Multicast-Aware High-Performance Wireless Network-on-Chip Architectures

Abstract: Today’s multiprocessor platforms employ the network-on-chip (NoC) architecture as the preferable communication backbone. Conventional NoCs are designed predominantly for unicast data exchanges. In such NoCs, the multicast traffic is generally handled by converting each multicast message to multiple unicast transmissions. Hence, applications dominated by multicast traffic experience high queuing latencies and significant performance penalties when running on systems designed with unicast-based NoC architectures. Various multicast mechanisms such as XY-tree multicast and path multicast have already been proposed to enhance the performance of the traditional wire line mesh NoC incorporating multicast traffic. However, even with such added features, the multichip nature of the wire line mesh NoC leads to high network latencies and thus limits the achievable system performance. In this paper, to sustain the high-bandwidth and high-throughput requirements of emerging applications, we propose the design of a wireless NoC (WiNoC) architecture incorporating necessary multicast support. By integrating congestion-aware multicast routing with network coding, the WiNoC is able to efficiently handle heavy multicast injections. For applications running with a broadcast heavy Hammer cache coherence protocol, the proposed multicast aware WiNoC achieves an average of 47% reduction in message latency compared with the XY-tree-based multicast-aware mesh NoC. This network level improvement translates into a 26% saving in full-system energy delay product. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

Existing System:

Conventional NoCs convert multicast messages into multiple unicast messages. This causes sharp increases in traffic injection rates, which ultimately lead to high queuing delays and poor system performance. To eliminate these queuing delays and improve the latency in delivering multicast messages in mesh NoCs, path multicast methodologies have been proposed. In a path multicast methodology, for each multicast message, the overall mesh network is divided into multiple destination regions. The number of such regions and the size of the each destination region is determined by the source node of the message and the variation of the adopted path multicast algorithm. For each destination region, one copy of the original multicast message is
created and this copy is routed across all the destinations of the region in ascending/descending order of the destination node addresses. As we show later, for cache-coherence-induced traffic patterns, the path multicast methodology can lead to large destination regions, which require a high number of hops to distribute the multicast messages. This ultimately leads to high network latency and poor system performance.

A dynamic path multicast mechanism where the network is recursively divided into multiple small destination regions is proposed. However, for all path multicast mechanisms, at each destination along a path, the header flit is needed to be repackaged with updated list of destinations. Hence, path multicast mechanism requires complex router architectures.

The XY-tree multicast mechanism for mesh NoCs has been proposed. Under this methodology, the multicast message is first forwarded from the source node to all the intermediate nodes lying in the same row. The message is then replicated at these intermediate nodes and a copy of the original message is forwarded to all the destinations lying in the same column. Unlike the path-based multicast mechanism, XY-tree multicast does not require a complex router and involves simple message forwarding.

For cache-coherence-induced traffic patterns, each multicast message is associated with a set of acknowledgement (ACK) messages that are transmitted from each multicast destination back to the source node. To meet the requirements of the cache coherence communication, an XY-tree multicast NoC incorporating an ACK aggregation network is proposed. A mesh NoC performing broadcasts over uncongested trees and incorporated with ACK aggregation and single-cycle multiport replication of broadcast flits is presented. These works demonstrate that with an efficient replication mechanism and with the use of ACK gathering networks, the performance of the many-core systems incorporating Hammer cache coherence protocol can be greatly enhanced. However, in these systems, the underlying NoC is still a mesh network. In a mesh NoC, the network latency is usually high due to the inherent multichip nature of the system. High network latencies cause undesired delays in forwarding the multicast messages as well as in collecting the ACKs, leading to stalled processor cycles. In addition, high multicast injection rates can lead to heavy congestion in a mesh network where the XY-tree multicast follows a single fixed path to transmit data.

Disadvantages:
Do Your Projects With Technology Experts…

- queuing latencies is high

**Proposed System:**

Cache coherence traffic patterns:

In a chip multiprocessor (CMP) platform, cache coherence protocols are used to ensure consistency among the multiplexed copies of shared data. Cache coherence protocols play a vital role in determining the nature of the on-chip network traffic. In general, selecting an appropriate cache coherence protocol for a CMP involves analyzing the area and traffic trade-offs associated with the protocol.

**Directory Protocol:**

In a directory-protocol-based CMP, each translation lookaside buffer entry is assigned to one of the cache controllers, called as the home node. The home node maintains the list of processing cores that have the most recent copy of the datablock. Whenever a processing core S needs to update a datablock or encounters a cache miss, it first communicates with the home node. For a read miss, if the requested data block is present on the chip, the home node forwards this request to the core D that has the most recent copy of the requested data. Core D responds by forwarding the data to the requested node (S).

**Hammer Protocol:**

In the Hammer protocol, similar to the directory protocol, each data block is assigned a home node. However, unlike the directory protocol, the home nodes in hammer protocols do not maintain the list of cores that are sharing the most recent copy of the data block. Instead, the home node broadcasts the read/write requests to all the cores in the system. This enables the Hammer protocol to have a low logic and memory overhead. Hence, the Hammer protocol enhances the scalability of the many-core system as it does not require large memory structures to keep the list of shared cores typically needed in directory-based protocols. However, the systems operating with the Hammer protocol require multicast-enabled NoCs to handle the heavy broadcast traffic.

**Analyzing Directory and Hammer Protocol Traffic:**
In order to design an efficient multicast-aware WiNoC incorporating the directory and Hammer protocols, we need to study the nature of the cache coherence traffic patterns. Hence, we discuss the traffic patterns induced by the directory and Hammer protocols. For these analyses, we consider a system consisting of 64 ALPHA cores operating with the two-level MESI directory and AMD HT cache coherence protocols.

MULTICAST-AWARE Wi NoC NETWORK DESIGN:

WiNoC Design Constraints:

In our WiNoC, each processing core is connected to a router and the routers are interconnected following a power law-based SW connectivity. Essentially, SW networks incorporate multiple short-range links and a few long-range shortcuts so as to minimize the overall inter-router hop count ($H_{avg}$). However, while adding links in SW NoCs, we need to follow certain restrictions. First, we should restrict the average number of ports ($K_{avg}$) per router to four so that the WiNoC does not introduce any additional router port overhead compared with a conventional mesh. Next, we should restrict the maximum number of ports in a router ($K_{max}$) so that no particular router becomes unnecessarily large. For a system size of 64 cores, the SW-based networks achieve the highest throughput with lowest energy dissipation when the maximum port count in a router is restricted to seven. Finally, for link lengths exceeding 6.25 mm in 28 nm CMOS technology node, wireless links are more efficient than the traditional metal wires in terms of energy delay product (EDP). Hence, WiNoCs should prefer using wireless links instead of wire line links for data exchanges exceeding a communication distance of 6.25 mm. However, depending on the available wireless resources and allowed area overhead, we can make only a limited number of the longest links wireless, while the others need to remain wire line.

Orthogonal Paths:

In order to facilitate the congestion-aware multicast routing, we propose that each non-wireless node in the system is connected to all the WIs in its own region via two orthogonal wire line paths, where none of the links in these two paths are being the same. This can be achieved by creating two orthogonal spanning trees for each region from the existing wire line connectivity.

Advantages:
queuing latencies is low

Software implementation:

- Modelsim
- Xilinx ISE