A Heuristic for Peak Power Constrained Design of Network-on-Chip (NoC) based Multimode Systems

Praveen Bhojwani, Rabi Mahapatra, Eun Jung Kim and Thomas Chen*
Computer Science Department, Texas A&M University, College Station, Texas
**IBM Microelectronics, Austin, Texas*
*{praveenb,rabi,ejkim}@cs.tamu.edu, *wtchen@us.ibm.com*

Abstract

Designing NoC-based systems has become increasingly complex with support for multiple functionalities. Decisions regarding interconnections between the heterogeneous system components and routing of system communication affect system performance and power consumption. This research provides a heuristic to determine the neighborhood configuration for each component. By controlling the communication bandwidth allocation, simulation results with synthetic and real workloads indicate that our heuristic is able to control the peak power consumption, but at cost of throughput degradation.

1. Introduction

With the shared bus - today's dominant interconnect template - not meeting the performance requirements of tomorrow's systems [1], a suitable replacement in the form of an on-chip packet-switched interconnection template has been suggested [1,2]. These NoCs are well suited for heterogeneous communication among cores in the SoC environment and will address performance and scalability requirements of future SoCs.

One of the facets of NoC-based system design addressed in this research is of *neighborhood determination* of the NoC's resources. Given the on-chip interconnect architecture, the system designer needs to make decisions as to how the resources in the SoC are interconnected and how communication in the system is routed. These decisions affect the power and performance of the system being developed. The authors in [3] provide an optimization problem formulation for the *mapping problem*, where they map a resource communication graph onto a target topology. Their objective is to reduce the total energy consumption. However, total energy reduction is not a sufficient objective to address other aspects of design constraints, such as *peak power* – which is critical to power constrained systems.

The neighborhood decisions are further complicated when the systems being developed are conglomerates of functionalities. These *multifunctional systems*, while providing multiple functionalities, have a consequential increase in the communication complexity within the system, making decisions more complex.

For a better portrayal of the system communication we consider the operation of the system in different *modes* for its multiple functionalities. Traditionally, when a system is referred to as being *multimode*, the mode is associated with the mode of operation of the resources that constitutes the system. For example, consider a multifunctional system such as the present day cell phone, which may behave as a MP3 player, digital recorder, PDA and cell phone. This system is not only capable of operating in the four basic modes of operation listed here, but can also operate as an MP3 player and a PDA at the same time. So the number of possible modes depends on the number of functionality. The resources that constitute the multifunctional system are capable of operating in more than a single mode of the system with differing operation/communication characteristics.

This research utilizes a *mode-based communication model* that allows for the abstraction of the multimode communication within the system. The information provided by the model aids in reducing the design space explored and yields a solution that conforms to the communication restrictions set by the design specifications. A design heuristic to aid in determining the immediate neighbors of a resource in a regular NoC topology and the routing of the communication within the system is presented here.

The solution obtained does not provide the exact mapping of resources onto the NoC. It only determines immediate neighbors of the resources in the SoC and

Proceedings of the 18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design (VLSID'05)
1063-9667/05 $20.00 © 2005 **IEEE**

routes taken by all communications in the system. This flexible solution provided can then be mapped onto the NoC template, depending on placement constraints that the system designer may have. A scenario where such flexibility may be essential is when a particular resource may need to be placed along the chip boundary for I/O operations. The solution obtained is verified through a network-on-a-chip system simulator, *NoCSim* [12].

Additionally the increasing dominance of the power consumption of these on-chip networks in present day systems, poses critical challenges that need to be addressed lest they become a bottleneck in the development of high performance systems [4]. The power consumption in the routers and the links of the Alpha 21364 microprocessor were found to be about 20% of the total power consumption [5]. In the MIT Raw on-chip network, the network components constitute 36% of the total power consumption [6]. These numbers indicate the significance of managing the interconnect power consumption.

When designing portable, battery constrained systems, the peak power consumption is a critical design time constraint. These constraints are set for reliability and power delivery purposes. As identified in [7,11], 50% of the electronic failures are temperature related, since circuit reliability is exponentially dependent on operation temperature.

When designing SoCs, predictable communication parameters allow for *off-line peak power constraining*. This research exploits this idea, and provides a bandwidth controlled peak power constraining scheme. The effect of the scheme in terms of cost and benefit is also presented.

The following section recaps some of the preliminary information that aid in understanding this research. The heuristic for determining the neighborhood and routing information for the communicating resources of the target system are presented in Section 3. Section 3 also presents the peak power constraining strategy. The experimental setup and results are enumerated in Section 4. We conclude and discuss future work in Section 5.

2. Preliminaries

2.1. NoC Architecture

Researchers have suggested the usage of regular layouts for the cores (processor cores, memory cores, etc.) constituting the SoC [2,9], i.e. utilizing topologies like the folded torus. The communication architecture, shown in Figure 1, for such systems consists of the basic building block, the *network tile*. These tiles are connected to an on-chip network that routes packets between them. Each tile may consist of one or more cores and would also have routing logic responsible for routing and forwarding the packets, based on the routing policy of the network.

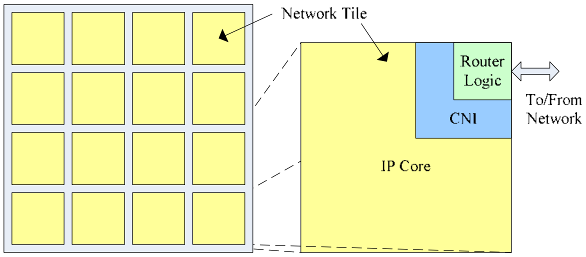


Figure 1: NoC Architecture

2.2. Power Model

Bit-level power models for on-chip networks have been utilized other researchers [3,8]. In this research, the power consumption of the NoC has been estimated at the per-flit level. Using the thermal time constant (TTC), we determine the number of flits that are present in the network during this time window and the consequent energy consumption of each of these flits. Flit traversal in the network can be broken down into a sequence of operations (1) buffer read, (2) switch traversal, (3) external link traversal and (4) buffer write [7]. The energy consumption of the flit will be:

$$E_{flit} = (n+1)[E_{BR} + E_{BW} + E_S] + nE_L$$

where, E_{flit} – energy consumption of flit, E_{BR} – buffer read energy, E_{BW} – buffer write energy, E_S – switch traversal energy, E_L – external link traversal energy, and n – number of hops to destination.

Hence the power consumption during TTC for t flits:

$$P = \frac{\sum_{t \in flits} E_{flit}(t)}{TTC}$$

where, P – power consumption, and $E_{flit}(t)$ – energy consumption for flit t.

Using energy/power values at 90nm technology from [13], we experiment to determine the effect of varying traffic loads on the energy distribution amongst the network tile components (buffer, switch and links). The energy distribution profile in Figure 2 emphasizes the contribution of the buffer operation energy to the total flit energy for the varying loads. With over 60% of the total energy contribution being made by the buffers, it stands in sharp contrast to the chip-to-chip link scenario, where the buffer contributes about 10% and the link contributes over 80% [10].

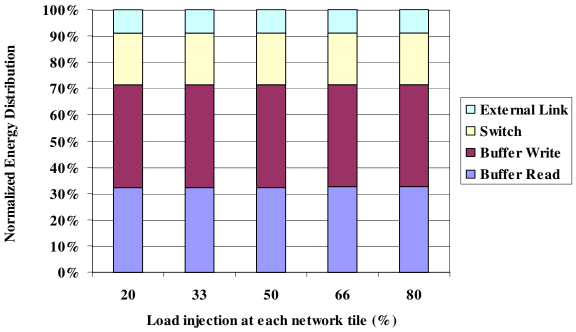


Figure 2. Energy distribution profile for different loads

2.3. Multimode System

A multimode system is defined by a set of a fixed number of modes. Each mode is characterized by a set of *communicating resources* (i.e. processing elements, memory elements, etc.) and their corresponding *communication characteristics*. The communication parameters considered are: i) bandwidth requirements and ii) data volume. A *mode characterization graph* enumerates communications in the various modes of operation of the system being developed. The following section utilizes this model.

3. Neighborhood determination for multimode systems

A crucial step in the design methodology, the neighborhood determination stage facilitates the determination of immediate neighbors of resources within the NoC topology and the communication routing between them. Power cost associated with the solution obtained is evaluated and depending on the *Peak Power Budget*, the communication bandwidth is throttled to constrain the peak power consumption.

3.1. Problem Formulation

The Mode Characterization Graph, **MCG**(**N**,**C**), shown in Figure 3(a), characterizes the communication characteristics of the target multimode system. The node set (**N**) enumerates the set of communicating resources and the edge set (**C**) specifies the communications in all modes. Communication edges are characterized by: i) *mode*: mode of operation, ii) *bandwidth required*: performance constraint, iii) *data volume*: amount of data to be transferred in each iteration, and iv) *injection load*: indicates the amount of load injected into the network by this communication pattern. It belongs to the range (0,1].

These communication edges are also referred to as secondary edges.

The Network Graph, **NG**(**T**,**L**), shown in Figure 3(b), provides the neighborhood characteristics of the network tile set(**T**). If a network edge `ne`∈ **L**, has an edge between `t1` and `t2` (`t1`, `t2`∈ **T**), then `t1` and `t2` are immediate neighbors in the NoC. The network edge `ne`, highlights the secondary edges that flow through it, and the link bandwidth that has been allocated to that communication edge.

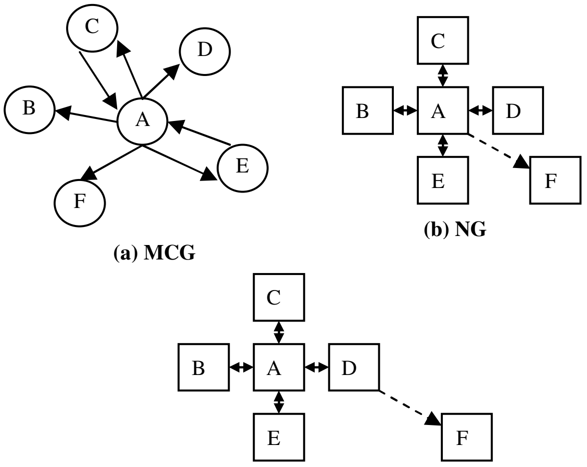
The neighborhood determination problem is framed as a transformation of the arbitrary cardinality **MCG** into the fixed cardinality **NG**. The constraint applied during transformation is *bandwidth reservation* from source to destination for each of the secondary edges.

3.2. Definitions and Operations

Definition 1: A *candidate* for transformation is a node in the MCG that has cardinality greater than the constraint set by the target topology.
Definition 2: A *critical edge* is a secondary edge that is selected by the heuristic to merge with one of the network edges.
Operation 1: The *merge operation* - `merge(se,ne)` - adds a secondary edge `se` to a network edge `ne`. During the merge operation, the heuristic attempts to allocate bandwidth to the secondary edge. If enough bandwidth is not available, the heuristic redistributes the link bandwidth amongst the secondary edges flowing through it. When the bandwidth of a secondary edge on `ne` is modified, its bandwidth is updated from source to the destination along all network edges it flows in.
Operation 2: The *rank operation* ranks the network edges with which a secondary edge would be merged. The ranking is based on three criteria:

- network edge already visited: This helps keep check on the cyclic propagation of secondary edges amongst the resources.
- network edge backtracked from: prevent wasted effort
- number of secondary edges - at the resource pointed to by the network edge - whose communication clashes with the mode of the candidate secondary edge.

Example: In the example in Figure 3, consider the given **MCG**. The target NoC topology is a 2D torus. We need to reduce the cardinality of resource A to 4 and route the extra edge through the other resources. The heuristic selects the four most critical outgoing edges of resource A and assigns them to the four vacant outgoing network edges. In this case, it has selected the edges to B, C, D and E to be the critical edges and hence places B, C, D and E as the immediate neighbors of A. The edge to F will be routed through the four outgoing edges, i.e. we need to *merge* the



(a) MCG

(b) NG

(c) NG

Figure 3: Merge Operation

communication of A to F through B, C, D or E. From Figure 3(c) we notice that the heuristic selects resource D to be the candidate through which this connection is to be routed. This selection is done based on the results obtained by the ranking operation.

3.3. Heuristic

The heuristic (see Figure 4) that has been shown here is iterative in nature. In each of the iterations we consider a resource (candidate) and attempt to make decisions on its immediate neighbors in the NoC. The objective of the heuristic is to merge these secondary edges with the network edges. These network edges are the network links between tiles in the NoC network.

When we attempt to merge a secondary edge with a network edge, there are three possible scenarios. In scenario 1, when the network edge has no destination specified yet, the merge operation will set the destination resource of the secondary edge as an immediate neighbor of the candidate. In scenario 2, the network edge has the same destination as that of the secondary edge. Here we just merge the communication and update the bandwidth for the different modes of communication on this network edge. In scenario 3, the destination of the network edge is not the same as that of the secondary edge. In this case too, we merge the communication with the network edge. Aside from that we also add a secondary edge to the destination resource pointed too by the network edge. This new secondary edge would have the same destination as that of the critical secondary edge being merged. These steps are performed for each un-merged secondary edge of each resource in the system.

Table 1. Heuristic Input Parameters

Power Budget	Thermal Time Constant	Number of VCs
Physical Link Properties (width, switching frequency)		
Target topology	Network size	Number of modes

```
while ( $n \in \mathbf{N}$ : n is a candidate )
  $SE_n$ = set of secondary edges of n
  $NE_n$ = set of network edges of n
  while ( $| SE_n | \neq \Phi$ )
    se = getCriticalSE()
    if ( $\exists$ ne $\in$ $NE_n$ : ne.dest = se.dest)
      merge(se,ne)
      $SE_n = SE_n - \{se\}$
      UpdateRoute(se)
    else if ( $\exists$ ne $\in$ $NE_n$ : ne.dest = $\Phi$)
      if ( N(ne.dest) supports a return edge to n)
        merge(se, ne)
        $SE_n = SE_n - \{se\}$
        UpdateRoute(se)
        setNetworkEdge (ne.dest, n)
        setNetworkEdge (n, ne.dest)
      else getMultihopReturnPath( )
    else
      rank ne $\in$ $NE_n$
      select best candidate for merge
      merge(se, ne)
      $SE_n = SE_n - \{se\}$
      $SE_{ne.dest} = SE_{n.dest} + \{se\}$
      UpdateRoute(se)
      if ( merge not possible )
        Backtrack2Predecessor( )
```

Figure 4. Neighborhood determination heuristic

3.4. Offline Peak Power Control

Another aspect that needs to be addressed in SoC design is that of peak power management. The peak power is the maximum power consumed during a TTC and is directly proportional to the number of flits traversing the on-chip network. It can be controlled in two ways: (i) dynamically, and (ii) statically. Dynamic peak power control would require the deployment of an on-line power control/management strategy. Some techniques that have been proposed are link voltage scaling [10], link on/off technique [14], and Powerherd [5]. But these techniques are mostly for off-chip networks. The cost associated with implementing such techniques on-chip might be too. For the off-line (static) peak power control, there are essentially two schemes: (1) admission control, and (2) communication scaling.

When designing SoCs, admission control cannot be used to reject communication since it is not a viable design option. So the appropriate solution would be to use the communication scaling scheme where the communication would be accepted, but the bandwidth allocated to it may not be equivalent to its demand.

The network model in [2], utilizes virtual channels and in this scenario, the link bandwidth is divided equally amongst them. So by modifying the number of virtual channels allocated to a communication, we can modify the bandwidth allocated to it and thereby its power consumption.

In the peak power constraining phase of the design, the bandwidth allocated to the communications is scaled down in the modes in which there is a peak power violation. The bandwidth allocation is reduced by scaling down the number of virtual channels that are allocated to the communications. Performance degradation in terms of throughput reduction is expected. We provide results to demonstrate the effect of the bandwidth control

4. Experimental setup and results

4.1. *NoCSim* verification test-bed

NoCSim is a network-on-chip system simulator developed using the popular system-level design programming language SystemC [15]. Table 2 provides an overview of the simulator features. The results obtained from the neighborhood detection heuristic are used to setup NoCSim and run simulations. These simulations will provide latency, power and throughput details that will validate the results of the heuristic.

4.2. Results

For our experiments, we used two test cases. For each test case we examine the peak power results for:

- Adhoc mapping solution: This solution is a random mapping of resources in the NoC. It is not the worst possible configuration.
- Expected peak power without bandwidth control: These peak power results are obtained from the neighborhood determination heuristic.
- Actual peak power without bandwidth control: These peak power results are obtained through *NoCSim* simulations.
- Expected peak power with bandwidth control: These peak power results are the expected peak power results due to the effect of the offline peak power control through bandwidth control.
- Actual peak power with bandwidth control: These peak power results are obtained through NoCSim simulations.

We also examine the effect of the peak power control on the throughput of the NoC.

Table 2. NoCSim features

Topology & Routing	2D torus, source routing
Types of traffic sources	Constant bit rate, Random Poisson distribution, Execution trace based
Flow control & buffer management	Wormhole, credit-based virtual channels
Configurable parameters	Number of VCs, Buffer depth of VCs, Network size
Power model	Currently setup for 90nm technology (leakage power not considered)[1]

Table 3. Test Case 1&2 specification summary

Classification	Synthetic test case	Multimedia System (MP3, MPEG, JPEG, ADPCM)
No. of resource	16	4
No. of modes	5	3
Total no. of comm. edges	39	21
Mode::No. of comm.. edges	0::6, 1::9, 2::8, 3::9, 4::7	0::8, 1::7, 2::6
Target Topology	2D torus (4x4)	2D torus (2x)
Network link Properties	128 bit wide, 1Ghz => 128Gbps	128 bit wide, 1Ghz=>128Gbps
Number of VCs/ Bandwidth per VC	4/32Gbps	4/Gbps
Peak Power Budget	1.5W	0.35W

The first test case was a synthetic case. Table 3 summarizes the specifications of the test case. Figure 5 shows the peak power consumption for each of the modes of operation. The adhoc mapping solutions consume a considerable amount of power (upto 2x). The neighborhood determination heuristic inherently constrains the amount of power consumed. Through bandwidth control we were able to reduce the peak power consumption to the 1.5W limit. Figure 6 shows the consequent degradation in throughput. An example of how performance degrades with peak power control can be seen in the case of mode 3 for test case 1. For a 40% reduction in peak power, there is a 40% reduction in throughput.

The second test case was a conglomeration of MP3 encoder-decoder, MPEG encoder, JPEG compression and ADPCM encoder-decoder. The system was configured to operate in three modes and the traffic loads have been scaled up to get results of interest. Table 3 summarizes the test case specification of this test case. Figure 7 illustrates the peak power consumption in the different modes. In mode 1, we observe that the peak power consumption for the adhoc solution is actually lower than that expected by the heuristic. But as mentioned earlier, this adhoc solution

[1] With migration towards smaller process technologies (90nm and below), *leakage power* becomes an important (constant) component towards total system power.

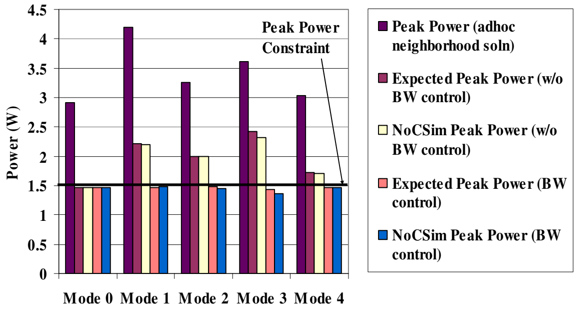


Figure 5. Test Case 1: Peak Power results for different NoC configurations (Peak Power constraint = 1.5W)

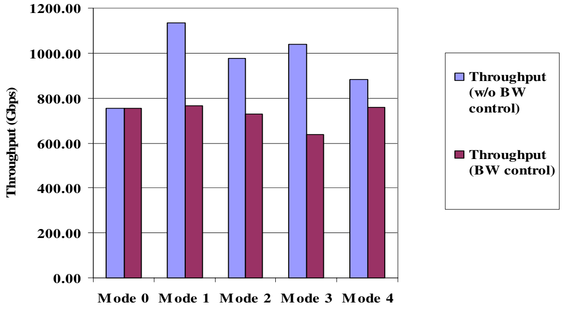


Figure 6. Test Case 1: Effect of Peak Power control on NoC throughput

was not setup to be the worst case scenario, hence it turned out to be a suitable configuration for mode 1 (but not for mode 0 and 2, else the heuristic would have determined it). The latency degradation for peak power control for test case 2 is shown in Figure 8.

5. Conclusions and future work

This paper presents a heuristic that aids in designing NoC based multimode systems. By not providing actual placements of the communicating cores, we provide the system designer with that degree of freedom and by controlling the bandwidth allocated to the communication in the on-chip network, we control the peak power consumption. The consequential tradeoff was the decrease in throughput and increased latency. We are currently developing a low cost, online power management strategy that would improve the throughput and reduce the latency.

6. References

[1] P.Guerrier and A.Greiner, "A generic architecture for on-chip packet-switched interconnections", Proc. IEEE DATE, 2000.

[2] W.J.Dally and B.Towles, "Route packets, not wires: On chip interconnection networks", Proc. ACM DAC, 2001.

[3] J.Hu and R.Marculescu, "Exploiting the routing flexibility for energy/performance aware mapping of regular NoC architectures", Proc. IEEE DATE, 2003.

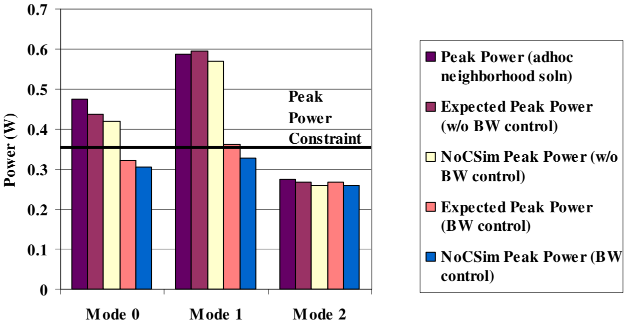


Figure 7. Test Case 2: Peak Power results for different NoC configurations (Peak power constraint = 0.35W)

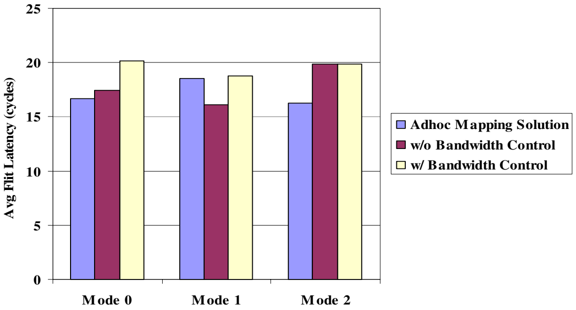


Figure 8. Test Case 2: Average Flit Latency

[4] V.Raghunathan, M.B.Srivastava and R.K.Gupta, "A survey of techniques for energy efficient on-chip communication", Proc. ACM/IEEE DAC, 2003.

[5] S.S.Mukherjee, et.al., "The Alpha 21364 network architecture", Proc. IEEE Micro, volume 22, issue 1, January/February 2002.

[6] H.-S.Wang, L.-S.Peh and S.Malik, "Power-driven design of router microarchitectures in on-chip network", Proc. Intl. Symp. Microarchitecture, November 2003, to appear.

[7] L.Shang, L.-S.Peh and N.K.Jha, "PowerHerd: dynamically satisfying peak power constraints in interconnection networks", Proc. Intl. Sym. Supercomputing (ICS), June 2003.

[8] T.Ye, L.Benini and G.De Micheli, "Analysis of power consumption on switch fabrics in network routers", Proc. ACM/IEEE DAC, 2002.

[9] S.Kumar, et al., "A Network on Chip architecture and design methodology", Proc. IEEE VLSI, April 2002.

[10] L.Shang, L.-S.Peh and N.K.Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks", Proc. HPCA, January 2003

[11] L.-T.Yeh and R.C.Chu, Thermal Management of Microelectronic Equipment: Heat Transfer Theory, Analysis Methods, and Design Practices. ASME Press, New York, NY, 2002.

[12] N.Swaminathan, P.Bhojwani and R.Mahapatra, “Communication architecture synthesis of packet-switched Network-on-Chip”, Technical Report # TR-CS-2002-08-0.

[13] IBM 90nm workbook.

[14] E.J.Kim, et.al., “Energy optimization techniques in cluster interconnects”, Proc. ISPLED 2003.

[15] "SystemC v.2.0.1 White Paper", www.systemc.org