Cube NoC based on Hybrid Topology

1R.Suraj and 2P.Chitra

1Research Scholar, Computer Science & Engineering, Thiagarajar College of Engineering, Madurai, India
2Assistant professor, Computer Science & Engineering, Thiagarajar College of Engineering, Madurai, India

ABSTRACT

In computing field, demand of speed is going beyond the limit, thus to satisfy the high-end needs, multi-core processors are playing its crucial role. As far as multicore is concerned, there are few difficulties to be optimized for the promising future processor infrastructure, SoC(System on chip) based processor is being replaced with NoC(Network on Chip). NoC is expected to overcome the limitation of SoC, with respect to scalability and communication overhead. They also offer high throughput and load distribution in the chip. This paper focuses on Cube based Network on Chip architecture modeled by 3D-NoC with hybrid topology and an optimized deterministic routing. NoC offers high bisection bandwidth and supports large number of cores, so layered cores where stacked to build up Cube based model. For Hybrid topology, planar mesh with diagonal links implanted. The major challenges in using cube based NoC are addressed by the basic block of topology, router design, route handling, deadlock avoidance and thermal issues. This model helps in obtaining a reliable short path between the processing elements over high flit traffic across router, failure of router and on thermal considerations.

INTRODUCTION

The performance of most digital system is limited by its communication. System-on-chip (SoC) architectures offers communication, both from physical wiring and distributed computational primitives. For larger SoC, chip capacity and buses does not scale well with the bandwidth, clocking frequency and power. A bus system has very limited concurrent communication capability, since only one device can drive a bus segment at a time. Wiring delays are dominating over gate delays, which favor short links. A bus based architecture has a critical performance and power bottleneck due to the scalability problem.

Above factors motivate the research on Network-on-Chip (NoC) that brings the techniques developed for macro-scale, multi-hop networks into a chip.

NoC

Network on chip architectures adopt design concepts and methodologies from computer networks. Interconnected routers with IP cores make a chip communication model. NoCs present better scalability of processing elements, improved performance and bandwidth of communication than shared bus. But implementing network on silicon chip requires broad research, as network architectures and protocols have to deal with the pros and cons of the silicon die. Optimal network on chip require unique methodologies for both switch designs and routing algorithm designs for deploying network on silicon. (Figure 1) shows the 2D NoC, in which planar mesh interconnect is implanted.
The design of these NOCs, requires satisfaction of multiple conflicting constraints that includes minimizing packet latency, reducing router area and reducing communication energy. In addition to basic packet transport, future NOCs will be expected to provide certain advanced services. In particular, quality-of-service (QoS) is emerging as a desirable feature due to the growing popularity of server consolidation, cloud computing, and real-time demands of SOCs (Grot et al., 2011).

3D NoC

Stacking multiple layers together using vertical links forms a 3D model. 3D NoC overcomes the limited scalability of 2D NoC over 2D plane by using short and fast vertical interconnects of 3D-IC’s. Figure 2 shows a typical 3D NoC which uses stacked 2D and short vertical links for interconnecting layers.

In the challenge of designing mixed signal chips which combine analog processing IP blocks, such as antenna and the digital IP blocks, such as microprocessors and memories, in conventional planar chip-making processes, analog IP blocks are kept on one layer plane, the digital IP blocks are placed on one or two, other layers and combined into a chip, which is termed as 3-D IC. Stacking dies reduces the wire length, increase the performance and reduces the power consumption.

DESIGN PARAMETERS

A NoC is a set of interconnected routers, with IP cores connected to these routers. NoC has four main components namely interconnect topology, router, switching logic and virtual channel. A router defines the path between input and output switch ports, when multiple requests arrive in parallel arbiters grant access to a given port. A buffer is used to store intermediate data and flow control module is used to regulate the data transfer to the next switch (Archana et al., 2012).

Electronic interconnection and packaging is mainly performed in a planar, 2D design style. Further efficiency and performance enhancement of electronic systems will require the use of 3D interconnection schemes. A basic reason for 3D-integration is system-size reduction. Traditional assembly technologies are based on 2D planar architectures. As Technology improves, memories and processors have become small and inexpensive (Dally and Towles, 2011).

Topology

An important part in design of NoCs, is choosing the most suitable NoC topology for a particular application and mapping of the application on to that topology. There are several standard topologies onto which an application can be mapped.

The 3D crossbar architecture (Lewis et al., 2009) significantly reduces power consumption and delay in every flow direction of the network.

NoC architectures where long-range links are inserted on top of a mesh network is proposed in (Gebali et al., 2009). Here, NoC is transformed to an application specific one, serving best the traffic, but it is limited to two dimensions. The presented a quantitative evaluation of 2D point-to-point and bus NoC interconnection approaches. In this work, an MPEG-2 implementation is studied, exploiting the aforementioned interconnection solutions and proved that the NoC-based solution scales very well in terms of area, performance and power consumption.
In (Tatas et al., 2013), an application specific 3D-NoC with mesh topology is proposed. The floorplanning follows 3D wiring of TSVsand merging of router optimises the topology. It reduces significant amount of hop count and achieves 74% power reduction. 3D integration is achieved by stacking a number of 2D layers.

Tsai (2010) proposed Bus and mesh topology were merged to hybrid topology and separated into multiple subsystems so that, high affinity IP cores are placed in same subsystem. AMBA bus architecture was used to integrate IP cores. A partition Mapping algorithm was proposed to reduce the communication latency. A good percentage of latency improvement was achieved, when the mesh size increases. The hybrid architecture was proved to be suitable for the system with unbalanced communication requirements.

Interconnection are the key to the success of future digital system. They are broadly classified as direct and indirect topologies. Each switch is connected to a single core in direct topologies. In indirect topologies, a set of cores are connected to a switch. Choosing the right topology involves minimizing average communication latency, power consumption, area, etc over various design objectives. Moreover, in the resulting NoC, the links should support the desired traffic between various cores. Thus the mapping of cores onto the NoCs, must satisfy the bandwidth constraints of the application.

**Router**

The router is responsible for receiving incoming packets, storing packets, routing those packets to specified output port and to forward packets to others routers.

A routing function determines the path of a packet from its source to the destination. Most networks use deterministic routing schemes, whose chief appeal is simplicity. In contrast, adaptive routing can boost throughput of a given topology, at the cost of additional storage and allocation complexity (Senthilkumar and Rajani, 2012).

**Switching**

The two main switching techniques are circuit and packet switching. The other switching such as, wormhole switching and virtual cut through(VCT) where developed by taking the cons and pros of circuit and packet switching. In the case of circuit switching, a path from the source to the destination of data is reserved prior to the transmission, for the duration of the data transmission and is accessible only by this set of data. Here, there is no need for the data to be packetized. Circuit switching, however, is inefficient in terms of link capacity utilisation and set-up latency unless used for long infrequent messages.

Packet switching (AV de Mello, 2010) is by far the most employed switching mechanism in NoCs. Packet switching requires the use of a switching mode, which defines how packets move through the routers (Liang, 2010).

Wormhole switching (Jin Liu, 2007) has been widely used in practical multi-computer and NoC, where small and faster router is needed. In wormhole switching, packets are broken up into flits for transmission flow control. Header flit contains routing information, and once a header is received at intermediate node, the routing decision is made and the header is forwarded to next node in absence of blocking. The remaining data flits follow the route traversed by the header flit. Input or output queues of a wormhole router only need to be able to store a few flits. The flits are pipelined through the network in a similar manner, as VCT in the absence of blocking due to a message is typically too large to be completely stored in a router.

<table>
<thead>
<tr>
<th>Packet</th>
<th>Head</th>
<th>Source Address</th>
<th>Target Address</th>
<th>Data</th>
<th>Sync flag</th>
<th>Trans id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flit Header</td>
<td>Head</td>
<td>Address to write Data</td>
<td>Seq no</td>
<td>Timer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flit 2</td>
<td>Head</td>
<td>Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flit n</td>
<td>Head</td>
<td>Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 3:** Packet and Flit header format
The packet and flit format is shown in Figure 3. The packet to be routed are split as flits for wormhole routing (Faizal et al., 2009). The flit has a header, message body and end of message body or the last flit of the message. Only one header flit is asserted for one message (as one packet), even if the size of the message is extremely large.

**Routing Algorithm**

Routing algorithm defines the path taken by a packet between source and target switches. They must prevent deadlock, livelock, and starvation situations.

Deadlock may be defined as a cyclic dependency among nodes requiring access to a set of resources, so that no forward progress can be made, no matter what sequence of events happens. Livelock refers to packets circulating the network without ever making any progress towards their destination. Starvation happens when a packet in a buffer requests an output channel and the packet is blocked because the output channel is always allocated to another packet.

An efficient gate-level implementation based on a mesh topology with multicast routing without virtual channels is proposed in (Dally, 1992). Multicast packets are injected to the network by sending multiple packet headers. Deadlocks in tree-based multicast routing are handled using a flit-by-flit round arbitration. A fair hold-release tagging mechanism guarantees lossless flit ejection in multiple destination nodes, even if the size of the multicast messages is long.

The authors in (Yuho Jin, Akram Ben Ahmed et al., 2009), studied the router architecture and network topology for communication behaviors in large-scale cache systems with Nonuniform Cache Architectures (NUCAs). Fast-LRU replacement strategy was used for L2 cache, where cache replacement overlaps with data request delivery. Halo network topology reduces the required links and improves inter process communication (IPC) speed above 30%. Look-Ahead-XYZ routing algorithm for 3D-NoC minimizes communication latency and power consumption. It used 3DOASIS-NoC system which follows wormhole routing to enable no-load bypass technique. The matrix arbiter scheduling grants permission to higher priority request in scheduling-matrix. Governed by two signals to detect about the buffer occupancy status, greatly reduces the latency.

Routing algorithms can be classified according to the three different criteria:

(i) where the routing decisions are taken

(ii) how a path is defined

(iii) the path length.

According to where routing decisions are taken, it is possible to classify the routing into source and distributed routing (Feihui Li et al., 2006). Depending how a path is defined, routing can be classified as deterministic or adaptive.

**Flow Control**

NoC router should implement per flit handling strategy with wide granularity. This requires an enhanced internal architecture that ensures from one hand a specific management according to a service classification and on the other hand, it enhances the routing process.

Flow control (Senthilkumar and Rajani, 2012) governs the flow of packets through the network by allocating channel bandwidth and buffer slots to packets. Conventional interconnects have traditionally employed bandwidth and storage allocation by Virtual Cut-Through (VCT) flow control. In contrast, NOCs have relied on flit-level flow control refining the allocation granularity to reduce the per-node storage requirements.

**Virtual Channel**

Network throughput can be improved by dividing the buffer storage connected with each network channel into several virtual channels (Dally, 1992). Each physical channel is associated with several small queues, rather than having single long queue. The virtual channels associated with one physical channel are allocated independently, but compete with each other for physical bandwidth. Virtual channels decouple buffer resources from transmission resources. This decoupling allows active messages to pass blocked messages using network bandwidth that would otherwise be left idle.
MATERIALS AND METHOD

Why to Cube Model

Model offers regular structure and non congested floorplanning. For enterprise processing solution, increasing the processing elements linearly will not make much difference. This approach allows the exponential increase of Processing Elements (PEs) and has an effective routing strategy and offers deadlock aware minimal hop network communication, fail-safe and thermal aware execution.

In traditional NoC architecture, router will be separate element and attached to switch in PEs. Figure 4 shows the OSI network model used in cube NoC. In the proposed cube model, each PE is equipped with switching and routing logic. The processing element with the switching and routing logic is shown in Figure 5. As in 2D NoC layer, PE can be implemented on one of the physical planes of the system, consisting of n*n PEs in each layer. For the 3-D system, n such layers are stacked and hence the total number of PEs will be n*(n*n).

The router has control logic to monitor the power and thermal impact, it also maintain an access control list which include the neighbor node power and thermal details. As flit traffic across the router increases the usage of power and subside it increase the thermal of processing element. So to avoid the bad impact of traffic the routing logic should be consistent over all sort of nature. Cube model offers a intelligent routing logic which helps to act according to the nature of traffic and failure.

Fig. 4: OSI model for NoC

Floor Planning

In Cube based NoC every element is arranged in 3D matrix form, every elements of processing block designed and developed as cubes[Fig 5,6]. Final production of NoC offers cube properties. Structure of NoC will be n*n*n matrix model, which grows in order of 3.

Processing Element with Router

In cube based model, router will be embedded with the processing element. Router has seven input and output ports with selector and arbiter. This model avoid unnecessary space and wire delay as its close to the IP of the processing Element.

Fig. 5: Processing element with router

Figure 6 shows an example cube of 3*(3*3) PEs and their interconnection.
**Topology**

Topology is developed for reliable flit transfer between the PEs. The proposed topology takes the advantages of mesh with a custom topology and it tends to be a hybrid one.

Mesh is the primary topology over the chip and there are diagonal fail safe link outer of the model. Figure 6 shows the diagonal connection established for cube model.

The topology switching used has seven links port in each PEs, one to connect the IP and the router, and six other to connect to adjacent PEs within the layer or if the router is in edge then in six, three ports connect to its diagonal PE router port.

**Routing Logic**

Packet transfer can take place either within the same layer or across the layers. According to the Figure 5 there are seven links denoted by LIP, LE, LW, LN, LS, LUP, and LDOWN. The routing algorithm has to determine the communication is within the same layer or up/down layer.

LIP – link between the IP and the switching port.
LE, LW, LN, LS - act as the port for communicating within the layer.
LUP, LDOWN - act as the port for communicating among the layers.
LE, LW, LN, LS, LUP, LDOWN- also used to communicate with diagonal nodes

XY routing is used, if the communication is within the same layer. If it’s to other layer, then the routing logic determines the shortest failsafe link and completes the transfer considering the traffic and thermal impact.

In moderated NoC environment due to flit traffic, power and thermal issues, router in PEs may fail to carry on the flit transfer across network. The proposed model can operate without interruption even under such situations. As it has diagonal link between bottom layer corner to top layer corner (Figure 6) even their some failures in intermediate node over mesh interconnect, diagonal link act as a failsafe communication for the model and offers many dedicated paths to proceed the operation. Let us have ‘n’ nodes in L number of layers.

In Cube model, 4 PE’s have failsafe diagonal links:

L(0,0,0) - L(n,n,n) , L(0,0,n) - L(n,0,0) , L(0,n,0) - L(n,0,n) and L(0,n,0) - L(n,n,0)
Each corner PE contribute 3 diagonal links over the input and output port. Considering our 3*3*3 model(Figure 6), model exhibits 12 (4 PE*3 link) failsafe links for flit transfer.

This enables a failsafe interconnect model for cube NoC.

If the local IP core at the bottom layer, wants to communicate to the IP core at the top layer, it does not need do a multiple hop by jumping layer by layer, instead jump to the top via diagonal link and do the needful action. Figure 7 gives the pseudo code for routing logic.

```
Declare source node as S(ZXY)
Define order of cube n
Cube elements ce=n^3
Access Control list of Router Traffic T
Choose destination node D(ZXY)

//Find the level/layer of source and destination
procedure level(S,D)
If curlevel = destlevel  //in same level/layer
samelevel(S,D)
else
  diagonal(S,D)
End If

Compute procedure level(source S,destination D)
// level of commutation is determined by Z value.
BEGIN
If S(Z) equal to D(Z) then
  Both in same layer/level
  //current level curlevel
  //destination level destlevel
End If

Compute procedure samelevel(source S,destination D)
BEGIN
If S(X) < D(X) & consider min. Traffic T then
  //Route packet to S(X+1),D(Y)
Else
  //Route packet to S(X-1),D(Y)
End If
If S(Y) < D(Y) & consider min. Traffic T then
  //Route packet to D(X),D(Y+1)
Else
  //Route packet to D(X),D(Y-1)
End If
End If

Compute procedure multilevel(source S,destination D)
//Determine the up/down
If curlevel<destlevel
  S(Z)=S(Z+1,X,Y) //do a hop to S(Z+1),thus it move to upper level
diagonal(S,D)
Else
  Route packet to S(Z-1,X,Y) //do a hop to S(Z-1),thus it move to lower level
```

diagonal(S,D)
END IF

//If there is high traffic on router or failure of router/PE the diagonal link is chosen
Compute procedure diagonal(source S, destination D)
BEGIN
Route packet to S to D
//switch to adaptive routing with minimal hop
Check with Access control list of Traffic T
//recursively run below until S reaches to D
Samelevel(S,D) //run to any corner of level so that diagonal link accessible)
Multilevel(S,D) //switch to other level

Fig. 7: Pseudo code for Routing Logic

For packets routed across layers, the router makes routing decision based on flit traffic and latency on port. When a packet is to be delivered between the routers in the same layer then, the router uses XY routing and finds the minimal hop path between them.

If the source and destination are in the different layer, then uses the hybrid cube routing for efficient transfer of flit. If the up/down link of the router is busy or not accessible, it will try to find an intermediate free route with a healthy vertical link at the same layer, which has a minimal distance to the destination, based on the routing table. An example for the routing is given below.

<table>
<thead>
<tr>
<th>020</th>
<th>021</th>
<th>013</th>
<th>120</th>
<th>121</th>
<th>113</th>
<th>220</th>
<th>221</th>
<th>222</th>
</tr>
</thead>
<tbody>
<tr>
<td>010</td>
<td>011</td>
<td>012</td>
<td>110</td>
<td>111</td>
<td>112</td>
<td>210</td>
<td>211</td>
<td>212</td>
</tr>
<tr>
<td>000</td>
<td>001</td>
<td>002</td>
<td>100</td>
<td>101</td>
<td>102</td>
<td>200</td>
<td>201</td>
<td>202</td>
</tr>
</tbody>
</table>

layer 0 | layer 1 | layer 2

Fig. 8: Matrix representation 3*3*3 PE

Figure 8 shows the organization of the PEs in each layer of the example given in Figure 6. Let us consider a path from source S(000) and destination D(012), it follows XY routing and take the router (001,011) to reach 012. An alternate path is to access through the diagonal link, (000,222,212,112,111,112,101,001,002,012) using hybrid cube routing.

Let us consider a path from the source S(000) at layer0 to the destination D(112) at layer1, since the level is nearby proceed with vertical link. Otherwise, if it takes many hops to reach the destination, consider the diagonal link. For this example, there are many solutions, best solution is (000,001,011,111,112) and (000,222,212,112), since the number of hop is less.

**EXPERIMENTAL SETUP**

To evaluate the proposed cube based NoC, adapted open source SystemC based Noxim simulator to inherit Cube model. As Noxim offer high flexibility for changing the architecture structures. At first it’s modified to accommodate 3D-noC properties by adding Z-Dimension. Then Router with seven input and output port was developed such a way each selector handles the input packets and the arbiter takes away the packet to output port such it was send only in a single path.

Routing Logic constructed to make the architectural model fruitful for failsafe (choose alternate path if there is link failure/busy/thermal issue) communication between PEs.
PERFORMANCE EVALUATION

The evaluation of cube model, for \( n^n^n \) \( [n=2,3,4] \) topology the Noxim Testbed analyzed with various traffic flow and distribution, buffer size, packet size and cycle count and compared with the XYZ,ZXY 3D-NoC model with same amount of PE and topology.

![Graph showing delay comparison](image)

**Fig. 8:** Impact of delay on various cube with random and shuffle traffic of Cube Routing vs XYZ vs ZXY

In cube model delay is slightly reduced due to availability of diagonal link. So it will make a impact on 3D-NoC using the proposed topology with routing logic.

<table>
<thead>
<tr>
<th>Routing</th>
<th>3D-Model</th>
<th>Traffic</th>
<th>Received Packets</th>
<th>Received Flits</th>
<th>Average Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cube (Mesh with Diagonal link)</td>
<td>2x2x2</td>
<td>Random</td>
<td>115</td>
<td>604</td>
<td>5.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shuffle</td>
<td>167</td>
<td>64</td>
<td>4.24</td>
</tr>
<tr>
<td></td>
<td>3x3x3</td>
<td>Random</td>
<td>350</td>
<td>1407</td>
<td>6.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shuffle</td>
<td>360</td>
<td>1418</td>
<td>7.1</td>
</tr>
<tr>
<td></td>
<td>4x4x4</td>
<td>Random</td>
<td>634</td>
<td>2496</td>
<td>8.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shuffle</td>
<td>677</td>
<td>2664</td>
<td>6.53</td>
</tr>
</tbody>
</table>

Table: Shows the result obtained on 4 buffer with poisson distribution, packet size min 2- max 6 for 500 cycles

CONCLUSION

3D NoC technology reduces the interconnect delays by stacking multiple layers on top of each by providing shorter vertical interconnect. A new Cube based NoC model is proposed, that provide shorter interconnects with hybrid topology. This also offers better scalability, with respect to processing elements. Incorporating router inside the PEs helps the floor planning simple. The interconnection proposed here, offers failsafe communication through the shortest link via optimized routing logic. Thermal issues have to be addressed for stacking the layers. As a future work, the strength of the model is to be compared with other 3D models for effective communication and develop thermal aware routing logic for cube based NoC model.
REFERENCES


