TRAFFIC-BALANCED IP MAPPING ALGORITHM FOR 2D-MESH ON-CHIP-NETWORKS

Ting-Jung Lin, Shu-Yen Lin, and An-Yeu (Andy) Wu

Department of Electrical Engineering, Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan ROC

ABSTRACT

Intellectual Properties (IPs) mapping algorithms for On-Chip-Networks (OCNs) allocate a set of IPs onto given network topologies. The existing mapping algorithms limit a single IP to connect to a single router. Hence, the IPs with large communication volumes will result in heavy traffic loads of certain routers. Those routers may become hot spots due to high power density, which affects the reliability of chips. In this paper, new Network Interfaces (NIs) were proposed to solve the aforementioned problem. Traffic-Balanced Mapping Algorithm (TBMAP) is also proposed based on the new NIs. The traffic loads then become more decentralized, and the traffic of all the routers on the chip can be balanced without sacrificing the networking performance. The TBMAP has short runtime to achieve balanced network traffic loads, which leads to the enhanced performance of OCNs. The experimental results show that at least 24% communication time is saved for real applications.

Index Terms— Interconnection networks, network interfaces

1. INTRODUCTION

On-Chip-Networks (OCNs) were proposed as a novel and practical communication platform for future System-on-Chip (SoC) design [1][2]. The existing ad hoc point-to-point interconnections or buses connections between IPs fail to be scalable. OCN provides an integrated solution to the incoming routing and scalability problems.

2D mesh topology was the most adopted for OCNs for its flexibility, scalability and regularity. Given a communication task graph (shown in Fig. 1(a)), the IP mapping algorithms automatically map each IP to a tile which connects to a router. The result is shown in Fig. 1(b).

The existing IP mapping algorithms limit a single IP to connect to a single router [5][6][7]. In some applications, there are drastic differences between communication volumes of different IPs, as shown in Fig. 1(a). The IPs with large amounts of communication volumes will result in heavy traffic loads of certain routers. For example, in Fig. 1(b), IP0 causes the high traffic load of R0. These routers may influence the performance of the system. Besides, during operations of the OCNs, those routers may become the hot spots due to high power density, which will affect the reliability of the chip [3][4].

To reduce the unbalanced traffic loads, we proposed new styles of NIs, Single-Router to Single-IP (SRSI), Single-Router to Multiple-IPs (SRMI) and Double-Router to Single-IP (DRSI). SRSI is similar as the existing styles of NI between one router and one IP. DRSI allows one IP to connect to two routers, and SRMI allows one router to connect to multiple IPs. Fig. 1(c) shows a mapping based on the proposed NIs. According to the communication task graph shown in Fig. 1(a), IP0 has especially large communication volume. With DRSI, the traffic of IP0 is distributed to R0 and R1. The maximum traffic load for a router is reduced from 1800 units to 1000. Since the traffic loads of these routers decrease, the networking performance is improved.

Based on the new NIs, we also proposed Traffic-Balanced Mapping Algorithm (TBMAP). TBMAP uses a modified version of branch-and-bound search to map all the IPs with different types of NIs onto 2D mesh OCNs [7].

Our contributions are summarized as follows:

- With the proposed NIs, the traffic of the routers in OCNs becomes balanced. And the number of routers adopted for an application can be reduced.
- TBMAP generates IP mapping with better performance. The experiments for real applications show the packets transmission time on OCNs is reduced at least by 24%.

Fig. 1 (a) An example of communication task graph. (b) An illustration of IP mapping. (c) IP mapping based on the new NIs.
2. RELATED WORKS

We introduce three previous works for IP mapping onto 2D mesh OCNs: NMAP [5], BMAP [6] and Energy and Performance Aware Mapping (EPAM) [7].

2.1. NMAP [5]

NMAP was proposed to find the mapping with the least latency under bandwidth constraints. The optimizer was based on simulated annealing. However, according to [6], the drawback is its large runtime complexity.

2.2. BMAP [6]

BMAP was introduced to minimize hardware cost. It adopted binomial IP merging to centralize the traffic loads, as shown in Fig. 2. The greedy binomial mapping algorithm shortened the runtime of mapping algorithm for its simplicity. Since the traffic is centralized to some routers, BMAP requires routers designed with different throughputs and frequencies. Only the routers with heavy traffic loads require better performances, and more hardware costs. BMAP reduced total traffic load, hardware cost and runtime. However, those routers with large traffic may become the hot spots due to high power density, which may affect the reliability and networking performance.

2.3. Energy and Performance Aware Mapping (EPAM) [7]

EPAM was proposed to find legal mappings with least total communication energy. The optimizer of the EPAM was based on branch-and-bound search trees. The internal nodes of the trees represent incomplete mapping results, and the leaf nodes represent all the possible solutions. Fig. 3 gives an example. Since a leaf node has the best energy cost, 80 mJ, other nodes with cost greater than 80mJ are discarded. The EPAM adopted the calculations of upper and lower bound for each incomplete mapping to decide whether to discard some subtrees in order to enhance efficiency. For an internal node, at least one complete solution following current incomplete mapping consumes energy less than or equal to the upper bound, while all the complete mapping results must consume more energy than the lower bound. Moreover, EPAM adopted priority queues with limited size to further control the growing of search trees.

The EPAM can save significant communication energy and runtime of algorithm. However, EPAM also fails to take balanced-traffic into account.

3. PROPOSED NI

The existing NI between one router and one IP was modified to allow more types of connections between IPs and routers for better traffic performance. Thus, three types of NIs are proposed.

3.1. Classifications

1) Single-Router to Single-IP (SRSI): A single IP connects to a single router. In Fig. 1(c), IP, is connected to R, by an SRSI.

2) Double-Routers to Single-IP (DRSI): In [11], the proposed configurable NIs allow a functional block to connect to different routers. To avoid modifications on the network topology or increasing routing complexity, we allow DRSI to connect to two neighboring routers at most. In Fig. 1(c), IP, connects to R, and R, by a DRSI.

3) Single-Router to Multiple-IPs (SRMI): Multiple IPs share an SRMI and connect to a same router by a local bus. An arbitrator [8] [9] or multiplexer [10] decide which IP or router is active to use the bus in each time slot. In Fig. 1(c), the connection between IP, IP, and R, gives an example of SRMI.

3.2. Architecture of NI

The architecture of the proposed NIs (SRSI, DRSI, SRMI) was designed. Our NIs are based on the functions mentioned in [2]. The functions of an NI include data packetization, header building, packet switch, wormhole routing, and data depacketization. The SRSI type NI structure is shown in Fig. 4. The synchronous interface is for handshaking. DRSI and SRMI are both based on SRSI.

For an IP with DRMI as shown in Fig. 5(a), we added a distance computing circuit. When sending, the circuit computes the distances between the destination and the two locally connected routers. It decides which router is used by choosing the one that causes less distance.

For SRMI shown in Fig. 5(b), we first added an arbiter. Upon receiving a flit, the arbiter decides which IP to get the
data according to the address in the flit header. On sending, SRMI checks which IPs request to send data. If the two request to send at the same time, the arbiter alternatively grants the two IPs the priority to send.

![Fig. 4 The architecture of SRSI.](image)

![Fig. 5 The architecture of (a) DRSI and (b) SRMI.](image)

### 4. TBMAP ALGORITHM

The TBMAP was proposed based on EPAM [7]. TBMAP can support SRSI, DRSI and SRMI. Extra steps of the TBMAP are shown in Fig. 6. The algorithm first decides the type of NI for each IP. The algorithm consists of two major conversions between different kinds of NIs.

1) SRSI to DRSI: An IP connected with a DRSI requires two routers. If there are available routers, the algorithm picks the IP with the largest communication volume and converts its NI into a DRSI. During the conversion, the sending and receiving traffic of the IP will be redistributed to the two connected routers.

2) Non-DRSI to SRMI: If the termination condition is not satisfied, the algorithm decides whether to merge two IPs or terminate the iterations. When two IPs are merged, they share a single router by an SIMR. Otherwise, the iterations terminate. We terminated the algorithm as the variance of the traffic loads decreases by a certain amount. The condition for termination can be customized.

Then the EPAM is modified for different types of NIs. Each type of NIs and the connected IPs occupy different numbers of tiles on 2D meshes, as shown in Table I.

**Table I. The number of tiles occupied by IPs with NIs**

<table>
<thead>
<tr>
<th>Type of connected NI</th>
<th>SRSI</th>
<th>SRMI</th>
<th>DRSI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of tiles occupied</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

5. EXPERIMENTAL RESULTS

We implemented the SRSI, DRSI and SRMI to prove their functionality. Then we applied the algorithms for IP mapping to real applications. The static simulations show that more balanced traffic is achieved by TBMAP. Besides, the dynamic simulations show the improvement of networking performance for these applications.

5.1. Implementation of NIs

The network environment is a 2D mesh with wormhole switching. All channels of routers have output queuing buffer to IP with depth of 4. We assumed that each flit contains 34 bits and each packet is formed with 5 flits. Table II shows the synthesis results with the least delay of the NIs via TSMC 0.13μm CMOS technology. Since the DRSI adopts two SRSI to connect to two routers, the area is nearly doubled. In SRMI, the MUX and DEMUX cause the most area overhead, which depends on the width of data. Besides, the delays of DRSI and SRMI are slightly greater than SRSI. The distance computing circuit in DRSI causes the circuit delay, while the MUX and DEMUX causes delay in SRMI. The hardware and timing overhead shows that the proposed NIs will not cause extra restrictions on router design.

**Table II. The synthesis results of the NIs**

<table>
<thead>
<tr>
<th>NI</th>
<th>Area(μm²)</th>
<th>Delay(ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRSI</td>
<td>51669</td>
<td>2.4</td>
</tr>
<tr>
<td>DRSI</td>
<td>103496</td>
<td>3.6</td>
</tr>
<tr>
<td>SRMI</td>
<td>93887</td>
<td>2.8</td>
</tr>
</tbody>
</table>

5.2. Static Simulations

We applied traffic models of two applications, MPEG-4 decoder [6], and a sample of video stream. Fig. 7 shows the network traffic loads of the MPEG-4 mapping results with different methods. It is shown that the traffic is more balanced in the case of TBMAP (Fig. 7(d)). The results of the other applications are similar.
Table III gives the averages (Avr.) and standard deviations (Std.) of traffic loads resulted from EPAM, BMAP and TBMAP for the two applications. The Std. of traffic loads resulting from TBMAP decrease at least 36.8% compared to BMAP, and 41.3% compared to EPAM. But since we make traffic loads more balanced on a chip, some data paths might not be the shortest possible ones. Therefore, the average traffic loads may be higher in some cases.

Table III: Avg. and Std. of the Traffic Loads

<table>
<thead>
<tr>
<th>Application</th>
<th>Algorithm</th>
<th>Avr. (Mbits)</th>
<th>Std. (Mbits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG-4</td>
<td>EPAM</td>
<td>891.3</td>
<td>1244.6</td>
</tr>
<tr>
<td></td>
<td>BMAP</td>
<td>1364.8</td>
<td>1156.1</td>
</tr>
<tr>
<td></td>
<td>TBMAP</td>
<td>910.3</td>
<td>730.1</td>
</tr>
<tr>
<td>VS (video stream)</td>
<td>EPAM</td>
<td>626.3</td>
<td>481.2</td>
</tr>
<tr>
<td></td>
<td>BMAP</td>
<td>664.5</td>
<td>474.8</td>
</tr>
<tr>
<td></td>
<td>TBMAP</td>
<td>666.8</td>
<td>99.2</td>
</tr>
</tbody>
</table>

5.3. Dynamic Simulations

To evaluate the realistic performance of the proposed algorithm, we have developed a simulator in SystemC language. We assume that one flit passing through one router takes one cycle. IPs generate packets at time intervals chosen from a negative exponential distribution. For each run of simulation, we discard the first 10,000 warm-up packets. After simulations, we obtain the time interval which is required to transmit the packets shown in Fig. 8. The experimental results show that compared to EPAM, TBMAP reduces 26.5% ~ 49.1% transmission time, and compared to BMAP, 23.9% ~ 49.3% reduction is shown.

6. CONCLUSIONS

With the proposed NIs, the traffic loads are decentralized and the traffic of all the routers on the chip becomes balanced. The networking performance of OCN is improved. The experiments show that at least 24% reduction of packets transmission time for real applications. Besides, TBMAP provides the flexibility to make a trade-off between runtime of algorithm and OCN performance.

Fig. 8. The total packets transmission time.

7. REFERENCE