A Deadlock Free Adaptive Routing Algorithm With Low-Power Scheduling And Dynamic Task Mapping for NoC

Venkata Sridhar .T[1], Dr. G. Chenchu Krishnaiah[2]

[1]Scholar, Dept. of E&C,VTU-Belgavi, India.
[2]Professor, Dept. of ECE, GKCE, Sullurupeta, India.

Abstract – *In order to reduce the routing latency in the Network-on-chips and to design a deadlock free packet communication in SoC with energy efficient scheduling of job and dynamic job mapping are implemented. Energy aware task scheduling algorithm allocates tasks on different elements under process and schedules the execution. Dynamic task mapping is applied to improve the performance of NoC and also to reduce both power consumption and delay in a non-preemptive network. The result of this proposed technique is more energy efficient and reduces latency. For the deadlock avoidance, an adaptive routing algorithm based on load on Virtual Channel (VC) was developed in two fashions. One is traditional single packet transfer adaptive routing algorithm and the second one is whole packet forward (WPF: Multiple packets). In this paper, a fully adaptive routing algorithm is presented (MWPF). Compared with TFA and WPF, our design achieves an average 65.43% and 35.54% latency improvement, respectively. 38.54% and 24.53% saturation throughput improvement in the standard synthetic traffic pattern.*

Key Words: **Dynamic job-map, scheduling, NoC, Virtual Channel, MWPF.**

1. INTRODUCTION

A Network-on-Chip (NoC) provides an energy efficient communication for Multi-Processor System-on-Chip (MPSoC). In integrated circuit, Network on chip (NoC) is used as communication subsystem. NoC technology plays major role in on chip communication. In System on Chip (SoC) it provides scalability and power efficiency, when compared with other designs [1]. The performance of network degraded by power consumption and latency. This is the important metric for NoC performance. These problems overcome by the dynamic task mapping by allocating tasks to processing elements on network. It will reduce the travelling distance of the task from source to destination, as well as it will reduce the power consumption on the network [2]. Energy efficient scheduling model is used in heterogeneous NoC to reduce the traffic delay as well as energy utilization of the network [3]. Further the routing strategy is major performance metric of the NoC in the SoC for communication.

1.1 Routing Algorithm

NoC have been proposed as a long-term on-chip communication for Multi-Processor System on Chip (MPSoC) microarchitecture [4]. NoC performance is related to the choice of a routing algorithm and the network topology. The most common NoC topology is a 2D-mesh or 3D-Mesh, consisting of a two-dimensional array of nodes. Processors, memory and other modules are located throughout the network and connected to local routers. In this topology, each router, except the boundary ones, is connected to four neighboring routers in the south, north, east and west directions, respectively. Within each router, buffers are provided for each input channel and a cross-bar switch is used to route the incoming packet to an appropriate output port [5]. Minimizing router overheads, NoCs tend to use simple router designs with limited VCs, shallow flit buffers per VC, short router pipeline stages, and messages with a limited number of flits [6]. The proposed routing algorithms for NoCs can be categorized in many different ways. For instance, depending on how a routing path is defined, routing can be classified as deterministic (static) and adaptive [7]. In a deterministic routing algorithm, the traversal path is determined by its source and destination alone and uses a fixed path for each pair of nodes, resulting in increased packet latency. Popular deterministic routing scheme for NoC include dimension-order (XY and YX) and source routing. In source routing, the source node makes static routing decisions for each router. In dimension-order routing (XY), the position of the mesh routers will be described by coordinates, the x-coordinate for the horizontal and the y coordinate for the vertical position; a packet is routed to the X-dimension first and then in Y-dimension to reach the destination.

1.2 Deadlock Avoidance

Deadlock is an abnormal network state in which a circular dependency relation is formed among the network resources; causing the routing of packets to be indefinitely postponed. NoC routing algorithms use one of the two solutions to handle the deadlock condition, namely deadlock avoidance or deadlock recovery. Deadlock avoidance theories impose additional constraints on the routing algorithms to ensure that deadlock can never occur. Deadlock recovery routing algorithms, detect and resolve the deadlock situation as it occurs [5].

In this paper, a deadlock avoidance fully adaptive routing algorithm (MWPF) based on WPF (multiple packets in a VC) is provided in which decrease overall packet latency to moderate full output buffer problem in WPF. The proposed routing algorithm first check output port, if it was full (because multiple packet can be held in a VC) increment a variable in VC allocator and uses another available output port that is produced by routing computation logic. Moreover, two threshold values are recorded: number of times a packet has met to the full output buffer and current packet latency. If these values for current packet exceeds the threshold value then uses a deterministic routing algorithm for these packets. Compared with TFA and WPF, our design achieves an average 65.43% and 35.54% latency improvement, respectively. 38.54% and 24.53% saturation throughput improvement in the standard synthetic traffic pattern. Compared with WPF an average 26.1% and 61.2% maximum latency reduction on SPLASH-2 benchmarks running on a 49-core CMP.

2. RELATED WORK

The goal of this research is to design an efficient task mapping algorithm with low chance of deadlock occurrence even on a high traffic density. Daniel Gebhardt et al [8] proposed the concept of NoC to optimize the power and communication delay on asynchronous network. It has four-phase routers and two-phase bundled-data links. ANetGen, is used to produce optimized result for the network by reducing energy and latency. Simulated annealing and force directed placement methods are used. System-C is used for traffic analysis.

Jia Zhan et al [9] have proposed to reduce the consumption of energy on NoC new approach was proposed, not to deviate the prescribed deadline of latency for real-time application. Maximum delay of every packets were calculated using standard approach which is established on network calculus, and evaluated the additional delay of the packet on network not to deviate the prescribed deadline is called worst case slack. Using this idea the slacks distributed to different traffic flows. And allocated different voltage as well as frequency to all routers.

James Harbin and Leandro Soares Indrusiak [2] have proposed a dynamic task mapping. The tasks are contention for network link to travel from source to destination. Then this contention will increase the power consumption and latency of the tasks on the network. These things are overcome by dynamic task mapping on non-preemptive network. The simulation results are compared to the preemptive network. Dynamic task mapping for non-preemptive network reduces power consumption and latency approximately 35% compared to preemptive network. The latency of high priority flow also reduced.

An important issue on deadlock avoidance fully adaptive routing algorithms is VC re-allocation scheme. A limitation of the traditional fully adaptive routing algorithms constructed based on Duato's theories [10-11] is: anytime, a VC can hold at most one packet, in other words, a VC can be re-allocated when it is empty (Conservative VC re-allocation scheme). Deterministic and partially adaptive routing algorithms [10, 11], use aggressive VC re-allocation: a VC can be reallocated as soon as a tail flit of the last packet arrives. On the other hand, in partially and deterministic routing algorithms, multiple packets can reside in a VC. Channel dependency graph in these algorithms is acyclic as a result deadlock can never occur.

3. PROPOSED WORK

The proposed concept of this paper is, energy aware task scheduling and dynamic job mapping for NoC Architecture with a low risk of deadlock occurrence even on critical situations. Job scheduling is performed using energy-aware scheduling algorithm. It allocates tasks on different elements under process and schedules the execution.

3.1 TSA With Energy Awareness

Do assume the following things for the ease of algorithm design.

Let J_i be the job

Let μ be the processing elements

Let E_{Ji}^{1} be the minimum energy consumption

Let E_{Ji}^{2} be the second energy consumption

Let $d(E) = E_{Ji}^{2} - E_{Ji}^{1}$ be the difference in energy

1) Initially, the following three metrics are estimated

E_{Ji} = energy consumption of J_i on different μ

T_{Ji} = execution time of J_i

$\sum T_{Ji}$ = mean execution time of J_i

2) Based on E_{Ji} and T_{Ji}, the following weight value is estimated

$$W_{Ji} = E_{Ji} \times T_{Ji} \qquad (1)$$

If W_{Ji} is high

Then

Higher priority is scheduled for respective J_i.

End if

Hence, using $\sum T_{Ji}$ of J_i, different queues for different paths are estimated and concerned queue is assigned for different jobs based on the computed weights.

3) With the estimated weight values, job deadline (JD_i) is estimated.

4) A ready job catalog (RJC) is generated for the jobs which have its precedent jobs are already scheduled.

5) For each $J_i \in$ RJC and μ , the initial completion time of J_i (FT_{Ji}) is estimated using the following equation.

$$FT_{Ji} = T_{DJi} + T_{Ji} \quad (2)$$

where T_{DJi} = ready time of data

T_{DJi} – The entire receipt of recent incoming time for communication transactions (CT) catalog to the respective job.

6) The transactions in CT are initially sorted using senders job completion time.

7) The path used for each transaction is estimated and the schedule table is set up by combining all occupied slots of its comprising paths/links.

8) Finally, the transaction is scheduled to the earliest time FT_{Ji} with respect to the current scheduling table of the path.

9) For every J_i in RJC, minimum FT_{Ji} is estimated using the following Eq

$$min_{FT} = \min FT_{Ji} \quad (3)$$

```
For any $J_i$,
If $min_{FT} \geq JD_i$
Then
        Job with maximum {$min_{FT}$ - $JD_i$} is
        selected and assigned to $\mu$
End if
          If all $J_i$ satisfies $min_{FT} < JD_i$.
          Then
                  Job list with $\mu$ is generated
                  Verify if $\mu$ in job list satisfies $min_{FT} \leq JD_i$
                  Estimate $d(E)$ for each $J_i$
                  Job with maximum $d(E)$ is
                  selected and it is assigned to $\mu$
          End if
```

10) The scheduling table of related μ and paths/links is updated.

11) Repeat this given scheduling process until all jobs are allocated and scheduled.

3.2 A Dynamically Mapping Algorithm for the Job

In Non-preemptive network, during data transmission low priority task may block the high priority task. To overcome this issue dynamic job mapping algorithm is applied to split the flows between the tasks which are contention for the network link and allow parallel transmission. In a non-preemptive network it optimizes the performance of NoC and also to reduce both energy consumption and delay. In order to avoid contention for network link the jobs are move around the network.

Let {P'_s} denote the list of PEs sorted in the ascending order of utilization.

Let {P''_s} denote the list of PEs sorted in the descending order of utilization.

Let ps $\in P'_s$ be the set of PEs that has utilization less than the threshold U_T.
Let cv_ps_i denotes the communication volume of task t_i on ps with other tasks.
Let cv_pt_i denotes the communication volume of task $t1_i$ on pt with other tasks

Algorithm

```
For each ps $\in P'_s$
    For each task $t_i$ scheduled on ps
        Calculate $cv\_ps_i$
        For each pt $\in P''_s$
            If capacity(Pt) > avail_capacity of $t_i$, then
                Calculate $cv\_pt_i$
                If $cv\_pt_i > cv\_ps_i$, then
                    Migrate $t_i$ to pt
                End of if
            End of if
        End of for
    End of for
End of for
```

The algorithm first lists out the set of PEs with least utilization. It then retrieves the list of tasks t_i assigned for those PEs and the estimates the communication volume cv_ps_i of each task t_i on these least utilized PEs. It again lists out the set of PEs with highest utilization. It the capacity of those Pes are sufficient to execute the tasks t_i, then the communication volume cv_pt_i of ti are again estimated on these highest utilized PEs. If cv_ps_i is less than cv_pt_i, then the corresponding tasks are migrated to those high utilized PEs. The main objective of this job mapping is to migrate the tasks from lightly loaded PEs to heavily loaded PEs which possess sufficient capacity.

This process of job mapping performed through resource management reduces the NoC routers complexity and enhances the overall network performance.

3.2 A Full Output Buffer Problem

To reduce packet latency, a deadlock avoidance fully adaptive routing algorithm (MWPF) is presented which moderates the negative impact of WPF on the packet latency because the full output buffer problem. In the previous, traditional fully adaptive routing algorithms use conservative VC reallocation scheme thus these algorithms can yield worse performance and VC utilization than deterministic and partially adaptive ones. WPF faces this problem with supporting multiple packets in a VC. Based on an experiment in WPF, most of the packets in NoCs are single flits and the remaining packets are 5 flits long. According to this observation, as regards, most of the packets in the network are single flits and these packets usually are back-to-back messages, WPF faces with full output buffer problem which a VC is filled with multiple packets and the current packet latency increases in the input port.

3.2 An MWPF: Fully Adaptive Routing Algorithm

To reduce the packet latency because the full output buffer problem in WPF, the proposed routing algorithm (MWPF) first check the output VC, a case of it was full, increment a variable in the VC allocator (FoB: Fully output Buffer) by one and uses the another available output port which is produced by the routing computation logic. If this port was full too, FoB increments by one again. Fig.-1 depicts our proposed method.

The number of times which second output port is full is very smaller than the first output port. Moreover, the routing computation logic uses two threshold values for FoB and the packet latency. If FoB and latency for the current packet are greater than the threshold values, then these packets are routed based on a deterministic routing algorithm such as dimension-order routing. Note that in Fig.-1, the east port of R4 is full thus algorithm uses another available output port (north port) and FoB parameter in VC allocator in packet __ is incremented. In the next hops, if the FoB and latency values for the packet __ are greater than the threshold value, then this packet is routed with a dimension-order routing.

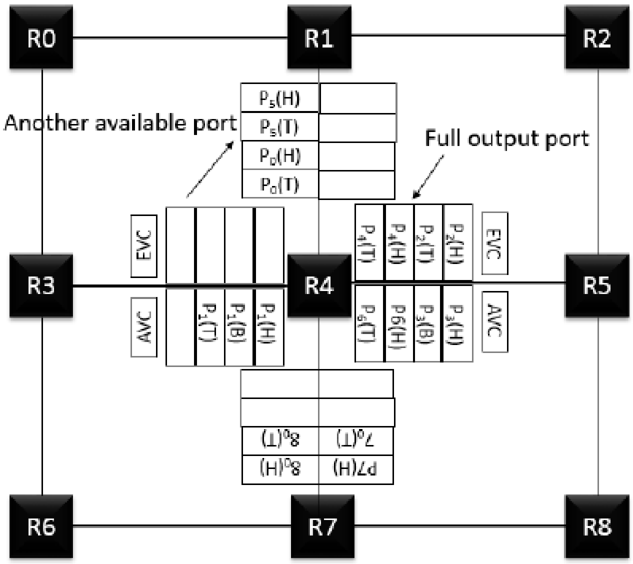


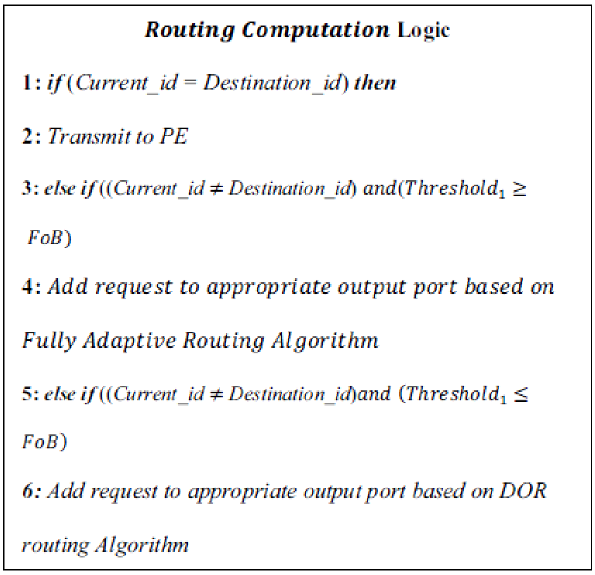
Fig -1: Routing Algorithm Proposed.

One of the important issues in MWPF is to find the optimum values for the thresholds. To obtain these threshold values, we evaluate performance (latency) of MWPF with different packet injection rate and uses different values for these thresholds. For the sake of simplicity, we find these thresholds for uniform traffic pattern and uses these values for other synthetic and real traffic patterns. Considering various experiments shows which optimum values for FoB and latency thresholds in uniform traffic are 20 and 400, respectively.

As regards the use of extra VC, increases the power consumption and the hardware overhead, the packets which are routed with dimension-order routing, have to use the existing VCs. These packets requests for all of the VCs in the output port (Escape and Adaptive VCs). In WPF, for proving that WPF is a deadlock avoidance routing algorithm, it uses this lemma: if the routing algorithm with conservative VC re-allocation is deadlock-avoidance, then applying WPF to forward packets into non-empty VCs will not lead to deadlock. Due to this fact, in our algorithm (MWPF), since exist any dependency between packets, as a result, MWPF is deadlock avoidance algorithm. On the other hand all the head flits in the network receive the VC heads.

3.3 RCL And VCA Unit

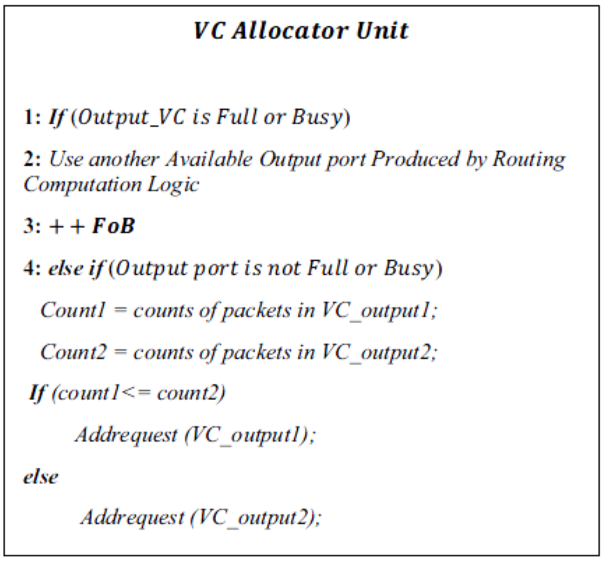
Following Fig.-2 and Fig.-3, discus about the routing computation logic of the proposed model and virtual channel allocation unit for the deadlock avoidance of the proposed model respectively. The pipelined NoC router consists of four stages: routing computation (RC), VC allocation (VA), switch allocation (SA) and switch traversal (ST). The speculative switch allocation is used to parallelize VA and SA. MWPF only requires simple modifications to the baseline VC allocator and routing computation logic. We use a separable VC allocator which is widely used due to its low complexity and high frequency [12]. In a separable VC allocator, each input VC determines which output VC of the selected output port to bid on in the first stage. The winning requests from the first stage then arbitrate for an output VC in the second stage.



```
Routing Computation Logic
if (Current_id = Destination_id) then
    Transmit to PE
else if ((Current_id ≠ Destination_id) and(Threshold₁ ≥ FoB)
    Add request to appropriate output port based on Fully Adaptive Routing Algorithm
else if ((Current_id ≠ Destination_id)and (Threshold₁ ≤ FoB)
    Add request to appropriate output port based on DOR routing Algorithm
```

Fig -2: RCL for the proposed MWPF-RA

We modify the first stage arbiters [12]. In routing computation stage, first latency and FoB values for input packet are compared with the threshold values. If these values are less than the threshold values, then the input packet will be routed according to fully adaptive routing algorithm, otherwise the input packet follows the DOR routing algorithm. The Pseudo-code for routing computation unit is shown in Fig.-2.



```
VC Allocator Unit

If (Output_VC is Full or Busy)
    Use another Available Output port Produced by Routing Computation Logic
    + + FoB
else if (Output port is not Full or Busy)
    Count1 = counts of packets in VC_output1;
    Count2 = counts of packets in VC_output2;
    If (count1<= count2)
        Addrequest (VC_output1);
    else
        Addrequest (VC_output2);
```

Fig -3: VCA for the proposed MWPF-RA

In VC allocation stage, the output VC is checked. If the output VC is full, FoB threshold value in the input packet incremented by one and uses another available output VC which is produced by the routing computation logic. If the output VC is not full, the number of packets inside two output VC that is produced in the routing phase is compared together and the input packet is sent to the output VC with the least number of packets. The Pseudo-code for the VC allocation unit is shown in Fig.-3.

4. SIMULATION RESULTS

The performance of the proposed Energy aware task scheduling and dynamic job mapping (EATSDJM) algorithm is evaluated with simulation on the network simulator (NS-2). Output is compared with Energy-Aware Communication and Task Scheduling (EACTS) technique [3]. 5x5 mesh topology used in this simulation model. Exponential traffic and CBR traffic based output described below chart-1 to chart-3 .

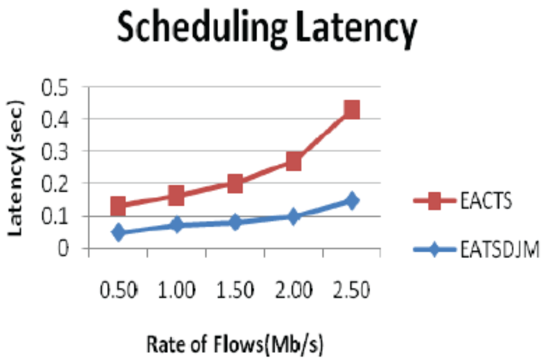


Chart -1: Latency Vs Rate of Flow.

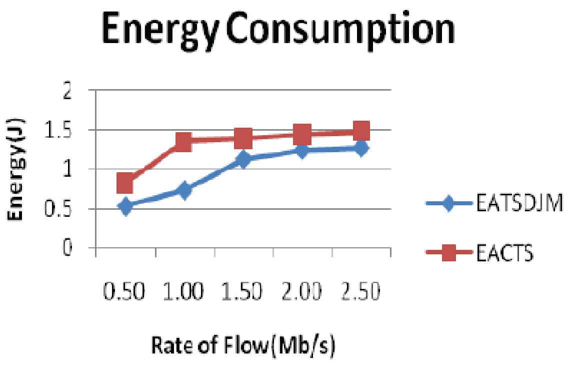


Chart -1: Energy Vs Rate of Flow.

Chart-1 shows the latency occurred in both EACTS and EATSDJM techniques. As seen from the figure, whenever the task size is increased, the latency increases linearly. However, since EATSDJM performs dynamic job mapping, in addition to the energy aware task scheduling, the latency is 34% less when compared to EACTS.

Chart-2 shows the average energy consumption at PEs measured by both EACTS and EATSDJM techniques. As seen from the figure, whenever the task size is increased, the energy consumption at PEs increases linearly. However, since EATSDJM performs dynamic job mapping, in addition to the energy aware task scheduling, the energy consumption is 25% less when compared to EACTS.

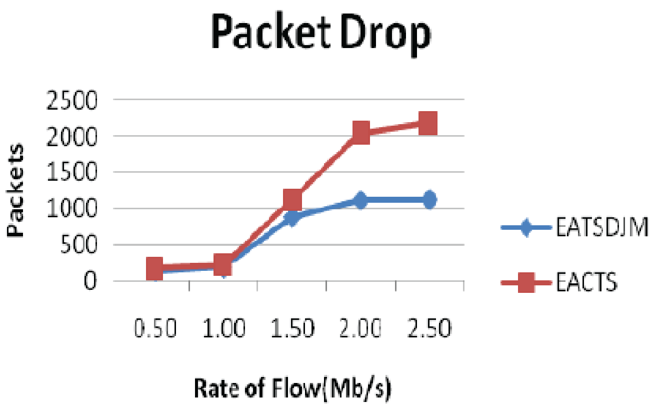


Chart -3: The Packet drop Vs Rate of Flow

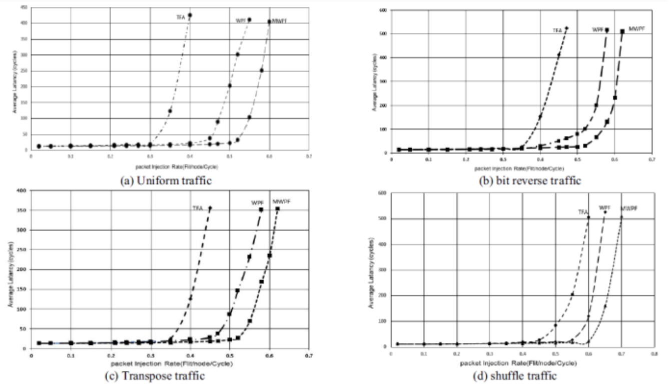
Chart-3 shows the packet drop occurred in both EATSDJM and EACTS. As seen from the figure, whenever the task size is increased, the packet drop increases linearly. EATSDJM outperforms EACTS in terms of packet drop by 31%, since it performs load balanced routing.

Further, for the deadlock avoidance in the different traffic conditions the performance of MWPF has been evaluated using a flit level, cycle accurate on-chip network simulator based on the Booksim. The modeled on-chip network configuration is a 2D mesh topology with a single-cycle channel. We evaluate our method with the synthetic traffic pattern and real traffic pattern. Our baseline configuration uses a 8 x 8 mesh with 2 VCs that are 4 flits deep. Moreover, the packet length is uniformly distributed between 1 flit and 5 flits. A baseline router has five input ports and five output ports.

Table-1: Network Configuration.

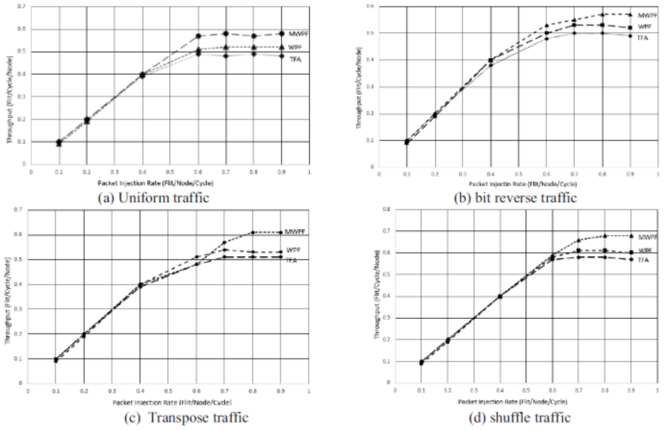
Characteristic	Baseline
Topology(mesh)	8 × 8
VCs	2 (1 Escape, 1 adaptive)
Flit buffers/VC	4
Packet length(flits)	Long:5, short: 1
Warm up cycles	10,000
Switching technology	wormhole
Router latency	1 cycles
Channel latency	1 cycles

Table -1 specifies the configuration parameters in our performance evaluation. We compare the performance of our proposed deadlock avoidance fully adaptive routing algorithm with WPF and TFA which uses conservative VC re-allocation. As a performance metric, we use latency, throughput and average hop count. Latency defined as the number of cycles between the initiation of a message issued by a source node and the time when the message is completely delivered to the destination node. While each source offers a particular amount of traffic to the network, the throughput, or accepted traffic, is the rate that traffic (bits/s) is delivered to the destination terminals. The distance between nodes, referred to as the hop count, is measured as the number of channels and nodes that a message must traverse on average to reach on destination. Fig.-4 and Fig.-5 shows the performance of the proposed model at various traffics.



(a) Uniform traffic
(b) bit reverse traffic
(c) Transpose traffic
(d) shuffle traffic

Fig -4: Performance of proposed model at different traffics.



(a) Uniform traffic
(b) bit reverse traffic
(c) Transpose traffic
(d) shuffle traffic

Fig -5: Throughput of proposed model at different traffics.

5. CONCLUSIONS

The proposed model "Energy aware task scheduling followed by dynamic job mapping" concept is developed for NoC architecture. The Energy aware task scheduling algorithm allocates tasks on different elements under process and schedules the execution. In non-preemptive network Dynamic task mapping is applied to improve the performance of NoC and also to reduce the energy (power) consumption and delay. The result from simulation shows that the proposed technique is more energy efficient and reduces latency.

Further, in this paper, a deadlock-avoidance fully adaptive routing algorithm named MWPF is developed for a mesh topology using two virtual channels along both directions. The proposed algorithm is based on WPF that allows multiple packets to reside in one VC concurrently; this algorithm improves VC utilization compare with traditional fully adaptive routing algorithm that use conservative VC reallocation. Compared with TFA and WPF, our design achieves an average 65.43% and 35.54% latency improvement, respectively. A 38.54% and 24.53% saturation throughput improvement in the standard synthetic traffic pattern.

MWPF is a unicast routing algorithm, the idea of multiple packet in a VC can be extended to the multicast routing algorithm.

REFERENCES

[1] Pascal T.W., Gerard J.M. Smit and Jens E. Becker, “Energy-Efficient NoC for Best-Effort Communication”, Field Programmable Logic and Applications, 2005, International Conference IEEE 2005.

[2] James H and Leandro S Indrusiak, “Dynamic Task Remapping For Power and Latency Performance Improvement In Priority-Based Non-Preemptive Networks On Chip”, Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), 2013 8th International Workshop IEEE 2013.

[3] J.Hu and R.Marculescu, “Energy-Aware Communication and Task Scheduling for Network-on-Chip Architectures under Real-Time Constraints”, Design, Automation and Test in Europe Conference and Exhibition 2004, Proceedings IEEE 2004.

[4] L. Benini, et al., “Networks on chips: a New SoC paradigm,” in proc. IEEE Computer, pp. 70-78, 2002.

[5] T. Wen-Chung, et al., “Non-Minimal, Turn-Model Based NoC Routing,” Microprocessors and Microsystems, 2012.

[6] G. Paul et al., “Regional congestion awareness for load balance in networks- on-chip,” in proc. HPCA, pp. 203-214, 2008.

[7] W. Ling, et al., “A Routing-Table-based Adaptive and Minimal Routing Scheme on Network-on-Chip Architectures,” in proc. HPCA, pp. 846-855, 2009.

[8] Daniel Gebhardt, Junbok You and Kenneth S. Stevens, “Design of anEnergy-Efficient Asynchronous NoC and Its Optimization Tools for Heterogeneous SoCs”, IEEE Transactions on Computer-Aided Design Of Integrated Circuits And Systems, Vol. 30, No. 9, September 2011.

[9] Jia Zhan, Nikolay Stoimenovy, Jin Ouyangz, Lothar Thiele, Vijaykrishnan Narayanan and Yuan Xie, “Designing Energy-Efficient NoC for Real-Time Embedded Systems through Slack Optimization”, Proceedings of the 50th Annual Design Automation Conference, ACM 2013.

[10] J. Duato., “A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks”, IEEE Trans on Parallel and Distributed Systems, pp. 1320-1331, 1993.

[11] J. Duato, “A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks”, IEEE Trans on Parallel and Distributed Systems, pp. 1055 –1067, 1995.

[12] S. Ma et al., “Whole Packet Forwarding: Efficient Design of Fully Adaptive Routing Algorithms for Networks-on-Chip,” in proc. HPCA, pp. 1-12, 2012.