PERFORMANCE EVALUATION IN COMMUNICATION DESIGN EXPLORATION FOR SoC DESIGN

S. Eslava, M. Strum, W.J. Chau
Microelectronics Laboratory, University of São Paulo
LME-EPUSP

seslava@lme.usp.br strum@lme.usp.br jcwang@lme.usp.br

ABSTRACT

The on-chip communication design of a SoC is a straightforward set of tasks. Traditional approaches start from very abstract models, usually analytical and they continue at the RTL level. Between these two abstraction levels there is a huge gap. TLM has been used the close this gap through the use of three different sub-levels. In this paper we present the use of two of these sub-levels. Including performance metrics that can be obtained at every sub-level and their application in the on-chip communication design. The performance evaluation is composed of two sub-tasks, estimate the metrics value and analyze the results. Previous works show how to estimate the performance of bus-based structures but they do not present analysis. In this paper we utilize the metrics value to define the quantity of buses, the element's mapping and the priorities for the elements of the on-chip communication structure.

1. INTRODUCTION

Designing system-on-chips (SoC) is a complex task. It can be simplified using the concept of *orthogonalization of concerns* [2]. This strategy allows, for instance to split the SoC design process into hardware and software or into function and architecture. It also allows to split the design into computation and communication.

The SoC on-chip communication design defines: 1) the type of communication structure (CS) (bus based or network-on-chip based); 2) Define the physical configuration parameters of the CS; 3) Define the logical attributes of the CS elements. This leads to a very wide design space exploration.

The impact of the on-chip communication structure on the SoC performance has been discussed by many authors [1, 2].

The on-chip communication design may be realized during different SoC design phases (different abstraction levels). The widely adopted RTL level offers good precision but requires a high development and simulation time effort. In contrast, the analytical abstraction level offers design speed but may be inaccurate and can hide important system details. The transaction level (TLM) is being considered a valuable alternative between these abstractions. Recent studies showed that the TLM abstraction may be further decomposed into TLM sub-levels. Three different TLM sub-levels have been identified [4]: 1) Untimed; 2) Estimated time; 3) Cycle time.

In [6] the on-chip communication design was performed using analytical model (Markov chains). They estimated the latency. This metric was used to define the bus width

In [7] the design was realized using a SystemC TLM model. They used a reduced set of performance metrics (transactions throughput, Arbitration effects). They compared different arbitration policies (TDMA, fixed priority, round-robin) and bus protocols. They do not present any correlation between the metrics results and parameters decision.

In [5] the design was performed at the RTL level. They do not present any performance metric. They design the SoC based tuning the parameters of a set of pre-defined components including a bus-based communication structure.

In this paper we show how the TLM sub-levels may be used to reduce the gap existing between the RTL and the analytical models through a progressive refinement design. Defining: 1) the number of bus instances (for a bus communication structure); and 2) the IP modules mapping (to the allocated buses) using performance metrics at the untimed TLM level and how to assign fixed priorities for each IP module (assuming that this type of arbitration policy has been adopted) using performance metrics at the estimated time TLM level.

This paper is structured as follows: Section 2 presents the use of TLM models for on-chip communication design. Section 3 presents the performance metrics used at the untimed and estimated sub-levels. Section 4 presents our results. Finally Section 5 presents the conclusions and future works.

2. ON-CHIP COMMUNICATION DESIGN

Figure 1 shows the different abstraction levels that may be used in a top-down design strategy. The X and Y axes represent the communication and the computation [8]. The grey zone represents the synthesis from RTL models. The figure shows nine possible abstractions.

Points labeled as G and H correspond to untimed computation behavior. At this sub-level, the hardware elements are not yet defined. This definition occurs at the transition between the untimed computation to the

* This work was supported by CNPq and FAPESP grants

estimated time computation. Hence those abstractions pairs are meaningless. The point labeled as I corresponds to the pair {untimed communication, cycle-level computation}. In the present approach, these two tasks are done at the same time

At the untimed sub-level the communication is represented as a set of ideal (point-to-point) channels. Read and write operations are meaningless. Channels contention does never occur. The computation is represented as a set of tasks that communicate between them as functions, messages and variables.

At the estimated timed sub-level, the communication is represented through shared channels. Bus contention may occur. An arbitration policy must exist. Operations have latency and it is estimated as one cycle per transferred bus-word. The model is functional accurate, but it is not signal accurate neither protocol accurate. The computation at this sub-level is composed by a set of hardware elements (masters and slaves).

At the cycle timed sub-level the communication is represented through a more detailed bus model. This is: 1) signal accurate (every signal present at the RTL representation must be present at this level); 2) protocol accurate (the model represent the protocol behavior of every bus); 3) functional accurate (every state and behavior must be modeled). The computational elements are modeled with the accuracy present at the communication.

These abstraction sub-levels may be used to gradually refine the communication structure design.

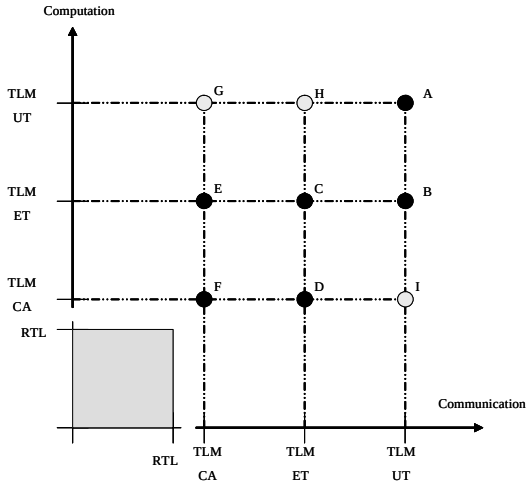

Figure 1. TLM Sub-levels

The on-chip communication design is composed of three tasks:

1) Define the CS type (bus or *NoC*);
2) Define physical elements;
    a. Define number of buses/type
    b. Element assignment.
    c. Define physical attributes/bus
3) Define the logical characteristics.

The starting point of the CS design is the hardware/software partitioning that produces the list of hardware elements. The partitioning is present at the estimated time sub-level. B point in figure 1. Our approach focus to an ASAP configuration parameters decision.

The first task defines the type of the on-chip communication, bus-based or NoC-based. This decision is based on an analysis of the requirement of the SoC. In this paper we will consider a bus based.

The second task defines the physical elements. This task can be done at different sub-levels. During this task two set of parameters are defined. The first set affects the SoC architecture. This is called IP elements mapping and is composed of; 1) Define the quantity of buses; 2) Do the elements assignment. These decisions can be done analyzing the communication characteristics between the hardware elements. The ASAP point to do this task is B (figure 1).

The second set of parameters adjusts the physical attributes of the buses: 1) Arbitration policy and its parameters (fixed priority (master ID)); 2) Bus Size. This sub-task needs models that represent the buses (they have already defined). The earliest point to do this sub-task is with communication estimated. (points C and D).

The third task defines the logical characteristics of the buses: 1) Type of bus: high or low throughput bus; 2) Protocols access bus; 3) Attributes. This task needs detailed buses models. These types of models are present at the cycle timed sub-level. The earliest point where this can be done are E and F

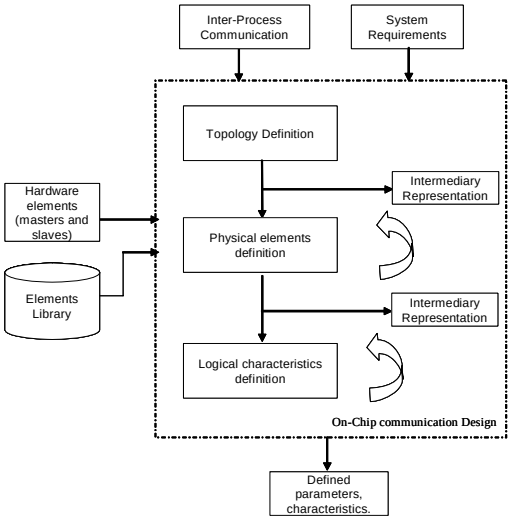

Figure 2. On-Chip communication design

3. PERFORMANCE METRICS

The performance evaluation is composed of two sub-tasks: 1) metrics estimation; 2) analysis.

3.1 Untimed sub-level metrics

Traffic Generation (*TG*): Equation 1 represents the full traffic generated by all the masters.

$$TG = \sum_{i=o}^{n} GD_i \cdots [Bytes] \qquad \text{Equation 1}$$

Where, GD_i represents the number of data bytes generated in each master element.

Channel Utilization Level (*CUL*): Equation 2 represents each dedicated channel activity.

$$CUL_{ij} = \frac{DC_{ij}}{\sum_{i=o}^{n} GD_i} = \frac{DC_{ij}}{TG} \cdots [\%] \quad \text{Equation 2}$$

Where, DC_{ij} represents the data bytes flowing through each dedicated channel. *I*f the traffic is bidirectional, C*UL* represents their sum (CUL_{ij}+ CUL_{ji}).

Untimed Communication Locality (UCL): Equation 3 represents the partial traffic between elements *i* and *j* with respect to the full traffic generated by master element *i*.

$$UCL_{ij} = \frac{DC_{ij}}{GD_i} \cdots [\%] \quad \text{Equation 3}$$

3.2 Estimated time sub-level metrics.

Average Latency (AL): Equation 4 represents the average over all read and write operations.

$$AL = \frac{\sum_{i=0}^{n} L_{Ti}}{\#\_Operations} \cdots [cycles] \quad \text{Equation4}$$

Where, Lt represents the latency of every operation.

Data Throughput (DT) (Equation 5).

$$DT = \frac{\sum_{i=0}^{n} GD_i}{ExT} \cdots \left[\frac{bytes}{cycle}\right] \quad \text{Equation 5}$$

Global utilization level (GUL): Equation 6 represents the utilization level of every bus.

$$GUL = \frac{TD_Q}{ExT} \cdots [\%] \quad \text{Equation 6}$$

Where, TDq represents the quantity of transactions.

Element participation (EP): Equation 7 represents the participation of every element in the global utilization of every bus.

$$EP_i = \frac{DG_i}{\sum_{i=0}^{n} DG_i} \cdots [\%] \quad \text{Equation 7}$$

These performance metrics per abstraction level were used to find the ASAP decision for configuration parameters decision.

4. RESULTS

We will illustrate our method showing how to define 3 configuration parameters: quantity of buses, elements mapping (at the untimed sub-level) and elements fixed priorities (at the estimate timed sub-level). Our system under analysis (SUA) is composed of 4 masters M1, M2, M3, M4 and two slaves S1, S2. The masters are parametrical traffic generators. One transaction consists of one read and one write operation. At the estimated time sub-level the master-master communication is represented as master-slave-master. Every data exchanged by two masters is stored at the memory by the master source and then loaded by the master target. This model is used by Shared memory and Message passing programming models.

4.1 Analysis at the untimed sub-level

The metrics at this sub-level serve to identify the intensity of the traffic among pairs of master/slave elements. The values shown in figure 3 were acquired from the untimed model simulation using the next traffic conditions.

- Time between transactions = Burst Size
- Burst size = M1=10; M2=20; M3=10; M4=20 (words)

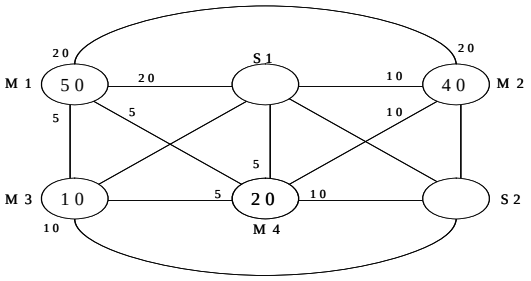

Figure 3. Example system architecture.

The total traffic generated by the system is obtained using the equation 1:

$$TG = 50 + 40 + 20 + 10$$

$$TG = 120KB$$

The channel utilization level is obtained using the equation 2.

$$CUL_{M1S1} = \frac{20}{120} = 16,6\%;..CUL_{M1M3} = \frac{5}{120} = 4,16\%;$$

$$CUL_{M1M4} = \frac{5}{120} = 4,16\%;..CUL_{S1M2} = \frac{10}{120} = 8,33\%;$$

$$CUL_{S1M4} = \frac{5}{120} = 4,16\%;..CUL_{M2M4} = \frac{10}{120} = 8,33\%$$

$$CUL_{M1M2} = \frac{20}{120} = 16,6\%;..CUL_{M2M1} = \frac{20}{120} = 16,6\%$$

$$CUL|_{M1M2}| = 33,33\%;.....CUL_{M3M4} = \frac{5}{120} = 4,16\%$$

$$CUL_{M3S2} = \frac{10}{120} = 8,33\%;..CUL_{M4S2} = \frac{10}{120} = 8,33\%$$

The communication locality is obtained with equation 3.

$$UCL_{M1S1} = \frac{20}{50} = 40\%;..UCL_{M1M2} = \frac{20}{50} = 40\%$$

$$UCL_{M1M3} = \frac{5}{50} = 10\%;..UCL_{M1M4} = \frac{5}{50} = 5\%$$

$$UCL_{M2M1} = \frac{20}{40} = 50\%;..UCL_{M2S1} = \frac{10}{40} = 25\%$$

$$UCL_{M2M4} = \frac{10}{40} = 25\%;..UCL_{M3S2} = \frac{10}{10} = 100\%$$

$$UCL_{M4S1} = \frac{5}{20} = 25\%;..UCL_{M4M3} = \frac{5}{20} = 25\%$$

$$UCL_{M4S2} = \frac{10}{20} = 50\%$$

Based on the communication locality it is possible to identify the master/slave pairs do not communicate between each other (M1 with S2; M2 with M3 and S2; M3 with S1) and those that communicate most (M1 with S1; M1 with M2; M2 with M1; M3 with S2; M4 with S2).

Elements that do not communicate will be mapped on different buses while those that most communicate will be mapped on the same bus. Applying these criteria the elements mapping becomes: M1 and S2: different buses.

- M3 and S2: same bus.
- M1 and M2: same bus.
- S1, M1 and M2: same bus.
- M4, M3 and S2: same bus.

Hence, there are two buses. These buses are interconnected through a bridge as shown in figure 4.

- Bus0 → M1, M2 and S1.
- Bus1 → M3, M4, and S2.

The M4 element can be mapped to the bus0. M4 has most traffic with the elements that are at the bus0. This option is used at the simulation D (architecture 1).

4.2 Analysis at estimated time sub-level

Four 100.000 cycles simulations were conducted. Simulations A, B and C use the architecture 0 of figure 4 and simulation D considered architecture 1. Every simulation has a different set of priorities (table 1). The metrics results are utilized to define the more suitable set of priorities.

These priorities were chosen to diminish the bottleneck caused by the bridges (higher priorities to the bridges) (simulations A, B and D); Give higher priority to the higher Traffic Generators. (Simulation A, C and D). The results obtained are presented at the tables 2, 3, 4 and 5.

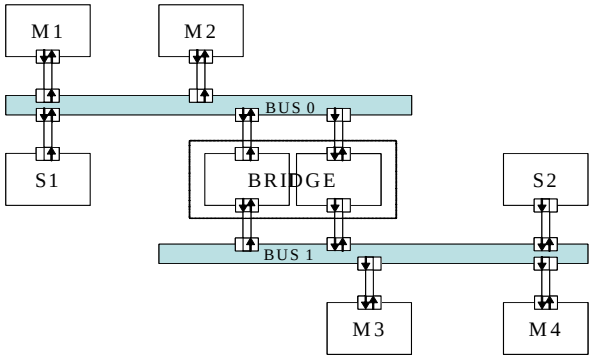

Figure 4. Architecture 0 based on the untimed metrics

	Bus 0	Bus 1
Simulation A	M1 = P2 M2 = P3 Bridge = P1	M3 = P2 M4 = P3 Bridge = P1
Simulation B	M1 = P3 M2 = P2 Bridge = P1	M3 = P3 M4 = P2 Bridge = P1
Simulation C	M1 = P1 M2 = P2 Bridge = P3	M3 = P2 M4 = P3 Bridge = P1
Simulation D	M1 = P2 M2 = P3 Bridge = P1 M4 = P4	M3 = P2 Bridge = P1

Table 1. Priorities of every simulation.

AL (cycles)	M1	M2	M3	M4
Simulation A	1,39	2,76	1,15	3,36
Simulation B	2,29	1,49	2,40	2,12
Simulation C	1,36	2,28	7,43	9,10
Simulation D	1,26	1,75	2,68	11,47

Table 2. AD results

DT (bytes/cycle)	Total
Simulation A	1,19
Simulation B	1,08
Simulation C	0,85
Simulation D	1,02

Table 3. DT results

MP (%)	M1	M2	M3	M4
Simulation A	35,01	17,64	31,68	15,65
Simulation B	27,87	26,29	23,53	22,30
Simulation C	49	27	13	10
Simulation D	43,29	26,09	23,36	7,24

Table 4. MP results

GUL (%)	Total
Simulation A	119,05
Simulation B	108,79
Simulation C	85,73
Simulation D	102,15

Table 5. GUL results.

The simulations A and B have the most GUL. This means that the most traffic is produced and consumed locally. The simulation C has the lowest GUL. So, this set of priorities is not used.

Simulation D presents an AL between 1,26 (M1) and 11,47 cycles (M4). This means an unbalanced architecture. So, architecture 1 is not used. The simulations A and B present the less AL results. The both simulations use the architecture 0 and the bridges have the bigger priority at both sides.

As both master M1 and M3 are high traffic generators, the simulation B presents a more uniform LA for both elements and their arbitration priorities are chosen.

5. CONCLUSIONS

On-chip communication design may be decomposed into three tasks. Each task defines a set of parameters. Their decisions affect the SoC architecture and the attributes of every bus. Our search focus on an ASAP decision of the parameters. ASAP means taking the decision at the highest possible level of abstraction.

We illustrated that the use of the performance metrics at different sub-levels allowed the definition of some of the bus-based parameters. The metrics obtained at the untimed sub-level defined the quantity of buses and the elements mapping to the different buses. The estimated time metrics were useful to define the more suitable set of arbitration priorities of the different buses. Future work pretends to identify new parameters that can be defined using the performance metrics presented. Expand this study including the cycle timed sub-level.

6. REFERENCES

[1] A.A. Jerraya, W. Wolf (Ed.), "*Multiprocessors System-on-Chip*", Morgan Kaufman Publishers, 2005

[2] K. Keutzer et al., "*System-Level Design: Orthogonalization of Concerns*", IEEE Transactions on computer-aided design of integrated circuits and systems, December 2000.

[3] "*AMBA AHB Cycle Level Interface Specification, Design Methodology and Tools*" ARM, 2003.

[4] S. Eslava, G.Lozada, M.Strum, J.C. Wang "*Performance estimation for on-chip communication structures at different TLM sub-levels*", XI WorkShop Iberchip, 2005

[5] N. E. Zergainoh, A. Baghdadi, A. Jerraya, "*Hardware/Software Codesign of On-chip Communication Architecture for Application-Specific Multiprocessor System-on-Chip"*, International Journal of embedded Systems Vol. 1 No. 12, September 2004.

[6] S. Pandey et al., "*High Level Hardware/Software Communication Estimation in Shared Mamory Architecture*", Symposium on Circuit and Systems (ISCAS), Japan, 2005.

[7] S. Pasricha, N. Dutt, M. Ben-Rodhane, "*Extending the transaction level modeling approach for fast communication architecture exploration*", DAC, June 2005.

[8] L. Cai, D. Gajski, "*Transaction Level Modeling: An Overview in System Level Design"*, CODES-ISSS, 2005.