# Packetized On-Chip Interconnect Communication Analysis for MPSoC

Terry Tao Ye
Computer Systems Lab
Stanford University
Stanford, CA 94305
taoye@stanford.edu

Luca Benini
DEIS
University of Bologna
Bologna, Italy
lbenini@deis.unibo.it

Giovanni De Micheli
Computer Systems Lab
Stanford University
Stanford, CA 94305
nanni@stanford.edu

## ABSTRACT

*Interconnect networks play a critical role in shared memory multi-processor systems-on-chip (MPSoC) designs. MPSoC performance and power consumption are greatly affected by the packet dataflows that are transported on the network. In this paper, by introducing a packetized on-chip communication power model, we discuss the packetization impact on MPSoC performance and power consumption. Particularly, we propose a quantitative analysis method to evaluate the relationship between different design options (cache, memory, packetization scheme, etc.) at the architectural level. From the benchmark experiments, we show that optimal performance and power tradeoff can be achieved by the selection of appropriate packet sizes.*

## 1. Introduction

Shared memory *multi-processor systems-on-chips* (MPSoCs) have been widely used in today's high performance embedded systems, such as *network processors* and *parallel media processors* (PMP). They combine the advantages of data processing parallelism of *multi-processors* (MP) and the high level integration of *systems-on-chip* (SoC). The MPSoC performance is not only determined by the capacity of the node processors (e.g. CPU speed, cache size, etc.), but it is also limited by the interconnect network that connects the processors and memories. Design and optimization of such interconnect network are critical for MPSoC performance.

MPSoC uses shared memories to exchange data between processors, and the exchanged data are transported from one processor to the other through the interconnect network. Dataflows are first packetized and then routed to their destinations. Packetized on-chip communication has many advantages over the ad-hoc wire routing in ASICs [1]. The signal integrity and interference are more controllable, multiple high-bandwidth parallel dataflows can be supported concurrently, and the systems are more modularized for IP reuse.

Performance and power consumption are the two most critical issues in MPSoC interconnect network design. On one hand, applications running on MPSoC platforms demand faster and more reliable on-chip communication, on the other hand, as the VLSI technology is moving quickly into the nano-meter domain, the energy dissipated by the network becomes a more and more significant contributor of total system energy consumption. Balancing performance and power will become a major design challenge in future MPSoC network design [3].

The dataflow traffic on MPSoC interconnect network comes from the processor-processor and processor-memory transactions. Therefore, the performance and power consumption of on-chip communication are not only determined by the physical aspects of the network (e.g. voltage swing, the wire delay and fan-out load capacitance, etc.), but are also dependent on the interactions between the processor nodes. Particularly, on-chip network traffic is coming from the following sources:

1. **Cache and memory transactions.** Every cache miss needs to fetch data from the shared memories, and consequently creates traffic on the interconnect.

2. **Cache coherence operations.** In MPSoC, one data may have multiple copies in the caches of different node processors. When the data in memory is updated, its cache copies also need to be updated or invalidated. This synchronization operation will create traffic on the interconnect as well.

3. **Packet segmentation overheads.** When dataflows are segmented into packets, traffic on the interconnect will carry additional overhead. The overhead is dependent on the packet size and header/tail size.

4. **Contentions between packets.** When there is contention between packets on the interconnect, the packets need to be re-routed to another datapath or buffered temporarily. This effect will again change the traffic pattern on the interconnect.

The above factors are not independent. Instead, the performance and power trade-off is determined by the interactions of all factors dynamically, and the variation of one factor will impact other factors. For example, the changes of packet size will affect the cache block size that can be updated during each memory access, and consequently change the cache miss rate.

While the MPSoC performance issues have been addressed by many researchers in the parallel computing field [4], the power consumption for on-chip network communication has not been quantitatively analyzed. Previous researches either use statistical traffic model, or calculate the power consumption in analytical methods [6][7][8]. Those researches did not address the packetization impact on the network power consumption, and they did not specify how on-chip network designers need to trade-off between different options in CPU, cache and memory designs at the architectural level.

In this paper, we introduce a MPSoC interconnect communication energy model and apply this model on RSIM, a multi-processor simulator [5]. We will analyze quantitatively the relationship between different packetization factors, and their impact on the power consumption as well as system performance. Furthermore, based on the analysis, we will show the trade-offs between MPSoC performance and its interconnect power consumption.

The paper is organized as follows: Section 2 describes briefly the basic architecture of MPSoC platform. Section 3 analyzes the sources and composition of the traffic on the MPSoC network. Section 4 lists the sources that contribute to MPSoC total power consumption. Section 5 gives the energy models that will be used on MPSoC network analysis. Section 6 explains the simulation platform and the experiments performed. Based on these experiments, Section 7 analyzes MPSoC performance trade-offs under different packetization schemes, followed by Section 8 on the power consumptions issues. As a summary, we will discuss how designers can trade-off performance with power in MPSoC network design in Section 9.

# 2. MPSoC Architecture

A typical MPSoC architecture is shown in Figure 1. It consists of several RISC processors connected by an interconnect network. Each processor is denoted as a "node processor" or simply referred as a "node".

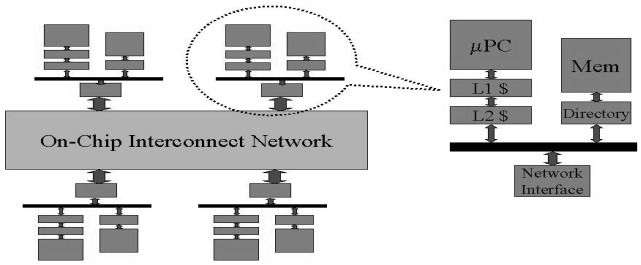


**Figure 1: MPSoC Architecture**

Each node has its own CPU/FPU and cache hierarchy (one level or two levels of cache). A read miss in L1 cache will create a L2 cache access, and a miss in L2 cache will then need a memory access. Both L1 and L2 may use write-through or write-back for cache updates.

The MPSoC uses shared memories, the memories are associated with each node, but they can be physically placed into one big memory block. The memories are globally addressed and accessible by the memory directory. When there is a miss in L2 cache, L2 cache will send a request packet across the network asking for memory access. The memory with the requested address will return a reply packet containing the data to the requesting node. When there is a cache write-through or write-back operation, the cache block that needs to be updated is encapsulated in a packet, and sent to the destination node where the corresponding memory resides.

*Cache coherence* is a very critical issue in MPSoC. Because one data may have several copies in different node caches, when the data in memory is updated, the stale data stored in each cache needs to be updated. There are two methods to solve the cache coherence problem: 1) *cache update* updates all copies in cache when the data in memory is updated; 2) *cache invalidate* invalidates all copies in cache. When next time the data is read, the read will become a miss and consequently need to fetch the updated data from the corresponding memory.

The interconnect network normally uses a mesh or torus topology [1]. The network performs point to point packet-based routing. The packets are routed in a wormhole fashion. Each node can arbitrate and distribute the packets to their destinations independently.

# 3. Packet Dataflow on Interconnect Network

Before we start to discuss the performance and power issues of MPSoC interconnect network, we need to first study the traffic on the network, particularly, we should analyze the composition of the packetized dataflows that are exchanged between MPSoC nodes.

Packets transported on the MPSoC network consist of three parts. The header contains the destination address, the source address, and the requested operation type (READ, WRITE, INVALIDATE, etc). The payload contains the transported data. The tail contains the error checking or correction code.

## 3.1 Types of Packets

From the previous sections, we know that the packets traveling on the network come from different sources, and they can be categorized into the following types:

**Memory access request packet.** The packet is induced by L2 cache miss that requests data fetch from memories. The header of these packets contains the destination address of the target memory (node ID and memory address) as well as the type of memory operation requested (memory READ, for example). Because there is no data being transported, the payload is empty.

**Cache coherence synchronization packet.** The packet is induced by the cache coherence operation from the memory. This type of packet comes from the updated memory, and it is sent to all caches that have a copy of the updated data. The packet header contains the memory tag and block address of the data. If the synchronization uses "update" method, the packet contains updated data as payload. If the synchronization uses "invalidate" method, the packet header contains the operation type (INVALIDATE, in this case), and the payload is empty.

**Data fetch packet.** It is the reply packet from memory, containing the requested data. The packet header contains the target address (the node ID of the cache requesting for the data). The data is contained in the packet payload.

**Data update packet**. This packet contains the data that will be written back to the memory. It comes from L2 cache that requests the memory write operation. The header of the packet contains the destination memory address, and the payload contains the data.

**IO and interrupt packet.** This packet is used by IO operations or interrupt operations. The header contains the destination address or node ID. If data exchange is involved, the payload contains the data.

## 3.2 Packet and Flit

In order to reduce the latency of packet transportation on the network, MPSoCs use wormhole routing scheme that can route the packets from one intermediate switch to the other (Fig. 2). In wormhole routing, packets are not traveling as a single unit, instead, packets are further segmented into *flits* (flow control unit). By segmenting packets into flits, one packet can occupy several intermediate node switches. Wormhole routing reduces the store-and-forward latencies of packet transportation [10].

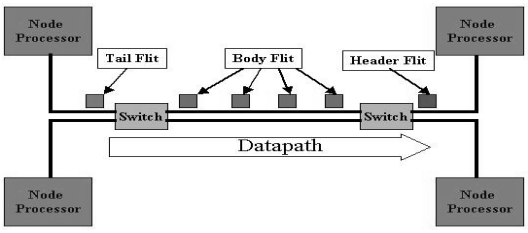


**Figure 2: Flit and Wormhole Routing of Packet**

When header flit arrives at a switch, if the destination port is available, the header flit will reserve the port. The consequent flits will flow through the reserved port, until the tail flit, which will release the destination port (Fig. 2).

## 3.3 Packet Size

From the above analysis, we can see most packets travel between memories and caches, except those packets involved in I/O and interrupt operations. Although packets of different types originate from different sources, the length of the packets is determined by the size of the payload. In reality, there are two differently sized packets on the MPSoC network, *short packet* and *long packet*, as described below.

*Short Packets* are the packets with no payloads, such as the memory access request packets and cache coherence packets (invalidate approach). These packets consist only header and tail. They have the length of 2 flits

*Long Packets* are the packets with payloads, such as the data fetch packets, the data update packets and the cache coherence packets used in update approach. These packets travel between caches and memories. The data contained in the payload are either from cache block, or they are sent back to the node cache to update the cache block. Normally, the payload size equals the cache block size, as shown in Fig. 3.

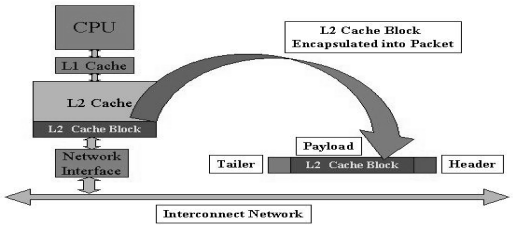


**Figure 3: Packet Size and Cache Block Size**

Packets with payload size different than the cache block size will increase cache miss penalty. The reasons are two: 1) If each cache block is segmented into different packets, it is not guaranteed that all packets will arrive at the same time, and consequently the cache block cannot be updated at the same time. 2) If several cache blocks are to be packed into one packet payload, the packet needs to hold its transmission until all the cache blocks are updated. This will again increase the cache miss delay penalty.

In our analysis, we assume all the long packets contain the payload of one cache block size. Therefore, the length of the long packets will determine the cache block size of each node processor.

# 4. MPSoC Power Consumption

The MPSoC power consumption originates from three sources: the node power consumption, the shared memory power consumption and the interconnect network power consumption.

## 4.1 Node power consumption

Node power consumption comes from the operations inside each node processor, these operations include:

**1. CPU and FPU operations.** Instructions like *ADD, MOV, SUB etc* consume power because these operations toggle the logic gates on the datapath of processor.

**2. L1 cache access.** L1 cache is built with fast SRAMs. When data is loaded or stored in the L1 cache, it consumes power.

**3. L2 cache access.** L2 cache is built with slower but larger SRAMs. Whenever there is a read miss in L1 cache, or when there is write back from L1 cache, L2 cache is accessed, and consequently consumes power.

## 4.2 Shared memory power consumption

Data miss in L2 cache requires data to be fetched from memory. Data write back from L2 cache also needs to update the memory. Both operations will dissipate power when accessing the memories.

## 4.3 Interconnect network power consumption

Operations like cache miss, data fetch, memory updates and cache synchronization all need to send packets on the interconnect network. When packets are transported on the network, energy is dissipated on the interconnect wires as well as the logic gates inside each switch. Both wires and logic gates need to be counted when we estimate the network power consumption.

Among the above three sources, the node power consumption and memory power consumption have been studied by many researches. In the following sections, we will only focus our analysis on the power consumption of interconnect networks. Later in this paper, when we compare the network power consumption with the total MPSoC power consumption, we will reference the results from other researches for node processor and memory power estimation.

# 5. Network Energy Modeling

## 5.1 Bit Energy of Packet

When a packet travels on the interconnect network, both the wires and logic gates on the datapath will toggle as the bit-stream flips its polarity. In this paper, we use an approach similar to the one presented in [12] to estimate the energy consumption for the packets traveling on the network.

We adopt the concept of bit energy $E_{bit}$ to estimate the energy consumed for each bit when the bit flips its polarity from previous bit in the bit stream. We further decompose the bit energy $E_{bit}$ into bit energy consumed on the interconnect wires $E_{W_{bit}}$ and the bit energy consumed on the logic gates inside the node switch $E_{S_{bit}}$.

The bit energy consumed on the interconnect wire can be estimated from the total load capacitance on the interconnect. The total load capacitance can be calculated from Thompson model, as described in [12].

The bit energy consumed on the switch logic gates can be estimated from Synopsys Power Compiler simulation. Without loss of generality, we use random bit-stream as the packet payload content. Details of the estimation can also be found in [12].

## 5.2 Packets and Hops

When the source node and destination node are not placed adjacent to each other on the network, a packet needs to travel several intermediate nodes until reaching the destination. We call each of the intermediate stages a *hop* (Fig. 4).

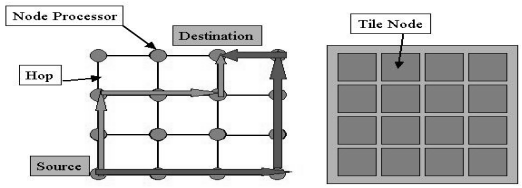


**Figure 4: Hops and Alternate Routes of Packets**

In the mesh or torus network, there are several different alternate datapaths between source and destination, as shown in Fig. 4. When contention occurs between packets, the packets may be rerouted to different datapaths. Therefore, packet datapath will vary dynamically according to the traffic condition. Packets with the same source and destination may not travel through the same number of hops, and they may not necessarily travel on the datapath with the minimum number of hops.

The number of hops a packet travels greatly affects the total energy consumption needed to transport the packet from source to destination. For every hop a packet travels, the interconnect wires between the nodes will be charged and discharged as the bit-stream flows by, and the logic gates inside the node switch will toggle.

We assume a tiled floorplan implementation for MPSoC, similar to those proposed by [1] and [2], as shown in Fig. 4. Each node processor is placed inside a tile, and the mesh network is routed in a regular topology. Without loss of generality, we can assume all the hops in mesh network have the same interconnect length. Therefore, if we pre-calculate the energy consumed by one packet on one hop, $E_{hop}$, by counting the number of hops a packet travels, we can estimate the total energy consumed by that packet.

We use the hop histogram to show the total energy consumption by the packet traffic. In Fig. 5 below, histograms of the packets traveling on an 8-processor SoC are shown. The 8 processors are connect by a 2-dimensional mesh interconnect network. The histograms are extracted from the trace file of a *quicksort* benchmark. The histogram has $n$ bins with *1, 2, .., n* hops, the bar on each bin shows the number of packets in each bin. We count long packets and short packets separately in the histograms.

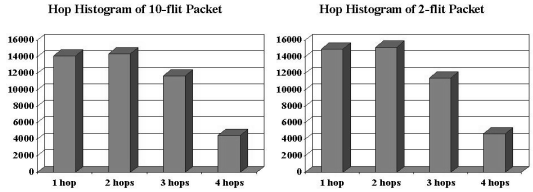


**Figure 5: Hop Histogram of Long and Short Packets**

Without loss of generality, we can assume packets of the same length will consume the same energy per hop. Using the hop histogram of the packets, we can calculate the total network energy consumption with the following equation (Eq. 1).

$$E_{packet} = \sum_{h=1}^{maxhops} h \times N(h)_{packet} \times L_{long} \times E_{flit} \quad (1)$$

$$+ \sum_{h=1}^{maxhops} h \times N(h)_{packet} \times L_{short} \times E_{flit} \quad (2)$$

where $N(h)_{packet}$ is the number of packets with $h$ number of hops in the histogram. $L_{long}$ and $L_{short}$ are the lengths of long and short packets respectively, in the unit of flit. $E_{flit}$ is the energy consumption for one flit on each hop. Because the packets are actually segmented into flits when they are transported on the network, we only need to calculate the energy consumption for one flit, $E_{flit}$. The energy of one packet per hop $E_{hop}$ can be calculated by multiplying the number of flits the packet contains.

# 6. Experiments

## 6.1 Platform

We use RSIM as our shared memory MPSoC simulation platform. Eight RISC processors are built into RSIM, they are connected by a 2-dimensional mesh interconnect network. The interconnect is 64-bit in width. Each node processor contains two levels of cache hierarchy. L1 cache is 16K bytes, and L2 cache is 64K bytes. Both L1 and L2 cache use write-through methods for memory updates. We use the *invalidate* approach for cache coherence synchronization. Wormhole routing is used, and the flit size is 8 bytes.

## 6.2 Energy Model

As discussed in Section 5, to calculate the power consumption on the network, we need to calculate the value of $E_{flit}$, which is the energy consumed by one flit traveling on one hop. We assume each tile of node processor is $2mm \times 2mm$ in dimension, and they are placed regularly on the floorplan, as shown in Fig. 4. We assume $0.13\mu m$ technology is used, and the wire load capacitance is 0.50fF per micron. Under these assumption, the energy consumed by one flit on one hop interconnect is 0.174nJ.

The energy consumed on the switch logic gates of one hop is calculated from Synopsys Power Compiler. We calculate the bit energy on the logic gates in a way similar to that used in [12]. We use $0.13\mu m$ standard cell library, and the energy consumed by one flit on one hop switch is 0.096nJ. Based on these calculation, the flit energy per hop $E_{flit} = 0.27nJ$.

## 6.3 Experiments and Benchmarks

We tested five applications on our RSIM MPSoC simulation platform, they are *sor*, *water*, *quicksort*, *lu* and *mp3d*. These applications are ported from the Stanford SPLASH project. To analyze how different packetization schemes will affect the performance and power, we change the dataflow with different packet sizes. The packet payload sizes are varied from 16Byte, 32Byte, 64Byte, 128Byte to 256Byte. Because the short packets are always 2-flit in length, therefore, the change of packet size is applied to long packets only. The results are discussed quantitatively in the following sections.

# 7. Packetization and MPSoC Performance

As we mentioned in Section 1, MPSoC performance is determined by many factors. Different packetization schemes have different impacts on these factors, and consequently, result in different performance metrics.

## 7.1 Cache Miss Rate

Changing the packet payload size (for long packets) will change the L2 cache block size that can be updated in one memory fetch. If we choose larger payload size, more cache contents can be updated. Therefore, the cache miss rate will decrease. This effect can be observed from Fig. 6. As the packet payload size increases, both the L1 cache (Fig. 6a) and L2 cache (Fig. 6b) miss rates decrease. Decreased cache miss rate will reduce the number of packets needed for memory access.

## 7.2 Cache Miss Penalty

Whenever there is a L2 cache miss, the missed cache block needs to be fetched from the memories. The latency associated with this fetch operation is called a miss penalty. When we estimate the cache miss penalty, we need to count all the delays occurred within the fetch operation. These delays include: 1) packetization delay, 2) interconnect delay, 3) store and forward delay on each hop for one flit, 4) arbitration delay, 5) memory access delay and 6) contention delay. Among these six factors, 2), 3) and 4) will not change significantly for packets with different sizes, because we use wormhole routing. However, delays on 1) and 5) will become longer because larger packets need longer time for packetization and memory access. Longer packets will actually cause more contention delay. This is because when wormhole routing is used, longer packet will hold more intermediate nodes during its transmission. Other packets have to wait in the buffer, or choose alternative datapaths, which are not necessarily the short routes. Combining all these factors,

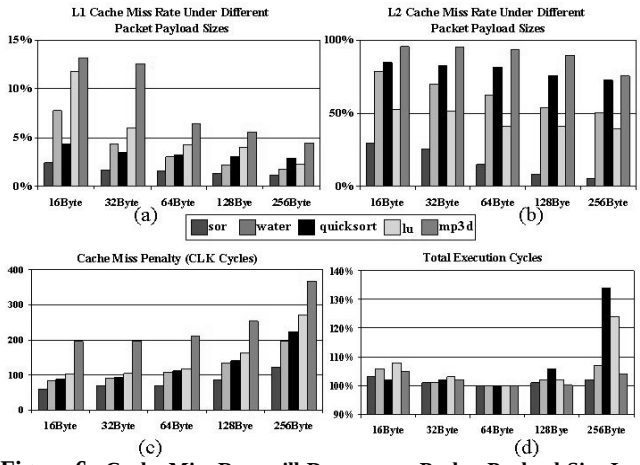


**Figure 6: Cache Miss Rate will Decrease as Packet Payload Size Increases**

the overall cache penalty will increase as the packet payload size increases, as shown from Fig. 6c.

## 7.3 Overall Performance

From the above analysis, we know that although larger payload size helps to decrease the cache miss rate, it will increase the cache miss latency. Combining these two factors, there exists an optimal payload size that can achieve the minimum execution time, as seen from Fig. 6d. In order to illustrate the variation of performance, we normalized the figure to the minimum execution time of each benchmark. In our experiments, all the five benchmarks achieve the best performance with 64 bytes of payload size.

# 8. Packetization and Power Consumption

Eq. 1 in Section 5 shows that the power consumption of packetized dataflow on MPSoC network is determined by the following three factors: 1) the number of packets on network, 2) the energy consumed by each packet on one hop, and 3) the number of hops each packet travels. Different packetization schemes will have different impact on these factors, and consequently affect the network power consumption. We summarize these effects and list them below.

1. Packets with larger payload size will decrease the cache miss rate and consequently decrease the number of packets on the network. This effect can be seen from Fig. 7a. It shows the average number of packets on the network (traffic density) at one clock cycle. As the packet size increases, the number of packets decreases accordingly. Actually, with the same packet size, the traffic density of different benchmarks is consistent with the miss penalty. By comparing Fig. 7a with Fig. 6c, we see that if the packet length stays the same, higher traffic density causes longer miss latency.

2. Larger packet size will increase the energy consumed per packet, because there are more bits in the payload.

3. As discussed in Section 7, larger packets will occupy the intermediate node switches for a longer time, and cause other packets to be re-routed to non-shortest datapaths. This leads to more contention that will increase the total number of hops needed for packets traveling from source to destination. This effect is shown in Fig.7b. It shows the average number of hops a packet travels between source and destination. As packet size (payload size) increases, more hops are needed to transport the packets.

Actually, increasing the cache block size will not decrease the

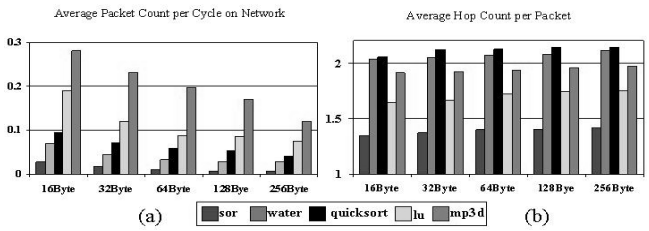


**Figure 7: Contention Occurrence Changes as Packet Payload Size Increases**

cache miss rate proportionally [11]. Therefore, the decrease of packet count cannot compensate the increase of energy consumed per packet caused by the increase of packet length. Larger packet size also increases the hop counts on the datapath. Fig. 9a shows the combined effects of these factors under different packet sizes. The values are normalized to the measurement of 16Byte. As packet size increases, energy consumption on the interconnect network will increase.

Although increase of packet size will increase the energy dissipated on the network, it will decrease the energy consumption on cache and memory. Because larger packet sizes will decrease the cache miss rate, both cache energy consumption and memory energy consumption will be reduced. This relationship can be seen from Fig. 8. It shows the energy consumption on cache and memory under different packet sizes respectively. The access energy of each cache and memory instruction is estimated based on the work from [13] and [14]. The energy in the figure is normalized to the value of 256Byte, which achieves the minimum energy consumption.

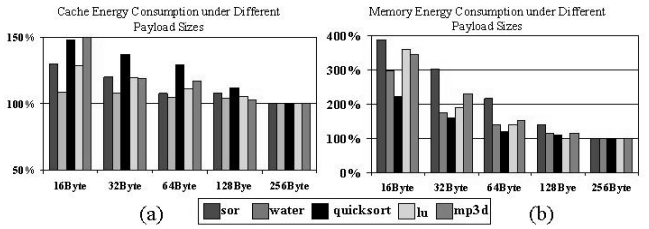


**Figure 8: Cache and Memory Energy Decrease as Packet Payload Size Increases**

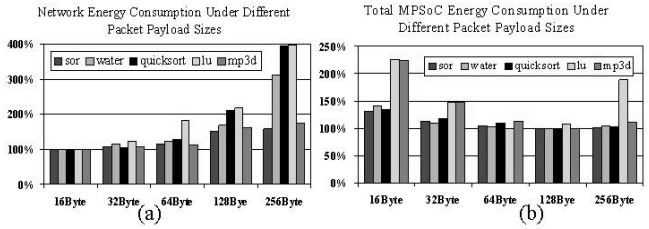


**Figure 9: Network and Total MPSoC Energy Consumption under Different Packet Payload Sizes**

The total energy dissipated on MPSoC comes from non-cache instructions (instructions that do not involve cache access) of each node processors, caches, shared memories as well as the interconnect network. In order to see the packetization impact on the total system energy consumption, we put all MPSoC energy contributors together and see how the energy changes under different packet sizes. The results are shown in Fig. 9b. From this figure, we can see the overall MPSoC energy will decrease as packets size increases. However, when the packets are too large, as in the case of 256Byte in the figure, the total MPSoC energy will increase. This is because

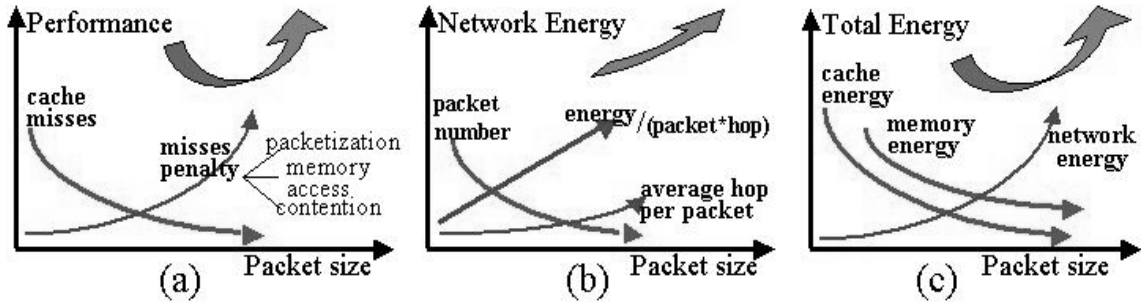


**Figure 10: Qualitative Analysis of Packet Size Impact**

when the packet is too large, the increase of interconnect network energy will outgrow the decrease of energy on cache, memory and non-cache instructions. In our simulation, the non-cache instruction energy consumption is estimated based on the techniques presented in [15], and it does not change significantly under different packet sizes.

## 9. Discussion

Although the specific measurement values in the experiments are technology and platform dependent, we believe the analysis will hold for different MPSoC implementations. We summarize our analysis qualitatively as follows (Fig. 10).

Large packet size decreases the cache miss rates of MPSoC but increases the miss penalty. The increase of miss penalty is caused by the increase of packetization delay, memory access delay, as well as contention delay on the network. As shown qualitatively in Fig. 10a, the cache miss rate saturates with the increase of packet size. Nevertheless, the miss penalty increases faster than linearly. Therefore, there exists an optimal packet size to achieve best performance.

The energy spent on the interconnect network increases as the packet size increases. Three factors play roles in this case (Fig. 10b). 1) Longer packets, i.e. larger cache lines, reduce the cache miss rate, hence reduce the packet count. Nevertheless, the packet count does not fall linearly with the increase of packet size. 2) The energy consumption per $packet \times hop$ increases in a linear fashion with the increase of packet length. If we ignore the overhead of packet header and tail, this increase is proportional to packet size. 3) The average number of hops per packet on the network also increases with the packet length. The combined effect causes the network energy to increase as the packet size increases.

The total MPSoC system energy is dominated by the sum of three factors as the packet size increases (Fig. 10c). 1) Cache energy will decrease. 2) Memory energy will decrease as well. 3) Network energy will increase over-linearly. In our benchmarks, the non-cache instruction energy does not change significantly. The overall trend depends on the breakdown among the three factors. Our experiments show that there exists a packet size that minimizes the overall energy consumption. Moreover, if the network energy contributes a major part of the total system energy consumption, which is expected to happen as VLSI technology moves to nanometer domain, the MPSoC energy will eventually increase with the packet size.

## 10. Conclusion

In this paper, we introduced a MPSoC interconnect network energy model and quantified the effect of packet size variation on performance and energy consumption. The analysis presented in this paper will help MPSoC designers to select the appropriate architectures and communication schemes in their system level design. Future research needs to address other aspects of packetized on-chip communication such as routing algorithms and re-transmission protocols.

## 11. Acknowledgment

The authors would like to acknowledge the support from MARCO/GSRC for this research.

## 12. REFERENCES

[1] Dally, William; Toles, Brian "Route Packets, Not Wires: On-Chip Interconnection Networks" *38th Design Automation Conference, 2001. Proceedings*

[2] Kumar, S. et. al "A network on chip architecture and design methodology" *VLSI on Annual Symposium, IEEE Computer Society ISVLSI 2002*

[3] Benini, Luca; De Micheli, Giovanni "Networks on Chips: A New Paradigm for System on Chip Design" *DATE Conference, 2002. Proceedings*

[4] Culler, D.E.; Singh, J.P.; Gupta, A.; "Parallel Computer Architecture: A Hardware/Software Approach" *Morgan Kaufmann Publishers* 1998

[5] Hughes, C.J.; Pai, V.S.; Ranganathan, P.; Adve, S.V. "Rsim: simulating shared-memory multiprocessors with ILP processors" *Computer , Volume: 35 Issue: 2 , Feb. 2002*

[6] Wassal, A.G.; Hasan, M.A. "Low-power system-level design of VLSI packet switching fabrics" *CAD of Integrated Circuits and Systems, IEEE Transactions on* , June 2001, Page(s): 723 -738

[7] C. Patel, S. Chai, S. Yalamanchili, D. Shimmel, "Power constrained design of multiprocessor interconnection networks," *IEEE International Conference on Computer Design*, pp. 408-416, 1997.

[8] Langen, D.; Brinkmann, A.; Ruckert, U. "High level estimation of the area and power consumption of on-chip interconnects," *13th IEEE International ASIC/SOC Conference, 2000. Proceedings.*.

[9] C. D. Thompson, "A Complexity Theory for VLSI, PhD thesis", *Carnegie-Mellon University*, August 1980

[10] Duato, J.; Yalamanchili, S.; Ni, L. "Interconnection Networks, an Engineering Approach" *IEEE Computer Society Press*, 1997

[11] Patterson, D.A.; Hennessy, J. "Computer Organization and Design, The Hardware/Software Interface" *Morgan Kaufmann Publishers*, 1998

[12] Ye, T.T.; Benini, L.; De Micheli, G. "Analysis of power consumption on switch fabrics in network routers" *Design Automation Conference, 2002. Proceedings. 39th , 2002*

[13] E. Geethanjali, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, "Memory System Energy: Influence of Hardware-Software Optimizations", Int'l Sym on Low Power Design and Electronics, July 2000.

[14] Wen-Tsong Shiue; Chakrabarti, C. "Memory exploration for low power, embedded systems", 36th Design Automation Conference, 1999. Proceedings.

[15] Zaccaria Vittorio, et. al "Energy Estimation and Optimization of Embedded VLIW Processors Based on Instruction Clustering" *39th DAC Proceeding*