# Pre-Synthesis Evaluation of Digital Bus Micro-Architectures

R. Garcia-Ramirez*, A. Chacon-Rodriguez*, C. Strydis†, and R. Rimolo-Donadio*
*Escuela de Ingeniería Electrónica, Instituto Tecnológico de Costa Rica
†Dept. of Neuroscience, Erasmus Medical Center, Rotterdam, The Netherlands
Email: {rgarcia, alchacon, rrimolo}@tec.ac.cr, c.strydis@erasmusmc.nl

***Abstract*—Buses are central building blocks in the architecture of digital systems. There are numerous standards for bus architectures and evaluation metrics in terms of data transfer rate, quality of service, and latency; however, it is not common to find metrics related to the physical features of bus implementations, such as power consumption and area in terms of their micro-architecture. This paper evaluate bus micro-architectures at pre-synthesis level, allowing for the comparison of alternative circuits implementing the same standard and thus providing estimations on the power consumption and area requirements. A metric is proposed to evaluate the bus implementation and its utilization is shown with generic serial and parallel buses, based on simulations with a 0.18$\mu$m CMOS standard cell library.**

***Index Terms*—Bus, Interconnects, Micro-Architecture, System-on-Chip, Very Large Scale Integration.**

## I. Introduction

Buses are the preferred interconnect architecture for the implementation of digital systems; there are many variations in terms of protocols and interfaces, for instance AMBA, STbus, Avalon, Core Connect, and Wishbone, to name some typically found in modern System-on-Chip (SoC) solutions [1].

The performance of interconnect networks is often evaluated by metrics such as latency, bandwidth, and throughput [2]. Latency is defined as the average time required by the packages of information to reach their final destination. Bandwidth is the amount data per unit time that travels through the bus, and throughput is the rate of data that is transmitted between entities capable to generate and consume data (here on referred as agents), which is lower than the bandwidth. Other metrics are intended to measure quality of service and fault tolerance.

All the previous metrics are affected by the protocol stack used, which defines the conformation of the flit[1], as well as by the general architecture of the bus [1]. Once these are fixed, the bus micro-architecture becomes the main factor determining its final power consumption and area. As such, in order to evaluate in detail the area and power of any particular micro-architectural implementation, it is necessary to carry out its design at least to the gate level.

Thus, it would be useful to have metrics allowing for the micro-architectural evaluation in an early stage, without requiring a full synthesis of a particular implementation. Analytical models intended to the analysis of delay [3], throughput [4] in Network on Chip (NoC), and methodologies for the evaluation of power at the CPU level [5] can be found in literature; but there are no metrics that can be used for the evaluation of the complexity of a micro-architecture for a given bus architecture and stack of communication protocols.

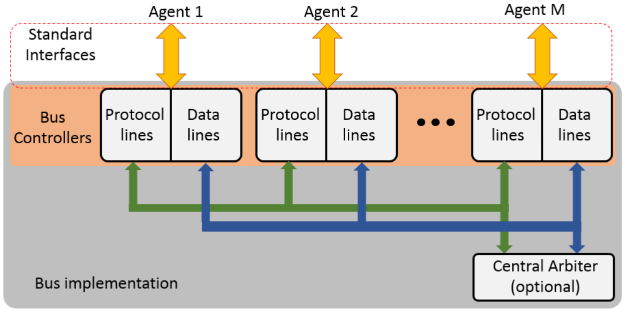

Fig. 1. Block diagram of a generic bus implementation.

This paper proposes a general metric for the evaluation of bus implementations. The proposed approach is applied for the evaluation of serial and parallel buses, which are compared in terms of area and power requirements. Metrics are validated against behavioral simulations using libraries of a 0.18$\mu$m CMOS process.

## II. Metric Proposal

A basic representation of the implementation of a bus interconnect is presented in Fig. 1. Every bus consists of bus controllers connected to a shared medium; the standard interface defined by the architecture is reflected in the connection between the bus controller and the agents. The shared lines connected to all the bus controllers can be classified as data lines, used for the transmission of payload, and control lines, intended for synchronization and routing. It is possible to add central structures (arbiters) for the synchronization of the communication between agents. Such structures normally require the routing of protocol and/or data lines between the central structure and the agents. Since the arbiter is a central block, the routing of lines from the bus controllers to this central block may not be acceptable for some system implementations where the bus controllers are physically distant, since this increases routing area and usually require premium routing tracks.

[1] Defined as the minimum package of information which can travel in the physical layer of the bus implementation

978-1-7281-3146-7/20/$31.00 ©2020 IEEE

The total silicon area of any bus depends on the area of the bus controllers ($A_D$), the area occupied by shared structures in the system ($A_S$), and the routing area ($A_R$) which will depend on the floor-plan of the specific implementation. The total area ($A_T$) for a bus with $M$ interfaces (or agents) can be described as

$$A_T = M \cdot (A_D) + A_S + A_R \tag{1}$$

In order to implement a bus architecture, an additional block to translate between the interfaces and protocols must be added to the bus controllers, so that in terms of area the translation between different bus implementations, it can be seen as a constant added to $A_D$.

There is an inverse relation between the number of lines shared by the bus controllers and the required implementation area $A_D$, since the use of centralized structures implies more shared signals. The use of distributed structures, in contrast, implies the duplication of logic structures in the bus controllers. In the case of the serial implementations, the reduction of the number of wires implies an increases of the area for the bus controllers. For the routing area $A_R$ and the area of the shared structures $A_S$, an inverse relation with the number of lines required for routing is appreciated. In terms of power consumption, one can postulate that the power consumption increases with the area of the system and its throughput, assuming the maximum utilization of the throughput of the micro-architecture in a pessimistic test scenario, since this implies the maximum activity factors for all the nodes.

Every bus has a "minimum latency" ($T_{min}$), consisting of the minimum time required to transmit a flit of information. Assuming that every $T_{min}$ it is possible to transmit $n$ bits of flit, with $k$ bits of payload and $n-k$ bits of overhead in the flit, it is possible to define the throughput of the bus ($R$) in bits/second as

$$R = \frac{k}{T_{min}} \tag{2}$$

Based on Eq. (2) and changing the minimum latency ($T_{min}$) to a minimum latency in clock cycles ($T_{C_{min}}$), where $T_{C_{min}} = T_{min}/T_{clk}$, it is possible to define the "Throughput per clock period" ($R_C$) in bits/T$_{clk}$ as

$$R_C = \frac{k}{T_{Cmin}} \tag{3}$$

If there are $L$ data/protocol transmission lines in a bus, a metric called "Efficiency per line per clock cycle" ($\eta_{LC}$) may be proposed to evaluate any bus implementation. This metric counts the number of useful bits transmitted by each shared transmission line of the bus per clock cycle, as given by

$$\eta_{LC} = \left(\frac{R_C}{L}\right) \tag{4}$$

Notice here that $L$ also accounts for the lines communicating each bus controller to the shared structures. For an ideal digital bus, $\eta_{LC} = 1$, i.e., every shared line in the bus sends one bit of payload every clock cycle, and therefore there are no lines or clock cycles "wasted" in the synchronization of the flits, or dedicated to the transmission of headers for the bus protocol (assuming the bus is working at its maximum transmission capacity). This metric considers inefficient implementations both in terms of the protocol and the micro-architecture. Using $\eta_{LC}$, it is therefore possible to evaluate, for instance, different micro-architectural implementations of the same standard in terms of their area of implementation before synthesis, assuming they work at the same clock frequency. Based on Eq. (1), one can postulate that:

$$\begin{aligned} A_D &\approx \alpha \cdot \eta_{LC} \\ A_S &\approx \beta \cdot (\eta_{LC})^{-1} \\ A_R &\approx \gamma \cdot (\eta_{LC})^{-1} \end{aligned}$$

where $\alpha$, $\beta$, and $\gamma$ are constants related to the micro-architecture and they must be estimated accordingly to a particular micro-architecture.

## III. Evaluation of the Proposed Metric

A SystemVerilog interconnect library was developed in order to generate the buses to be evaluated. Once the RTL is generated from this custom library, it is embedded into a testbed, intended to reach the highest possible dynamic power consumption for the micro-architecture, while also reaching the maximum data transfer rate of the system. This generates a pessimistic power scenario. The results of the activity factors for every signal are recorded and feed into an RTL compiler and synthesizer tool, in order to generate a gate level netlist, from which accurate area and power estimations may be extracted.

Four micro-architectures were implemented, serial and parallel buses with and without central arbiters. In the serial bus without central arbiter ($SB$), one line transmits data and three are used for signaling; in the serial bus with central arbiter ($SBA$), two additional connections per bus agent to the arbiter are required. The parallel versions without arbiter ($PB$) has a wire per bit in the flit and additional bits for the synchronization of the data flow, whereas, the version with central arbiter ($PBA$) require additional $n$ plus two bits per bus controller to coordinate with the central arbiter.

Table I compares $T_{Cmin}$, $L$ and $\eta_{LC}$ for each bus. In every bus in this work, $k$ is equal to the number of bits in the flit minus eight corresponding to the identifier of the intended receiver of the message. In the $SB$ implementation, the synchronization between agents is accomplished using a "token pass" strategy; a wire called "turn_change" is used to pass the token between agents in a round robin arbitration style, signaling when one agent finishes using the bus; an additional wire called "bus_busy" is shared between agents to inform when the bus is used and the data is transmitted serially using one wire. In order to transmit a single flit, four clock cycles are required for the synchronization of the data and an additional clock cycle s required for each one of the bits in the flit. Based on this, we can conclude that for $SB$, $T_{Cmin} = n + 4$ and $L = 3$.

For $SBA$, the transmission synchronization between agents is centralized in the arbiter, using "request" and "grant" signals; also, a single line is used for the transmission of data;

TABLE I
PARAMETERS FOR THE CALCULATION OF THE "EFFICIENCY PER LINE PER CLOCK CYCLE" ($\eta_{LC}$).

| **Serial** | **SB** | **SBA** |
|---|---|---|
| $T_{Cmin}$ | $n+4$ | $n+3$ |
| $L$ | 3 | $1+2\cdot M$ |
| $\eta_{LC}$ | $\frac{n-8}{3\cdot(n+4)}$ | $\frac{n-8}{(1+2\cdot M)\cdot(n+3)}$ |
| **Parallel** | **PB** | **PBA** |
| $T_{Cmin}$ | 5 | 5 |
| $L$ | n+3 | $n+log_2(M)+M\cdot(n+1)$ |
| $\eta_{LC}$ | $\frac{n-8}{(n+3)\cdot 5}$ | $\frac{n-8}{(n+2+log_2(M)+M\cdot(n+1))\cdot 5}$ |

3 cycles are required for the synchronization of the agents and an additional cycle is required for the transmission of each bit in the flit; therefore, for $SBA$ assuming "$M$" agents connected to the bus, we can say that $T_{Cmin} = n+3$ and $L = 1+2\cdot M$.

For $PB$, the implementation is equivalent to the one described for $SB$ but, instead of having one line for the serial data, it requires one line per bit in the flit to transmit in parallel; in this case, $T_{Cmin} = 5$, with four cycles for synchronization and one for the transmission of the flit, and $L = n+3$. $PBA$ is different from $SBA$ because, instead of simply using "request" and "grant" signals in the arbiter, routing and flit transmission are centralized in the arbiter structure to simplify the controllers. Each bus controller share one line per bit in the flit (with a size of "$n$") for the incoming data and two additional control signals: "push", used to signal the receiver controller that the data can be saved and "pop", used to signal the driver controller that the data has been transmitted. Instead of passing the token using only one line, the token is signaled by a binary number of pointing to the ID of the controller which hold it; all the previously described signals are driven by the central arbiter. Additionally each controller must drive $n$ lines to the arbiter with the output data plus one additional line signaling if there are pending flits to send. We have that $L = n+2+log_2(M)+M(n+1)$ and the number of cycles required to send one flit is the same as in $PB$, so $TC_min = 5$. Finally, $\eta_{LC}$ is estimated using Eq. (4).

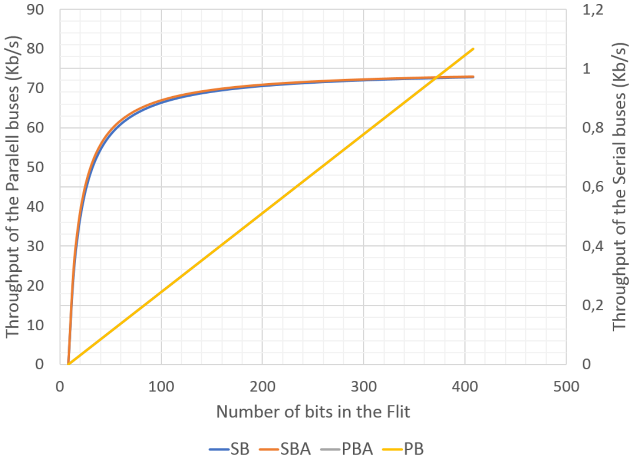

Fig. 2. PB, PBA, SB and SBA bandwidths assuming a 1kHz system clock with 4 agents connected to the bus.

Assuming a 1kHz clock and four agents in the system the transfer rates for the serial and the parallel buses are presented in Fig. 2. We use here 1KHz in order to get a result that may be easily extrapolated. Based on $\eta_{LC}$, one can estimate $R$ for each one of the implementations using

$$R_c \frac{\eta_{LC}\cdot L}{T_{clk}} \tag{5}$$

from where it is easy to show that, in the best of cases, as the size of the flit increases, $R_c$ is constrained for serial buses to the clock frequency of the clock. Meanwhile, for parallel buses, the transmission rate increases linearly with the number of bits in the flit, with a slope of $1/(R_C \cdot T_{CLK})$.

Figure 2 shows that the throughput for both serial buses is equal. Both parallel buses exhibit equal throughput as well. However Fig. 3 shows that $\eta_{LC}$ is better for buses with a distributed arbitration. This because they require fewer routing between agents, while in the cases with central arbiter the additional lines required to communicate the central block with every agent in the system negatively impacts the metric. The proposed metric $\eta_{LC}$ is limited to a finite value as the number of bits in the flit increase. These maximum values can be estimated as:

$$\lim_{n\to\infty} \eta_{LC}(n) \tag{6}$$

with $\eta_{LC}$ converging to $1/(1+2\cdot M)$ for the $SBA$ bus implementation and to $1/3$ for the $SB$, as the number of bits in the flit increases; for the $PBA$ bus, $\eta_{LC}$ grows asymptotically to $1/(5\cdot(1+M))$, while for $PB$ it grows asymptotically to $1/(5)$. Notice that, for buses with central arbiter, $\eta_{LC}$ decays with the number of agents connected, because the number of routed lines increases while, for the distributed implementations, it converges to a constant because of the fixed number of shared lines.

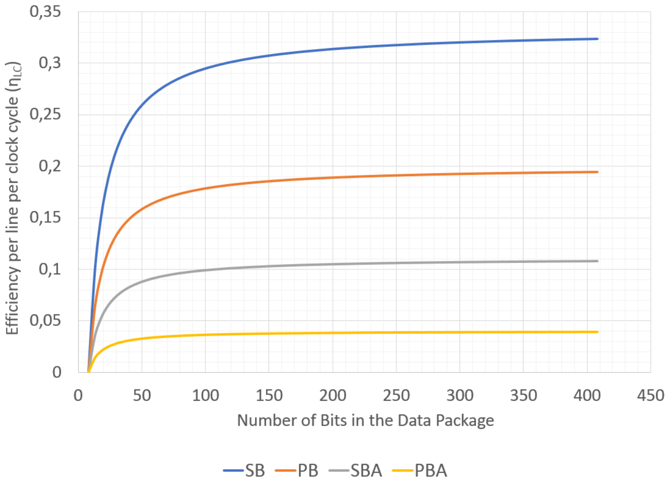

Fig. 3. Comparison of $\eta_{LC}$ for each bus, assuming $M = 4$. Observe how $\eta_{LC}$ converges to $1/(1+2\cdot M)$ for a $SBA$ bus implementation and to $1/4$ for a $SB$, as the number of bits in the flit increase; in the cases of the parallel buses, $\eta_{LC}$ for a $PBA$ converges to $1/(5\cdot(1+M))$ and to $1/(5)$ for a $PB$.

Based on Eq. (1), the $\eta_{LC}$ presented in Table I and the relations postulated for $\alpha$, $\beta$ and $\gamma$, it is possible to infer an equation for the area of each micro-architecture; notice that the routing area $A_R$ depends on the floor-plan of the intended implementation, and its analysis is left as future work. Solving for $n = 32$ and taking into account only the cell placement area, the area for the different buses may be inferred as a function of the number of agents connected to the bus ($M$), such that

$$A_{SB} \approx \frac{2\alpha}{9} \cdot M + \frac{9\beta}{2} \tag{7}$$

$$A_{PB} \approx \frac{24 \cdot \alpha}{175} \cdot M + \frac{175 \cdot \beta}{24} \tag{8}$$

$$A_{SBA} \approx \frac{35 \cdot \beta}{12} \cdot M + \frac{12\alpha}{35} + \frac{35 \cdot \beta}{24} - \frac{12 \cdot \alpha}{70M + 35} \tag{9}$$

$$A_{PBA} \approx \frac{11\beta M}{11} + \frac{8\alpha}{11} + \frac{3\beta}{2} - \frac{96\alpha}{121M + 132} \tag{10}$$

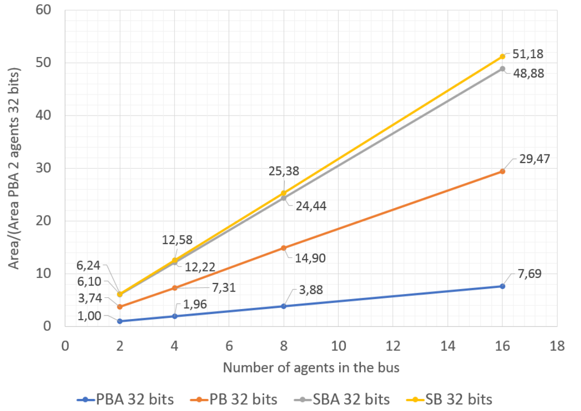

Fig. 4. Comparison of total cell area required for four 32-bit buses (@20MHz clock), using relaxed timing convergence constraints with a 0.18$\mu$m standard cell library. Area of the buses is normalized to the area of a $PBA$ with two agents.

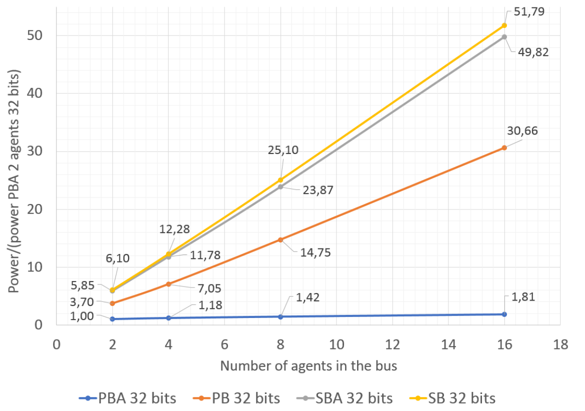

Fig. 5. Comparison of the dynamic power consumption for four buses (@20MHz clock), using relaxed timing convergence constraints with a 0.18$\mu$m standard cell library. Dynamic power consumption of the buses is normalized to the dynamic power of a $PBA$ with two agents.

Depending on the design's fabrication technology, standard cells libraries, clock speed, placement and routing used, the area of the interfaces may change even using the same RTL description. In this work all the buses run at a 20MHz clock, using typical conditions for the standard cells library in power tests. One may assume that all the implementations scale similarly if any of the previously mentioned conditions vary, keeping the results presented in this work relevant. This statement requires nonetheless further exploration.

For the distributed implementations ($SB$ and $PB$), the area equations predict a linear growth of the cell placement area, as the number of agents connected to the bus increases and, in the case of buses with central arbiter, an additional term that is inversely proportional to the number of agents is added to the linear equation. These terms converge to zero as the number of agents increase, so an approximate linear behavior is also expected. Figure 4 compares the total cell area required for different 32-bit buses, with their area normalized to that of the $PBA$ with two agents. Notice how the predicted linear behavior matches the simulated one. Figure 5 compares power consumption. Here, leakage power is not significant and dynamic power is dominant. As expected, there is a close relationship between area and power consumption. The values of $\alpha$ and $\beta$ for area and power approximations may be calculated using Eqs. (7-10) and a linear regression of the simulated data.

## IV. Conclusions

A metric called "Efficiency per line per clock cycle" ($\eta_{LC}$) is presented as the first attempt to provide a semi-analytical approach for pre-silicon evaluation of digital on chip bus micro-architectures. Based on post synthesis simulations, we have found that the proposed metric in terms of area and power consumption offers an early estimate for designers when choosing a particular bus architecture, without the need to have a working RTL design. Further research is required to fully validate the metric using different floor-plans, synthesis algorithms, variations in the number of bits in the flit, and bus protocols.

## References

[1] M. Mitić and M. Stojčev, "An overview of on-chip buses," *Facta universitatis-series: Electronics and Energetics*, vol. 19, no. 3, pp. 405–428, 2006.

[2] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Evaluation of mp-soc interconnect architectures: a case study," in *4th IEEE International Workshop on System-on-Chip for Real-Time Applications*, July 2004, pp. 253–256.

[3] H. Li, X. Liu, W. He, J. Li, and W. Dou, "End-to-end delay analysis in wireless network coding: A network calculus-based approach," in *2011 31st International Conference on Distributed Computing Systems*, June 2011, pp. 47–56.

[4] M. Bakhouya, S. Suboh, J. Gaber, and T. El-Ghazawi, "Analytical modeling and evaluation of on-chip interconnects using network calculus," in *2009 3rd ACM/IEEE International Symposium on Networks-on-Chip*, May 2009, pp. 74–79.

[5] V. Zyuban and P. Strenski, "Unified methodology for resolving power-performance tradeoffs at the microarchitectural and circuit levels," in *Proceedings of the 2002 International Symposium on Low Power Electronics and Design*, ser. ISLPED '02. New York, NY, USA: ACM, 2002, pp. 166–171. [Online]. Available: http://doi.acm.org/10.1145/566408.566451