

System Level Interconnect Design for Network-on-Chip Using Interconnect IPs

Jian Liu
jianliu@imit.kth.se
+46 8 790 41 97

Meigen Shen
mgshen@imit.kth.se
+46 8 790 41 06

Li-Rong Zheng
lrzheng@imit.kth.se
+46 8 790 41 04

Hannu Tenhunen
hannu@imit.kth.se
+46 8 790 41 19

Laboratory of Electronics and Computer Systems (LECS)
Royal Institute of Technology (KTH)
Electrum 229, SE-164-40 Kista, Stockholm, Sweden

ABSTRACT

As technology scales down, the interconnect for on-chip global communication becomes the delay bottleneck. In order to provide well-controlled global wire delay and efficient global communication, a Network-on-Chip (NoC) architecture was proposed by different authors [1][5][6]. NoC uses Interconnect Intellectual Property (IIP) to connect different resources. In a bottom up approach, this paper first studies the NoC system parameters constrained by the interconnections. Predictions on scaled system parameters such as clock frequency, resource size, global communication bandwidth and inter-resource delay are made for future technologies. Based on these parameters, a global wire planning scheme is proposed. At last, the main IIP modules are described and one possible transmission scheme is demonstrated and simulated.

Categories and Subject Descriptors

B.7.1 [Integrated Circuits]: Types and Design Styles – *advanced technologies, VLSI (very large scale integration)*.

General Terms

Performance, Design, Reliability, Theory.

Keywords

Network on chip, interconnect, Interconnect IP, bandwidth optimization.

1. INTRODUCTION

Interconnect has been the major design constraint in deep sub-micron (DSM) circuits. The downscaled wire size, increased aspect ratio, combined with higher signal speed cause many signal integrity challenges and time closure problems. Traditionally, these issues are tackled mainly from an electrical design point of view. Recent studies show that the problem also

can be coped with interconnect-centric system architectures [1][5][6]. One such emerging architecture is the Network-on-Chip (NoC). The NoC architecture is a data packet based communication network on a single chip. It scales from a few dozens to several hundreds resources. A resource may be a processor core, a DSP core, an FPGA block, or any other intellectual property (IP) block. The resources are connected by Interconnection IPs. The structured network wiring gives well-controlled electrical parameters and enables reusing of building blocks. Clearly, any topology that fully connects the resources can be used for the network. However, a two-dimensional mesh topology turns out to be simple and effective [1][9]. Thus, the following study will be based on this specific topology.

The NoC uses IIPs to provide a reliable and efficient communication platform for user-specified resources. Conceptually, the NoC resources are connected by IIPs to form a two-dimensional mesh as shown in Figure 1. Each IIP is connected to its four closest neighbors and to its corresponding resource. The data from one resource is first passed to the IIP attached to the resource. The IIP then packets the data and routes the data packets onto the appropriate link, see also paragraph 4 for more detail.

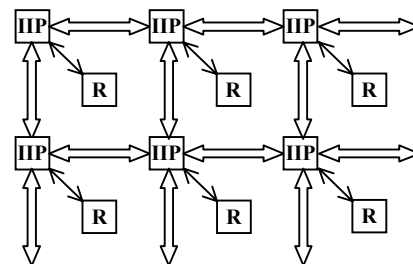


Figure 1. The 2D-mesh NoC backbone with Resources (R) and Interconnect IPs (IIP).

At a high level, the NoC architecture and IIPs must provide transparent and efficient inter-resource communication. In paragraph 4, the different layers in NoC, the main IIP modules and one possible transmission scheme are described and simulated. As the NoC is targeted to future deep submicron (DSM) and nanometer technologies, the following questions related to physical constraints are also interesting: what is the appropriate size of each synchronous resource; how many resources can be integrated in one chip in future technologies;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SLIP'03, April 5–6, 2003, Monterey, California, USA.

Copyright 2003 ACM 1-58113-627-7/03/0004...\$5.00.

how fast can signals travel from one resource to another through the on-chip micro-communication network and how to get an optimal data bandwidth with limited wire resource. In paragraph 2, we use empirical rules to derive the gate delays for future DSM technologies, which is followed by an estimation of the maximum clock frequency and the corresponding resource size. In paragraph 3, the inter-resource delay is studied, expressions for maximum inter-resource bandwidth are derived and a global wire planning scheme providing maximum bandwidth is proposed.

The NoC is a typical interconnect-centric architecture, which means that the wire planning is the first design step. In this early planning stage, detailed system parameters for the wires are often unknown, making it impractical to consider layout-related properties such as 3D multiplayer interconnections. Therefore, a simpler wire model is used below. When the planning is done and various requirements on the wires, such as delay and noise level, are determined, a dynamic interconnect model can be used to generate a wire structure meeting these requirements in later design phases. One dynamic interconnect model using 3D capacitance, resistance and inductance is described in [13]. Similar CAD tools like Magma's FixedTiming [www.magma.com] are also emerging commercially.

2. GLOBAL WIRE PLANNING FOR NOC

The performance of interconnections is a major concern in scaled technologies. Under scaling, the gate delay decreases. However, the global wires do not scale in length since they communicate signals across the chip. For these wires, the delay per unit length can be kept constant if optimal repeaters are used [4]. In the following study, we assume that global wires are reserved for global communications and semi-global wires/local wires are used within a resource. To estimate the size of each resource, we first find the typical gate delay, which determines the maximum clock rate using an empirical approach. The maximum size of the resource can then be estimated under assumption that in a synchronous resource, a signal must travel from one corner of the resource to the opposite within one clock cycle.

2.1 Technology Scaling and Gate Delay

Since four is the typical average gate connectivity, "fan-out-of-four inverter delay", or simply FO4 is a reasonable parameter to be used for measuring gate delays. As the name suggests, an FO4 is the delay through an inverter driving four identical copies. In a 0.18- μm technology, an FO4 is about 90 ps under worst-case environmental conditions (high temperature and low Vdd). Ron Ho [6] pointed out that, historically, gates have scaled linearly with technology, and an accurate model of recent FO4 delays has been $360 \cdot L_{gate}$ ps at typical and $500 \cdot L_{gate}$ ps under worst-case environmental conditions. After studying today's existing nanometer scale devices, he also predicts that this trend will continue for future generations of transistors, which means $500 \cdot L_{gate}$ ps is a lower limit for future FO4 delays. This model of gate delay will be used later when estimating clock cycle time and comparing with wiring delays.

2.2 Clock Cycle Analysis

A resource in a NoC can run at different speed. To study how the clock cycle within a NoC resource scales with the gate delay, we

first examine the relationship between clock cycle and FO4 delay. Recent Pentium4 micro architecture and the aggressive Compaq/DEC alpha chips have 14 to 16 FO4s per clock cycle. Older processors, for example PentiumPro/II, run at 20 to 40 FO4s per clock cycle. It shows that the number of FO4s required in a clock cycle decreases as the technology scales down. Extrapolating historical data would lead to 6-8 FO4s per clock cycle within a few generations [6]. However, such fast-cycling machines pose many difficulties. With 6-8 FO4s per clock cycle, clock skew of a few FO4s would be extreme hard to manage. Furthermore, generating a clock of 8 FO4s per clock cycle is a difficult task since the rise and fall time of a clock wave take more than 2 FO4s to fully transition. With these difficulties in consideration, a clock cycle of 20 FO4s is projected for a cost-performance NoC resource and 10 FO4s for a high-performance one. Thus, with 0.05- μm technology, the clock cycle becomes $20 \cdot 500 \cdot 0.05 = 500$ ps for a cost-performance NoC resource, giving a clock frequency of 2 GHz. Table 1 shows projected clock frequencies for some different technologies.

Table 1. Projected clock frequencies for NoC resources under worse-case FO4 delays.

	0.18- μm	0.13- μm	0.10- μm	0.07- μm	0.05- μm
Cost Perf. (GHz)	0.56	0.77	1.0	1.4	2.0
High Perf. (GHz)	1.1	1.5	2.0	2.9	4.0

2.3 NoC Resource Size Estimation

Knowing the projected clock cycle, the maximum size of a synchronous NoC resource is limited by the wiring delays since the clock signal must be able to traverse 2 resource edges within a clock cycle (assuming the resource is quadratic) in the worst case, see Figure 2.

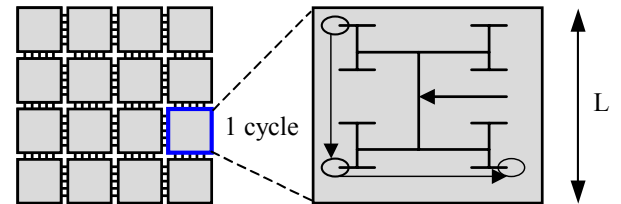


Figure 2. The worst-case delay in a resource.

The wiring delay of a distributed RC line can be modeled as:

$$T_{wire} = 0.4rc l^2$$

Here T_{wire} is the wiring delay, l is the wire length, r is the resistance per unit length and c is the capacitance per unit length. This is a very good approximation and is reported to be accurate to within 4% for a very wide range of r and c [10]. Knowing the clock cycle time and RC delay model, the maximum resource size satisfies:

$$\begin{aligned} \max_wiring_delay &< clock_cycle \\ \Rightarrow 0.4rc(2L)^2 &< clock_cycle \end{aligned}$$

Here, L is the maximum resource edge length. The clock cycle estimation is described in previous section and qualified

predictions on wire resistance and capacitance for future technologies are available in a number of different papers.

The RC -model given above shows that the wiring delay grows quadratically with wire length. To reduce the delay for semi-global and global wires, a long line can be broken into shorter sections, with a repeater (an inverter) driving each section, see Figure 3. This makes the total wire delay equal to the number of repeated sections multiplied by the individual section delay:

$$T_{total} = k \cdot (T_{drv} + 0.4 \cdot rc(l/k)^2)$$

Now, a first order model of the driver (repeater), with lumped output resistance and input capacitance, gives the driver delay as:

$$T_{drv} = 0.7 \frac{R}{h} (hC_0 + hC_g + c \frac{l}{k}) + 0.7r \frac{l}{k} hC_g$$

Here, R is the resistance of a minimum sized inverter, C_0 and C_g are diffusion and gate capacitances of a minimum sized inverter and r and c are wire resistance and capacitance per unit length.

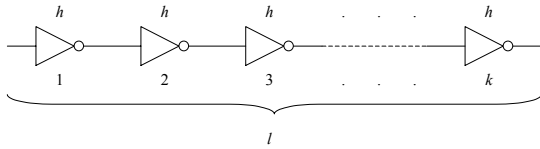


Figure 3. A long wire with k repeaters, each with a size of h times the minimum sized inverter.

The expression above for the total delay can be minimized and the minimum delay per unit length can be shown to be $2.13\sqrt{rcFO1}$ ps/mm [6][11]. Here, FO1 stands for fan-out-of-one delay and $1FO4 \approx 3FO1$. The time for a signal to traverse 2 resource edge lengths should be less than a clock cycle, suggesting the inequality $4.26 \cdot L \cdot \sqrt{rcFO1} < 1 \text{ clock_cycle}$. Using the predicted future semi-global wire (with a width of approximately 3.5 times the minimum feature size) parameters provided in [11], as shown in Table 2, the maximum synchronous resource size and the number of resources on a single chip are calculated and listed in Table 3.

Table 2. Wire parameters for different technologies.

Wire Type	Parameter	0.18- μm	0.13- μm	0.10- μm	0.07- μm	0.05- μm
Semi-Global	R (ohm/mm)	107	185	317	611	1196
	c (fF/mm)	331	268	208	170	155

The resistance and capacitance used to calculate Table 3 are for semi-global wire, since the semi-global wire is normally used within a resource. Routing with global wires within a resource would allow larger resource size, since global wires, in general, have lower resistance and therefore also smaller delay per unit length than semi-global wires. From the table, we have that the maximum size of a synchronous high performance resource is 1.5 mm using 0.05 μm technology. For a cost performance resource with a cycle time of 20 FO4s, twice as long as the high

performance resource cycle time, the maximum resource size is also twice as large.

Table 3. Maximum resource size and number of resources on a single chip, with different technologies.

	Technology	0.18- μm	0.13- μm	0.10- μm	0.07- μm	0.05- μm
	Chip Size (mm)	20	21	23	25	28
High Performance	Max Resource Size	6.5	4.7	3.5	2.4	1.5
	Nr of Resources	9	20	42	112	350
Cost Performance	Max Resource Size	13	9.3	7.1	4.7	3.0
	Nr of Resources	2	5	10	28	87

It should be noticed that the analysis made above is valid for single wires. Crosstalk effects are not taken into consideration. If many wires are in parallel and switch simultaneously, the delay will be higher for unfavorable switch patterns, requiring smaller resource size. Therefore, the derived maximum resource size above should be seen as an upper bound.

3. INTER-RESOURCE BANDWIDTH

3.1 Inter-Resource Delay

The inter-resource communication link will most likely consist of a large number of parallel wires, with uniform coupling over most of the wire length. For such closely coupled parallel wire structures, the crosstalk effects are considerable and cannot be neglected. Hence, the single wire model used in previous section is not valid here. Instead, the model shown in Figure 4 is used. Each wire is modeled as a distributed RC line with total resistance R , total self-capacitance C_s , and total coupling capacitance C_c uniformly distributed over the whole line.

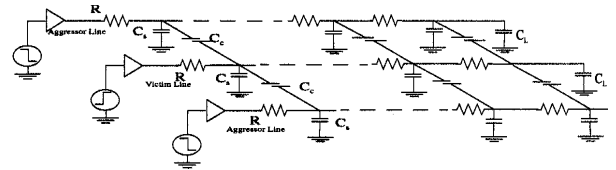


Figure 4. Distributed RC lines with uniform coupling.

The effect of crosstalk on the delay depends on the switching pattern of the aggressor (adjacent) lines. Most often, static timing models that take crosstalk into account are based on a *switch factor*. To model the crosstalk effects, the coupling capacitance is multiplied by this switch factor, which takes the value between 0 and 2 for the best and worst case respectively. In Figure 4, suppose that the victim line in the middle switches up from zero to one, the switching pattern that gives rise to the worst case delay on the victim line is when the two aggressor lines switch down from one to zero (almost) simultaneously [10]. The worst-case delay is then given by:

$$t_{0.5} = 0.7R_{drv}(C_s + 4.4C_c + C_{drv}) + R(0.4C_s + 1.5C_c + 0.7C_{drv})$$

Here, $t_{0.5}$ is the delay for step response to reach 50% point, R_{drv} is the driver (minimum sized inverter) output resistance and C_{drv} is the driver capacitance. Similar to the single wire case, the second term in this expression grows quadratically with the wire length. Inserting repeaters reduces the total wire delay. As shown

In Figure 5, a long wire is broken into k sections, with an h -sized repeater driving each section. For each section, the driver has a lumped resistance of R_{drv}/h and capacitance of $h \cdot C_{drv}$, the wire has a distributed resistance of R/k and self-capacitance C_s/k , the mutual capacitance becomes C_c/k between two adjacent lines.

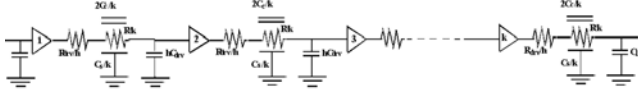


Figure 5. Insertion of repeaters in a uniformly coupled RC line.

Applying the formula for worst-case delay for each section, the total wire delay becomes:

$$t_{0.5} = k \left[0.7 \frac{R_{drv}}{h} \left(\frac{C_s}{k} + hC_{drv} + 4.4 \frac{C_c}{k} \right) + \frac{R}{k} \left(0.4 \frac{C_s}{k} + 1.5 \frac{C_c}{k} + 0.7hC_{drv} \right) \right]$$

To obtain the optimal k and h value, the partial derivatives are equaled to zero, giving [10]:

$$\frac{\partial t_{0.5}}{\partial k} = 0 \Rightarrow k_{opt} = \sqrt{\frac{0.4RC_s + 1.5RC_c}{0.7R_{drv}C_{drv}}}$$

$$\frac{\partial t_{0.5}}{\partial h} = 0 \Rightarrow h_{opt} = \sqrt{\frac{0.7R_{drv}C_s + 3.1R_{drv}C_c}{0.7RC_{drv}}}$$

Now, the optimal value of k must be a positive integer. Using the minimum sized inverter resistance and capacitance from [8], as shown in Table 4, the optimal k and h values are calculated and listed in Table 5. If the optimal k is not an integer, both of the two closest integers are used and corresponding delays are compared to each other in order to find the smallest delay.

Table 4. Resistance and capacitance of minimum sized inverter for different technologies.

	0.18- μm	0.13- μm	0.10- μm	0.07- μm	0.05- μm
Inv. Resistance (ohm)	9020	10560	11370	13710	15080
Inv. Capacitance (fF)	1.795	1.267	0.996	0.709	0.532

From Table 5, we see that the optimal size of the repeaters is large and the number of sections does not seem to be very significant for the delay. The increased number of repeaters only gives marginal improvement in delay. This means that the trade-off between the number of repeaters and the delay should be considered. Also, since the distance between two adjacent switches is one resource edge (neglecting the overhead areas for switches), it might be preferable to not to choose the largest possible resource size. By doing so, the area consuming and power hungry repeaters can be avoided. From this point of view, the resource size should be chosen such that $k=1$ gives the minimum delay. Comparing Table 3 and Table 5, we can clearly see that the largest possible resource sizes require repeaters to reach the minimum delay.

Table 5. Optimal size of the repeaters, h , optimal number of sections, k , closest integer values of k and corresponding delay per unit length.

Technology	0.18- μm	0.13- μm	0.10- μm	0.07- μm	0.05- μm
Optimal h	322	296	226	187	154
Optimal k (1/mm)	0.99	1.30	1.66	2.28	3.33
Integer k (1/mm)	1	1	1	2	3
Total Delay (ps/mm)	65.5	73.2	83.7	91.8	110
Integer k (1/mm)	1	2	2	3	4
Total Delay (ps/mm)	65.5	71.3	76.0	90.1	108

3.2 Inter-Resource Bandwidth Estimation

The wire delay limits the inter-resource bandwidth and distance. To see how these quantities are related, we first assume that a good signal has duration of at least $3t_r$, where t_r is the time for a rising signal to rise from 10% to 90% of its final value. Usually, for RC delays, 0-50% time $t_{0.5} = 0.69\tau$ and $t_r = 2.2\tau$ [12], where τ is the RC time constant. Thus, the bandwidth of a single wire is limited by $\frac{1}{9t_{0.5}}$. Figure 6 shows the allowed maximum length of a global wire at different bandwidths, with and without repeaters. Clearly, for same technology and wire length, wires with repeaters can have higher bandwidth due to their low propagation delay. For an inter-resource distance of 1.5 mm with 0.05- μm technology (assuming that the resources are close to each other and the inter-resource distance is therefore equal to the resource size), the bandwidth between two adjacent resources is estimated to 0.6 Gbps per global wire without repeaters.

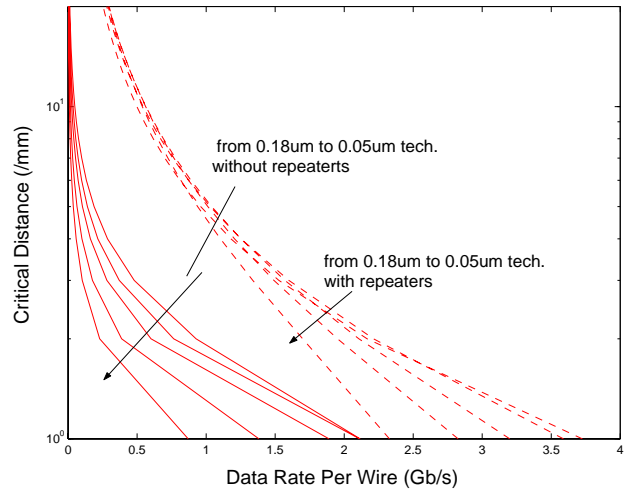


Figure 6. Maximum length of a global wire for different bandwidths and technologies, with and without repeaters.

3.3 Variable Wire Width and Spacing

In the previous paragraph, fixed predictions are used as future wire parameters. In a real process, the wire width and pitch is typically limited by the minimum feature size of the technology. As long as this condition is fulfilled, the wire width and spacing can be varied freely to maximize the inter-resource bandwidth.

For a given total width of the wires, the choice of wire width and pitch decides the total bandwidth. Clearly, wider wires and larger spacing give higher bandwidth per conductor. But the number of conductors allowed in the given total width is also smaller.

Using simulations, Dinesh [10] shows the optimal wire width and spacing with different constraints. For a total wire width of 15 μm , using copper wires with technology dependent constant $\beta = 1.65$, minimum wire width $w = 0.1 \mu\text{m}$, minimum distance between two adjacent wires $s = 0.1 \mu\text{m}$, distance between the signal wires and ground plane $h = 0.2 \mu\text{m}$, wire thickness $t = 0.21 \mu\text{m}$ (giving an aspect ratio of 2.1), the optimal number of wires is 19 if ideal drivers are assumed and no repeaters are used. Using real inverters with output impedance of a minimum sized inverter 7 $\text{k}\Omega$, input capacitance of the same inverter 1 fF and optimal repeater insertion, maximum number of wires allowed (75) also gives the maximum total bandwidth on 20 Gbps.

4. THE INTERCONNECT IP

So far, we have made global wire planning for NoC. To make NoC meaningful and attractive to use, the communication between resources needs to be transparent, the interface between a resource and an IIP needs to be standardized and the IIPs must provide efficient and reliable communication services. Using layered communication architecture and standardized IIPs can fulfill these requirements. Similar to a computer network with layered communication protocols, the NoC is a layered network. The lowest four layers: transport layer, network layer, link layer and physical layer are a part of the NoC backbone and reside outside of the resources.

The different layers mentioned above are implemented in different modules. These layers, together with the modules that implement them, form the Interconnect Intellectual Property (IIP) as shown in Figure 7. The IIP provides the services inter-resource communication relies on.

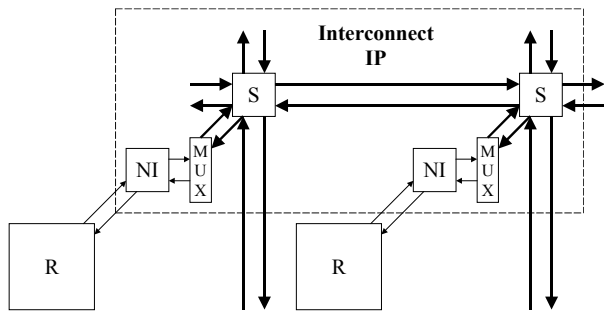


Figure 7. The Interconnect IP modules. R=Resource, NI=Network Interface and S=Switch.

4.1 The Network Interface and MUX Unit

4.1.1 NI Functionality

The Network Interface (NI) works in the transport layer. It is responsible for assemble/reassemble messages from/to multiple packets. As described in paragraph 3.3, the optimal number of wires between two switches may vary depending on technology parameters and different constraints. In Figure 7, a bold arrow

directly connected to a switch denotes a link with the wire configuration (number of wires, wire width etc) that maximizes the inter-resource (switch) bandwidth. Each link handles traffic in only one direction so bi-directional communication requires two links. Different resources may have different number of input and output signals. The NI controls the Multiplexing/Demultiplexing unit (MUX) to map the input and output signals of the resources to/from the switch-to-switch link. It should be noticed that the MUX-to-Switch link width and Switch-to-Switch link width are the same to reduce the complexity of the switch.

4.1.2 One NI Example

One example on the mapping is shown in Figure 8. Here, the transmitting resource is a 64 bits CPU and the link width between the switches is 8 wires. The receiving resource is a main memory (MM) located somewhere else. Furthermore, a packet size of 64 bits of which 32 are header bits is used in this example. This packet structure is used just for demonstration. It may be redefined as the communication protocols are defined in more detail and the traffic model more thoroughly analyzed.

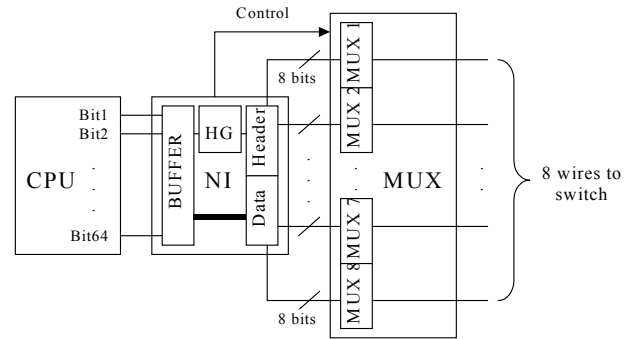


Figure 8. NI architecture.

Before data transmitting, the CPU puts one 64 bits special NOTIFICATION message on its output, notifying the NI that data transmission is to be initiated. Depending on the message content (data type, fault tolerance level, priority, etc), there may or may not be a handshake process between the CPU and the NI. Any additional information besides the first message can also be sent under this handshake process. From the NOTIFICATION or the additional messages, the Header Generator unit (HG) extracts the destination ID (or address) and other useful information for packeting of the data traffic later.

Once the Header Generator is ready, the CPU can start to put data on its output lines, just like with a traditional 64-bits data bus. The data is first stored in the NI input buffer. Since the user data in each packet is 32 bits, two packets are needed to accommodate the stored data. The packeting process starts with the HG writes the 32 header bits into an output buffer in the NI. Simultaneously, the data bits 1 through 32 can be written to the user data part of the output buffer in parallel, as shown in Figure 8. When both header and data bits are written, the first packet is ready to be sent. In a similar manner, the header bits generated by the HG and the data bits 33 through 64 are written into the second data packet. Clearly, the bandwidth between the NI input and output buffer should be at least twice as high as the bandwidth between the CPU and the NI.

When sending a packet, the NI puts the 64 bits packet on its 64 output lines. These lines are divided into 8 groups with 8 lines in each. Each group of lines is connected to the input of an 8×1 multiplexor in the Multiplexing/Demultiplexing unit. The multiplexors are controlled by the NI and the 8 bits data is serialized and transmitted further onto the link connected to the switch. In this way, the MUX unit partially serializes data packets with the same speed as the NI generates them, makes the total MUX-to-Switch bandwidth equal to the NI-to-MUX bandwidth. This means that the MUX-to-Switch bandwidth should accommodate at least twice the CPU (cache) to main memory traffic throughput. With a typical CPU (cache) to main memory throughput in 1 Gbps range [3], the traffic load to the switch generated by the CPU resource is 2 Gbps, which makes the lower bound of the MUX-to-switch and switch-to-switch bandwidth.

The receiving of packet is a reverse process to the transmitting. The MUX unit demultiplexes the data from a switch and passes to the Network Interface. The NI then extracts the user data and sends to the receiving resource, main memory in this example. However, the extracted user data might not be able to be sent right away since a packet only contains 32 bits of user data. The other 32 bits from the CPU is still needed before it can be sent. This property requires low delay variation between the packets and (somewhat) in-order delivery of the packets. Alternatively, the packet size can be increased, for example to 96 bits so that a whole CPU word can be carried in one single packet.

4.1.3 NI Implementation and Simulation

To verify the logic function of the Network Interface and the MUX unit, simulation is carried out. A simplified version of NI and MUX unit is simulated using FPGA components as shown in Figure 9. In order to emphasize the actual data transmission, the Header Generator unit in the Network Interface is not included here. At the transmitting side, the 64 bits CPU data is first stored in a 64 bits wide and 32 words deep FIFO buffer, which represents the NI output buffer connected to the MUX units. To make the schematics simple and foreseeable, only the lowest 16 bits are multiplexed and transmitted. On the receiving side, the data is demultiplexed and transmitted to the main memory.

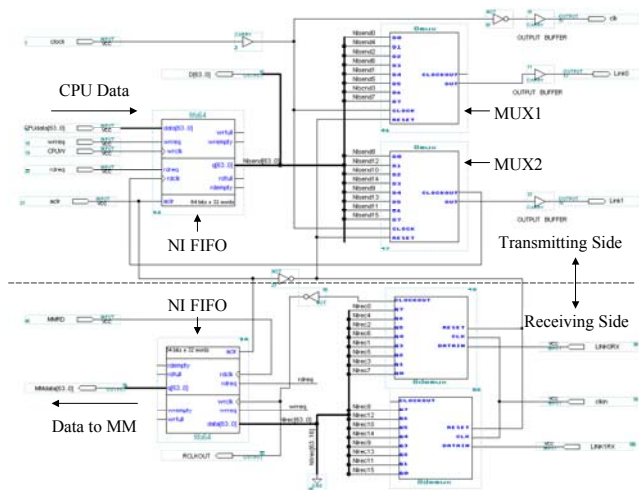


Figure 9. Schematics of simplified NI and MUX unit.

The simulation result is shown in Figure 10. Clearly, the decimal data 1111, 2222... is properly transmitted from the sending side to the receiving side, which proves the correctness of the transmission scheme.

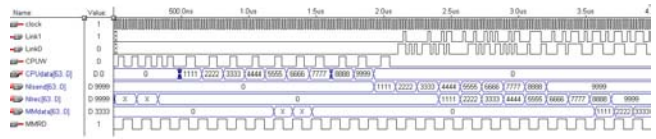


Figure 10. Simulation result of the simplified NI and MUX.

4.2 The Switch and Network Taxonomy

The switch is the other important component in IIP and has a central function in NoC. Responsible for routing data packets, it implements the network (sending resource-to-receiving resource routing) and link layer (switch-to-switch routing). In the example from paragraph 4.1.2, when receiving a data packet, the switch extracts the header information¹, makes routing decision based on the header information and current traffic load (to avoid congestion) and performs appropriate action (put the packet onto a link, delay the packet, drop the packet, etc).

So far, the NoC has been described as a communication network based on data packets and the high-level logic function of the switch is routing the packets. For different network cores, different approaches may be used for data packet routing. In the following text, the traditional telecommunications network taxonomy (also apply on NoC), which determines the low-level architecture and implementation of the switch, will be studied.

As shown in Figure 11, a traditional telecommunications network either employs circuit or packet switching. A link in a circuit switched network can use either FDM (frequency-division multiplexing) or TDM (time-division multiplexing) while packet switched networks are either virtual circuit (VC) networks or datagram networks [7]. This classification can be generalized and apply on any network core, including NoC.

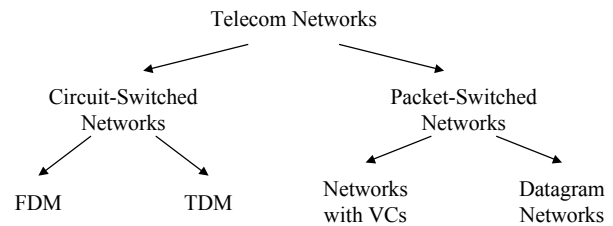


Figure 11. Telecommunication network taxonomy.

4.2.1 Circuit Switching and Philips Aethereal NoC

Even circuit switched network can transmit data in small data packets. The only difference compared to packet switching is that a circuit switched network requires a dedicated end-to-end *circuit* (with a guaranteed constant bandwidth) between the transmitting and the receiving end. As the “circuit” is an abstract concept,

¹ Mainly destination address if datagram based switching policy used, virtual circuit number if virtual circuit based switching policy used, and priority information bits if any.

most of the time, it is not a physical end-to-end wire, but can span over many links. In a telecommunications network, the circuit is typically implemented with either frequency-division multiplexing (FDM) or time-division multiplexing (TDM) in each link [7]. With FDM, the frequency spectrum of a link is shared among the connections across the link. Obviously, the FDM is not suitable for NoC. For TDM on the other hand, time is divided into frames of fixed duration, and each frame is divided into a fixed number of time slots as shown in Figure 12. When the network establishes a connection (or circuit) across a TDM link, the network dedicates a certain number of time slots in every frame to the connection. These slots are dedicated for the sole use of that connection, with some time slots available for use (in every frame) to transmit the connection's data [7].

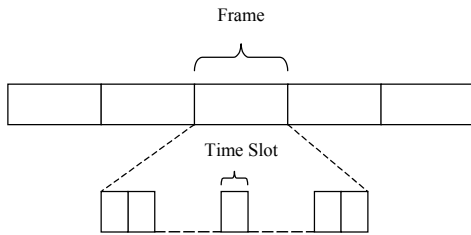


Figure 12. Circuit realization with TDM.

The \mathcal{A} etheral Network on Chip developed at Philips Research is based on the time-division multiplexed circuit switching approach described above [2]. Here, the network provides two different kinds of services to support differentiated data traffic: guaranteed throughput (GT) and best-offer (BE) traffic. For the GT traffic, a connection needs to be established before the actual transmission can take place. When establishing a connection, the switches reserve a number of time slots on each link along the path from the sending resource to the receiving resource. This connection-oriented service has many advantages. First, the congestion control mechanism is built-in in the connection establishing process, resulting in contention-free traffic. Second, the time slots are fixed in each time frame, meaning that the delay of a data packet between two consecutive switches is bounded by a time frame. The total delay is then constant and bounded by the number of hops between the two ends multiplied with the time frame. At last, since the delay is (approximately) constant for each GT packet, the data packets will also be received in order. The best-offer traffic is connectionless. It uses unutilized time slots to transmit data packets. More detailed information on the \mathcal{A} etheral NoC can be found in [2].

4.2.2 Packet Switching

Depending on the routing method, packet switched networks are divided into virtual circuit (VC) networks and datagram networks. The virtual circuit approach is connection-oriented and resembles the circuit switching. Both packet switched VC network and circuit switched network are suitable for uniform data traffic with long lifetime. For other bursty traffic, the connection management will tend to be computationally demanding and occupy a large portion of the bandwidth. They also require that the switches maintain the state information, resulting in more complex switch architecture and signaling scheme between switches.

To reduce the switch complexity and therefore also the area overhead of the network, a datagram based switching policy is used in our NoC approach. That is, the switches are state- and memoryless, each packet is treated independently, with no reference to preceding packets. This approach more easily adapts to changes in the network such as congestion and dead links. However, it does not guarantee that packets with same source and destination will follow the same route. Consequently, the delay of packets with same source and destination may vary and packets may also arrive out of order, requiring buffering element at the receiving end.

5. CONCLUSIONS

In this paper, we studied the NoC system parameters and Interconnect Intellectual Property in NoC. Predictions on future technology feature size, clock speed in a synchronous resource, maximum NoC resource size, optimal global communication bandwidth and inter-resource distance, are made. These quantities are closely related to each other. The technology determines the gate delay, which in turn determines the maximum clock frequency. The maximum resource size can then be derived from the obtained clock frequency and the semi-global wire delay. Finally, the global communication bandwidth is limited by the distance between resources and the global wire delay. Providing estimations on these system parameters, this paper provides a global wire planning scheme using the IIPs and can be used as a guideline for NoC system architecture definition. This can be demonstrated in a numerical example: for a NoC in 50-nm technology, the clock frequency is estimated to be 4 GHz for a high-performance synchronous resource with an edge length of 1.5 μm . With an inter-resource distance of 1.5 μm , there is room for about 350 such resources on a single chip of 28 \times 28 μm^2 . The bandwidth between two adjacent resources is estimated to be 0.6 Gbps per global wire without using repeaters.

The IIPs connect different resources in NoC. The main components in an IIP are the Network Interface and the switch. As the number of wires for optimal global communication bandwidth might not be the same as the number of input/output signal lines to/from a resource, the Network Interface is needed. It also assembles/reassembles data stream from/to a resource. The switch has the function of routing the data packets to their destination. For different types of underlying network cores, different switch architectures and routing policies are possible. Simulation shows that a multiplexing/demultiplexing transmission scheme of the IIP is feasible, independent of the switch implementation.

Future work evolves packet definition, reliable communication mechanism and switch architecture. Furthermore, applications that fully utilize the services provided by NoC need to be developed. At last, performance evaluation and estimation on area overhead of the packet switched network are needed to compare it to a more conventional bus structure and dedicated wires.

6. ACKNOWLEDGEMENTS

This work is partly funded by the SOCWARE and COMPLAIN project. Productive discussions with Dinesh Pamunuwa and supportive advising from Axel Jantsch and Johnny Öberg have been of great importance for this work and are gratefully acknowledged.

7. REFERENCES

- [1] Dally W. J. and B. Towles. Route Packets, Not Wires: On-Chip Interconnection Networks. Design Automation Conference, 2001, Proceedings, 684-689.
- [2] Goossens K. Guaranteeing the Quality of Services in Network on Chip. Chapter 4 in "Network on Chip", Kluwer (March 2003). <http://www.dcs.ed.ac.uk/home/kgg/2003-networksonchip-chap4.pdf>
- [3] Guerrier P. and Greiner A. A Generic Architecture for On-Chip Packet-Switched Interconnections. Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings, 2000, 250-256.
- [4] Ho R, Mai K. W. and Horowitz M. The Future of Wires. Proceedings of The IEEE, vol. 89, no. 4 (April 2001).
- [5] Hemani A., Jantsch A., Kumar S., Postula A., Öberg J., Millberg M. and Lindqvist D. Network on Chip: An Architecture for Billion Transistor Era. Proceeding of the IEEE NorChip Conference (November 2000).
- [6] Jantsch Axel. Network on Chip. Proceedings of the Conference Radio vetenskap och Kommunikation, Stockholm (June 2002).
- [7] Kurose J. F., Ross K. W. Computer Networking: A Top-Down Approach Featuring the Internet. Addison Wesley Longman, Inc (2001).
- [8] Maheshwari A., Srinivasaraghavan S. and Burleson W. Quantifying the Impact of Current-Sensing on Interconnect Delay Trends. ASIC/SOC Conference, 2002. 15th Annual IEEE International (2002), 461-465.
- [9] Nilsson E. Design and Implementation of a Hot-potato Switch in Network on Chip. Master of Science thesis, Laboratory of Electronics and Computer Systems, Royal Institute of Technology (KTH), Sweden (June 2002).
- [10] Pamunuwa D., Zheng L-R. and Tenhunen H. Optimising Bandwidth Over Deep Sub-micron Interconnect. Proc. of the 2002 IEEE International Symposium on Circuits And Systems (ISCAS), Scottsdale, Arizona, USA (May 2002).
- [11] Tenhunen H. Workshop "Systems on Chip, Systems in Package", ESSCIRC 2001, Villach Austria (Sep 2001).
- [12] Zheng L-R. Design, Analysis and Integration of Mixed-Signal Systems for Signal and Power Integrity. PhD thesis, Laboratory of Electronics and Computer Systems, Royal Institute of Technology (KTH), Sweden (2001).
- [13] Zheng L-R. Design and Analysis of Power Integrity in Deep Submicron System-on-Chip Circuits. Analog Integrated Circuits and Signal Processing, 30, 2002, 15-29.