithm reduces latency and energy consumption.

In [12], V. Tsoutsouras et al. presented distributed task mapping (DRTRM), based on controllers and local managers. It uses the manager core, initial and controller cores. The initial core picks cores for mapping, resources are managed through the manager core, and the activity of the platform is controlled through the controller core. It results in reducing communication overhead. In [13], the Segmented Brute-force Mapping (SBMAP) algorithm is proposed. Cores with maximum communications are separated into many segments and arranged to obtain minimum cost, and then the remaining parameters are calculated otherwise system moves to the next segments. This algorithm iteratively hunts for the optimum mapping for different sections of the task graph. It reduces the communication energy and computational complexity of the NoC design. In [14], authors presented Particle Swarm Optimization (PSO) which mimics the behaviours of birds flocking and fish schooling. It has multiple solutions known as particles that cooperate with their neighbour particles and result in problem space. Each solution in the search space is like a particle that has a fitness value. The bigger task is divided into small sizes which are assigned to PE according to their nature. IP mapping is done to achieve the least power consumption. In [15], L shaped isolated mapping approach (Liso) is presented, and applications with the same sources are divided into isolated regions to have non-interference communication. Each one of the tasks is mapped to only one core in NoC. It results in higher performance and reduced communication interference. In [16], authors proposed a self-adaptive chicken swarm optimization algorithm. It provides the advantage over swarm population control parameters by not setting them before the start of the algorithm. They are changed during the simulation. Initial mapping is done by randomly selecting the core. Global Best Swarm is then estimated through the fitness function. Rooster, Hen, and Chicken are ranked on the basis of the fitness function. Not best R, H, and C move into new velocity, and new solution is generated. This algorithm gives best communication cost. In [17], an improved ant colony mapping technique is proposed, at first tasks are divided based on their task size. The substructure of NoC maps the task when the scale of task is small otherwise they are arranged in different clusters and tasks are divided according to their dependencies if they are using the same resource node, then algorithm is applied to the resource node for task mapping. It reduces the overall latency of the NoC network, communication power.

The work given in [18] uses a Genetic Algorithm (GA) and a multipurpose-based mapping approach to various topologies of 2D NoC. In addition, GA works well in heuristics and performs efficiently at exploration. In [19], a Fibonacci tree Optimization (FTOS) Strategy is proposed for the scheduling problem of the wireless sensor network. It outperforms in terms of energy consumption as compared to differential evolution (DE), PSO, and Artificial Bee Colony (ABC). In [20], the authors presented a reliability-based technique. The graph presented is divided into its sub-graphs, to lessen the transmission flow. As a product, traffic within every graph increases while communication flow is reduced between the sub-graphs, this technique efficiently routes redundant communication packets through the non-uniform traffic distribution across network channels. In [21], a heuristic-based mapping approach is proposed which offers better-optimized results over network performance metric.

III. Sparrow Search algorithm (SSA)

Sparrows are swarming birds and are of different species, common in various areas of the world, and love to live in places where people live. Furthermore, they are omnivorous birds and mostly eat grains, insects, seeds, and weeds. Sparrows are also known to be perennial birds. Unlike other small birds, sparrows are very clever and have a strong memory. Their speed ranges from 38km/h to 50 km/hr whereas they are 14-16 cm in length and are very aggressive. Sparrows are of two type's producers and scroungers; which 75% are producers and 25% are scroungers. SSA is built on the hunting behaviour of sparrows; Producers search for food sources and scroungers get their food from them. There is evidence that sparrows usually adopt flexible behavioural strategies and switch between scroungers and producers [22]. Individual sparrow monitors the behaviour of other sparrows in the group. On the other hand, in a flock of birds, attackers compete to increase the predation rate compete with their companions for higher food intake [23]. In addition, the individual energy reserve can play an important role as sparrows choose different feeding strategies and low-energy sparrows scrounge more. Sparrows at the end of the group can easily be attacked by predators. Animals in the centre can approach their neighbours to minimize their range of influence. All sparrows show an instinct to be always vigilant and curious about everything. For example, when a sparrow finds a predator, one or more sparrows will sing and the whole group will fly away to the safe zone. For mathematical modelling sparrow search algorithm rules are summarized as:

1. Producers usually store high energy and provide directions to foraging areas for all scroungers. It is the responsibility of producers to identify areas where abundant sources of food are available. The amount of energy reserve depends on individual fitness value.

2. When the sparrow finds a predator, the individual sparrow begins to chirp as a warning signal. If the alert level is above the safety threshold value, the producer must move all populations to the safe zone.

3. Every sparrow in the flock can be a producer as long as they search for a better food source, but the number of scroungers and producers in the total population remains the same.

4. Higher energy level sparrows act as producers. To gain more energy several starving scroungers move to new places to find food.

5. Scroungers follow producers who can find the best food source. Meanwhile, some scroungers may continually monitor producers and increase their predation rates by competing for food.

6. Sparrows at the end of the group move quickly towards a safe place and get a better place for themselves in the case of any danger. On the other

hand, the sparrow in the centre of the group walks randomly and approaches other sparrows.

IV. MAPPING USING SSA

In the NoC design, the Network Task Graph (NTG) represents the application and is scheduled by the IP core scheduler available via the Network Core Graph (NCG). The NoC Architecture Graph (NAG) then transforms the NCG with an efficient mapping algorithm and maps to the NoC topology.

A. Problem formulation

Network Task Graph (NTG): The NTG is a directed graph where N = N (T, C), and each vertex of the graph represents a task ($t_i \in T$; i = 1, 2, 3, ...) for application computing resources, this task is associated with energy consumption, run time, and resource deadlines. Task i is known as the predecessor and task j is known as the successor of task i. The directed arc ($c_{(i,j)} \in C$; where, i = 1, 2, 3 ...; and j = 1, 2, 3,...) is the amount of data or the communication information between the communication task (t_i, t_j).

Network core graph (NCG): The NCG is the directed graph where G = G (P, A) and the nodes P = P =$P_0, P_1, P_2, \dots, P_n$ shows the processing element. A = $A_{(i,j)}$ indicates the bandwidth and characteristic parameters between P_i and P_j and.

B. NoC mapping

The producers and scroungers initial population is produced randomly using the initial mapping technique. Parameters of the proposed algorithm include the total number of sparrows (n), the maximum number of iterations (Max T), the total number of producers(P), alarm value (R_2) and safety threshold (ST) are initialized at Time t=0 and the weight of the task graph is generated at a given time. Communication Cost (CC) which is the fitness value of the current best sparrow is calculated. A sparrow with a minimum fitness value is known as the current best sparrow and the maximum fitness value gives the current worst sparrow. CC is calculated in section V using equation 6. Later on, sparrow location (in terms of safety and food) for producers and scroungers in search space are updated using equations given in section VI. If a new location is better than before (X_{best}) and fitness value > f_g (global best) the optimized mapping is obtained.

V. MODELS USED FOR METRIC ANALYSIS

The mathematical models given below are for calculating energy, latency, throughput, and power for SSA on NoC architecture.

A. Bit Energy Model

This model is used for the NoC platform [30]. The Bit Energy (E_{bit}) is used to find the energy consumption of every bit.

$$E_{bit} = E_k + E_s \qquad (1)$$

Where, E_{bit} represent energy used by one bit of data in sending it from the sender node to receiver nodes including the switch's energy. E_k gives energy to the link and E_s is Switch energy. The average network energy consumed in transmitting data from sender tile t_i to receiver tile t_j is calculated by the formula:

$$E_{(ti,tj)} = E_k * (N_t - 1) + E_s * N_h \qquad (2)$$

Where, N_t is Manhattan distance from sender to receiver node and is given by:

$$N_t = \left|x_i - x_j\right| + \left|y_i - y_j\right| \qquad (3)$$

(x,y) are used here because we are working on two-dimensional network. Total energy consumed by the entire network is given by:

$$E_{total} = \sum_{i,j}^{N_h} \left(E_{t_i,t_j} * B_{t_i,t_j}\right) \qquad (4)$$

Where, B_{t_i,t_j} is the bandwidth of the tiles.

$$E_{total=} \sum_{i,j}^{N_h} (E_k * (N_t - 1) + E_s * N_t) * B_{t_i,t_j} \qquad (5)$$

The Communication Cost (CC) of the NoC network is calculated by:

$$CC = \sum_{i,j}^{N_h} \left(N_t * B_{t_i,t_j}\right) \qquad (6)$$

Our goal for this research is to find a suitable mapping approach that minimizes energy and cost of the network. In this study, the energy and cost of applications on NoC will be used as performance indicators for various applications mapped to NoC architecture.

B. CMOS Cell Library Model

This model uses cell data from the CMOS library to calculate timing and power estimates [24]. The proposed algorithm uses this model to calculate the power consumption, throughput, average latency and performance of NoC.

$$L_{avg} = \frac{1}{N}[(\sum_{i=1}^{N}(1/Ni)][\sum_{j=1}^{N}(L_{i,j})] \quad (7)$$

Where, L_{avg} is the average latency, $L_{i,j}$ is the latency of the packet, Ni is packets received by the processor and a total number of processors in the NoC network is indicated by N. Average throughput is given as (T_{avg}).

$$T_{avg} = \frac{1}{N(T_{sim} - T_{war})}\sum_{i=1}^{N} N_i \quad (8)$$

Where, T_{sim} is Simulation time and T_{war} is the Simulation warm-up time.The average power of a network is calculated as:

$$P_{avg} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} N_i \, [\,(1-\alpha_{i,j})P_{w_{inactive,j}} + \alpha_{I,j}\, P_{wactive,j}] \quad (9)$$

Where, $P_{w_{inactive}}$ is the in-active power of post-layout and $P_{wactive}$ is the active power of post layout. $\alpha_{i,j}$ is the active component of the network. The average energy consumed in the network by each packet is given by:

$$E_{avg} = \frac{T_{sim} - T_{wrm}}{NN_P}\sum_{i=1}^{N}\sum_{j=1}^{N}[(1 - \alpha_{i,j})P_{winact,j} + \alpha_{i,j}P_{wact,j}] \quad (10)$$

Where, N_P is the total number of data packets in the network.

VI. PROPOSED ALGORITHM

A. Base core selection for initial mapping

The following steps are followed for base core selection:

Step I. Select the cores randomly from NTG.

Step II. Identify the direct connection of all available cores for mapping with the base core by using matrix N.

$$N = \begin{cases} 1 & if\ (z_i, z_j) = t_{i,j} \epsilon\ T \\ 0 & otherwise \end{cases} \quad (11)$$

Step III. Calculate the average communication cost (C_i) and weight (w_i) of each core by the formula given below:

$$C_i = \sum_{t_{ij}\epsilon T}\frac{w_{ij}}{N(z_i)} \quad (12)$$

$$w_i = \sum_{t_{ij}\epsilon T} w_{i,j} \quad (13)$$

w_{ij} represents weight between the initial and final core.

Step IV. The hop count is calculated between the initial to the final core by the formula given below:

$$H_{count} = [H_{ij}] \quad (14)$$

Where, $H_{ij} = Min(N(z_i, z_i))$. $N(z_i, z_i)$ shows the number of hops between the initial to the final core.

Step V. From another cluster using K-nearest neighbor method an edge exists between z_i, z_i if they have each other on their nearest neighbor list.

B. SSA algorithm

Virtual sparrows are used to find food in search space. In SSA, producers with high fitness value are prioritized in the search process when obtaining food. Producers are responsible for migration control of the entire population and searching for food. Therefore, producers can forage in a wider range for searching food than scroungers. According to rules 1 and 2 given in section (3) the location of producers is updated in every iteration by the formula given below:

$$X_{i,j}^{t+1} = \begin{cases} X^t_{i,j}.\exp\left(\frac{-i}{t_{max}.\propto}\right) & \text{if } R_2 < S_{th} \\ X^t_{i,j} + R.L & \text{if } R_2 \geq S_{th} \end{cases} \quad (15)$$

For R2 < S_{th} means no predator is nearby and the producer can enter into a wider search area. For R2 ≥ S_{th} means that predators are sensed by some sparrows in response whole group of sparrows flew to a safe zone. The current iteration is represented by "t" at the ith iteration. t_{max} represent the maximum number of iterations. $X^t_{i,j}$ represents of ith sparrows jth dimension, j=1,2,3, … d. α represents a random number α ∈ (0, 1]. S_{th} ∈ [0.5, 1.0], is the safety threshold value, R_2 is the alarm value and $R_2 \in [0, 1]$. R is a matrix of 1×d in which each element is 1. Q obeys normal distribution. The scroungers enforce rules 4 and 5 given in section (III). Some scroungers keep an eye on the producers more often. As soon as the producer finds a good food location, scroungers leaves his current position and competes for food. If they win, they immediately get the producer's food, otherwise, they follows the rule 5. The updated position of scroungers is given as:

$$X_{i,j}^{t+1} = \begin{cases} Q.\exp\left(\frac{X_{worst}^{t} - X_{i,j}^{t}}{\propto.iter_{max}}\right) & if\ i > n/2 \\ X_p^{t+1} + \left|X_{i,j}^{t} - X_p^{t+1}\right| T^{+}.R & otherwise \end{cases} \tag{16}$$

Where; the optimum position occupied by the producer is represented by X_p. $T^{+} = T^{T}(\ TT^{T})^{-1})$. X_{worst} shows the current worst location. 'T' defines a matrix of order $1 \times d$, each element inside the matrix is randomly assigned 1 or −1. When i> $n/2$, the scrounger which has the worst fitness value is certainly starving. In the simulation, it is expected that these danger-conscious sparrows are only 10% to 20% of the overall sparrow population. These sparrows' initial position is generated randomly within the population itself. According to rule 6, a mathematical model is stated as:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^{t} + \gamma.\left|X_{i,j}^{t} + X_{best}^{t}\right| & if\ f_i > f_g \\ X_{i,j}^{t} + L.\left(\frac{\left|X_{i,j}^{t} - X_{worst}^{t}\right|}{(f_i - f_w) + \theta}\right) & if\ f_i = f_g \end{cases} \tag{17}$$

Here, X_{best} is the global optimal current location. L ∈ [-1, 1] is a random number. γ as a step size control parameter with a variance of 1 and mean of 0. f_g and f_w are currently the highest and lowest fitness values , f_i is fitness of Current sparrows. θ is the minimum constant to avoid division by zero errors. $f_i > f_g$ shows that the sparrow is at the end of the group. X_{best} represents a central location in the population and is safe. K specifies the path of movement of the sparrow and is also step size control parameter. $f_i = f_g$ indicates that the in the center of the population the sparrow is well aware of the danger and needs to approach other sparrows. Furthermore, a new location is updated, if the new position is better than the already updated position then the required result is achieved, the Current best position (X_{best}) and f_g is obtained as result. The flow graph of the SSA algorithm is given in figure 2.

VII. Results and Discussion

In this section, we present the results of a performance analysis of 2D NoC SSA for six standard NoC benchmarks including MPEG4, MWD, VOPD, MP3enc MP3dec, 263dec MP3dec and 263enc MP3dec, as given in Table 1. 4*4is the size of the network for all given benchmarks. For comparison the network size is the same. Application VOPD task graph consists of 16 sub-tasks. These 16 tasks are mapped on a 4*4 NoC mesh network. Whereas, in the case of other available benchmarks a certain number of routers are placed idle.

A. Experimental setup

To estimate the performance of the presented mapping algorithm SSA, six standard NoC benchmarks are used and several experiments were conducted. The presented algorithm was tested for 2D NoC with other nature-inspired algorithms such as Integer Linear Programming (ILP), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Simulated Annealing (SA), Bat Algorithm (BA), and Chicken Swarm Optimization (CSO).

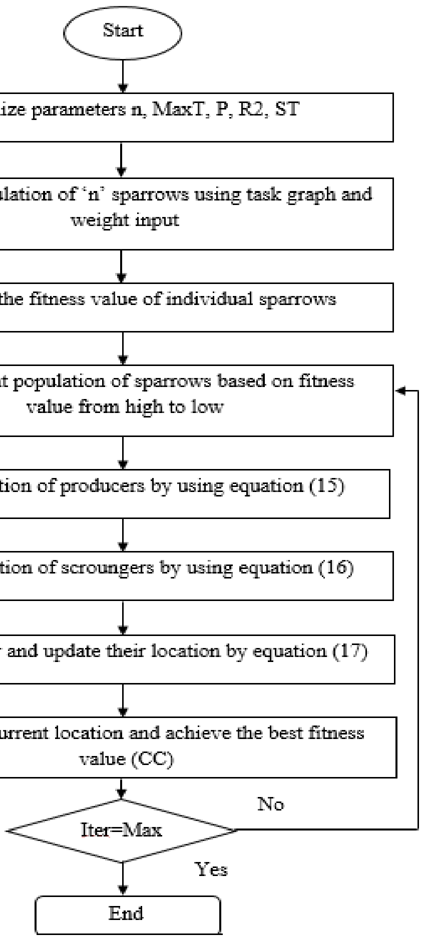
Start
Initialize parameters n, MaxT, P, R2, ST
Generate population of 'n' sparrows using task graph and weight input
Compute the fitness value of individual sparrows
Rank present population of sparrows based on fitness value from high to low
Update position of producers by using equation (15)
Update position of scroungers by using equation (16)
Select scouter and update their location by equation (17)
Update the current location and achieve the best fitness value (CC)
Iter=Max
No
Yes
End

Fig. 2 Flow Chart of Sparrow Search algorithm

Table1. Details of Standard NoC benchmarks with mesh sizes

Benchmark used	**Number of nodes**	**Number of edges**	**Size of mesh**
VOPD	16	21	4X4
MPEG	12	26	4X4
MWD	12	13	4X4
MP3encMP3 dec	13	14	4X4
263encMP3dec	12	12	4X4
263encMP3dec	14	15	4X4

The code for the proposed SSA algorithm is in Python and after getting the mapping for certain applications for simulation purposes we used it NockTweak Simulator. The simulations are performed on a corei7 16 GB RAM 4.9 GHz processor.

Table 2. Requirements of the Simulation environment.

Type of Network	Torus and 2-D Mesh
Type of platform	Embedded / Synthetic traffic
Embedded application	MPEG, VOPD, MWD etc.
Type of Packet delivery	Without ACK
Sending ACK policy	Optimally send ACK
Distribution of Packet	Exponential
Length of fixed packet	10 (flits)
Rate of Flit injection	0.1,0.2, 0.3,….1.0 (flits/cycle/node)
Mapping algorithm used	Random/ Proposed
Type of router	Wormhole pipeline
Routing algorithm	XY dimension
Selection of output channel	XY-Ordered
Size of buffer	8
Type of pipeline	8
Stages of Pipeline	4(RC, VA, SA, XB)
Input voltage	1(V)
Frequency of input clock	1000 MHZ
Frequency of operating clock	1000

B. Performance of SSA for 2D NoC against communication cost and simulation time

In this section of the paper, the overall performance of the presented algorithm SSA over communication cost is calculated and analyzed against already present mapping algorithms. Table 4 shows the valuation of communication cost. As ILP is observed as one of the best capable mapping algorithms in the exact mapping algorithm method for the computation of CC, our algorithm is compared with ILP and other algorithms. Table 3 shows the deviation percentage for ILP based methods for 2D NoC. The proposed mapping approach SSA yields the best results compared to other naturally inspired algorithms, as the results in Table 4 shows.

Table 3. Percentage of communication cost deviation from ILP

Mapping algorithm	**Percentage of deviation**	
	VOPD	**MPEG**
SA	2.7	-
ACO	-	1.9
GA	0.0	5.7
PSO	2	0.0
BA	0.0	0.0
CSO	0.0	0.0

The analysis of the mapping approach SSA simulation time is obtained and compared to other existing nature encouraged mapping algorithms and results are presented in Table 5 results clearly shows that SSA outperforms in case of simulation time as it consumes less time as compared to other algorithms while running for six standard benchmark/ embedded application. In comparison with other mapping algorithms, the proposed SSA takes 72% less simulation time.

C. Average power dissipation analysis

To calculate the efficiency of the presented nature inspired meta-heuristic Sparrow Search Algorithm, a power minimization analysis was also performed on given standard six benchmarks. It shows that SSA outperformed other existing mapping techniques and the average percentage of improvement in power minimization with other nature encouraged algorithms.

Table 6 shows the total power consumption in Watts (W) results of the 2D 4*4 mesh for six standard NoC benchmarks. The average power minimization improvement of the proposed algorithm is evident from the results in Table 6. Our proposed algorithm consumes less power as compared to already present algorithms including ILP, SA, ACO, GA, PSO, CSO, and BA.

It is obvious that the improvement of power minimization of our algorithm SSA is 3.6%, 24.2%, 19.40%, 27.59%, 18.98%, 12.81% and 4.83% over ILP, ACO, PSO, SA, GA, BA and CSO, respectively.

D. Average network latency analysis of the network

To check the performance of the network the analysis of latency on the network is also performed using SSA. It was done through different traffic patterns on the mesh topology. There are two types of traffic which are generated to check the average network latency of network. Traffics are tornado and random. Uniform patterns distribute the traffic in a uniform manner which helps to balance the load. In uniform random traffic all the packets are equally distributed among networks but in case of tornado traffic, traffic originates in the middle of the network.

The analysis of mesh-based NoC network is via the NoCTweak simulator using XY routing. Graphs of figures 3 and 4 shows the performance of SSA for network latency and shows that SSA has performed better as compared to other mapping algorithms. It is clear from graph of figure 3 that SSA outperformed PSO, GA, and CSO by 12.02%, 16.97% and 4.75%, respectively, for uniform random traffic pattern and graph of figure 4 shows that SSA outperformed PSO, GA, and CSO by 25.01%, 14.28%, and 6.32%, respectively, for tornado traffic patterns.

Table 4. Communication Cost using the SSA algorithm.

Mapping algorithm	**VOPD**	**MPEG-4**	**MWD**	**MP3enc MP3dec**	**263enc MP3dec**	**263dec MP3dec**
2D ILP[25]	4119	3567	1120	17.021	230.407	19.823
SA[26]	4231	3567	1451	-	-	-
ACO[27]	-	3633	-	17.231	-	-
GA[28]	4119	3772	1321	17.133	230.698	19.911
PSO[29]	3567	3567	1120	17.021	230.407	19.823
BA[30]	4119	3567	1122	17.834	231.450	19.936
CSO[16]	4119	3567	1122	17.021	230.407	19.823
Proposed algorıthm	4119	3567	1120	17.020	230.4069	19.823

Table 5. Simulation time (s) of 2D NoC against standard NoC benchmarks

Mapping algorithm	**VOPD**	**MPEG-4**	**MWD**	**MP3enc MP3dec**	**263enc MP3dec**	**263dec MP3dec**
SA	3878.527	-	197.541	-	-	-
ACO	-	18.652	-	1196.856	-	-
GA	3.925	3.234	3.420	3.194	3.185	3.174
PSO	3.785	3.465	3.432	3.194	3.185	3.188
BA	2.231	2.925	2.894	2.653	2.345	2.350
CSO	2.231	2.010	1.996	1.785	1.527	1.511
Proposed algorithm	1.81	1.21	1.735	1.499	1.124	1.009

Table 6. Total power consumption in watts (W)

Mapping algorithm	Power dissipation (W)					
	VOPD	MPEG-4	MWD	MP3enc MP3dec	263enc MP3dec	263dec MP3dec
2D ILP	1.528	1.137	1.012	1.228	1.286	1.211
SA	1.971	1.478	1.256	1.590	1.697	1.877
ACO	1.920	1.423	1.218	1.498	1.599	1.738
GA	1.843	1.356	1.109	1.507	1.445	1.561
PSO	1.841	1.357	1.112	1.507	1.445	1.561
BA	1.634	1.247	1.110	1.486	1.323	1.313
CSO	1.518	1.219	1.023	1.228	1.286	1.198
Proposed algorithm	1.310	1.20	1.013	1.217	1.265	1.147

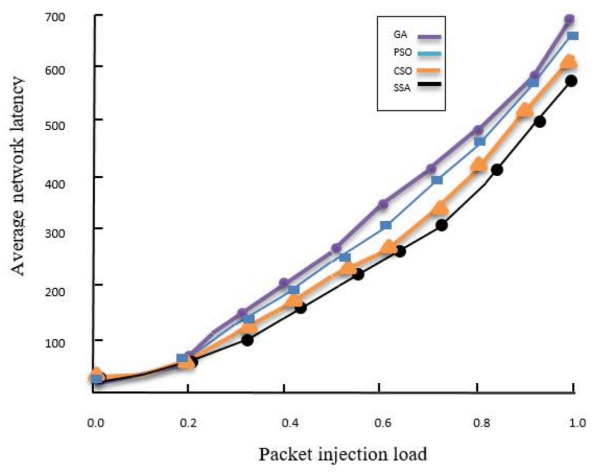

Fig 3. Average network latency for random traffic

VIII. Conclusion

This research work presents a nature inspired state-of-the art optimized mapping algorithm known as Sparrow Search Algorithm (SSA). This algorithm is used for optimized mapping for implementing task graph onto 2D NoC architecture using mesh topology.

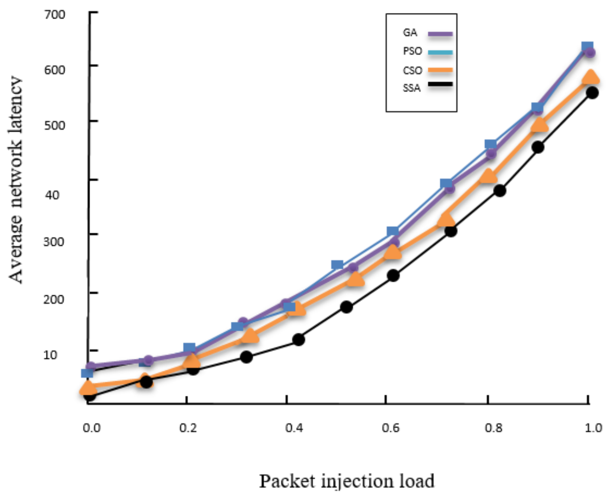

Fig 4. Average network latency for tornado traffic

The performance of proposed algorithm is obtained using six of the standard NoC benchmarks and is compared with already present state-of-the art optimization algorithms. Results of SSA given in above section are compared with ILP, BA, PSO, ACO, and CSO and our proposed algorithm outperformed in all parameters including power dissipation where our algorithm shows improvement of 3.6%, 24.2%, 19.40%, 27.59%, 18.98%, 12.81% and 4.83% over ILP, ACO, PSO, SA, GA, BA and CSO respectively and also proposed algorithm takes less simulation time as compared to other algorithms.

This research work provides a road to further research can be used for 3D NoC using other network topologies, other mapping algorithms are used with it to make a hybrid mapping technique for further improvement. Lastly SSA can be used to compute the other performance parameters including reliability and area using either 2D or 3D NoC architecture.

ACKNOWLEDGMENT

This research work was done for my master's thesis. I would like to express my sincere gratitude and appreciation to University of Engineering and Technology Taxila, Pakistan. I would also like to thank Prof Dr. Muhammad Iram Baig for his assistance in writing this article.

REFERENCES

[1] W.-C. Tsai, Y.-C. Lan, Y.-H. Hu, and S.-J. Chen, "Networks on chips: structure and design methodologies," J. *Electr. Comput. Eng.,* vol. 2012, 2012.

[2] S. Tosun, "Cluster-based application mapping method for Network-on-Chip," *Adv. Eng. Softw.,* vol. 42, no. 10, pp. 868–874, 2011.

[3] M. Fattah et al., "Mixed-criticality run-time task mapping for noc-based many-core systems," in 2014 *22nd Euromicro international conference on parallel, distributed, and network-based processing,* 2014, pp. 458–465.

[4] M.-H. Haghbayan, A. Kanduri, A.-M. Rahmani, P. Liljeberg, A. Jantsch, and H. Tenhunen, "Mappro: Proactive runtime mapping for dynamic workloads by quantifying ripple effect of applications on networks-on-chip*," in Proceedings of the 9th International Symposium on Networks-on-Chip, 2015,* pp. 1–8.

[5] L.-T. Huang, H. Dong, J.-S. Wang, M. Daneshtalab, and G.-J. Li, "Wena: Deterministic run-time task mapping for performance improvement in many-core embedded systems," *IEEE Embed. Syst. Lett.,* vol. 7, no. 4, pp. 93–96, 2015.

[6] H.-L. Chao, S.-Y. Tung, and P.-A. Hsiung, "Dynamic task mapping with congestion speculation for reconfigurable network-on-chip," *ACM Trans. Reconfigurable Technol. Syst. TRETS,* vol. 10, no. 1, pp. 1–25, 2016.

[7] M. F. Reza, D. Zhao, and M. Bayoumi, "Dark silicon-power-thermal aware runtime mapping and configuration in heterogeneous many-core NoC," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS),* 2017, pp. 1–4.

[8] E. L. de Souza Carvalho, N. L. V. Calazans, and F. G. Moraes, "Dynamic task mapping for MPSoCs," *IEEE Des. Test Comput,* vol. 27, no. 5, pp. 26–35, 2010.

[9] M. Mandelli, A. Amory, L. Ost, and F. G. Moraes, "Multi-task dynamic mapping onto NoC-based MPSoCs," in Proceedings of the *24th symposium on Integrated circuits and systems design,* 2011, pp. 191–196.

[10] S. Tosun, "New heuristic algorithms for energy aware application mapping and routing on mesh-based NoCs," *J. Syst. Archit.,* vol. 57, no. 1, pp. 69–78, 2011.

[11] C.-H. Huang, C.-Y. Chen, and H.-Y. Huang, "Hierarchical and dependency-aware task mapping for NoC-based systems," in 2018 *11th International Workshop on Network on Chip Architectures (NoCArc),* 2018, pp. 1–6.

[12] V. Tsoutsouras, I. Anagnostopoulos, D. Masouros, and D. Soudris, "A hierarchical distributed runtime resource management scheme for NoC-based many-cores," *ACM Trans. Embed. Comput. Syst. TECS,* vol. 17, no. 3, pp. 1–26, 2018.

[13] S. Khan, S. Anjum, U. A. Gulzari, T. Umer, and B.-S. Kim, "Bandwidth-constrained multi-objective segmented brute-force algorithm for efficient mapping of embedded applications on NoC architecture," *IEEE Access,* vol. 6, pp. 11242–11254, 2017.

[14] B. Chopard and M. Tomassini, "Particle swarm optimization," in An Introduction to Metaheuristics for Optimization, Springer, 2018, pp. 97–102.

[15] M. S. Sadeghi, S. B. Sarmadi, and S. Hessabi, "Toward on-chip network security using runtime isolation mapping," *ACM Trans. Archit. Code Optim. TACO,* vol. 16, no. 3, pp. 1–25, 2019.

[16] A. Alagarsamy, L. Gopalakrishnan, S. Mahilmaran, and S.-B. Ko, "A self-adaptive mapping approach for network on chip with low power consumption," *IEEE Access,* vol. 7, pp. 84066–84081, 2019.

[17] J. Fang, T. Yu, and Z. Wei, "Improved ant colony algorithm based on task scale in network on chip

(NoC) mapping," *Electronics*, vol. 9, no. 1, p. 6, 2019.

[18] A. A. Morgan, H. Elmiligi, M. W. El-Kharashi, and F. Gebali, "Unified multi-objective mapping and architecture customisation of networks-on-chip," *IET Comput. Digit. Tech.,* vol. 7, no. 6, pp. 282–293, 2013.

[19] L. Wu and H. Cai, "Energy-Efficient Adaptive Sensing Scheduling in Wireless Sensor Networks Using Fibonacci Tree Optimization Algorithm," *Sensors*, vol. 21, no. 15, p. 5002, 2021.

[20] A. Patooghy, H. Tabkhi, and S. G. Miremadi, "RMAP: A reliability-aware application mapping for network-on-chips," in 2010 *third International conference on dependability, 2010,* pp. 112–117.

[21] P. K. Sahu and S. Chattopadhyay, "A survey on application mapping strategies for network-on-chip design," *J. Syst. Archit.,* vol. 59, no. 1, pp. 60–76, 2013.

[22] Z. Barta, A. Liker, and F. Mónus, "The effects of predation risk on the use of social foraging tactics," *Anim. Behav.,* vol. 67, no. 2, pp. 301–308, 2004.

[23] R. Budgey and S. Hutton, "Three dimensional bird flock structure and its implications for birdstrike tolerence in aircraft," *Proc Int Bird Strike Comm.,* vol. 24, pp. 307–320, 1998.

[24] A. T. Tran and B. Baas, "NoCTweak: a highly parameterizable simulator for early exploration of performance and energy of networks on-chip," *VLSI Comput. Lab ECE Dep. Univ. Calif. Davis Tech Rep ECE-VCL*-2012-2, 2012.

[25] S. Tosun, O. Ozturk, and M. Ozen, "An ILP formulation for application mapping onto network-on-chips," *in 2009 International Conference on Application of Information and Communication Technologies, 2009,* pp. 1–5.

[26] C. Marcon, A. Borin, A. Susin, L. Carro, and F. Wagner, "Time and energy efficient mapping of embedded applications onto NoCs," *in Proceedings of the 2005 Asia and South Pacific Design Automation Conference,* 2005, pp. 33–38.

[27] F. Ferrandi, P. L. Lanzi, C. Pilato, D. Sciuto, and A. Tumeo, "Ant colony heuristic for mapping and scheduling tasks and communications on heterogeneous embedded systems," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.,* vol. 29, no. 6, pp. 911–924, 2010.

[28] G. Ascia, V. Catania, and M. Palesi, "Multi-objective mapping for mesh-based NoC architectures," *in Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, 2004*, pp. 182–187.

[29] P. K. Sahu, T. Shah, K. Manna, and S. Chattopadhyay, "Application mapping onto mesh-based network-on-chip using discrete particle swarm optimization," *IEEE Trans. Very Large Scale Integr. VLSI Syst.,* vol. 22, no. 2, pp. 300–312, 2013.

[30] J. Li et al., "Bat algorithm based low power mapping methods for 3D network-on-chips," *in National Conference of Theoretical Computer Science, 2017,* pp. 277–295.