Power Analysis of System-Level On-Chip Communication Architectures

Kanishka Lahiri and Anand Raghunathan
{klahiri,anand}@nec-labs.com
NEC Laboratories America, Princeton, NJ

ABSTRACT

For complex System-on-chips (SoCs) fabricated in nanometer technologies, the system-level on-chip communication architecture is emerging as a significant source of power consumption. Managing and optimizing this important component of SoC power requires a detailed understanding of the characteristics of its power consumption.

Various power estimation and low-power design techniques have been proposed for the global interconnects that form part of SoC communication architectures (*e.g.*, low-swing buses, bus encoding, *etc*). While effective, they only address a limited part of communication architecture power consumption. A state-of-the-art communication architecture, viewed in its entirety, is quite complex, comprising several components, such as bus interfaces, arbiters, bridges, decoders, and multiplexers, in addition to the global bus lines. Relatively little research has focused on analyzing and comparing the power consumed by different components of the communication architecture.

In this work, we present a systematic evaluation and analysis of the power consumed by a state-of-the-art communication architecture (the AMBA on-chip bus), using a commercial design flow. We focus on developing a quantitative understanding of the relative contributions of different communication architecture components to its power consumption, and the factors on which they depend. We decompose the communication architecture power into power consumed by logic components (such as arbiters, decoders, bus bridges), global bus lines (that carry address, data, and control information), and bus interfaces. We also perform studies that analyze the impact of varying application traffic characteristics, and varying SoC complexity, on communication architecture power. Based on our analyses, we evaluate different techniques for reducing the power consumed by the on-chip communication architecture, and compare their effectiveness in achieving power savings at the system level. In addition to quantitatively reinforcing the view that on-chip communication is an important target for system-level power optimization, our work demonstrates (i) the importance of considering the communication architecture in its entirety, and (ii) the opportunities that exist for power reduction through careful communication architecture design.

Category and Subject Descriptors: C.5.4 [VLSI Systems]
General Terms: Design, Experimentation
Keywords: Communication Architectures, System-on-Chip, Low-Power Design, Power Analysis, Network-on-Chip

1. INTRODUCTION

Increasing System-on-Chip (SoC) complexity, coupled with poor scaling trends of global interconnect, are together making on-chip communication a bottleneck to improving overall system performance and power consumption [1, 2]. As a result of this trend, recent research has focused on analyzing and optimizing the power consumed by global interconnect wires, which represent one part of the on-chip communication architecture. However, a state-of-the-art communication architecture, viewed in its entirety, is significantly more complex, comprising numerous components, such as arbiters, bridges, decoders, data/address multiplexers, control logic, and bus interfaces, in addition to the interconnect wires. The amount of logic in a complex communication architecture can easily rival an embedded processor of moderate complexity. Very little work has addressed analyzing the nature of power consumption in the communication architecture as a whole.

Table 1: Power consumption of SoC components @ 200 Mhz

System Component	Part Name	Power (mW)
Embedded Processor	ARM946E-S[1]	60
Memory Controller	DW_ahb_memctl[2]	29.1
On-Chip Bus	DW_amba[2]	22.6
Cache	ARM946E-S[1]	36
Interrupt Controller	DW_ahb_ictl[2]	2.6
UART	DW_apb_uart[2]	4.1

[1] http://www.arm.com/products/CPUs/ARM946ES.html
[2] Gate-level measurements of Synopsys Designware Library Cores

To highlight the need for studying communication architectures and their power requirements, we compare the power consumed by a typical communication architecture to the power consumed by other system components in Table 1. The table presents data obtained from gate-level power measurements, and manufacturer data sheets, of several commercial SoC components, including a complete communication architecture (the AMBA on-chip bus [3]). The table indicates that the communication architecture can consume significantly more power than many system components, and in fact, its power is comparable in magnitude to well-known primary sources of power consumption (*e.g.*, processors, caches). With increasing system complexity, and shrinking feature size, the relative contribution of the communication architecture can only be expected to increase, motivating the need to better understand and optimize communication architecture power consumption.

1.1 Paper Overview and Contributions

In this paper, we present a systematic evaluation and analysis of the power consumption of a commercial, state-of-the-art, on-chip communication architecture (the AMBA on-chip bus [3]), using a commercial design flow. We focus on developing a quantitative understanding of the relative contributions of different parts of the communication architecture to its overall power consumption, and the factors on which they depend. We decompose the communication architecture power into power consumed by logic components (such as arbiters, decoders, bus bridges), global bus lines (that carry address, data, and control information), and bus interfaces. We also analyze the impact of varying application traffic characteristics, as well as the impact of varying SoC complexity, on the power consumed by the communication architecture.

In addition to quantitatively reinforcing the view that on-chip communication is an important target for system-level power optimization, our studies demonstrate the importance of considering the communication architecture in its entirety (and not just individual components, such as interconnect wires) for power optimization. To support this viewpoint, we consider various alternatives for communication

CODES+ISSS'04, September 8–10, 2004, Stockholm, Sweden.

architecture power reduction, and based on the insights gained from our studies, compare their ability to achieve power savings at the system level.

The rest of this paper is organized as follows. In Section 2, we provide background on the AMBA on-chip bus, and the experimental methodology used. In Section 3, we present a structural breakdown of the power consumed by the AMBA architecture. In Section 4, we analyze the impact of application traffic characteristics, and in Section 5, we analyze the impact of varying system complexity, on AMBA power consumption. In Section 6, we evaluate different techniques for communication architecture power optimization, and conclude in Section 7.

1.2 Related Work

There is a large body of work that addresses power consumption of global bus lines [4, 5, 6, 7, 8, 9, 10], or the interaction of global bus lines with other components, such as processor caches [11, 12]. As mentioned earlier, these techniques do not consider the power consumption of the communication architecture in its entirety. Recognizing this, recent work has addressed communication architecture components other than global wires. Such work includes techniques for analyzing and optimizing the power consumed in on-chip routers [13, 14], and enabling power-conscious routing in complex communication architectures [15]. Many of these techniques are targeted towards on-chip networks, and do not directly apply to bus-based architectures, which are the focus of our work. Integration of power models of certain AMBA components into a transaction-level modeling framework was explored in [16]. In contrast, our work aims at understanding the different sources of power consumption in the AMBA architecture, through detailed (gate-level) power analysis.

To the knowledge of the authors, this paper is the first attempt to experimentally quantify the different factors that contribute to the power consumption of a commercial, bus-based, SoC communication architecture. We chose the AMBA architecture since it is one of the most popular commercial on-chip communication architectures in use today.

2. BACKGROUND

In this section, we first present an overview of the AMBA on-chip bus architecture, highlighting its different components, and their functions. We next describe the methodology used for the experimental studies described later in this paper.

2.1 AMBA On-Chip Bus: Overview

For our studies, we used an implementation of the AMBA on-chip bus that is available in the Synopsys Designware library [17]. Complete details of AMBA functionality may be obtained from [3, 17]. Figure 1 depicts the class of AMBA architectures that we consider. The AMBA architecture is based on a hierarchy of buses. The AHB (Advanced High Performance Bus) integrates high performance components (*e.g.*, processors, DMA controllers). It is a pipelined bus, implying that address and data belonging to different transactions may overlap in time. The bus can support upto 16 masters (components that can initiate bus transactions) and upto 16 slaves (components that only respond to transactions initiated by a master). All the slaves are memory mapped. For each transfer, the decoder generates slave select signals to notify the correct slave. Multiplexers properly route address, write data, and control parameters from the masters to the slaves, as well as slave responses and read data, from the slaves back to the masters. The arbiter regulates access to the shared bus using a configurable arbitration scheme. Burst transactions enable a master to perform a sequence of transfers without requiring arbitration for each transfer. The AHB is connected to the APB (Advanced Peripheral Bus) via the AHB-APB bridge. The bridge behaves as a slave to the AHB, and as a master to the APB. The APB supports only one master, is not pipelined, and

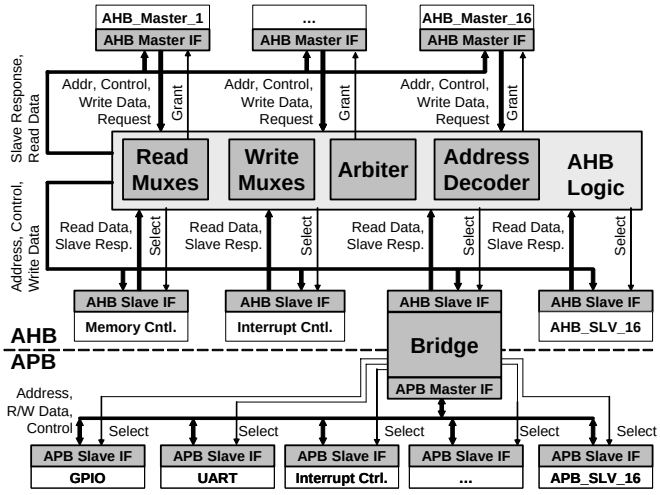

Figure 1: Overview of the AMBA on-chip bus

can integrate upto 16 peripherals (*e.g.*, GPIO interface, UART). Any master/slave that communicates over the AMBA architecture is enhanced with AHB and/or APB interfaces. Finally, the physical connectivity between components is achieved using global interconnect wires, that carry address, data and control values between masters and slaves.

2.2 Power Estimation Methodology

The power estimation methodology used in this work is illustrated in Figure 2. Architectural configuration parameters of the AMBA bus (*e.g.*, address width, data width, number of masters, number of slaves) are specified to Synopsys CoreTools [18], which generates synthesizable RT-level descriptions of the bus hardware. In our experiments, while the number of masters and slaves vary, the address and data widths are held constant at typical values (32-bit addresses, 32-bit AHB data width, and 16-bit APB data width). The arbiter is configured to use static priorities, a scheme wherein access rights are determined by examining the set of pending requests from masters, and selecting the one with the highest (fixed) priority. Synopsys Design Compiler [19] is used to synthesize the bus hardware to the gate level, using CB-12L, NEC's low-power 0.15 μm standard cell library [20]. Vectors to drive power estimation are obtained via RT/gate level simulations using the Modelsim simulator [21]. The systems into which the AMBA bus is integrated consist of a variable number of master and slave components, and numerous peripherals, such as a memory controller, interrupt controllers, *etc*. Transaction-level testbenches are constructed using programmable

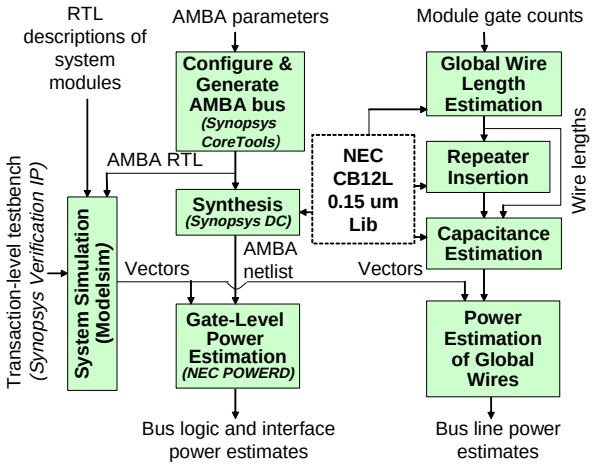

Figure 2: Methodology for communication architecture power estimation

functional models of masters for bus traffic generation. Power simulation vectors are provided to POWERD, NEC's gate-level power estimator [22] to obtain power measurements for different components. Pre-layout wire length estimation techniques are used to compute global wire lengths [23, 24], which take into account area estimates and pin counts for each system component. Wire capacitances are extracted from the technology library, and the number of repeaters and their sizes are calculated using a delay-optimal algorithm described in [25]. The capacitances are used to calibrate a transition-count based power model for estimating the power consumed by bus lines [11].

3. STRUCTURAL BREAKDOWN OF AMBA POWER

In this section, we present a relative comparison of the power consumed by different parts of the AMBA on-chip bus. For this study, we used a configuration of the AMBA bus that integrates 2 AHB masters, 2 AHB slaves (a memory controller and an interrupt controller), an AHB-APB bridge, and three APB peripherals, including a General Purpose Input/Output (GPIO) device, an interrupt controller, and an I2C (external bus) interface. The testbench consists of an even mix of different types of transactions, including burst transactions on the AHB, and regular transactions between AHB masters and APB slaves. Reads and writes are equally distributed, and addresses are generated uniformly at random, while staying within valid regions of the memory address map. The computed global wirelengths lie between 1.5 and 3 mm. The total power consumed by the communication architecture for this study was 12 mW, and the breakdown of the power among the various components of the AMBA bus is presented in Figure 3. The components whose power is shown in the figure include the AHB arbiter, multiplexers, and address decoder, the AHB-APB bridge (which also implements the APB bus logic), the AHB and APB address, data, and control bus lines, two AHB master interfaces, two AHB slave interfaces (corresponding to the memory controller and the interrupt controller), and three APB slave interfaces (corresponding to the GPIO, interrupt controller, and I2C peripherals).

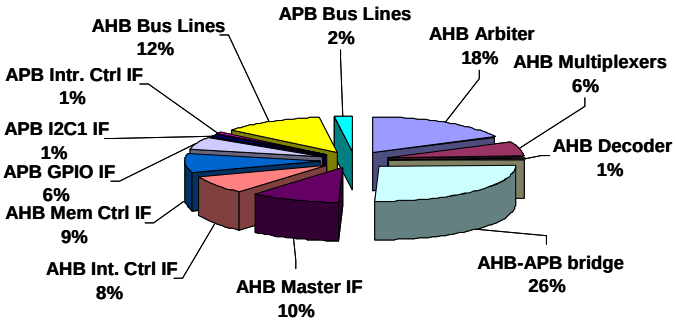

Figure 3: Breakdown of power in the AMBA on-chip bus

From Figure 3, we observe that the total power consumed by the communication architecture is not dominated by any single component, but has several important contributors. We next group together some of the components in order to gain a better understanding of the relative contributions to the overall power consumption.

- The power consumed by the bus lines (considering both the AHB and APB together) represents 14% of the total power consumed by the communication architecture. The power consumed on the APB bus lines is relatively small since (a) it operates at half the clock frequency, (b) it is not pipelined, and (c) it has a narrow data width (16-bit). Also, all APB transfers are associated with AHB transfers, but not vice versa.
- In contrast, the total power consumed by the logic components (arbiter, multiplexers, decoder, and bridge) constitutes about 51% of the total communication architecture power.
- The power consumed by bus interfaces (master and slave) constitutes the remaining 35%.

In summary, our analysis demonstrates that while a large portion of the power consumed by the communication architecture is dominated by logic (86%), it is spread across numerous components. Hence, in order to address communication architecture power consumption, the contributions of several different components needs to be addressed. As discussed in detail later in this paper, approaches that focus on specific components in isolation (*e.g.*, global wires) may have limited impact at the system level. In the next two sections, we consider different factors on which the communication architecture power depends. We examine the impact of variable traffic characteristics, and variable system complexity, on the power consumed by the AMBA on-chip bus.

4. IMPACT OF ON-CHIP COMMUNICATION TRAFFIC CHARACTERISTICS

In this section, we analyze how application traffic characteristics influence communication architecture power consumption. In the following subsections, we consider the quantitative impact of variations in (i) spatial distribution of communication transactions, (ii) transaction pipelining, and (iii) different transaction types, on communication architecture power.

4.1 Spatial Distribution of Transactions

Since all the devices attached to the AMBA bus are memory mapped, the spatial distribution of communication transactions is determined by the address values generated by each master. For this study, a set of test programs were written for an AHB master to generate single (non-burst) write transfers (NONSEQ transfers, in AMBA terminology) to each of a set of AHB/APB slaves, choosing addresses at random from within the address space of each slave. In each case, the power consumed by the entire communication architecture was measured. Figure 4 presents a few illustrative cases. Bars 1 and 2 correspond to traffic directed towards APB slaves (the GPIO Interface and the APB Interrupt Controller, respectively), while bars 3 and 4 correspond to traffic directed towards AHB slaves (the AHB Interrupt Controller and Memory Controller, respectively).

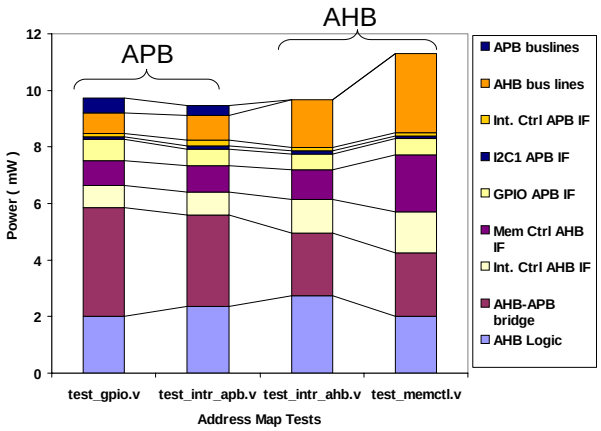

Figure 4: Comparison of communication architecture power consumption for different target slaves

From the figure, we observe that with variation in the address values (or target slaves), the total power consumed by the communication architecture varies appreciably (upto 19%), but more importantly, the relative contributions of each component varies even more

significantly. We next discuss the power consumed by each component in further detail.

Bus Lines: The power consumed by the AHB bus lines varies significantly (3.8X) across the 4 tests. In bar 4, its contribution is the largest, owing to the larger size of the address map of the memory controller (which results in higher switching activity on the address bus). The power consumed by the AHB bus lines during APB transfers (bars 1 and 2) is, on average, 2.3X less than during pure AHB transfers (bars 3 and 4). This may seem counter-intuitive, since all APB traffic also goes through the AHB bus. The reason behind this is that APB slaves are only 16 bits wide, causing the more significant AHB data bus lines to remain idle during APB transfers. The power consumed by the APB lines, while only 4% in the first two cases, is zero in the latter two cases, due to the absence of APB traffic.

Bus Bridge: The contribution of the AHB-APB bridge varies significantly across the different cases, from 22% (2.2 mW in bar 3), upto 40% (3.85 mW in bar 1). In bars 3 and 4, the bridge behaves as an AHB slave that merely observes AHB traffic, but does not respond to it. Hence, the APB related logic (decoding, multiplexing, mastering) remains idle. In the first two cases, however, the bridge's AHB slave interface is always selected, which results in activation of the bridging and APB mastering logic, causing higher power consumption.

Slave Interfaces: The contribution of the slave interfaces varies, depending on whether or not the corresponding device is selected. For example, the power consumed in the memory controller's AHB interface varies from 18% (2.02 mW in bar 4) to 9% (0.87 mW in bar 1). It should be noted that even when a slave device is not selected, the power contribution of its bus interface is appreciable. For example, when all the traffic is directed towards APB slaves, the power consumed by AHB slave interfaces (excluding the bridge) is as high as 18%. In addition, the power consumed by a slave interface can vary significantly, even when the slave is not selected. For example, in bars 1 and 4, the AHB Interrupt Controller interface is never selected, but a variation of as much as 1.8X is observed. In fact, its power consumption in bar 4 is 22% higher than in bar 3 (where the AHB Interrupt Controller is selected). This apparent anomaly is attributed to the dependence of the slave interface power on the switching activity of the address and data values observed at its inputs, even when it is not selected. The high switching activity is also borne out by the significantly higher power consumption on the AHB bus lines in bar 4 as compared to bar 3.

AHB Bus Logic: In these experiments, the variation in the power consumed by the AHB logic is less significant. This is because the AHB logic power is in large part dominated by the arbiter, which is not affected by variations in slave access profiles.

4.2 Pipelined Transactions

AHB transfers are split into an address phase, in which address and control signals are asserted by the master, and a data phase, in which data is written to (or read from) a slave device. The AHB supports pipelined transfers by allowing the address phase of one transfer to overlap with the data phase of the previous transfer. This enables the initiation and completion of one transfer per clock cycle.

In this study, we compare the power consumed by transactions that exploit the AMBA architecture's pipelining capability, versus transactions that execute sequentially (without pipelining). Intuitively, we expect the pipelined case to consume more power (owing to the processing of multiple transactions simultaneously). We study the power consumed when transferring a 2 KB block of data from a master to a slave under two scenarios. In the first case, single word transfers are initiated every clock cycle, in a pipelined fashion. In this case, each transaction takes two cycles to complete. In the second case, each transaction is allowed to complete before initiating a new transaction. In this case, each transaction takes four cycles to complete (request, grant, address, and data phases each consume one cycle). For this discussion, we focus on the AHB subsystem, since the APB does not support pipelined transactions.

The results of this experiment are depicted in bars 1 and 2 of Figure 5. We observe that the power consumed by the non-pipelined traffic is 1.5X *higher* than the power consumed by pipelined traffic. The higher power dissipation is reflected in almost all the components: the bus lines (2X), multiplexers (4X), and arbiter (1.5X). Upon further analysis, we found three reasons that explain this result.

In the non-pipelined case, there are many unutilized bus cycles, during which the bus logic drives the address and data buses to zero, resulting in relatively high switching activity. In the pipelined case however, every cycle is utilized and hence, successive values on the bus are correlated, resulting in lower switching activity. In addition, in the non-pipelined case, global control lines between the master and the bus logic undergo frequent switching activity, owing to assertion/de-assertion of requests, grants, and transaction parameter values. In the pipelined case, these lines hold their values, since on every clock cycle, a new transaction starts with identical parameters. While the above arguments explain the disparity in the power consumption of the bus lines and multiplexers, the difference in the power consumed by the arbiter requires further explanation. In the non-pipelined case, upon successful transaction initiation, the request lines are de-asserted and are then re-asserted for the next transaction. This causes the inputs to the arbiter to change frequently, resulting in higher power consumed by the arbiter, as it computes which master should be granted next. In the pipelined case, the inputs to the arbiter rarely change, leading to lower power consumption.

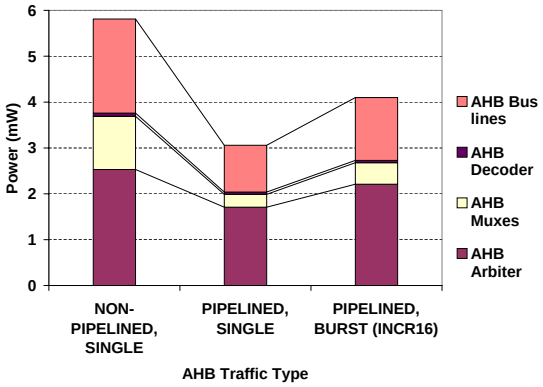

Figure 5: Comparison of power consumption of different types of communication traffic

4.3 Burst Transactions

The AHB supports the use of different transaction types. These include *single* transactions (denoted by NONSEQ), wherein the master requests access for the bus to carry out a single transfer (8, 16 or 32-bit), and a variety of *burst* transactions, wherein the master is granted the bus for a sequence of transfers. For each transaction, the type is controlled by the HBURST parameter, whose value determines the burst length (4, 8, 16 or undefined), and whether or not the burst wraps at certain address boundaries. In this study, we repeat the experiment described in the previous subsection, but use non-wrapping, 16-word bursts (denoted by INCR16) to transfer the 2 KB block of data. We compare the power consumed during this transfer with pipelined, single transactions. Note that, within a burst, the transactions are always pipelined.

Figure 5 presents the results for the burst case in bar 3, alongside the results described in the previous subsection. From the figure, we observe that the power consumed during the burst transfer is 34% higher than the power consumed during pipelined, single transfers. This is due to a 29% increase in arbiter power, a 68% increase in multiplexer power, and a 34% increase in bus line power.

There are two factors that influence the difference in power consumption between these cases. The first factor is related to the additional functionality executed by the arbiter during a burst transaction, which keeps track of information regarding the progress of the burst,

such as a count of the number of words transfered. This state is maintained by the arbiter to ensure that the burst can be restarted in case it has to be terminated by some external event. The second factor (which influences both arbiter and bus line power) is the manner in which bus control signals are asserted in the two cases. In the first case (pipelined, single transfers), control lines and request/grant values remain unchanged throughout the experiment. In the burst transfer case, there is activity on the control lines within bursts (to indicate start, progress, and end of a burst), and between bursts (assertion/deassertion of control signals at the start/end of each burst). This causes higher power consumption in the control lines and arbiter.

5. IMPACT OF SYSTEM COMPLEXITY

We next analyze how the power consumption of the communication architecture varies with system complexity. For this study, we scale system complexity along two natural dimensions from the communication architecture standpoint, namely, the number of masters and the number of slaves that are connected to it. For each system , we apply the methodology of Figure 2 to generate the communication architecture hardware and analyze its power consumption. For fair comparison, all the testbenches in this study generate traffic consisting of pipelined, single, write transactions, with contiguous addresses, uniformly distributed data values, and no wasted bus cycles. For systems with multiple masters, testbenches were designed to create maximum contention for access to the bus.

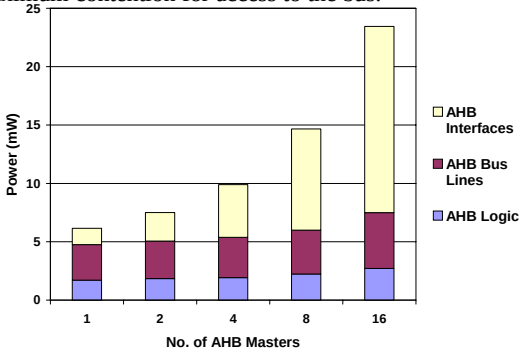

Figure 6: Communication architecture power consumption versus increasing system complexity

The results of these studies for increasing number of masters are presented in Figure 6. For brevity, we omit the results for varying number of slaves, which demonstrate similar trends. From the figure, we observe that the power consumed by the interface logic exhibits the most significant variation with system complexity, scaling proportionally to the number of components. The relative contribution of the interfaces increases from 29% of the total for the 1 master system, to 67% for the 16 master system. While the power consumed by the bus lines also increases with increasing system complexity (by 57%), owing to longer wire lengths and larger number of repeaters, their relative contribution to the total *decreases* from 49% (in the 1 master system), to 19% (in the 16 master system). The impact on bus logic was observed to differ, depending on whether system complexity is scaled in terms of the number of masters or slaves. While increasing the number of slaves has little impact on the bus logic power, increasing the number of masters results in a 59% increase (from 1.7 mW to 2.71 mW), owing to increased arbitration and multiplexer complexity.

6. COMMUNICATION ARCHITECTURE POWER REDUCTION TECHNIQUES

Having presented studies that analyze the sources of power consumption in a communication architecture, we next discuss opportunities for power reduction. We consider a suite of techniques for optimizing communication architecture power: bus encoding, segmented bus design, interface power management, and traffic sequencing. The purpose of this discussion is to obtain an understanding of which portions of the communication architecture are addressed by each technique, and to quanitfy the impact of these techniques on communication architecture power.

We consider an example in which two masters communicate with the AHB memory controller in an interleaved manner. In addition, one of them executes a sequence of transactions with the APB GPIO peripheral. The power breakdown corresponding to this case is shown in bar 1 of Figure 7. In the following subsections, we use this as a base case for comparison, and analyze each of the above techniques in terms of the power reductions they achieve.

6.1 Bus Encoding

Numerous encoding schemes have been proposed to reduce switching activity on global buses (*e.g.*, [5, 6, 9, 26]). While the impact of bus encoding on the power consumed by bus lines has been well documented, here we are interested in its impact on the power consumption of the communication architecture as a whole. In order to evaluate the savings via bus encoding on the AMBA on-chip bus, we consider an scheme that reduces address bus transitions by 60%, and data bus transitions by 20%. Note that, a simple scheme, in which the encoder/decoder circuitry resides between the bus interfaces and the bus lines affects only the bus line power. In a more sophisticated scheme, the codecs are implemented within each bus interface. In such a scheme, since the signals received/transmitted by the bus interface are encoded, there is reduced activity within each bus interface, leading to additional power savings.

We analyzed the data collected for the example AMBA-based system, and estimated the potential power savings achievable by the latter scheme. From bar 2 of Figure 7, we observe that while the technique achieves a 30% reduction in the power consumed by bus lines, the overall communication architecture power reduction is only 14%. The impact is relatively small, owing to communication architecture components whose power is unaffected by bus encoding, such as the control lines, bus arbiter, bus bridge, and idle bus interfaces.

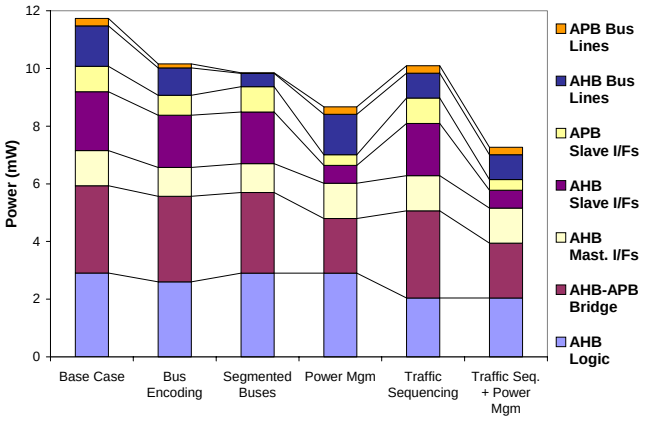

Figure 7: Comparison of different approaches for communication architecture power reduction

6.2 Segmented Buses

One of the causes of power dissipation in the communication architecture is the broadcast nature of address and data values to all the slaves attached to the bus, which results in unnecessary switching in long capacitive bus lines and slave interfaces. A solution to this problem is to use a segmented bus architecture [7, 8]. In this architecture, long nets (such as the address bus) that fan out to multiple components, are split into segments, with neighboring segments connected by control logic (*e.g.*, pass transistors). The slave select logic (implemented in the bus address decoder) can be enhanced to activate only those segments that are required for each transaction, thereby propa-

gating the address/data values to only the relevant slave. While this approach can achieve power savings, it may result in increased delay on the bus lines, due to the additional control logic. The approach could also be applied to regulate activation of bus segments when slave responses and read data are propagated back to the masters.

We evaluated the potential impact of applying such a strategy on the slave side, by analyzing the data collected from the example AMBA-based system. Bar 3 of Figure 7 indicates that using a segmented bus, an overall savings of 16% are achieved. This is due to 70% savings in bus line power, and 11% savings in interface power. The impact on interface power (and hence on the overall power) is less significant, due to the relatively high power consumed by the interfaces even when there is no activity on their inputs.

6.3 Power Management

Several power management techniques lend themselves naturally to reducing communication architecture power. For example, idle slave interfaces could be shut down, to be woken up on-demand, by the bus logic (arbiter). However, slave wake-up latencies may subject masters to additional delay before receiving grants. While time-out based techniques may be effective, we observe that the communication architecture could make better informed power management decisions (when to shut down/wake up components), based on system-level information. For example, the bus arbiter may be capable of predicting idle times for slaves, based on information that it possesses about the set of current and pending transactions. During the progress of a fixed length, locked burst, it might conclude that all the slaves, except the one participating in the burst, are likely to remain idle for a certain number of cycles; hence it can save power by shutting them down. The notion of shutting down interfaces can be applied to entire levels of the bus hierarchy. For example, if the AHB slave interface of the AHB-APB bridge is inactive for prolonged periods, the APB subsystem may be shutdown.

An analysis of the potential savings of interface and hierarchy shutdown for the example AMBA-based system reveals a power reduction of 26% (bar 4 of Figure 7), due to savings in slave interfaces and the bridge. Significantly larger savings may be achievable for systems comprising many more slaves and/or hierarchy levels.

6.4 Traffic Sequencing

In our experiments, we observed that the interleaving of transactions from different masters leads to increased power consumption due to more frequent arbitration, and increased switching activity on the bus lines. Traffic sequencing is a proposed approach that aims to reduce the extent to which multiple masters obtain interleaved access to the bus. This could be achieved either by modifying the application itself, or by employing a more sophisticated bus protocol in which masters are provided with feedback regarding the current state of the bus. For example, if the bus is busy executing a sequence of pipelined transfers, a competing master could "back-off" and retry at a later time. We analyzed the advantage that such a technique might provide (bar 5, Figure 7) for our AMBA-based system. While the overall savings is only 14%, it is worth noting that this is the only technique that significantly reduces the AHB logic power (30%).

6.5 Summary

The above analysis suggests that, in practice, none of the described techniques may be individually capable of achieving large power savings for the entire communication architecture. This is because the different techniques, while effective in terms of reducing the power consumption of certain components, may have little or no impact on the rest of the communication architecture. However, we observe that most of the techniques are complementary. For example, an analysis of the benefits of combining traffic sequencing with power management indicates that almost 40% savings may be achievable (bar 6 of Figure 7). Our studies show that significant opportunities for power reduction exist, through appropriate combination of strategies that address the power consumption of different parts of the communication architecture.

7. CONCLUSIONS

This work demonstrated that communication architecture power is a subject that needs to be addressed through careful consideration of all its constituent components. We presented a first-of-its-kind study to analyze the power consumed by a commercial SoC communication architecture, and quantitatively analyzed the various factors on which it depends. Based on the insights gained, we evaluated a suite of power reduction techniques, and discussed their effectiveness in optimizing communication architecture power.

8. REFERENCES

[1] R. Ho, K. W. Mai, and M. A. Horowitz, "The Future of Wires," *Proc. IEEE*, vol. 89, pp. 490–504, Apr. 2001.

[2] D. Sylvester and K. Keutzer, "A Global Wiring Paradigm for Deep Submicron Design," *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 242–252, Feb. 2000.

[3] "AMBA 2.0 Specification." `http://www.arm.com/armtech/AMBA`.

[4] P. P. Sotiriadis and A. P. Chandrakasan, "A Bus Energy Model for Deep Submicron Technology," *IEEE Trans. VLSI Systems*, vol. 10, pp. 341–350, June 2002.

[5] M. R. Stan and W. P. Burleson, "Bus Invert Coding for Low Power I/O," *IEEE Trans. VLSI Systems*, vol. 3, pp. 49–58, Mar. 1995.

[6] L. Benini, A. Macii, M. Poncino, and R. Scarsi, "Architectures and Synthesis Algorithms for Power-efficient Bus Interfaces," *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 969–980, Sept. 2000.

[7] C.-T. Hsieh and M. Pedram, "Architectural Power Optimization by Bus Splitting," *IEEE Trans. Computer-Aided Design*, vol. 21, pp. 408–414, Apr. 2002.

[8] J. Y. Chen, W. B. Jone, J. S. Wang, H. I. Lu, and T. F. Chen, "Segmented Bus Design for Low Power," *IEEE Trans. VLSI Systems*, vol. 7, pp. 25–29, Mar. 1999.

[9] T. Lv, J. Henkel, H. Lekatsas, and W. Wolf, "A Dictionary-based En/decoding Scheme for Low-power Data buses," *IEEE Trans. VLSI Systems*, vol. 11, pp. 943–951, Oct. 2003.

[10] S. Osbourne, A. T. Erdogan, T. Arlsan, and D. Robinson, "Bus Encoding Architecture for Low-power Implementation of an AMBA-based SoC Platform," in *Proc. IEE*, pp. 152–156, July 2002.

[11] T. D. Givargis, F. Vahid, and J. Henkel, "Evaluating Power Consumption of Parameterized Cache and Bus Architectures in System-on-a-Chip Designs," *IEEE Trans. VLSI Systems*, vol. 9, pp. 500–508, Aug. 2001.

[12] W. Fornaciari, D. Sciuto, and C. Silvano, "Power Estimation for Architectural Exploration of HW/SW Communication on System-Level Buses," in *Proc. Int. Symp. on HW/SW Codesign*, pp. 152–256, May 1999.

[13] T. T. Ye, L. Benini, and G. D. Micheli, "Analysis of Power Consumption on Switch Fabrics in Network Routers," in *Proc. Design Automation Conf.*, pp. 524–529, June 2002.

[14] H. Wang, X. Zhu, L. S. Peh, and S. Malik, "Orion: A Power-Performance Simulator for Interconnection Networks," in *Int. Symp. on Microarchitecture*, pp. 294–305, Nov. 2002.

[15] J. Hu and R. Marculescu, "Exploiting the Routing Flexibility for Energy/Performance Aware Mapping of Regular NoC Architectures," in *Proc. Design Automation & Test Europe (DATE) Conf.*, Feb. 2004.

[16] M. Caldari, M. Conti, M. Coppola, P. Crippa, S. Orcioni, L. Pieralisi, and C. Turchetti, "System-Level Power Analysis Methodology Applied to the AMBA AHB Bus [SoC Applications]," in *Proc. Design Automation & Test Europe (DATE) Conf.*, pp. 32–37, Mar. 2003.

[17] "Designware Intellectual Property, Synopsys Inc." `http://www.synopsys.com/products/designware/`.

[18] "Designware IP Reuse Tools, Synopsys Inc." `http://www.synopsys.com/products/designware/ipreuse_tools.html`.

[19] "RTL Synthesis, Synopsys Inc." `http://www.synopsys.com/products/logic/`.

[20] "CB-12 Family L/M/H Type for Cell Based ICs." `http://www.necel.com/cbic/en/product/cb12.html`.

[21] "Modelsim 5.7e." `http://www.model.com`.

[22] *NEC OpenCAD V 5.2 Users Manual*. NEC Electronics, Inc., Jan. 1999.

[23] D. Sylvester and C. Hu, "Analytical Modeling and Characterization of Deep-Submicrometer Interconnect," *Proc. IEEE*, vol. 89, pp. 634–664, May 2001.

[24] W. E. Donath, "Placement and Average Interconnection Lengths for Computer Logic," *IEEE Trans. Circuits and Systems*, vol. 26, pp. 272–277, Apr. 1979.

[25] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley, Menlo Park, CA, 1990.

[26] P. P. Sotiriadis and A. P. Chandrakasan, "Bus Energy Minimization by Transition Pattern Coding (TPC) in Deep Sub-Micron Technologies," in *Proc. Int. Conf. Computer-Aided Design*, pp. 322–327, Nov. 2000.