Energy models for network-on-chip components

Bhat, S.

Award date:
2006

TECHNISCHE UNIVERSITEIT EINDHOVEN
Department of Mathematics and Computer Science

ENERGY MODELS FOR NETWORK-ON-CHIP COMPONENTS

By
Shubha Bhat

Supervisors:
Dr. Ir. Twan Basten
Dr. Ir. Marc Geilen
Ir. Sander Stuijk

Eindhoven, Dec 2005

ACKNOWLEDGEMENTS

Foremost, I would like to thank my supervisor Dr. Ir. Twan Basten, Associate Professor at the Electrical Engineering department of the Technical University Eindhoven, Netherlands, for providing me with an opportunity to carry out my master thesis project at the department of electrical engineering at the university. I thank him for his support and encouragement throughout the project.

I also thank Dr. Ir. Marc Geilen, Assistant Professor at the Electrical Engineering department of the Technical University Eindhoven, Netherlands, for his valuable suggestions and ideas regarding my project during the discussions in the PROMES meetings.

I am especially grateful to Ir. Sander Stuijk, who guided me throughout the project and cleared all my doubts and questions patiently.

I would also like to thank all the PROMES group members, Dr. Ir. Bart Theelen, Amir Hossein Ghamarian, M.Sc. and Dr. Mohammad Mousavi, M.Sc. for their valuable suggestions throughout the project.

I thank my husband Aravind Bhat and our families back in India for giving me the support and encouragement throughout my study period.

Finally, I would like to thank all my colleagues at the electrical department and my friends for their love and support.

A. ABSTRACT

Today with the advent of new VLSI processing technologies, System-on-Chip (SoC) design is gaining prominence in order to achieve faster time to market, reduced costs and flexible solutions. Present day embedded multimedia applications are becoming more computation intensive due to the large number of integrated functions .Often such applications are mapped onto mobile systems that need to operate with a low energy consumption.

It is seen that multiprocessor Systems-on-Chip (MPSoC) offer a superior performance and lower energy consumption than single processor systems. Networks-on-Chip (NoCs) are considered as an interconnection mechanism between the various IP blocks in an MPSoC. NoCs are preferred over traditional buses, as NoCs offer predictable timing behaviour and can easily scale without degrading the performance.

As modern systems are often mobile, energy consumption is an important design criterion. Therefore accurate estimates of the energy consumption of the implementation must be made early in the design process. This requires an energy model for various system components. Current state of the art energy estimation flows for MPSoCs take a path all the way to the low implementation levels taking into account all these low level details. During design-space exploration (DSE), it would be difficult to follow the entire time consuming flow for each potential solution. Hence, it is important to have abstract energy models of various components in the system and the system as a whole.

In this thesis, we focus on the NoC interconnect. We present energy models at an intermediate abstraction level for all NoC components in a typical SoC, namely links, FIFO buffers and routers and show how these models can be used to estimate the energy consumption of a complete NoC. The energy model of an individual component is expressed as a simple parameterized expression. These models are validated comprehensively by means of simulations carried out in MAGMA. The models can be used to compute the overall energy consumption for the communication part of the chip without carrying out time-consuming simulations. The abstraction level of the energy models is such that only a few parameter values need to be known. Examples are the frequency of changes in the incoming data and read and write rates. These parameters are application dependent and are less easily influenced by the designer. Also some of the architectural parameters such as FIFO sizes, flit size, number of input and output ports etc, that can be chosen by the designer, can be taken into account. Thus, the models are at a higher abstraction level than the gate and transistor level models used in MAGMA.

Our experiments on FIFO buffers exemplify that the energy consumption increases linearly with the number of FIFO places. Our results on router energy consumption show that the arbiter and the crossbar consume similar amounts of energy in the router. Furthermore, it is also seen that the energy consumed by the link doubles with doubling the link length. Our experiments suggest that, for a given fixed or average switching activity in the transferred data, the energy consumption is linear in the number of hops in the network, which is the commonly used abstract energy model for NoCs.

The work started in this thesis ultimately provides designers with concrete numbers concerning the power and energy consumption of SoCs enabling them to optimize their designs and perform trade-offs and take decisions through the predictions made based on the developed models preferably at an early stage in the design of a system.

B. TABLE OF CONTENTS

C. ABBREVIATIONS

ATM	Asynchronous Transfer Mode
DRAM	Dynamic Random Access Memory
CMOS	Complementary Metal Oxide Semiconductor
CA	Communication Assist
DSE	Design Space Exploration
FIFO	First in First out
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
IP	Intellectual Property
ITRS	International Technical Roadmap for Semiconductors
MPSoC	Multiprocessor System-on-chip
MSE	Mean Square Error
NoC	Network-on-chip
NMOS	N-Channel Metal Oxide Semiconductor
OSI	Open System Interconnect
PMOS	P-Channel Metal Oxide Semiconductor
QoS	Quality of Service
RTL	Register Transfer Level
SoC	System-on-chip
SRAM	Static Random Access Memory
TDMA	Time Division Multiple Access
VCD	Value Change Dump

D. LIST OF FIGURES

E. LIST OF TABLES

F. LIST OF PARAMETERS

C	:	Load Capacitance
V	:	Supply Voltage
f	:	Operating frequency
α_L	:	Link/wire switching activity, i.e., the number of 0 to 1 and 1 to 0 transitions per time unit, normalized by the clock frequency
α_C	:	Cross-coupling switching activity
I_{short}	:	Short circuit current
τ	:	Short circuit period
$I_{bias,\, wire}$	:	Reverse-bias current
$I_{leak,\, gate}$	:	Leakage current
N	:	Number of wires in a link
n_r	:	Number of routers
P_{Link}	:	Link Power
α_F	:	bit flips in FIFO data
r	:	Rate of read and write actions
n	:	Number of FIFO places

Chapter 1 Introduction

1.1. Problem Definition

With technology scaling downwards, SoCs are getting more complex with additional power, area and speed constraints incorporating several processing components in a single chip. Today embedded multimedia applications are becoming more computation intensive due to the large number of functions incorporated within them. Often such applications are mapped onto mobile systems that are typically battery operated and support a wide range of applications so they have to be flexible as well as energy-efficient.

It is seen that multiprocessor Systems-on-Chip (MPSoC) offer a superior performance and lower energy consumption than single processor systems. A certain computational task accomplished by a single processor at a certain frequency can be achieved by multiprocessors in parallel with reduced frequency and voltage and thereby reduced energy consumption at the same amount of time. Figure 1 depicts a tile-based MPSoC. A tile normally is composed of a processor, a memory unit, communication assist and a network interface. A communication assist is a small controller that performs buffer accesses on behalf of the network. Most importantly, it decouples computation from communication.

For such systems, energy consumption is an important design criterion. Hence accurate estimates of the energy consumption of the implementation must be made early in the design process. This requires an energy model for various components in the system.

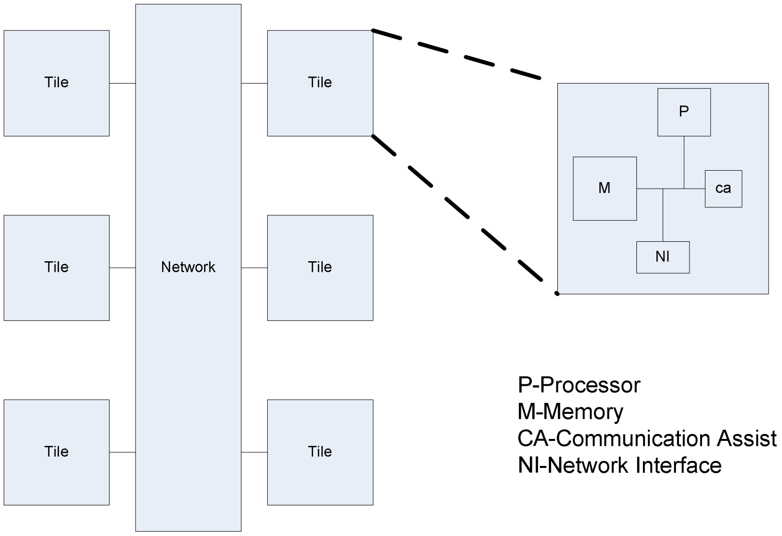


Figure 1: Architectural template of a Multiprocessor SoC

Total energy consumption is basically composed of two main components namely, the dynamic and the static or the leakage power. It is assumed that the leakage power will dominate the dynamic power consumption for technologies below 90nm (ITRS Roadmap, 2001-2002, Figure 5). Additionally, current applications are becoming more computation intensive and therefore more functions are added to the current systems. These increased computational requirements have led to the integration of multiple components on a chip operating at higher clock frequencies which has consequently resulted in the increase of overall energy consumption. Nevertheless, irrespective of the technology, it is necessary to have a prior knowledge of the total energy consumption on a chip.

This information can be used during the design to minimise energy consumption or to make trade-offs with other aspects. Information about energy consumption needs to be available in the form of parameterised models. Such models provide a basis to optimize energy consumption of SoC components and to set the parameters of the design of a SoC.

Further, the current state of the art flow takes a path all the way to the low implementation levels taking into account all the low level details. During DSE, it wouldn't be ideal to follow the entire flow for each instance. Hence, it is important to have abstract but accurate energy models for the various components in the system.

1.2. Energy vs. Power consumption

The term 'Power consumption' refers to the rate at which energy is consumed over time.. The term 'Energy consumption' is often of interest in the case of portable devices operated by battery. In such devices, the amount of energy needed to perform a computation is a more useful measure than the power consumption.

Consider two cases, one with high power consumption and another with low power consumption. Both these approaches require the same amount of energy to complete some operation. The solution requiring high power would simply be faster than the other.

Hence in this thesis, when referring to power consumption, it means power consumed at one moment in time, whereas energy consumption means the energy consumed over a period of time, typically for some given task or operation.

1.3. Networks-on-chip

A Network-on-chip (NoC) is an efficient on-chip communication architecture for System-on-Chip architectures. It enables integration of a large number of computational and storage blocks on a single chip. Traditionally communication between processing elements was based on buses. But for large multiprocessor SoCs with many processing elements, it is expected that the bus will become a bottleneck from a performance, scalability and energy point of view [41][43]. Therefore the idea of networks on chip has evolved which consists of a set of routers interconnected by links. Figure 2 shows a typical 3x3 mesh topology with each network interface connecting one IP to a router.

End to end communication between IP blocks is accomplished by the exchange of messages. These messages are further broken down into packets. Packets are the standard form of representing the information for communication. Packets are split into several flits, which is the basic unit of data transfer in a NoC.

There are two main parts of the interconnection network, the services and the communication system. A typical service is a bidirectional communication channel with reserved bandwidth that offer throughput and latency guarantees. The communication system is what supports the transfer of information from source to destination in the NoC structure or the network topology. A routing algorithm determines the route of a packet from source to destination.

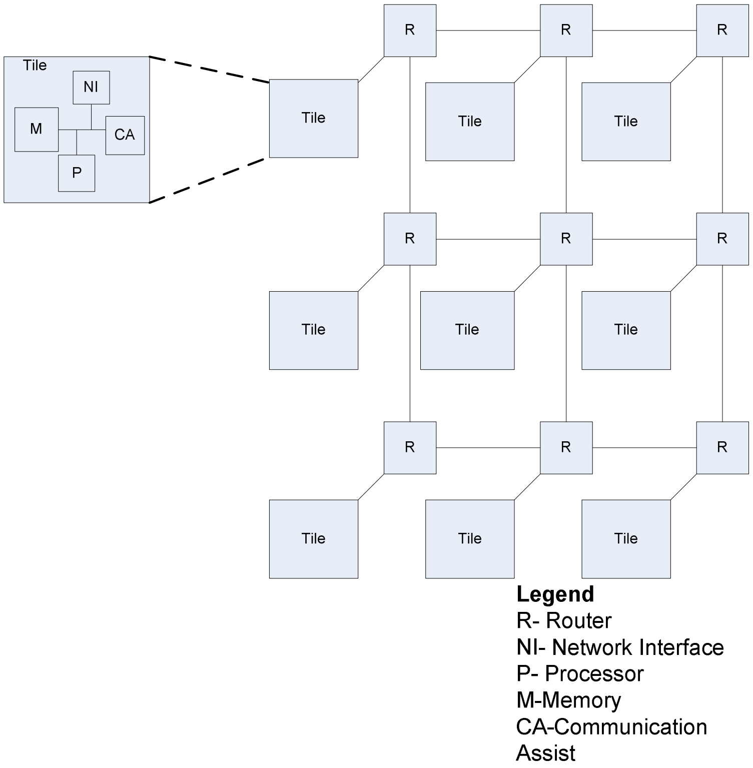


Figure 2: NoC with 3x3 mesh topology

1.4. NoC Components

A typical NoC (Figure 2) is composed of multiple routers and network interfaces (NI) which connects the IP blocks to the network. Each tile is composed of a processor, communication assist (CA), memory and a network interface as mentioned in the previous section. Routers and NIs are connected through links. A router routes the packets along the network. A network interface makes the IP view (protocol) compatible to the router view on communication (packets). Each NI can connect multiple IP blocks to the router network. Each router is composed of FIFO buffers, an arbiter and a crossbar to route the packets to the destination. FIFOs are also available in the NI. In this thesis, we will therefore focus on FIFOs, routers and links.

1.5. Goal

The main goal of this thesis is to develop energy models for Network-on-Chip (NoC) components that characterize the energy consumption of these systems and their components. These models should allow the prediction of the energy consumption of NoC-based systems. To achieve this goal, it is necessary to investigate and profile the energy consumption of various processing, communication and storage components that form a part of NoC-based systems. In order to estimate the energy consumed by components in isolation, we need to build energy models and analyze the different parameters that contribute to the overall energy consumption of the component. Therefore, the resulting model thus developed will allow designers to predict the energy consumed by these components. As mentioned, this thesis focuses on models for routers, FIFOs and interconnecting links.

Several power and energy models have been proposed by different people from the SoC community, each focusing on a specific issue. It is necessary to get an in-depth knowledge of these traditional energy models through an extensive literature study. Through this literature study, it is intended to answer and analyze several questions such as those concerning the level of abstraction dealt with while estimating energy consumption of individual components, the correctness of models in terms of relative and absolute accuracy and their impact on a system as a whole.

1.6. Approach and Contribution

With reference to the goal mentioned in Section 1.5, NoC components are designed and implemented in Verilog at RT level and thereafter energy models of these components are built. Energy and power measurements are then carried out on these designs to validate the models.

Initially the existing analytical model of a link is reviewed as in the literature. However a concrete energy model for the link is given only at the time of developing a compositional energy model for an entire NoC. Further, an n-place FIFO is designed and an abstract energy model is proposed for the same comprising several architectural and application specific parameters. The same procedure is followed to develop an abstract model for the router and its components in isolation. Finally a compositional energy model of the whole NoC is developed.

Therefore to summarise, in the current thesis, energy models for NoC components are built at a higher level of abstraction, i.e, at the transaction level. It is further intended as a future work that energy models for computational components also be developed and integrated with the NoC components to build models for the SoC as a whole.

1.7. Thesis Overview

The thesis is organized as follows. In Chapter 2, a literature survey of energy and power models is done with references to the literature and resulting in a classification of energy and power models based on various criteria. Chapter 3 deals with the energy model of the links; however at this stage only a mathematical model for the links is specified and this is validated after implementing the routers which act as realistic input and output blocks connected via an interconnecting link. Chapter 4 deals with the design and energy model of FIFO buffers along with experimental results. Chapter 5 explains the energy model of a router along with the experimental results. Chapter 1 deals with a compositional energy model that integrates all NoC components explained in the previous chapters. Chapter 1 summarizes the conclusion of the thesis together with the ideas for future recommendations.

Chapter 2 **Literature Survey of Power Models**

2.1. Introduction

There has always been a need to estimate and analyse the power consumption on a chip in order to optimise the chip design for energy consumption or to map applications onto systems to achieve low power. With this objective, power models are developed at various abstraction levels for SoCs and especially for NoC-based SoCs.

The concept of networks-on-chip is often compared with that of the fundamental concepts of communication networks and more specifically with the OSI seven layer model [2][4]. Figure 3 depicts the seven layers and their respective importance in the field of NoCs. More specifically for a NoC architecture, the last four layers are very important. Therefore, we can classify NoCs in terms of the OSI layers.

Secondly, at the lowest level of abstraction, bits of data are sent through links (bunch of wires) from one processing node to another via intermediate routers. Here, the energy consumed is the sum of the energy consumed by the links (through the wires), nodes (processing elements) and intermediate routers which in turn include the power consumed by internal components such as buffers, arbiters and crossbars during switching action.

At higher levels, energy consumed due to the sending of one bit of data from one router to another via the links is a function of the number of hops (or the number of routers) and the number of links. The energy consumed by the bit is given by:

$E_{bit} = n_{routers} * E_{router} + n_{links} * E_{link}$

This is typical to networks where a router is connected to each processing element.

Thus, energy models are dealt with at different abstraction levels normally focussing on the application and at times focussing on the architecture. It is often challenging or simply impossible to take into consideration all these aspects and abstraction levels; however a reasonable compromise can lead to efficient models.

In this chapter, details concerning the classification of energy models at various levels of abstraction and at various component levels are studied together with some related work done by different people. Three different tabular representations of the classifications along with the references are made so as to locate the references common to all the three classifications (shown in section 2.2), thereby investigating the development of energy models at different levels and focussing at different aspects. We also identify those energy models that focus on one particular area.

2.2. Classification of Power Models

In this section, a classification of Power/Energy models is made based on three essential categories. They are as follows:

1. Comparison of OSI layers with NoCs.
2. Component level
   a. Communication elements
      i. Routers
      ii. Links
   b. Computational elements
      i. Processors- Instruction level

ii. Memories

c. Other devices

i. Peripherals

ii. I/O interfaces

3. Different levels of abstraction
   a. Algorithms (Functionalities)
   b. Transaction level
   c. Register transfer level
   d. Gate level
   e. Transistor level

2.2.1. Comparison of OSI layers with NoCs

Figure 3 below depicts the seven OSI layers [3][4] and their comparison with on-chip communication. The level of depth and accuracy increases as we move towards the physical layer. The lower three layers are concerned with the architecture of the network while the top four layers provide communication services to applications.

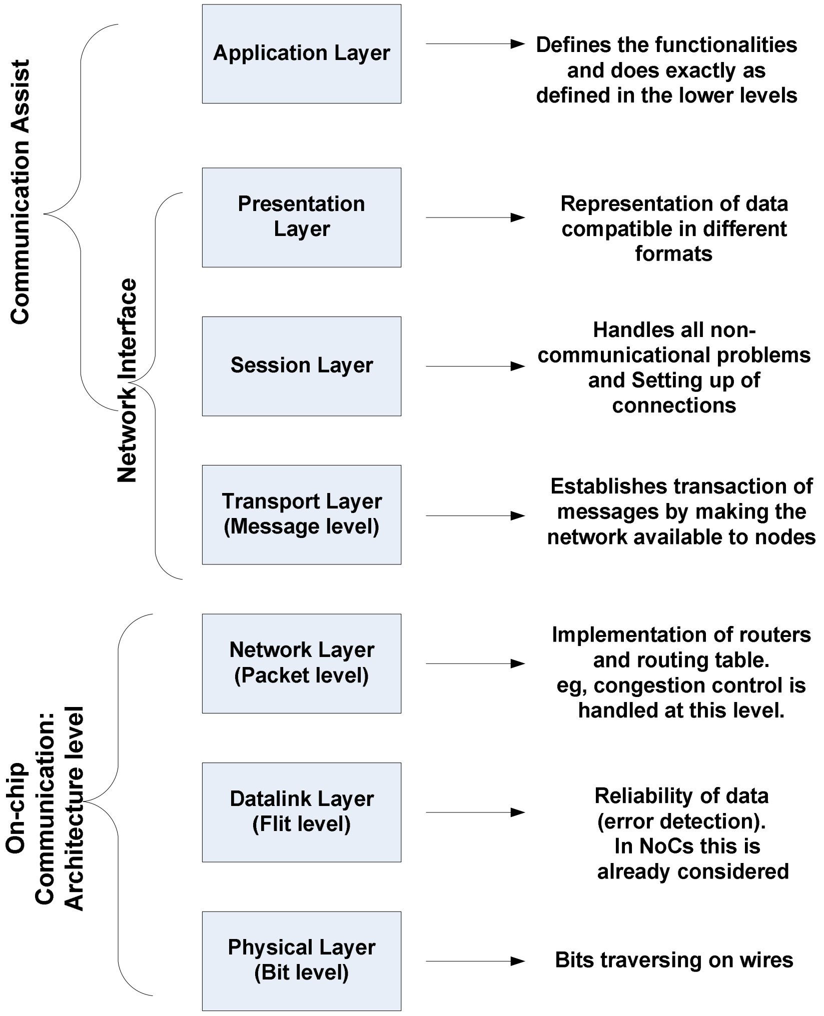

Communication Assist
Network Interface
On-chip Communication: Architecture level
Application Layer
Defines the functionalities and does exactly as defined in the lower levels
Presentation Layer
Representation of data compatible in different formats
Session Layer
Handles all non-communicational problems and Setting up of connections
Transport Layer (Message level)
Establishes transaction of messages by making the network available to nodes
Network Layer (Packet level)
Implementation of routers and routing table. eg, congestion control is handled at this level.
Datalink Layer (Flit level)
Reliability of data (error detection). In NoCs this is already considered
Physical Layer (Bit level)
Bits traversing on wires

Figure 3: OSI layers with reference to Network-on-Chip components

Table 1 shows the classification of NoCs when compared to the OSI layers with references. In the table, we observe that most models focus on the lower levels. No work is done on the higher levels, because most of the network related issues are handled at low levels, while the top three layers are more concerned with the application and the operating system in general. Therefore no power models are developed in this area.

OSI seven layers	References
Application layer	-
Presentation layer	-
Session layer	-
Transport layer (messages) - packetization issues	[15] – network power model based on message flows injected into the network.
Network layer (packets)	[5] – Packet switching fabrics,
Datalink layer (flits)	[15] – modelling based on flits traversing in a link.
Physical layer (bit level)	[1],[6],[7]

Table 1: Classification based on comparison of OSI layers with NoCs

2.2.2. Component level

We can broadly distinguish the components on a chip into three classes. They are communicational or NoC components, computational elements and other additional devices. Further in this classification we illustrate some references that describe power models for the system as a whole. We define a system as a collective whole of many sub-components including system software, whereas the application is treated separately. Table 2 provides the classification, discussed in more detail below.

2.2.2.1. SoC and Application Models

Energy models at this level are built considering the SoC as a whole. In [9], they used cycle accurate power models for an audio decoder application and in general their focus was on energy optimization for multimedia applications and device drivers again dealing at low level. The power models of some of the components were developed from the datasheets of the same obtained from the manufacturers. To deal with other SoC components, they used the validated power models developed by other authors. In [45], a cycle accurate multiprocessor System-on-Chip power simulation platform called GRAPES was developed.

In [15], a comparison of various benchmark applications based on their power consumption was made.

2.2.2.2. Communication (NoC) Components

At this level, energy models are built for various NoC components [1] such as links, buffers, routers, arbiters and controllers.

In [1], the authors performed experiments on register transfer (RT) level models to carry out trade-off analysis between the amount of power consumed by various components versus performance parameters such as injection rate of packets into the network, the number of virtual channels. In [1], the power models were dealt with at a low level of abstraction i.e., at gate level and hence focussed on issues concerning both dynamic and leakage power consumption.

In [5],[6] and [7], energy models were developed on various cross-bar based routers. The starting point of their experiments was based on extensive analysis carried out to study the power consumption on routers. While [15] was more related to power models for links.

2.2.2.3. Computational elements

In [8], the work focuses on different register architectures in order to obtain low cost solutions especially in media processing applications without degrading the performance at transistor level. They focussed on different register architectures in order to obtain low cost solutions especially in media processing applications without degrading the performance.

In [11], the authors claimed that significant power savings can be achieved using various optimization techniques particularly on processors at the instruction level (concerning the number and type of instructions executed). They established that techniques such as data-code transformation and hence the number of instructions (measured as current drawn by individual instructions, [12]) have significant impact on processor power consumption.

In [13] and [14], the authors also dealt with power models at instruction level where they redefined the claim made by [12] that stated that total power consumption was the sum (function of current drawn) of all the instructions. They ([13] and [14]) proved that the power consumption was due to the circuit changes between consecutive executed instructions and this information needs to be taken into account in particular to establish accuracy.

In [10], the authors propose high-level analytical power models (for both on-chip and off-chip components) with FPGA-based co-processors achieving reasonable accuracy (≈12%). They mainly focussed on input-output traffic between local memory and processing elements.

2.2.2.4. Other Components

Apart from NoC and computational components, focus has been on other SoC components such as peripheral devices. Profiling the power-consuming components is a basis to this approach.

In [9], the work was focussed around power models for components such as device drivers at cycle accurate level. The power models of some of the components were developed from the datasheets of the same obtained from the manufacturers.

Components				
SoC (System as a whole)	Application	Communication elements NoC	Computational elements	other
[9], [45]	[15] – Comparison of benchmark applications.	[1] - Physical links and routers	[8]- Registers for processors	[9] - audio & video devices
	[8]- Media processing applications.	[5] – Routers	[10]-Processors at Instruction level (loop transformation); Memory models developed based on its internal parameters.	
		[6] – Routers and interconnection architectures	[11] – Processors: Data reuse transformation at instruction level	
		[7] – Routers & links	[12] – Processors: at instruction level.	
		[15] – link utilization.	[13] – Processors at instruction level based on switching actions. [14] – Same as [13].	

Table 2: Classification based on component aspects

2.2.3. Different levels of abstraction

Energy models are also classified based on different levels of abstraction. At the highest level of abstraction is the algorithm level with different functions and processes. This level emulates the functionalities performed at the lower levels. Subsequent to this is the transaction level, where processes correspond to events. Each write or read transaction is considered as an event and do not go into details up to the register transfer level. The models at these two levels represent the system level and are less accurate but also require less simulation time. The next level lower in the abstraction is the register transfer level. The transfer of data is at register and wire level, but does not deal further to gate or transistor level. The models at this level are very accurate but need more time to simulate. Next to this level are the transistor or gate levels which represent the model at circuit level and are more accurate than any of the other levels; however these models are extremely timing consuming.

In [7], energy models were developed based on an appropriate algorithm that mapped a given IP to a regular network architecture which is deterministic and deadlock-free using worm-hole routing.

At transaction level, energy models are dealt with at a system level and principally without going into the hardware details of individual components. In [5], power models were developed aiming at a low power NoC solution and observing the system performance on shared memory ATM based switch fabrics. Further, they emphasized the importance of moving power optimization processes from circuit level to system level.

In [6], transaction level power models were developed for various cross-bar based routers. The starting point of the experiments was based on extensive analysis carried out to study the power consumption of routers. Both [5] and [6] deal with the transaction level and focus only on dynamic power consumption and neglect leakage power. The main reason for this was due to the complexity involved in computing the leakage power. Secondly, the processing technology employed made it possible to neglect the leakage component. Most importantly, they aimed at improving performance metrics such as throughput and delay at system level.

In [11],[12],[13] and [14] energy models were developed for processors at instruction level. They measured the power consumed by individual instructions as the current drawn by each instruction along with the impact caused by circuit changes due to the execution of consecutive instructions.

In [45], a Multiprocessor Systems on-Chip power simulation platform called GRAPES was developed at cycle accurate level. Several optimization techniques of the synchronization mechanism for MPSoC based NoC were explored.

In [1], power models were developed at the Register transfer level to carry out trade-off analysis between power consumed by various components and parameters such as size of packet, length and width of physical links, number and depth of virtual channels and switching technique.

Level of Abstraction	**References**
Algorithm level (functionalities)	[7]
Instruction level	[11][12][13][14]
Cycle Accurate level	[45]
Transaction level	[5],[6],[15]
Register transfer level	[1]
Gate level	[9]
Transistor level	[8]

Table 3: Classification based on levels of abstraction

2.3. Conclusion

We can thus infer from the above description that power models are developed at various levels of abstraction and targeted for particular components that are communicational or computational. Based on these models, predictions of power consumption are made in order to optimise systems and to map applications onto systems to achieve low power.

From the literature study it was observed that some of the energy and power models are too low level (e.g. transistor level) and are also too expensive and time consuming to evaluate, while some of them are too abstract and are not accurate enough. Some power models describe only one or a few components of a SoC which are not usable for the SoC design as a whole. Therefore, we intend to target an intermediate level of abstraction that leads to models usable for the evaluation of the SoC as a whole during architectural design space exploration.

In this thesis, NoC components are designed and implemented at Register transfer level, which is at a low level. However the energy models are developed at a higher abstraction level than the RT level and to be precise, they are represented at transaction level. It is further intended as a future work that energy models for computational components also be developed and integrated with the NoC components to build models for the SoC as a whole.

Chapter 3 **Power Model for Links**

3.1. Introduction

We are aware of the fact that the evolution of Multiprocessor System-on-chip (SoC) designs have resulted in a number of NoC-based interconnect architectures [6][18][21]. These aim to achieve an improved communication system compared to the traditional architectures in terms of energy consumed, throughput and latency. On the other hand, current technologies are migrating towards achieving smaller device sizes and shortened interconnects. However, these short interconnects are posing a serious threat in achieving low power solutions in terms of additional heat dissipated on the system due to the decreased spacing between the wires and increased load on the wires due to increased frequencies. Therefore, in this chapter, we analyze the different issues concerning the power consumed in the interconnect wire by means of a detailed study of power models. These models can not only give us an insight in how power models are developed in general but can also help us in identifying the key parameters responsible for the power consumption of the wires in a chip.

3.2. Energy Consumption in a Wire

In order to calculate the energy consumed in an interconnect wire we need to study its analog or electrical behavior. In a wire, significant energy is consumed during the switching activity (charging and discharging) of the capacitor, which is normally referred to as the switching power. However, a portion of energy is also consumed as a result of a short circuit during switching activity at the driving gate's output; this is referred to as internal power. Both of these together form the dynamic energy. This contributes to about 70-90% of the total energy consumption under normal operating conditions. While a portion of energy is lost due to the leakage power irrespective of the switching activity and state of the gate [19], this is referred to as the static energy. The static component contributes to about 10-30% of the total energy consumption.

At processing technologies until 90 nm, it has been seen that dynamic power is the major power consuming component [23] (refer Figure 5) but at technologies below 90nm, this leakage power begins to build up, due to the fact that at lower technologies the supply voltage and thereby the threshold voltage is also reduced consequently. This definitely reduces the dynamic power consumed but on the other hand the static power increases due to the increased leakage currents from the non-conducting gates flowing from source to ground which is more prominent with dropping voltages due to its exponential dependency on threshold voltage [24]. As a result, the power consumptions due to the leakage current are increased.

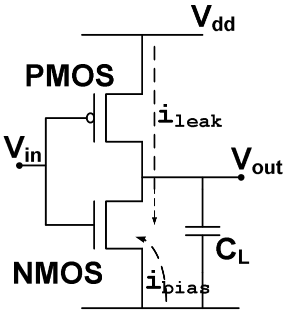
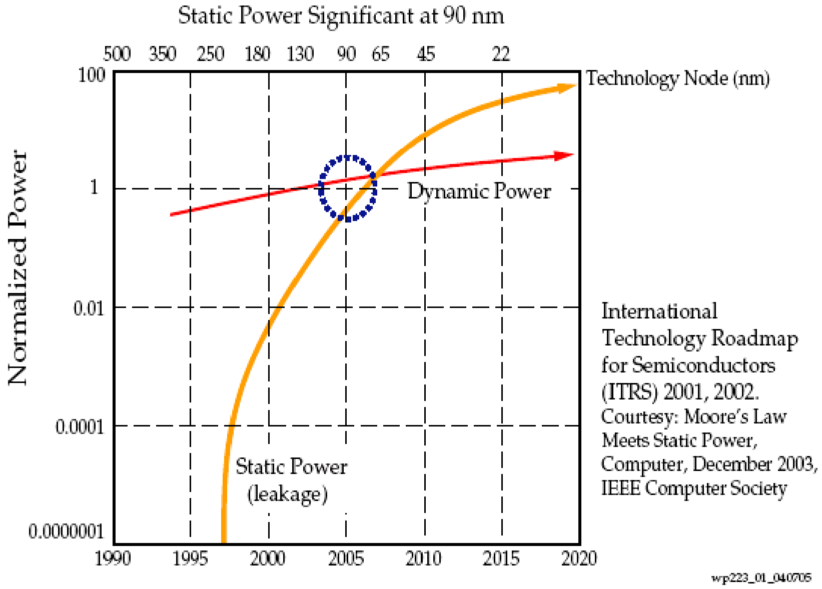


Figure 4: CMOS Circuit showing leakage current and reverse bias current

In order to understand the dynamic and static energy components more precisely, they will be dealt with in detail in section 3.3 focusing on a single interconnect wire.

In addition to the above mentioned components of power consumption, when we consider a larger interconnect system such as a NoC with wires adjacent to each other (owing to their physical routing scheme), energy is consumed due to cross-coupling and self coupling effects [17],[22]. The former is caused by interference of activities between the neighboring wires while the latter is caused by interference between different events on the same wire. The coupling capacitance in particular is significant with shrinking technologies. Due to the decreased spacing between wires, the cross coupling capacitances are increased thereby increasing the power consumption. Hence we will focus on the cross coupling effects in section 3.4 In section 3.5, the total power consumption of a link will be calculated.


Static Power Significant at 90 nm

International Technology Roadmap for Semiconductors (ITRS) 2001, 2002.
Courtesy: Moore's Law Meets Static Power, Computer, December 2003, IEEE Computer Society

Figure 5: 2001, 2002 ITRS prediction of static and dynamic power consumption trends

3.3. Power equations for a gate driven wire

As mentioned in the earlier section, power consumption in a gate driven interconnect wire is a combination of dynamic and static power.

$$P = \frac{1}{2} C_L V^2 \alpha_L f + \tau \alpha_L V I_{short} f + V I_{bias,\, wire} + V I_{leak,\, gate}$$

Here the first component is the dynamic power consumption caused by charging and discharging of capacitive load on the gate's output where 'C_L' is the load capacitance, 'V' is the supply voltage, 'α_L' is the switching activity of the gate and f is the operating frequency of the system. Switching activity is defined as the number of 0 to 1 and 1 to 0 transitions, normalized by the operating frequency. The switching activity constant eventually captures the above mentioned self-coupling effect.

The second component is also a part of the dynamic power consumed by a momentary short circuit caused between supply and ground when the gate's output is switching, Figure 7. Here τ is the short time period during which the short circuit current I_{short} flows between source and ground.

The third component is the static power consumed when there is no switching activity and specifically as a result of the circuit structure, the wire, [25] and $I_{bias,\,wire}$ is the current flowing from the wire to its substrate (irrespective of the gate).

The fourth component, also a part of the static power, is the leakage power consumed by the gate due to the leakage current flowing again from source to ground but regardless of the gate's state and switching activity.

Finally, note that the capacitance and current constants in the above model depend on the length of the wire, i.e., the above model implicitly gives the power dissipation of a wire per unit length.

3.3.1 Traditional ½ CV^2 Model to calculate the Dynamic Energy Consumption on a Single wire

The ½CV^2 model [22] has been used commonly to measure the interconnect energy consumption. This model is based on the assumption that energy is consumed in every rise or fall transition of an event [20]. In this model we concentrate only on a single wire and neglect the coupling effect caused by the neighboring wires. The effect of coupling capacitors will be dealt with in section 3.4. Further, in this model, we also neglect the effect of resistance, assuming that the capacitor requires sufficiently long time to charge. The energy consumed by the wire during one transition from low to high and back (during charging and discharging of the capacitance) is given by $E = C_L V^2{}_{dd}$, where C_L is the total load capacitance and V_{dd} is the supply voltage. This can be derived as follows.

Consider a simple interconnect wire model as shown in Figure 6 with a load capacitance C_L which is the sum of the wire capacitance C_{wire} and the capacitance of the input gate, an inverter driving the wire. The two gates is a combination of PMOS and CMOS transistors as shown in the Figure 7.

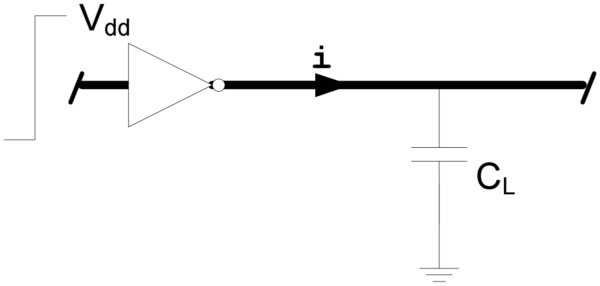


Figure 6: A simple interconnect wire model

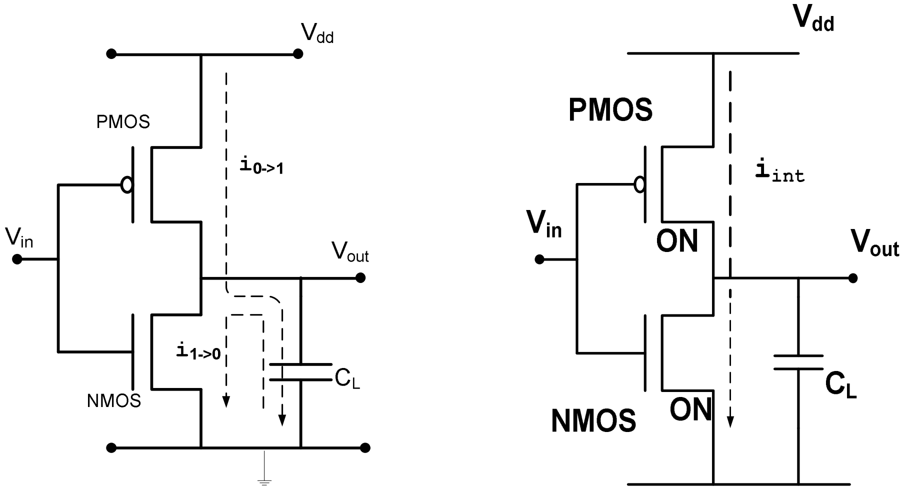


Figure 7: CMOS Model of an inverter cell depicting switching power (left) and internal power (right)

When the voltage at the input switches from low to high i.e., from 0 to V_{dd}, the capacitor gets charged from 0 to V_{dd} in T amount of time (this T can be assumed sufficiently large). Here we assume a zero rise and fall time, which means switching between low to high or vice versa is instantaneous and ideally the transistors in the driving gate are never ON at the same time. During the transition from low to high, the current i flows from the power supply to the capacitor via the PMOS transistor. The energy consumed from the power supply during this transition can be calculated as follows:

The power consumed during low to high transition is given by:

$$P(t) = i(t)V_{dd} \,,$$

Where V_{dd} is the supply voltage needed to charge the capacitor from 0 to V_{dd} and i is the current drawn from the power supply to charge the capacitor. P and i both depend on the time t. The current i equals:

$$i(t) = C_L \frac{dV_{out}}{dt}$$

The energy that is drawn from the power supply can be derived as follows:

$$E_{0->1} = \int_0^T P(t)dt \;\; = \int_0^T i(t)V_{dd} \cdot dt$$

$$= C_L V_{dd} \int_0^T \frac{dV_{out}}{dt} dt$$

$$= C_L V_{dd} \int_0^{V_{dd}} dV_{out}.$$

$$= C_L V_{dd}^2$$

Also the energy stored in a capacitor can be calculated as follows:

$$E_{capacitor} = \int_0^T i(t).V_{out}.dt \quad = \int_0^T C_L \frac{dV_{out}}{dt} V_{out} dt$$

$$= C_L \int_0^{V_{dd}} V_{out}.dV_{out}.$$

$$= \frac{1}{2} C_L V_{dd}^2$$

From the above equation, we can say that only half of the energy is used to charge the capacitor during low to high transition. The other half of the energy is dissipated by the PMOS transistor. During transition from high to low, no energy is drawn from the supply and the capacitor discharges its energy into the NMOS transistor. Thus, the total amount of energy drawn from the power supply during both the transitions together (switching activities) equals $C_L V_{dd}^2$ and $1/2 C_L V_{dd}^2$ on average per transition.

In general, we also need to take into account how often the circuit switches. For simplicity, from now on, V_{dd} will be replaced by V. Thus the switching power is given by $P_{switching} = \frac{1}{2} \alpha_L C_L V^2 f$, which is the first component in the formula given in section 3.3. This can be captured in a constant α_L, which is called the switching factor and has a value between 0 to 1 representing the switching activity relative to the clock frequency f. Thus by knowing how often a circuit is switched, we may calculate the total power consumed in a wire.

While deriving the power equations in an interconnect we assumed that the switching between low to high and vice versa happens instantaneously. This is an ideal situation. But in reality, this never happens and there is always a sloping transition from low to high and vice versa. During this period, there is a small leakage current flowing from the source to ground, when the outputs of the gates are switching. At one moment both the transistors in the driving gate are conducting and this short-circuit current flows until a stable condition is reached. This gives the following component in the equation of section 3.3

$P_{short} = \tau \alpha_L V I_{short} f$,

where τ is the short period during which this leakage occurs and I_{short} is the short circuit current flowing from source to ground.

3.3.2Static Power Consumption on a Single wire

Static power is the power dissipated by a gate or a wire when it is inactive or in static state. The static power is mostly influenced by the structure of the circuit. Ideally the static current of the CMOS circuit is equal to zero as the NMOS and PMOS devices are never on simultaneously in steady-state operation. But, a leakage current flowing through the reverse biased junctions of the transistor, located between the source and the ground causes the static power dissipation. When voltage falls below a certain threshold voltage, these static powers are consumed which prevent the gate from completely turning off. This power is drawn from the supply and more specific from the battery when referring to hand-held devices. Also, static power is dissipated when current leaks between the diffusion layers and the substrate. The static power dissipation can be express by the equation:

$P_{static} = VI_{bias,\, wire} + VI_{leak,\, gate}$.

The leakage power is thus assumed to be a serious threat at technologies below 90nm, due to reduced threshold voltages.

3.4 Power Equations incorporating Cross-Coupling effects

Figure 8 shows a simple interconnect model comprising three wires with their respective load capacitance and in addition a mutual (cross) capacitance between the wires. This cross coupling capacitance results in increased consumption of power. The effect due to this cross capacitance is often termed cross-talk.

This cross-talk effect is prominently seen when the wires are switching to different output values and when one of the two wires switches, while the other doesn't and both have different output values.

The power model for a 3-wire interconnect and therefore in general for an N-wire interconnect can be derived as follows.

Consider a 3-wire interconnect as shown in Figure 8 with C_C the cross coupling capacitance with a relative switching activity α_C between wires. In order to study the impact of cross-coupling, we must also take into account the self capacitance which is the result of switching activity within the wire. This is necessary due to the fact that cross coupling occurs as a result of opposite switching activities between the adjacent wires. Let C_L be the load capacitance and α_L the switching activity caused due to switching within the wire.

Here we choose three wires because mutual coupling can be better understood with three wires as we can focus on the middle wire with mutual coupling effect separated by a distance 'd' from its two adjacent neighbors. The capacitance on the middle wire is $C = C_L + 2C_C$, where C_L is its (middle wire) load capacitance and $2C_C$ is the coupling capacitance of both the neighboring wires. From the equation above it appears as if the coupling effect is dominating over the self coupling capacitance. However, when considering the total power consumption due to transitions on all the three wires as a link, we notice the following:

$$C_{total} = 3C_L + 2C_C$$

And therefore in general for an N-wire interconnect, we obtain $C_{total} = NC_L + (N-1)C_C$

Therefore knowing the total capacitance including cross-coupling and self-coupling of a link, ignoring the static power and the effect caused by the momentary short-circuit for the current discussion, we obtain the power equations for N wires as shown below:

$$P_{switching} = \frac{1}{2}(NC_L\alpha_L + (N-1)C_C\alpha_C)fV^2$$

Assuming N≈N-1 for a large interconnect, we obtain

$$P_{switching} \cong \frac{1}{2}N(C_L\alpha_L + C_C\alpha_C)fV^2$$

From the equations above, it is obvious that the coupling effect cannot be neglected if Cc is large. Now, the important thing to analyse is how large is C_C with respect to C_L. From the studies performed by [17] on a 0.25µm process technology, it is observed that the values of C_L are two or three times the values of C_C for a distance of say 1µm separating the wires. So assuming equal α_L and α_C, the cross-coupling effect is responsible for 25-33% of the switching power. When moving towards 0.09µm or 90nm technology and the typical values of distances separating the wires are reducing below 1µm (current trend 0.2µm) and for such distances, the typical values of C_C are much more than C_L (factor of 2 or 3). This gives an indication of the threat due to coupling capacitance for the future technologies.

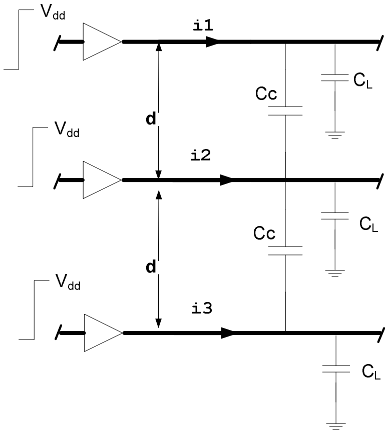


Figure 8: 3 Interconnect wires with mutual capacitance

3.5 Power model for a link

From the power models derived in section 3.3 for a single wire and the power model considering the cross-coupling effect for N-wire interconnect in section 3.4, we may conclude the total power for an N-wire link per unit length as follows:

$$P_{link} = \frac{1}{2} N V^2 (C_L \alpha_L + C_C \alpha_C) f + N \tau \alpha_L V I_{short} . f + N.(V I_{bias,\ wire} + V I_{leak,\ gate})$$

where N is the total number of wires in the link, C_L and C_C are the self and coupling capacitance of a wire and neighboring wires respectively, α_L is the switching activity on a wire and α_C is the switching activity with respect to the adjacent wires, τ is the short circuit period, V is the supply voltage, f is the clock frequency and I_{short}, $I_{bias,\ wire}$ and $I_{leak,gate}$ are currents as explained before. Again, to summarize, the first two components are the dynamic components and the last component is the static component. To be more precise, the first component is due to switching activity within a wire and between wires, the second component is a result of short-circuit of the gate during switching.

Chapter 4 Design and Power Model of Register-based FIFOs

4.1. Introduction

FIFO, acronym for “First in First out”, is a concept used to describe the behaviour of a buffer. As the name says, it works according to the first-come first-serve principle.

A FIFO fundamentally consists of some storage elements from which data can be read or written to. The storage element may either be an SRAM [33][27] (static random access memory) or a DRAM [26] (dynamic random access memory) or a set of registers (flip-flops) or any other form of storage.

SRAM’s are static in nature and provide data as long as power is provided. SRAM’s are mainly used for their speed.

DRAM’s are employed as separate individual memories due to their small size and lower cost than SRAM’s. But they can be power hungry on account of their need for periodic refreshing.

Alternatively, registers are better than SRAMs and DRAMs for realising small memories due to their small size, cost effectiveness and simplistic organization. Current trend in NoC-based SoC’s [28] is the usage of register-based buffers as they consume less energy than DRAM or SRAM solutions.

In general, irrespective of the type of memory used, buffering basically helps in managing the data traffic during congestion of packets (increased traffic) through the links in a network and during contention of resources (a situation where two or more sources compete for the same resource at the same time) in a large network.

In this chapter, we therefore analyse and focus on the power consumption of a router based FIFO buffer using registers. The energy consumed also depends on various factors and shall be discussed in this chapter and later on, a power model is given. In order to observe and analyse the effect of various parameters on energy consumption of a FIFO, several experiments are performed on FIFO buffers of various sizes.

4.2. Implementation of Register based FIFO Buffers

As explained earlier, FIFOs can either be SRAMs, DRAMs or registers. We focus on the latter solution which is the standard for NoCs. When focussing to the internals of a FIFO, the data (group of flits or words) is either multiplexed-demultiplexed (as in the current implementation described in section 4.2.1) or shifted using shift registers (refer to section 4.2.3) to the output of the FIFO [29][30][31]. We discuss these two alternatives briefly in subsequent subsections along with their pros and cons.

4.2.1 Implementation of 4-place FIFO Buffers

Figure 9 depicts the internal block diagram of a 4-place FIFO buffer. It is mainly composed of a FIFO control unit, a 1-to-4 channel input unit (similar to a de-multiplexer), a 4-to-1 channel output unit (similar to a multiplexer) and four registers to store the data.

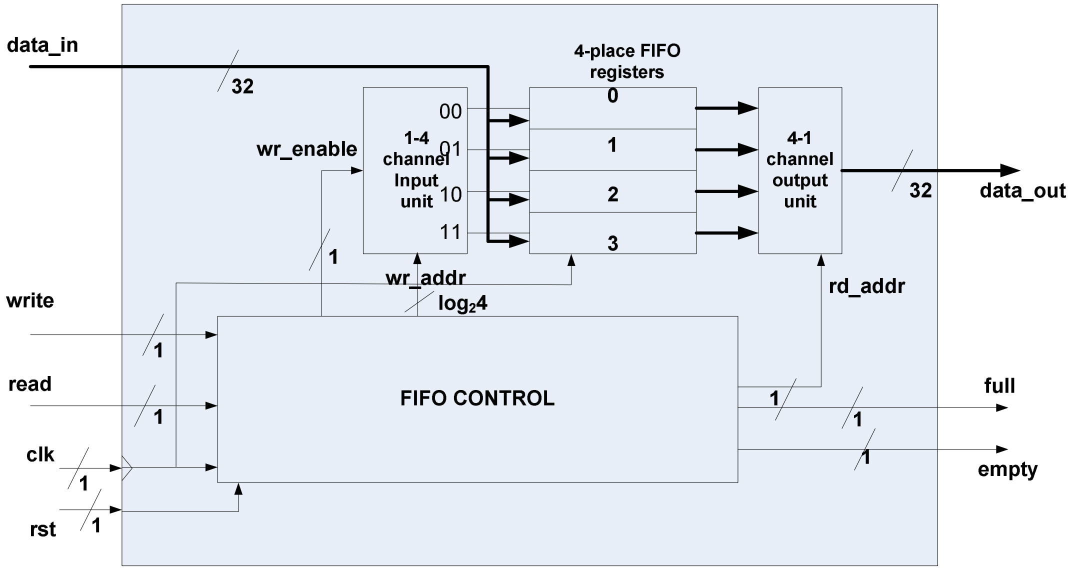


Figure 9: 4-place FIFO buffer

The interfaces to the FIFO at the input side are the read, write and input data signals apart from the clock and the reset signal, while at the output, the interfaces are the output data, full and empty signals. Internally there are signals such as the write and read address pointers and a write enable signal which are controlled by the FIFO control unit. In addition, the reset signal is used to initialize the write and read address pointers to a steady state, which is the empty state of the FIFO, to avoid abnormal behaviour of the FIFO. Further, the usage of the clock is such that changes occur only at the rising or falling clock edge depending on the requirements.

As explained earlier, the FIFO control unit controls the functioning of the whole FIFO. It controls and updates the read and write address pointers. It also informs the outside world about the full and empty status of the FIFO. The 1-to-4 channel input unit lets the data to be written onto the correct FIFO place based on the write address generated from the FIFO control unit only when both the write action and write enable are true. Similarly, the 4-to-1 channel output unit lets the data to be read from the correct location based on the read address generated from the FIFO control unit.

The data arrives at the FIFO and with the write action issued, the data is written into the appropriate register place. The right location of the FIFO place is notified with the write enable signal together with the write signal going high. This is to ensure that the write action takes place only when at least one of the FIFO places is free, so that the data is not overwritten before a read action has taken place. Similarly, when the read action is issued, the data is read from the correct location, which in turn is controlled by the FIFO control unit. It ensures that the FIFO has at least one entry. When the FIFO is empty, on a read action, the previous data read will be revealed again.

4.2.2 Working of a 4-place FIFO buffer in terms of a behavioural FSM

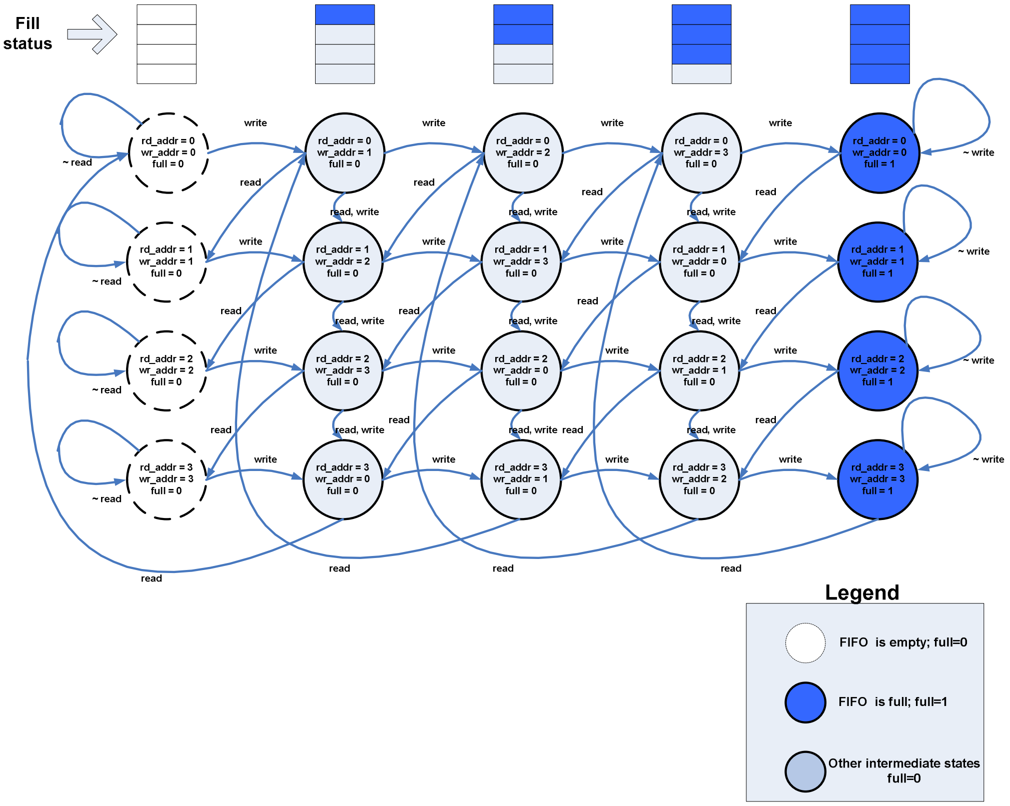

Fill status
rd_addr = 0 wr_addr = 0 full = 0
rd_addr = 0 wr_addr = 1 full = 0
rd_addr = 0 wr_addr = 2 full = 0
rd_addr = 0 wr_addr = 3 full = 0
rd_addr = 0 wr_addr = 0 full = 1
rd_addr = 1 wr_addr = 1 full = 0
rd_addr = 1 wr_addr = 2 full = 0
rd_addr = 1 wr_addr = 3 full = 0
rd_addr = 1 wr_addr = 0 full = 0
rd_addr = 1 wr_addr = 1 full = 1
rd_addr = 2 wr_addr = 2 full = 0
rd_addr = 2 wr_addr = 3 full = 0
rd_addr = 2 wr_addr = 0 full = 0
rd_addr = 2 wr_addr = 1 full = 0
rd_addr = 2 wr_addr = 2 full = 1
rd_addr = 3 wr_addr = 3 full = 0
rd_addr = 3 wr_addr = 0 full = 0
rd_addr = 3 wr_addr = 1 full = 0
rd_addr = 3 wr_addr = 2 full = 0
rd_addr = 3 wr_addr = 3 full = 1
write
read
~ read
read, write
~ write
Legend
FIFO is empty; full=0
FIFO is full; full=1
Other intermediate states full=0

Figure 10: FSM of a 4-place FIFO buffer

The FIFO acts as a counter modulo the FIFO size. The states depicted here are ranked according to their fill status from left to right ranging from empty, one entry, and so on until the FIFO is full. Each of these states has four occurrences, depending on the precise value of the read and write pointers. If the FIFO is either empty or full, the read and write pointers point to the same location in the FIFO. Hence we define a variable full that is zero when empty and one when full. The five main FIFO states are represented as, 0(full=0), 1(full=0), 2(full=0), 3(full=0) and 0(full=1). Note that we differentiate the first and the last state with the status of the variable full. Further from Figure 10, we see that write and read actions can be performed simultaneously, provided the FIFO is neither full nor empty.

4.2.3 Alternate model of a FIFO buffer using shift registers

FIFO buffers can also be implemented using shift registers, refer Figure 11 [30]. Shift registers are primarily used to transfer data linearly from input to output. Fundamentally, shift registers are a set of registers arranged in a linear order with inputs connected to outputs in such a way that the data is shifted along its line when activated.

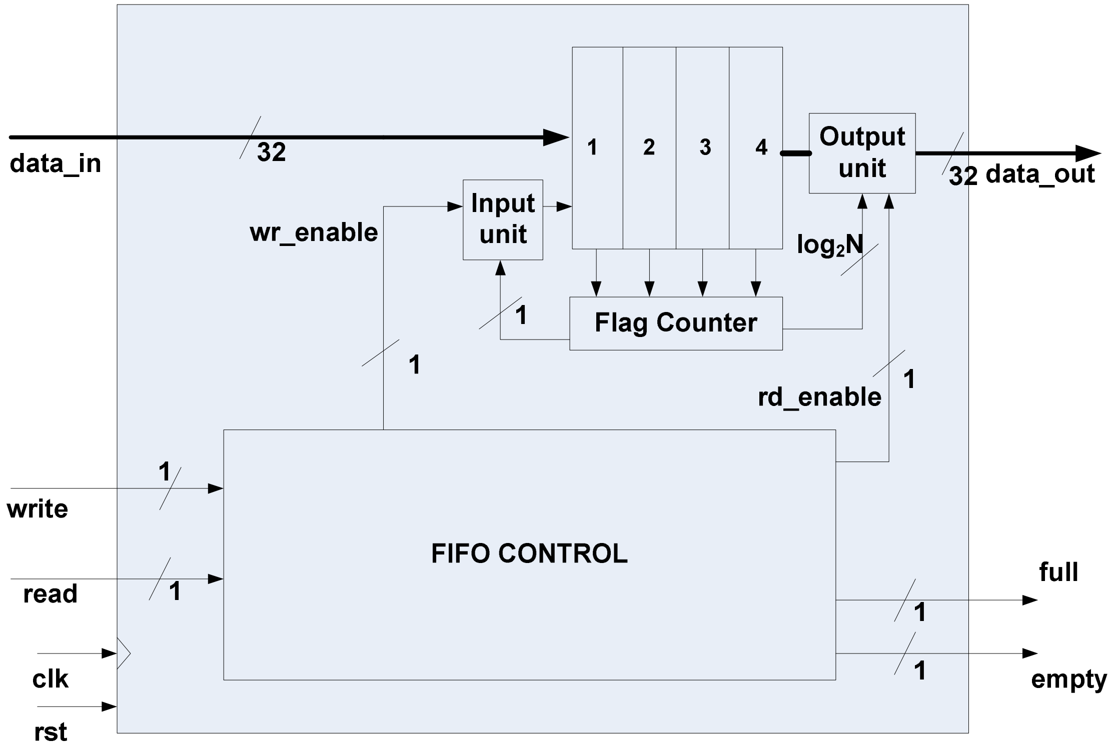


Figure 11: FIFO model with shift registers

Initially when a write action is issued, with the arrival of the data at the input, the previously available data in the FIFO is shifted one position towards the output depending upon the availability of space in the buffer. On a read action, data is forwarded towards the output of the FIFO from the first location containing valid data. The information regarding space availability and location of data is given by the flag counter.

4.2.4 Pros and Cons of the above two architectures

A FIFO buffer as implemented in section 4.2.1 is a better option over a FIFO architecture as implemented in section 4.2.3 as there is no delay in extracting the data from the FIFO buffer. In the latter case, the data has to shift as many places as FIFO buffers and this introduces latency. A drawback of the former architecture is that it requires multiplexers and de-multiplexers sized according to the number of FIFO places. This adds to area and power consumed.

Shift registers are useful if a small number of shift stages is used (<8), otherwise, it may lead to overloading of the data path if too many stages are connected. Even in case of a low number of shift stages, for e.g., 4 stages, we see that there is a latency of 4 clock cycles for data to traverse through the shift registers, assuming that the time taken by the data to shift between successive places is one clock cycle. Thus it is almost never recommended when compared to the FIFO buffers as shown in Figure 10.

4.3. Power model for a FIFO

Power consumption in a FIFO is due to reading and writing the data, clock activity, internal short circuit and leakage. The first four components form the dynamic part of the power consumption. This dynamic component can be further divided into power consumption due to switching activity and due to internal short-circuits that occur during switching. Hence the total power consumption is modelled as follows.

$$P_{total} = P_{write} + P_{read} + P_{clk} + P_{int} + P_{leak}$$

The first three components P_{read} , P_{write} and P_{clk} form the switching power of the dynamic component. Further P_{read} and P_{write} are influenced by the power consumed by the control part and due to the data part, the switching activity. The control part includes the power consumed by the control bits, i.e., the write and read signals and the FIFO control unit. The data part includes the power consumed due to registers and the flipping of bits.

In a write operation, the power consumed per time unit is the result of the power consumed by the FIFO control unit combined with the 1-to-n channel input unit in decoding the write address, along with the power consumed to store the incoming data in the FIFO.

$$P_{write} = r_w.P_{ctrl} + \alpha_F.P_{store}$$

Here, r_w is the rate at which write actions occur and α_F is the amount of 0 to 1 and 1 to 0 bit flips occurring between the incoming data and 'F' is a subscript for FIFO. P_{ctrl} and P_{store} are the average power consumed at the control unit for one write action and the power consumed to store the data in the FIFO respectively. Note that $\alpha_F.r_w$ in this context is the same parameter as α_L in the previous chapter.

Similarly in a read operation, the power consumed per time unit is the result of the power consumed by the FIFO control unit combined with the 4-to-1 channel output unit in encoding the read address along with the power consumed to retrieve the data from the FIFO.

$$P_{read} = r_r.P_{ctrl} + \alpha_F.P._{retreive}$$

Here, r_r is the rate at which read actions occur and α_F is the amount of 0 to 1 and 1 to 0 bit flips occurring between the outgoing data. In the long run, $r_r = r_w = r$. P_{ctrl} and $P_{retrieve}$ is the power consumed at the control unit and the power consumed to retrieve the data from the FIFO respectively. It is assumed that the average power consumed for the control of one read action equals the power consumed for controlling one write action.

Apart from the power consumed by write and read activities, there is another component namely the P_{clk} . Clock activity causes permanent constant power consumption due to switching activity.

The other component of the dynamic power is the internal power P_{int} which is the result of the internal short circuits occurring during switching of bits in the incoming data, the switching of read and write actions and clock activity. Therefore P_{int} can be calculated as follows,

$$P_{int} = k1.r + k2.\alpha_F + k3$$

Here, r is the rate of read and write actions and α_F is the amount of bit flips. The constants k1 and k2 are respectively the average internal power consumed for the control of read and write actions and due to bit changes in data and k3 is due to the clock activity. This varies with varying clock frequencies. The static power component, i.e., the leakage power P_{leak}, is assumed to be a constant.

Substituting the detailed power models for the individual components in P_{total}, we get

$$P_{total} = (2P_{ctrl} + k_1)r + \alpha_F(P_{store} + P_{retrieve} + k_2) + k_3 + P_{clk} + P_{leak}$$

4.4. Experiments and Results

4.4.1. Functional Verification of the model

In order to verify the FIFO model, several experiments are performed with different input vectors, different write and read patterns and different cases, some of which are mentioned below. On the whole the complete functional behavior is verified.

1. Full and empty status of the buffers.
2. Switching activity of the data bits (only zeros, only ones, different combinations of zeros and ones).
3. Alternate read and write actions.
4. Bursts of read and write actions.

A testbench is written to validate the functionality of the design with different situations and different test vectors. Both the design and the testbench are implemented in Verilog.

Initially, 32-bit sized input words are written to the FIFO and consecutively these words are read and they are compared in order to verify the correctness of the design. In addition, it is also necessary to verify the operation of a FIFO during certain conditions especially when the FIFO is full and a write action is issued and likewise, when a FIFO is empty and a read action is issued. Interestingly, the counter behavior of the FIFO also needs to be verified. That is to say, as the write or read pointer reaches the end of the FIFO, it is incremented and set back to the first position and in this way acts as a counter modulo the FIFO size.

4.4.2 Power measurements and Power Analysis in MAGMA Blast Chip

Power analysis of various components on chip can be done using Blast chip from MAGMA [16]. Figure 12 shows a typical flow of MAGMA Blast chip's power analysis is possible after fixing the netlist and later throughout the flow. But as the design flow progresses, the wire capacitance gets increasingly accurate and thereby the power analysis also becomes more accurate. One of the important pieces of information concerning the energy consumption is the amount of charges and discharges of capacitors. In Blast Chip, this information can be obtained when switching activity values at each node are assigned. Another method to set the activities is by importing a VCD (Value Change dump) file that is generated from a simulator using a testbench.

The basic power analysis flow includes the following

- Importing the power models from .lib
- Importing the design or the model in Verilog either at RT level or Netlist
- Configuring the switching activity
- Propagating the switching activity
- Performing power analysis

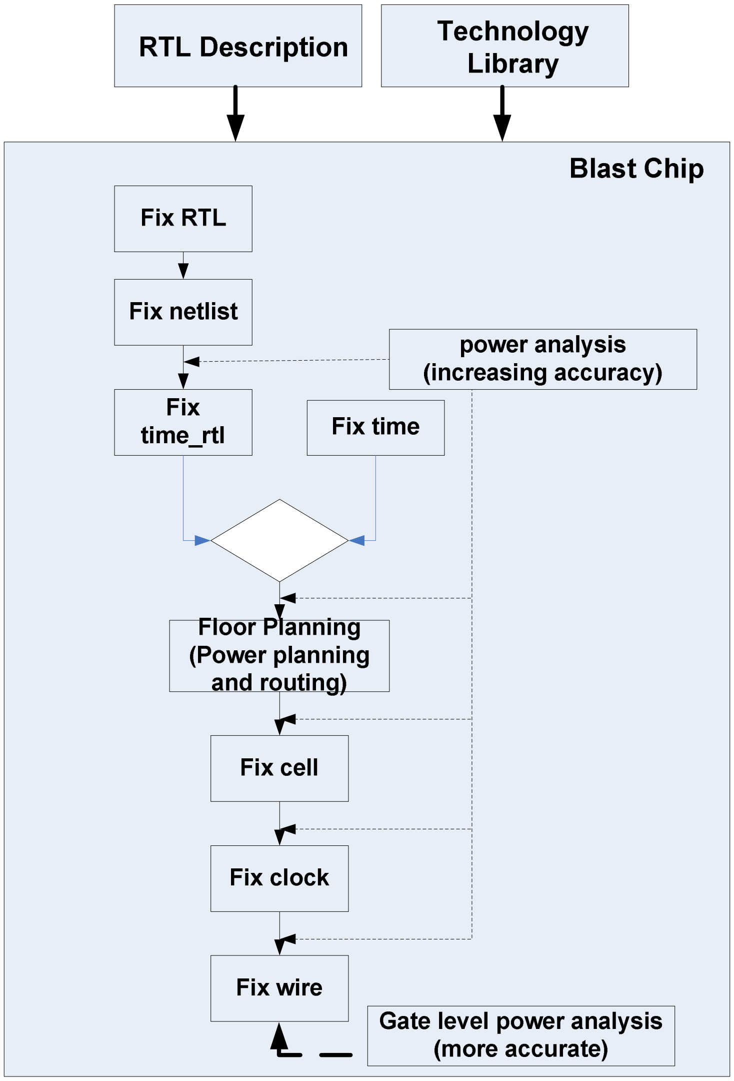

RTL Description
Technology Library
Blast Chip
Fix RTL
Fix netlist
power analysis (increasing accuracy)
Fix time_rtl
Fix time
Floor Planning (Power planning and routing)
Fix cell
Fix clock
Fix wire
Gate level power analysis (more accurate)

Figure 12: MAGMA Blast Chip design flow

A power model describes potential power dissipation of a system in terms of different transition states and events. Once the power characterization data of each of the models is imported from the library, the switching activities of a circuit needs to be specified. The switching activities are basically determined by two parameters, the toggle rate and the probability that the signal value is logic ‘1’. In this way, we can measure the power consumed at the output. These parameters can be set to desired values representing the rate at which the transitions occur and the probability of a logic value being high respectively. These switching activities propagate through the combinational logic between sequential elements. The switching activity information calculated during propagation is stored at the pins and reported.

4.4.3 Power measurements for the FIFO model in MAGMA

Power measurements for the FIFO model are carried out using the MAGMA power simulator. Each of the input signal activities are manually fed to the MAGMA power simulator to generate power numbers. An attempt to generate power numbers using VCD (value change dump) files was also done, but this resulted in wrong results. MAGMA could not correctly read the VCD files. It computed incorrect switching activity. Hence power measurements are carried out by individually setting activities of all input and output nets and propagating them through the combinational logic between sequential elements.

Propagation of the activity through a logic cell can be done by setting the probabilities and toggle rates of the input and output signals [35]. The probability that the signals have logic '1' and the toggle rates are set according to the two variables 'r' and 'α_F' introduced in the previous section. Basically, the input and output signals of the model are classified under two parts, namely the control part and data part. The former is due to the activities occurring at the control part mainly from the write and read signal modeled by 'r', while the latter is a result of switching activity of the incoming data bits, modeled by 'α_F'.

For instance, r=0.5 would imply that write and read signals would be active high for 50% of the total time and therefore the probability that the write and read signals have logic '1' is 0.5 and the toggle rate would be half of the clock frequency. A value of α_F =0.75 would imply that 75% of the data bits would flip with each incoming data. Thus we set the probability of the input and output data bits as 0.5 and the toggle rates to three quarters of the toggle rate of the write signal. Thus the toggle rates of each of the individual input and output nets of the entire circuit are set in accordance with the clock frequency.

Unfortunately, r and α_F are parameters that cannot be influenced much by the NoC designer, as they are application dependent. On the other hand, the number of FIFO places is an important design parameter. Therefore, it is useful to express power consumption of a FIFO as a function of the number of places in the FIFO, which is done in section 4.4.3.5.

The power numbers obtained from these experiments are analysed and compared with the mathematical power model. In this way, we can fill in the constants occurring in the mathematical model from the power numbers obtained from the MAGMA power simulator by means of several experiments. Thus, in the end, we can arrive at a power model for the FIFO in co-ordinance with MAGMA and thus in the future replace the MAGMA power simulations with the mathematical model in the early phases of SoC design. In sections 4.4.3.1 to 4.4.3.4, different experiments are conducted to measure internal power, leakage and switching power, and to experiment with different FIFO sizes.

4.4.3.1 Experiments for Measurement of Internal power

The experiments are initially performed at a clock frequency of 100 MHz but are also repeated at a clock frequency of 500 MHz. This is done for three main reasons. First of all, the effect of increasing clock frequency on power consumption had to be observed. Secondly, it was seen that until 500 MHz, the leakage power is under tolerable limits, above which it starts to increase linearly. Lastly, 500 MHz is considered to be the ideal operating clock frequency as used by most of the current standard NoC designs.

Table 4 depicts the power numbers from MAGMA measurements along with the predicted numbers from the power model at 500 MHz of clock frequency given below. The first two columns are respectively the probability of write or read actions representing the control part, 'r', and the probability of the bit flips in the incoming data representing the switching in the data part, 'α_F'. In the long run, we assume that the number of write actions equals the read actions.

	Probability of writes/reads (r)	Probability of switching data(α_F)	Leakage power (µW)	Internal power(µW) (actual)	Internal power(µW) (predicted)
1.	0,25	0,25	9,6	100,3	107,466
2.	0,25	0,5	9,6	121,4	144,591
3.	0,25	0,75	9,6	142,5	181,716
4.	0,25	1	9,6	163,6	218,841
5.	0,5	0,25	9,6	142,4	169,265
6.	0,5	0,5	9,6	184,6	206,39
7.	0,5	0,75	9,6	230,7	243,515
8.	0,5	1	9,6	273	280,64
9.	0,75	0,25	9,6	192,4	231,064
10.	0,75	0,5	9,6	255,7	268,189
11.	0,75	0,75	9,6	319	305,314
12.	0,75	1	9,6	382,4	342,439
13.	1	0,25	9,6	238,2	292,863
14.	1	0,5	9,6	322,6	329,988
15.	1	0,75	9,6	407,1	367,113
16.	1	1	9,6	491,5	404,238
				%Mean square Error	13,68%

Table 4: Power numbers from MAGMA measurements against predicted numbers from the power model for the internal power @ 500 MHz of clock frequency

Figure 13 depicts the 3D representation of the internal power plotted against r and α_F for all values, as depicted in the table for a 4-place FIFO at 500MHz. Figure 13 represents only the realistic values of r and α_F.

From the measured power numbers, we subsequently carry out multiple regression analysis using the least square error method [36] in MATLAB, to derive the constants of the power model for internal power.

Multiple regression estimates the outcomes (dependent variables; here internal power) which may be affected by more than one control parameter (independent variables; here r and α_F) or there may be more than one control parameter being changed at the same time.

The constants thus obtained by this method are back- substituted in the model to obtain the predicted values for internal power as shown in Table 4 and yield a minimum square error (MSE). This MSE should be within the tolerable limit. From our experiments, we obtain the following values for the constants.

$k_1 = 247.19\mu W$; $k_2 = 148.5\mu W$; and $k_3 = 8.542\mu W$.

The mean square error thus obtained is approx 13.68% which is acceptable. This error is relative to the accuracy of the MAGMA prediction. Generally, for most energy models at architecture level as referred in the literature [42], an accuracy of 25 % is accepted.

Therefore after filling the constants in the power model for internal power as shown below, $P_{\text{int}} = k_1.r + k_2.\alpha_F + k_3$ we get

$P_{int} = 247.196.r + 148.5.\alpha_F + 8.542$ for a clock frequency at 500 MHz.

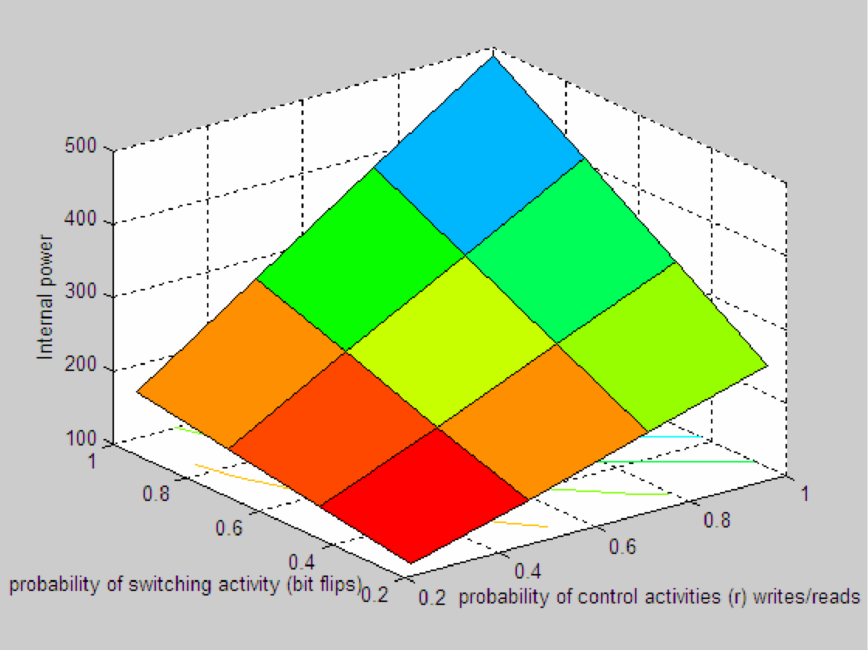


Figure 13: 3D graph of internal power plotted against α_F and r for a 4-place FIFO at 500MHz

Additionally, we also plot the actual internal power along with the predicted internal power to observe the deviation of measured power from the model.

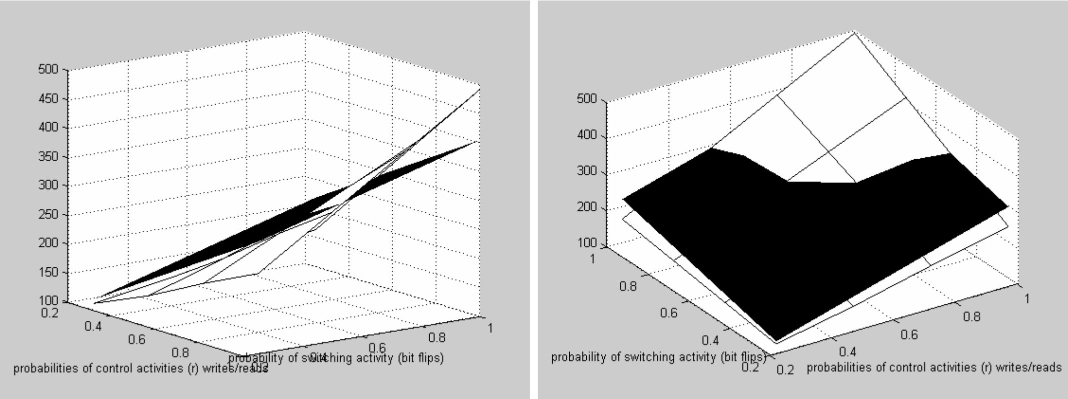


Figure 14: Overlapped 3D graph of actual (white) vs predicted (black) internal power against α_F and r for a 4-place FIFO at 500 MHz in two different angles

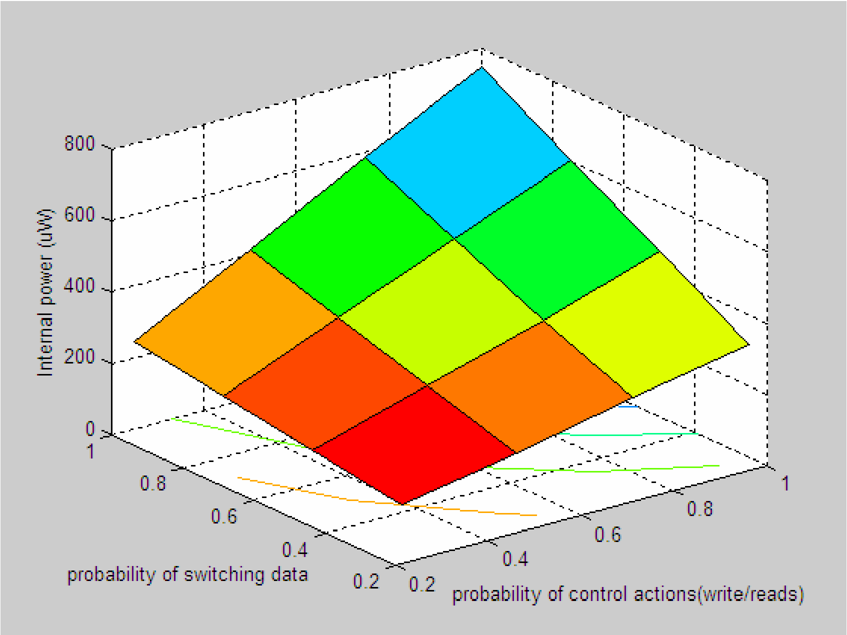


Figure 15: 3D graph of internal power plotted against α_F and r for an 8-place FIFO at 500MHz

4.4.3.2 Experiments for Measurement of Leakage power

Leakage power is the static power consumption by a gate due to the reverse bias current flowing from the source to ground, regardless of the state of the gate and switching activity. In the power model of section 4.3, leakage power is a constant (for a given clock frequency). The measurements detailed in Table 4 confirm that this is correct. In the remainder, we analyse the effect of leakage power for varying clock frequencies.

	Clk Frequency (MHz)	Leakage Power (µW)	Area (µm2)	Power density mW /µm2
1.	100	8,7	4598	1,89
2.	300	8,9	4599	1,93
3.	500	9,6	4617	2,07
4.	1000	16,2	4775	3,39
5.	1500	21,1	4881	4.32

Table 5: Leakage power numbers against clock frequency for a 4-place FIFO

	Clock Frequency (MHz)	Leakage power (µW)	Area (µm2)	Power density mW /µm2
1.	100	15,8	8733	1,81
2.	300	15,7	8725	1,79
3.	500	16,6	8744	1,89
4.	1000	27,7	9088	3,04
5.	1500	34,4	9232	3.72

Table 6: Leakage power vs clock frequency for an 8-place FIFO buffer

It is easily inferred from Figure 16 that as the clock frequency is increased, both for the 4-place and the 8-place FIFO, the leakage power remains almost constant until a certain point and later on increases linearly.

As the clock frequency is increased, the clock skew and delay specs become tighter. Clock skew is the difference between clock signal arrival times between various sub-components on a chip. For increased clock frequencies, a negative slack occurs. Slack is the amount of margin by which timing constraints are met. A negative slack on the critical path or the slowest path means that the circuit is not met for the given cycle time (or the maximum operating clock frequency). Thus the slack must always be positive. For e.g., if 5% skew was tolerable at lower frequency of say 100 MHz, it becomes 0.5% at a clock frequency of 1 GHz. To achieve the new specs, the clock buffers will have to be upsized (to drive the clock edges faster). This upsizing costs an area of almost 4% and also leakage power, as larger transistors consume more leakage power.

In the current design, the maximum clock frequency with which it can operate is 1.5 GHz, above which the design fails as it doesn't meet the timing constraints.

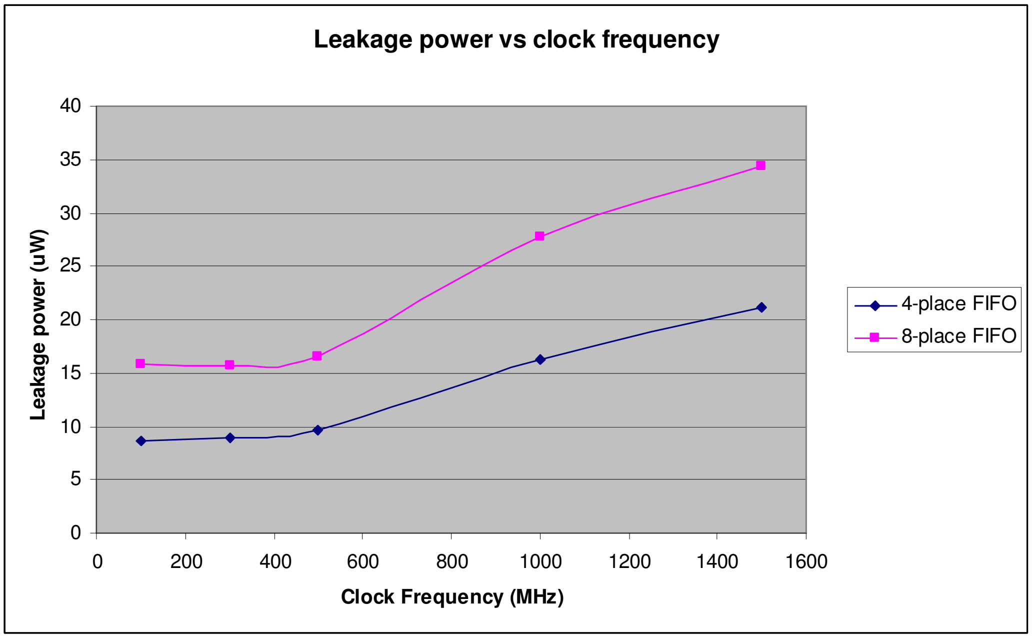


Figure 16: Graph representing leakage power vs. clock frequency for a 4-place FIFO and an 8-place FIFO

4.4.3.3 Experiments for Measurement of switching power

Switching power is basically due to charging and discharging of capacitors. It mainly consists of the power consumed as a result of writing and reading the data and an additional component, namely the clock power, consumed by the toggling of the clock. Table 7 shows the switching power consumption for various values of r and α_F for a 4-place FIFO at 500MHz.

	Probability of writes/reads (r)	**Probability of switching data(α_F)**	**switching power (µW)**
1.	0	0	12,5
2.	0,25	0,25	23,9
3.	0,25	0,5	27,1
4.	0,25	0,75	27,3
5.	0,25	1	30,4
6.	0,5	0,25	38,3
7.	0,5	0,5	41,7
8.	0,5	0,75	45,2
9.	0,5	1	48,4
10.	0,75	0,25	56,3
11.	0,75	0,5	59,2
12.	0,75	0,75	61,6
13.	0,75	1	66,3
14.	1	0,25	62,9
15.	1	0,5	69,6
16.	1	0,75	77,9
17.	1	1	84,2

Table 7: Power numbers from MAGMA measurements for switching power and leakage power for 4-place FIFO @ 500 MHz

The power consumed on a write signal includes the power consumed to store the incoming data with different switching activity in the FIFO buffers and the activities at the control part as shown below:

$$P_{write} = r.P_{ctrl} + \alpha_F.P_{store}$$

$$P_{read} = r.P_{ctrl} + \alpha_F.P._{retreive}$$

The individual components of P_{write} and P_{read} are obtained by summing up the individual cells contributing to the write and read actions from the detailed power report obtained after simulations from MAGMA. The individual components, P_{store}, $P_{retreive}$ and P_{ctrl} are measured for a 4-place FIFO and generalised for all values of r and α_F as follows:

$$P_{store} = 12.33 \text{ µW}$$

$$P_{retrieve} = 13 \text{ µW}$$

$$P_{ctrl} = 23.35 \text{ µW}$$

We fill in the values for different constants for the power model for P_{write} and P_{read}, we get

$$P_{write} = 12.33\alpha_F + 23.35r \text{ and } P_{read} = 13\alpha_F + 23.35r$$

	Probability of writes/reads (r)	**Probability of switching data(α_F)**	**P_{write} (actual) (µW)**	**P_{write} (predicted) (µW)**
1.	0,25	0,25	5,7	8,915
2.	0,25	0,5	7,3	11,9975
3.	0,25	0,75	8,15	15,08
4.	0,25	1	8,95	18,1625
5.	0,5	0,25	14	14,7475
6.	0,5	0,5	15,75	17,83
7.	0,5	0,75	16,35	20,9125
8.	0,5	1	17,95	23,995
9.	0,75	0,25	20,8	20,58
10.	0,75	0,5	21,9	23,6625
11.	0,75	0,75	24,55	26,745
12.	0,75	1	26,9	29,8275
13.	1	0,25	25,8	26,4125
14.	1	0,5	29,15	29,495
15.	1	0,75	32,7	32,5775
16.	1	1	35,85	35,66
			% Mean Square Error	**13,8%**

Table 8: Actual and Predicted values for Power consumption due to write activity for a 4-place FIFO at 500MHz

After a detailed power analysis, we get power numbers for P_{write} as shown in Table 8. Note that these models are with respect to a 4-place FIFO.

In order to measure the accuracy, the % mean square error is calculated, and an error of 13.8% is obtained. A similar result can be obtained for P_{read}. This shows that the models for P_{write} and P_{read} are reasonably accurate.

The third component of the switching power is the P_{clk}, which is obtained when there is neither switching activity of the incoming data nor the write or read actions. This is observed after conducting experiments with values of r and α_F equal to 0 as shown from experiment 1 of Table 7. Thus we get,

$P_{clk} = 12.5$ µW.

Through literature study [37], it is found that the clock power accounts to about 75 % of the total switching power. This is basically on account of the clock network power which consists of several buffers and repeaters. It is found that large clock trees consume more clock power. In the current power measurements, it is observed that P_{clk} amounts from 18 to 50% of the switching power. This difference is accounted for by the amount of logic available in the design.

4.4.3.4 Experiments for power measurements for a 4-size FIFO

Thus to summarize, the total power consumption for a 4-place FIFO according to the abstract power model is as follows (see section 4.3):

$P_{total} = (2P_{ctrl} + k_1)r + \alpha_F(P_{store} + P_{retrieve} + k_2) + k_3 + P_{clk} + P_{leak}$

We get the concrete power model for a 4-size FIFO by substituting all the concrete results from the previous subsections as follows:

$$P_{total} = 173.83\alpha_F + 293.89r + 30.642$$

This model is then compared with that of the measured power numbers for the total power consumption obtained from MAGMA. Therefore, Table 9 shows the actual and the predicted numbers for the total power consumption, resulting in a mean square error of 13.39% which is quite acceptable.

	Probability of writes/reads (r)	**Probability of switching data(α_F)**	**P_{total} (actual) (µW)**	**P_{total} (predicted) (µW)**
1.	0,25	0,25	126,1	147,5735
2.	0,25	0,5	158,1	191,031
3.	0,25	0,75	171,8	234,4885
4.	0,25	1	203,6	277,946
5.	0,5	0,25	194,2	221,0475
6.	0,5	0,5	239,8	264,505
7.	0,5	0,75	285,5	307,9625
8.	0,5	1	330,9	351,42
9.	0,75	0,25	270,6	294,5215
10.	0,75	0,5	339,2	337,979
11.	0,75	0,75	390,2	381,4365
12.	0,75	1	458,3	424,894
13.	1	0,25	306,3	367,9955
14.	1	0,5	397,5	411,453
15.	1	0,75	494,5	454,9105
16.	1	1	585,3	498,368
			% Mean square Error =	13.39%

Table 9: Actual and predicted power numbers for the total power consumption for a 4-size at 500 MHz

4.4.3.5 Experiments for power measurement for different FIFO sizes

The number of places in a FIFO buffer is a parameter that is explored in order to model the power consumption of a FIFO as a function of the number of places.

Figure 18 depicts the graphs of power consumption for different FIFO sizes for different values of r and α_F. We see that the power consumption is more or less linear with respect to the number of FIFO places.

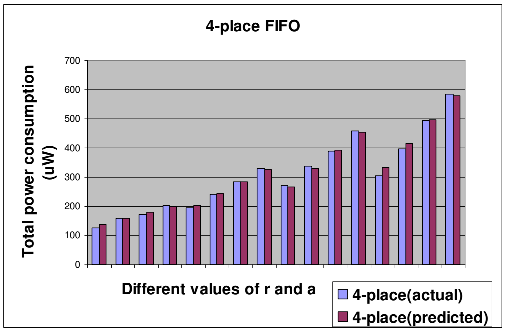
We can therefore derive the FIFO power consumption as a function of r, α_F and n by means of linear regression, as shown below:

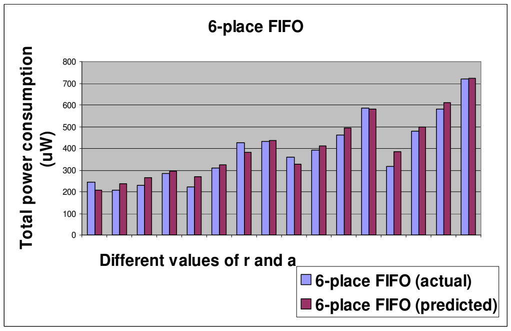
$$P_{total} = n(21.73\alpha_F + 36.73r + 7.66) + 153.73\alpha_F + 113.93r$$

Therefore after building up the concrete power model, accuracy of the model is measured and compared with that of the power numbers obtained from the experiments. Table 10 shows the actual and the predicted numbers for the total power consumption for a 4-place FIFO at 500 MHz. Similarly, this is done for different values of FIFO sizes. A graphical representation is given in Figure 17. Note that the predicted results in Table 10 deviate to some extent from the results of Table 9 because the two underlying models have been derived in different ways. The concrete 4-place FIFO model has been derived in a compositional way from the component building blocks of the 4-place FIFO, whereas the above model for the n-place FIFO is obtained via direct linear regression on all the measurements for all the FIFOs of various sizes. Note that concrete numbers for the 4-size FIFO is more accurate than the generic model. The average mean square error of the overall generic model therefore is around 19% which is quite acceptable.

	Probability of writes/reads (r)	**Probability of switching data(α_F)**	**P_{total} (actual) (µW)**	**P_{total} (predicted) (µW)**
1.	0,25	0,25	126,1	155,9525
2.	0,25	0,5	158,1	216,1025
3.	0,25	0,75	171,8	276,2525
4.	0,25	1	203,6	336,4025
5.	0,5	0,25	194,2	221,115
6.	0,5	0,5	239,8	281,265
7.	0,5	0,75	285,5	341,415
8.	0,5	1	330,9	401,565
9.	0,75	0,25	270,6	286,2775
10.	0,75	0,5	339,2	346,4275
11.	0,75	0,75	390,2	406,5775
12.	0,75	1	458,3	466,7275
13.	1	0,25	306,3	351,44
14.	1	0,5	397,5	411,59
15.	1	0,75	494,5	471,74
16.	1	1	585,3	531,89
		% Mean Square Error		**19%**

Table 10: Actual and Predicted values for total power consumption for a 4-place FIFO at 500MHz





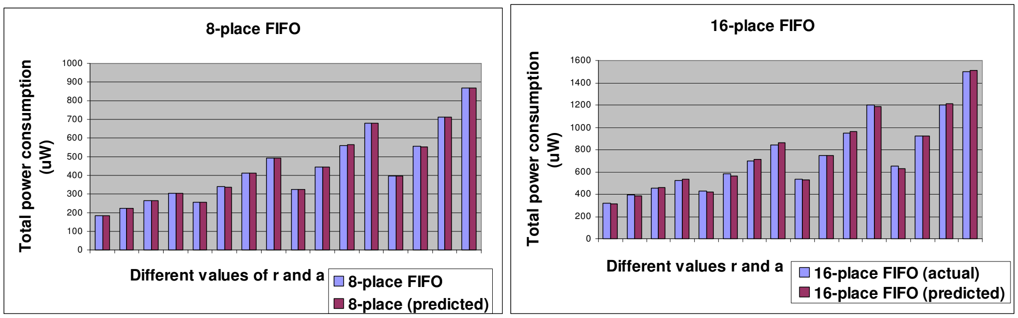


Figure 17: Actual and predicted values of total power consumption for different values of r and a for different FIFO sizes

Further, it is observed that a FIFO with size 6, which is not a factor of 2^n, sometimes consumes more power than an 8-place FIFO (shown in dotted circle in the graph below). It was also observed that the power consumed by the control part of a 6-place FIFO is 30% more than that of an 8-place FIFO. The reason behind this increase is the wastage of power in the unused bits for a 6-place FIFO. Additionally it utilises 22% more area in the control part due to additional logic needed to implement the counter wrap around, although the overall area of an 8-place FIFO is still more than the 6-place FIFO.

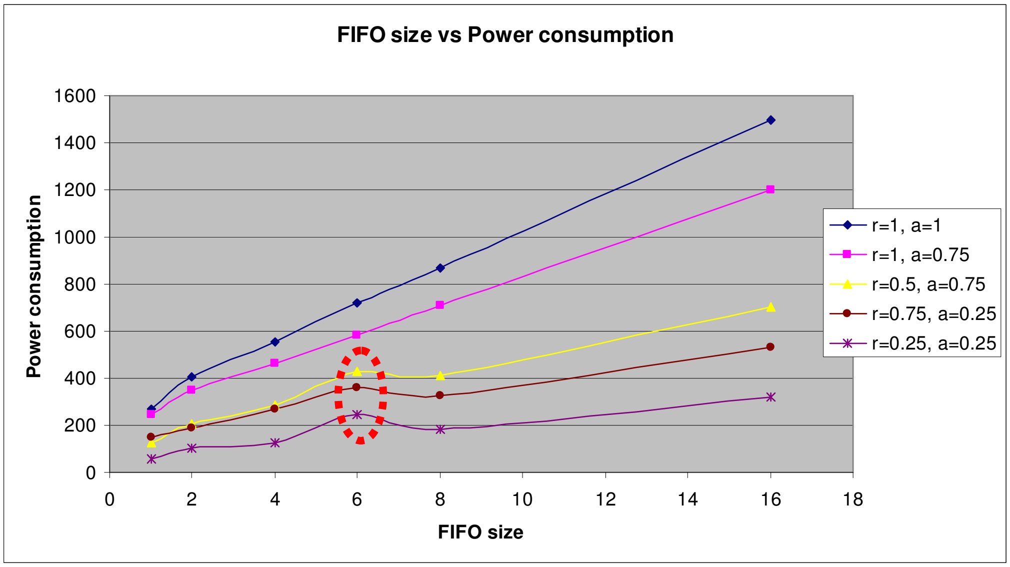


Figure 18: Power consumption of different FIFO sizes

4.5. Analysis of FIFO power consumption

Pie charts of the final power model are plotted to observe the distribution of various components of power for typical values of r and α_F. Figure 19 shows the pie chart of the three main components of power consumption as specified by MAGMA for a 4-place FIFO. From both the charts, we observe that the internal power (part of dynamic power) is the main component of the total power consumption at 500 MHz. The reason for this is mainly attributed to the sizes of the cells available in the MAGMA power library. Smaller cells need to be driven for a larger time, as a result the short circuit period that determines the internal power consumption also increases.

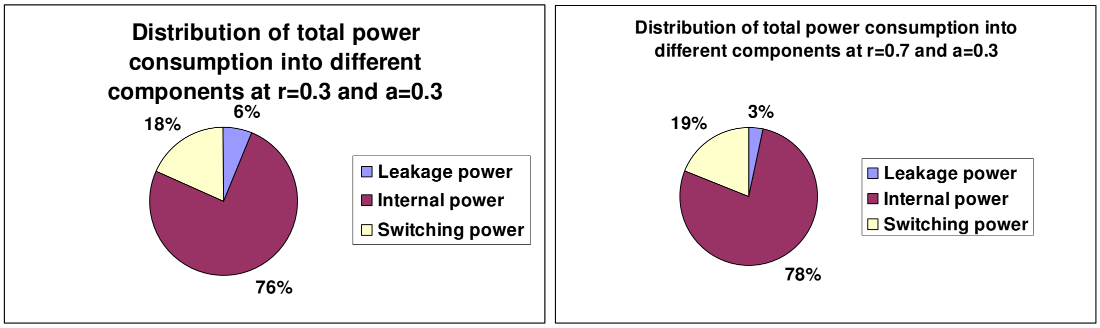


Figure 19: Pie chart showing the distribution of different components of power with typical values of r and α_F

Figure 20 refines Figure 19 in the sense that it breaks down the switching power into three parts. Here, P_{clk} is a constant and therefore its effect is more seen when the switching activities and read and write rates are small. While during high switching activity of the incoming data bits and high read and write rates, its effect is lessened.

Figure 21 shows the pie chart of the switching and control components of power consumption. The figure in the right part of the chart shows that the control part consumes more power than the switching part due to the increased control activity.

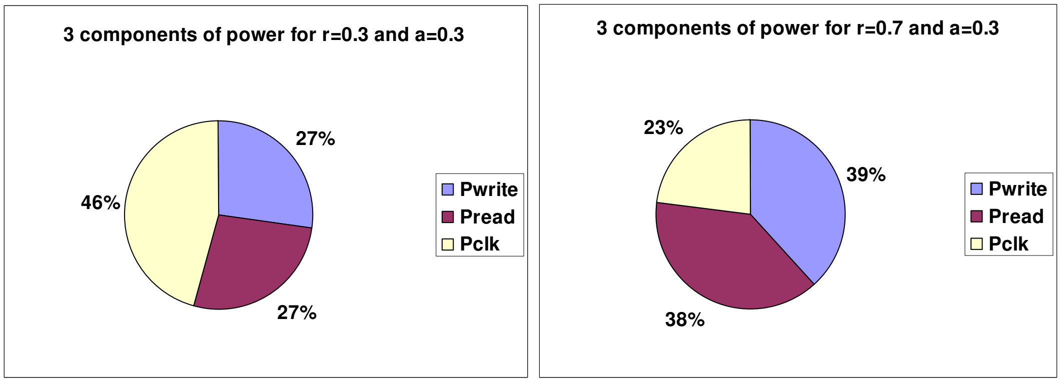


Figure 20: Pie chart showing the three main components of switching power with typical values of r and α_F

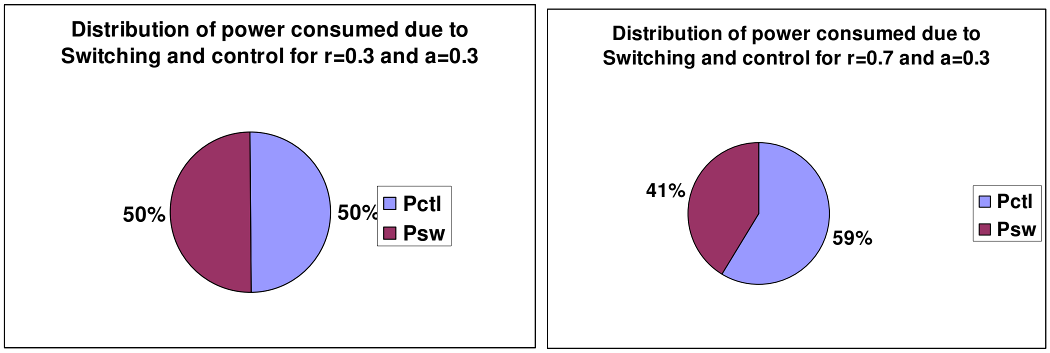


Figure 21: Pie chart showing the switching and control component of power with typical values of r and α_F

Chapter 5 **Design and Energy Model of a Router**

5.1. Introduction

In typical networks-on-chip, processing nodes are connected by means of intermediate routers that forward packets to their destinations by means of a routing mechanism. Having intermediate inter-processing routers is more useful than one global router, for the reason that this enables efficient local forwarding and management of packets. However, these routers contribute significantly to the overall power consumption on a chip. From the studies performed by MIT [38] on a 16-tile NoC, it is found that the processing elements including the routers consumed around 36% of the total power consumption, out of which around 40% of power was consumed by the routers. This motivates to develop an energy model for a typical NoC-based router. Therefore, in this chapter, we will discuss the state of the art of NoC routers (section 5.2) followed with the architecture of the router used for our experiments (section 5.3), and finally the energy model for a typical network-on-chip router (section 5.4). Note that here we resort to an energy model instead of a power model, because we are interested in the amount of energy used to transfer one flit from the input of the router to the router output.

5.2. State of the art of Routers

In this section, state of the art of NoCs and their routers are discussed. The details are obtained from the literature survey done by a research group at the University of Brazil [40]. The most important details concerning the router are shown here. This provides a benchmark for designers to design a router architecture taking into consideration several parameters as shown below.

NoC	Topology / Routing	Word Size	Buffering	QoS Support
***SPIN* - 2000**	Fat-tree / Deterministic and adaptive	32 bits data + 4 bits control	Input queues + 2 shared output queues	-
***aSOC* - 2000**	2D mesh (scalable) / Determined by application	32 bits	None	Circuit-switching (no wormhole)
Dally - 2001	Folded 2D torus / XY source	256 bits data + 38 bits control	Input queues	GT - virtual channels
***Nostrum* - 2001**	2D mesh (scalable) / Hot potato	128 bits data + 10 bits control	Input and output queues	-
Sgroi - 2001	2D mesh / NA	18 bits data + 2 bits control	-	-
***Octagon*- 2001**	Chordal ring / Distributed And adaptive	Variable data + 3 bits control	-	Circuit-switching

Marescaux - 2002	2D torus (scalable) / XY blocking, hop-based, deterministic	16 bits data + 3 bits control	Input queues	2 virtual channels (to avoid deadlock)
Bartic – 2003	Arbitrary (parameterizable links) / Deterministic, virtual-cut-through	Variable data size (n bits) + 2 bits control / link	Output queues	Injection rate control, congestion control
***Æthereal* - 2002**	2D mesh / Source	32 bits + 2 bits control	Input queues	GT,BE (circuit switching)
***Eclipse* - 2002**	2D sparse hierarchical mesh / NA	68 bits	Output queues	-
***Proteo* - 2002**	Bi-directional ring / NA	Variable control and data sizes	Input and output queues	-
SOCIN - 2002	2D mesh (scalable) / XY source	n bits data + 4 bits control (parameterizable)	Input queues (parameterize-able)	No
***SoCBus* - 2002**	2D mesh / XY adaptive	16 bits data + 3 bits control	Single position input and output buffers	Circuit-switching
QNOC - 2003	2D mesh regular or irregular / XY	16 bits data + 10 bits control (parameterizable)	Input queues (parameterize-able) + Output queues (single position)	GT - virtual channels, (4 different traffic)
T-SoC – 2003	Fat-tree / Adaptive	38 bits maximum	Input and output queues	GT - 4 virtual channels
Xpipes - 2002	Arbitrary (design time) / Source static (street sign)	32, 64 or 128 bits	Virtual output queues	No
***Hermes –* 2003**	2D mesh (scalable) / XY	8 bits data + 2 bits control (parameterizable)	Input queues (parameterize-able)	No

- : No information available

GT : Guaranteed Throughput
BE : Best Effort

Table 11: State of the art NoC Routers

A typical router is composed of input buffers and output buffers to store packets during contention, an arbiter to arbitrate the incoming and outgoing packets and a cross-bar to route the packets to the required output.

From the above table, we see that a 2D mesh topology with XY routing is most commonly used due to its simplicity and ease of network scalability. Word sizes range from 8 to 64 bits which are similar to the data width of the current processor architectures. Both input and output buffering are common choices. Input buffering implies a single queue per input. However having an additional output buffering increases the total cost and area. Therefore, we avoid the use of output buffering and confine only to input buffers. The most commonly found form of guaranteeing QoS in NoCs is through the use of circuit switching. In circuit switching, a path is established before the packets are sent by allocating a sequence of channels between source and target. The disadvantage of this approach is that the bandwidth is wasted especially if the communication path is not used at every moment during the period the connection is established. In addition, since most approaches combine circuit switching with best effort techniques, this consequently results in the increase of the router area. However, virtual channels are one way to achieve QoS without compromising the bandwidth especially with TDMA techniques [44].

5.3. Router Architecture

The router architecture that is implemented is shown in Figure 22. It is composed of input FIFO buffers, a fully connected crossbar and an arbiter. The components are not optimised but rather simplified to measure the energy consumed to transfer a flit through a router. If the components were to be optimised, it would only influence in the constants in the energy model being different whereas the energy model would still remain the same.

A 3-place FIFO buffer equivalent to the size of one flit is used at the input. Each flit is made of three 34-bit words (32 data bits and 2 control bits). It can contain headers (start of packet), payload and trailers (end of packet). Thus, the FIFO buffer is just sufficient for storage of one flit at the input. However, as the words vacate the buffer, new flits can arrive, but can move across the router only after the previous flit has completely moved out of the router. In a situation without contention, this is acceptable. Absence of contention is a reasonable assumption for some connection-based services, such as the GT service of *Æ*thereal. The choices for our router design are in fact consistent with the *Æ*thereal NoC design, which is a state of the art NoC.

A 5x5 fully connected crossbar with multiplexers is used to aggregate every input to the outputs. A 5x5 router is typical in a mesh topology with an IP block connected to each router. There might not necessarily be any communication between some inputs and outputs. In such cases the crossbar design can be optimised to improve its performance. Each multiplexer is controlled by the arbiter that determines which input should be directed to the output.

Arbitration of the router occurs at a granularity of three words (or a flit). The routing mechanism is source routing, i.e., the header contains the path information from source to destination. Each router removes as many bits ($\log_2 N$, N being the number of outputs) from the path as necessary to determine to which output the packet must go. If the word is a packet header, the arbiter assigns a route to the data word and also saves a copy of the route in its memory. The following words or the payload and the trailer subsequently follow the same path. Once the trailer is detected and sent to its destination, the memory is refreshed with the next incoming flit.

A one word output buffer is used in order to increase the potential operating frequency of the router.

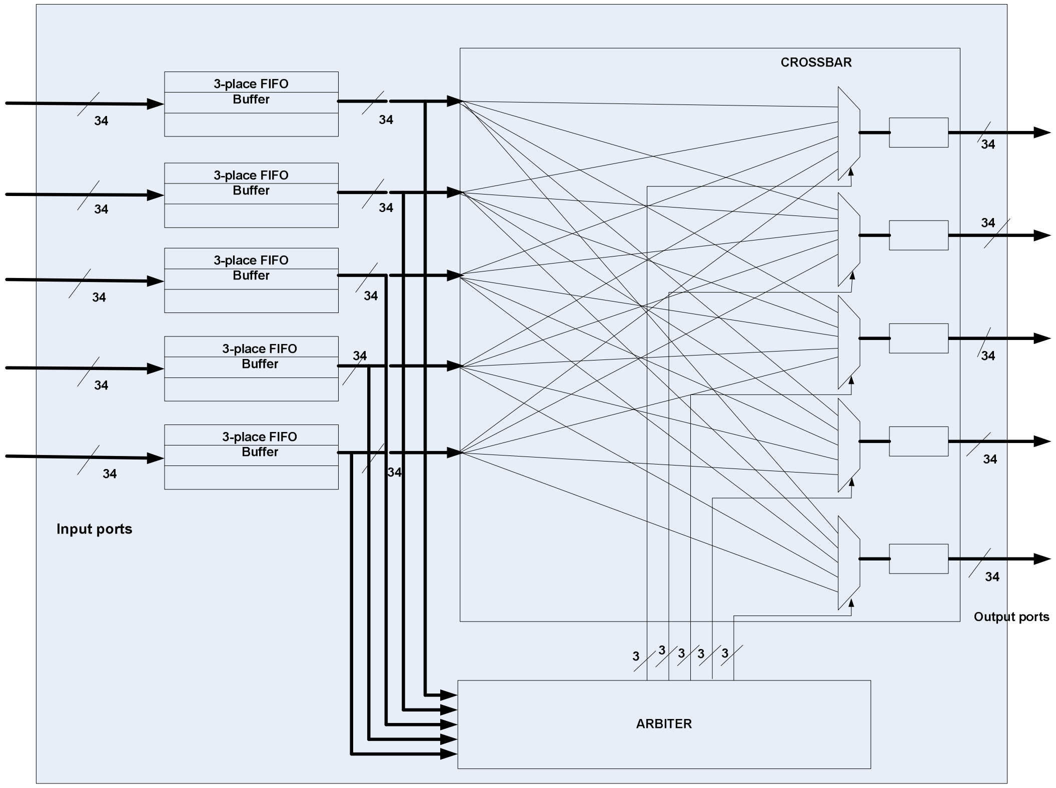


Figure 22: Router architecture

5.4. Router Energy Model

In this section, an energy model for a router is developed based on Figure 22. This section also briefly describes the sources of energy consumption. Recall that we are interested in determining the energy consumed to transfer one flit from input to output.

The total energy consumption in a router for the transfer of one flit is a combination of the individual components given as follows

$E_{router} = E_{FIFO} + E_{arbiter} + E_{crossbar}$

In the current router, there are five 3-place FIFO buffers. However, only one is involved in the transfer of one flit from input to output.

Energy consumed at the arbiter is due to the arbitration action which includes setting up of a path for the packets to traverse from the input port to the output of the router based on the control bits.

Energy consumed at the crossbar is from the packets being routed from input ports to the output based on the path allocated by the arbiter.

Energy consumed at the input FIFO buffers are mainly due to the storing of incoming packets in the buffers and specifically due to read and write accesses at each FIFO.

5.5. Experiments for Energy Measurements on a Router

Energy estimation of the router logic is performed by modeling the design in Verilog. The synthesized Verilog design is then annotated using different test patterns representing the switching activity ranging from best case (no activity) to worst case (continuous activity) to typical case (50% activity) [43]. Here energy is measured in MAGMA as shown in Table 12 on a per second basis. Therefore in order to measure the energy consumed for a flit to pass from the input to outputs of the router, we need to measure the number of flits passing per second. The design of the router is such that it can transfer one flit every three clock cycles (each clock cycle is 2ns). Therefore a total of 166 million flits pass from the input to the output in a second, if we assume a utilization of 100%. For our energy measurements this can be achieved by assuming a write rate r=1.

Table 12 shows the energy measurements of a router and its components at a clock frequency of 500 MHz.

Router

Switching activity (α_R)	Leakage power (mW)	Internal power(mW)	Switching power (mW)	Total energy (mWsec)
0	0,0512	2,2	1,3	3,5
0,25	0,0513	2,5	1,3	3,9
0,5	0,051	2,8	1,6	4,4
0,75	0,0513	3	1,7	4,8
1	0,0513	3,2	1,5	4,8

FIFO

Switching activity (α_F)	Leakage power (µW)	Internal power(µW)	Switching power (µW)	Total energy (µWsec)
0	8,1	59,8	15,3	83,3
0,25	11,7	195,3	64,8	271,7
0,5	9,9	250,8	55,8	316
0,75	9,4	325,9	79,7	415,5
1	11,7	398,6	85,8	496,1

Crossbar

Switching activity (α_C)	Leakage power (µW)	Internal power(mW)	Switching power (mW)	Total energy (mWsec)
0	4,8	0	0	0,0048
0,25	4,8	0,765	0,423	1,2
0,5	4,9	1	0,647	1,7
0,75	5	1,3	0,903	2,2
1	5,1	1,5	1,2	2,7

Arbiter				
Switching activity (α_A)	**Leakage power (µW)**	**Internal power(mW)**	**Switching power (mW)**	**Total energy (mWsec)**
0	4,7	0,606	0,264	0,875
0,25	4,7	0,845	0,452	1,3
0,5	4,7	0,848	0,454	1,3
0,75	4,7	0,853	0,457	1,3
1	4,7	0,855	0,458	1,3

Table 12: Power/Energy measurements of Router and its components@ 500MHz clock freq

5.5.1. FIFO Energy model

Since the power model of an n-place FIFO model is already explained in Chapter 4, only the final energy model (in µWsec) of an n-place FIFOs is given. Note that the model for the 34-bit wide FIFO turns out to be the same as the model for the 32-bit wide FIFO of Chapter 4. The effect of the two extra bits has a negligible effect on the constants.

$$E_{total} = n(21.73\alpha_F + 36.73r + 7.66) + 153.73\alpha_F + 113.93r$$

Note that a 1-flit buffer corresponds to a 3-place buffer. For a 3-place FIFO, the energy model is as follows:

$$E_{total} = \alpha_F 218.92 + 224.12r + 22.98$$

Where α_F is the switching activity of the data bits and r is the rate of writes and reads.

5.5.2. Crossbar Energy Model

Energy consumed by the crossbar can be further broken down into three components as shown below:

$$E_{crossbar} = E_{int} + E_{sw} + E_{leak}$$

Replacing each of the three components with a relation expressed in terms of α_C, the switching activity of the data bits, we get the following energy model (in mWsec) for a fully connected crossbar (5x5 crossbar).

$$E_{crossbar} = 1.002\alpha_C + 0.515 + 1.0348\alpha_C + 0.1465 + 0.005$$

Simplifying the above expression, we get the energy model for the crossbar as:

$$E_{crossbar} = 2.0368\alpha_C + 0.6665$$

5.5.3. Arbiter Energy Model

The energy model for the arbiter can be represented in a similar way as shown for the crossbar:

$$E_{arbiter} = E_{int} + E_{sw} + E_{leak}$$

Each of these components is expressed as a function of α_A, the switching activity of the data bits entering the arbiter. Therefore the above relation can be substituted with constants and a function of α_A as follows:

$$E_{arbiter} = 0.014\alpha_A + 0.8415 + 0.0084\alpha_A + 0.45 + 0.0047$$

Simplifying the above expression, we get the energy model (in mWsec) for the arbiter as:

$$E_{arbiter} = 0.0224\alpha_A + 1.2962$$

5.5.4. Router Energy Model

Combining the energy models for the FIFO, arbiter and crossbar, we get the following router energy model (in mWsec):

$$E_{router} = (\alpha_F 0.218 + 0.224)r + 0.022 + 2.0368\alpha_C + 0.6665 + 0.0224\alpha_A + 1.2962$$

Here $\alpha_F = \alpha_C = \alpha_R$, same switching activity in the FIFO and the crossbar, while $\alpha_A = 0.66\alpha_R$, that is only twice every three words, a new data arrives at the arbiter. The two control bits of the three words vary from 11 to 10 to 01 which represents a GT header, 2 payloads and an end of flit respectively. Thus on an average 2 bits out of 3 vary for the three words or for every flit.

Substituting r=1 for our current experiments, indicating 100% utilization of the router, we get the energy model for a router in mWsec as follows:

$$E_{router} = \alpha_R 2.26 + 2.2087$$

A regression analysis is performed on the router energy model as shown in the previous section with that of the energy numbers as shown in Table 12, we get a mean square error of 14% which is quite acceptable. Optimising the router design would only cause a change in the constants while the energy model remains the same.

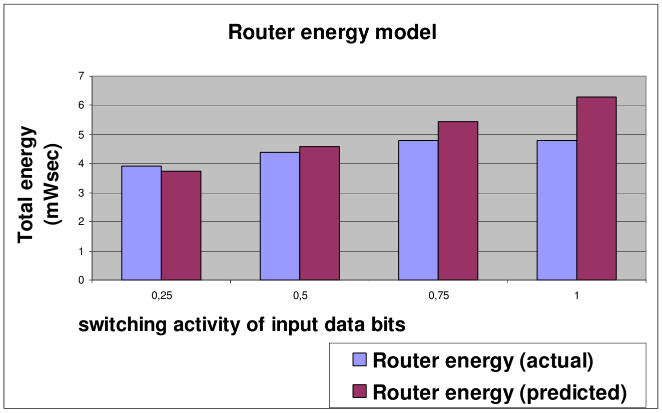


Figure 23: Actual and Predicted values of energy numbers of a router for different values for r and a

In order to arrive at an average energy model for the transfer of a flit through the router, we take the number of flits passing through the router per second as calculated above and deduce the energy model in nJ to transfer a single flit.

$$E_{router} = \alpha_R 0.013 + 0.133$$

5.6. Analysis of the Energy Measurements on a Router

From the pie chart of Figure 24, we can deduce that at 25% of input data switching activity, the energy consumed by the arbiter and the crossbar is almost the same, while the FIFO consumes around 10% of energy. Considering the extreme case of 100% of the incoming data switching activity, the crossbar becomes the bottleneck attributed to its architecture and consumes 60% of the total energy consumption followed with the arbiter with 29% of overall energy consumption while the FIFO consumes 11% of overall energy.

Another important observation was that the energy numbers measured for the 3-place FIFO used in the router matched according to the linear behaviour as observed in the section 4.4.3.4 of Chapter 4.

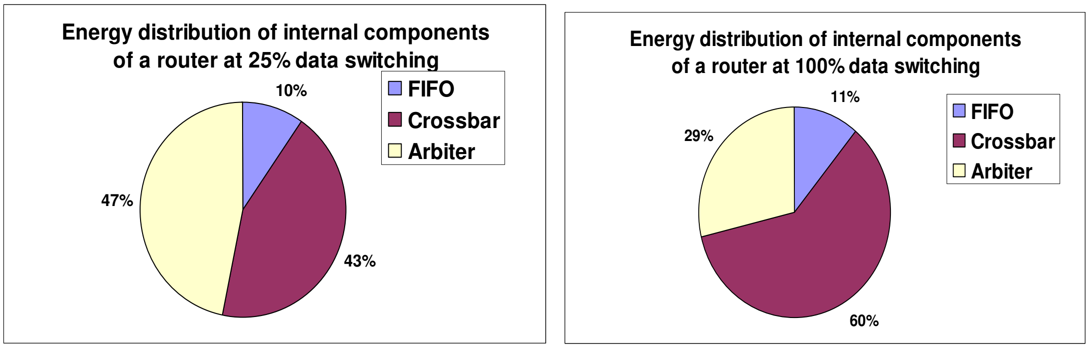


Figure 24: Pie chart of energy distribution for internal components of a router

Chapter 6 Integrated Energy Model for Routers, FIFOs and Links

In this chapter, all the components developed individually are integrated to build a network. The components comprise the routers connected together with links. By means of an integrated model and prior energy measurements of an individual router we can deduce the energy consumed by the links.

In Chapter 3, a mathematical model of a link is built. However, to experimentally validate the model, it was necessary to have realistic input and output blocks connected via the links, and at that time, a complete model such as a router was not yet designed. Hence it was decided to postpone the energy measurements for the links until this chapter. By doing so, it is ensured that the link energy consumption numbers are obtained using realistic input and output blocks.

6.1. Integrated Model: Architecture

As shown in the Figure 25 below, a complete model is built integrating two routers via a link. Each router is composed of input FIFOs, arbiter and crossbar. In order to measure the energy consumed by the link, the two routers are placed far apart in the chip. We also perform an experiment with three routers connected by two links.

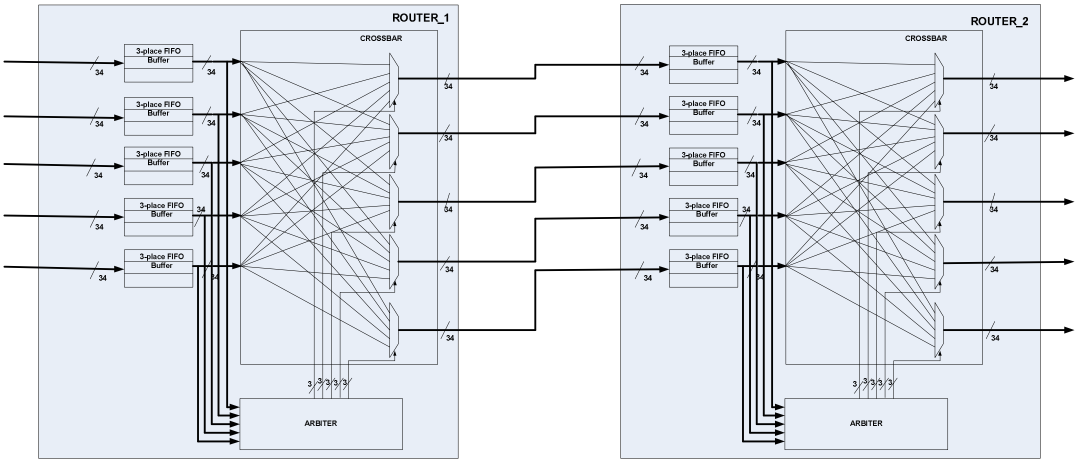


Figure 25: Integrated model of routers connected with links

6.2. Link Energy Model

In section 3.5, we have derived a link power model. Summarising the same in terms of energy per second we get the following:

$$E_{link} = \frac{1}{2} N V^2 (C_L \alpha_L + C_C \alpha_C) f + N \tau \alpha_L V I_{short} . f + N.(V I_{bias,\, wire} + V I_{leak,\, gate})$$

where N is the total number of wires in the link, C_L and C_C are the self and coupling capacitance of a wire and neighboring wires respectively, α_L is the switching activity on a wire and α_C is the switching activity with respect to the adjacent wires, τ is the short circuit period, V is the supply voltage and I_{short}, $I_{bias,\ wire}$ and $I_{leak,gate}$ are currents as explained before.

6.3. Compositional Energy Model

The abstract energy model for the whole NoC is as shown below:

$$E_{network} = n_r E_{router} + (n_r - 1) E_{link} \text{ ,}$$

where n_r is the number of routers.

6.4. Energy Measurements

As mentioned above, experiments are carried out to measure the energy consumed for the transfer of one flit by the integrated system as a whole as well as the energy consumed by the link. The distance between the routers is modified in the layout via explicit global placement. By varying the distances between the two routers, the effect of the wire length is measured.

Table 13 depicts the energy measurements for the whole integrated model with two routers at 50 MHz.

Integrated model energy numbers at a frequency of 50MHz						
	Switching activity (α)	**Leakage power (µW)**	**Internal power(mW)**	**Switching power (mW)**	**Total energy (mWsec)**	**Total energy (nJ/flit)**
1.	0,25	95,1	2,9	2,1	5,1	0,307
2.	0,5	96,8	3,1	2,4	5,6	0,337
3.	0,75	96,9	3,5	2,8	6,3	0,379
4.	1,0	97,8	3,7	3,1	6,9	0,415
link length (Distance between routers) =						500µm
Integrated model energy numbers at a frequency of 50MHz						
	Switching activity (α)	**Leakage power (µW)**	**Internal power(mW)**	**Switching power (mW)**	**Total energy (mWsec)**	**Total energy (nJ/flit)**
5.	1,0	97,6	3,9	4,4	8,4	0,506
link length (Distance between routers) =						1000µm

Table 13: Energy numbers for an integrated model with 2 routers at 50 MHz

Integrated model (with 2 routers) energy numbers at 100MHz						
	Switching activity (α)	**Leakage power (µW)**	**Internal power(mW)**	**Switching power (mW)**	**Total energy (mWsec)**	**Total energy (nJ/flit)**
6.	0,5	95,6	1,1	9,2	10,3	0,309
link length (Distance between routers) =						2000 µm
Integrated model (with 3 routers) energy numbers at 100MHz						
7.	0,5	147,2	1,2	17,7	18,9	0,567
link length (Distance between routers) =						2000 µm

Table 14: Energy numbers of integrated models with 2 and 3 routers at 100 MHz

The current trend in the dimensions of the network is such that the distance between two tiles is assumed to be around 2mm with each tile occupying an area of 4 mm^2. These numbers are based on area specifications of IP cores. This distance is estimated by calculating the individual area of the cores of a tile and summing them up to obtain the area of a single tile. We assume that the area of each IP core in a tile is 0.84 mm^2[46]. Further we assume that there are at most four IP cores within a tile, so the total area of the tile can be approximated to 4 mm^2. Thus the distance between the tiles is kept equivalent to the width of a tile. Therefore experiments were performed with realistic dimensions placing the two routers separated by a distance of 2 mm (refer experiment 6) and later with three routers (refer experiment 7).

Router energy measurements were performed at a frequency of 100 MHz and from the total energy consumption for the integrated model as in experiment 6 of Table 14, we deduce the energy consumed for a link.

Router energy numbers at a frequency of 100MHz						
	Switching activity (α)	Leakage power (µW)	Internal power(mW)	Switching power (mW)	Total energy (mWsec)	Total energy (nJ/flit)
8.	1.0	16,2	2,6	0,653	3,4	0,102
9.	0,75	16,2	2,6	0,616	3,2	0,096
10.	0,5	16,2	2,6	0,590	3,0	0,090
Link energy at a frequency of 100MHz						
11.	1,0	62,8	1,0	8,5	9,5	0,285
12.	0,5	62,8	1,0	3,3	4,3	0,129

Table 15: Energy numbers for Router and link at 100 MHz

6.5. Energy Distribution Analysis

From the energy measurements as shown in Figure 26, it is seen that the link is responsible for 16% and the routers for 84% of the total energy consumption with the two routers separated by a distance of 500 µm. As the distance between the routers is doubled from 500 µm to 1000 µm, the link consumes 31% of the total energy consumption, i.e., the energy linearly increases with the length of the link. Thus it is seen that the routers placed far apart consumed more energy than the routers placed closer to each other. This is because, to drive long wires large driving gates and repeaters are necessary and the increased switching activities in these wires consume energy.

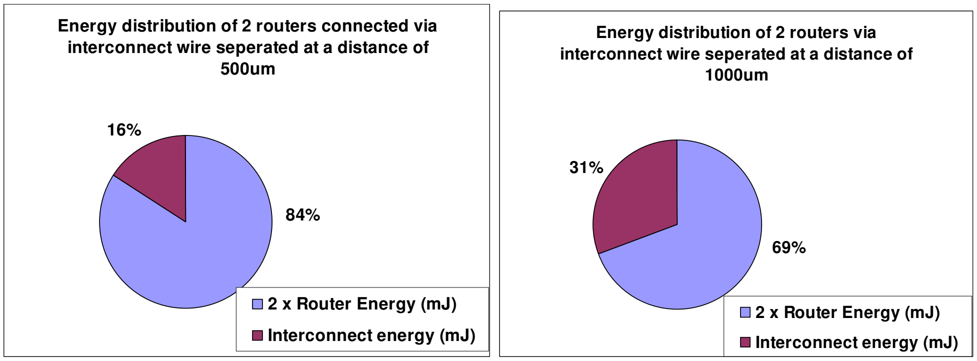


Figure 26: Energy distribution of the two routers put together vs link measured at two different distances at 50 MHz

6.6. Concrete Link Model

From the mathematical model for the link as given in section 6.2, we derive an abstract model for the link energy per flit transfer as shown below:

$$E_{link} = N(c_1\alpha_L + c_2) + N\alpha_L c_3 + Nc_4$$

where N is the number of wires in the link, α_L is the link switching activity and c_1, c_2, c_3 and c_4 are constants.

From the energy measurements as shown in Table 15, we can derive a concrete energy model for the 34-bit wide link as follows:

$E_{link} = 0.312\alpha_L - 0.027$

More measurements would be needed to determine the accuracy of this model, and to generalize it to take the link-width into account.

6.7. Concrete Energy model for the NoC as a whole

Initially a concrete energy model for a router and a link can be developed based on our experiments for a single hop network. This model can then be used to obtain energy numbers for a 2-hop model which is then validated with the energy numbers obtained from the experiments.

That is, a concrete energy model for the transfer of one flit in the network can be obtained after substituting the results of experiment 6 of Table 14 in the abstract energy model as explained in section 6.3:

$E_{network} = n_r E_{router} + (n_r\text{-}1)E_{link} = 0.090n_r + 0.129(n_r\text{-}1)$.

In order to measure the accuracy of the model, we substitute $n_r = 3$ in the model. We then get $E_{network} = 0.528$, which gives an acceptable error of 6.8%. Thus we see that the energy numbers in Table 14 are in line with the abstract energy model given in the section 6.3. Therefore the models suggest that the energy consumption for the transfer of one flit through a NoC is, as expected, linearly dependent on the number of hops in the network.

Chapter 7 **Conclusion and Future Recommendation**

In this thesis a compositional energy model of a simple communication architecture is developed besides the energy and power models of individual NoC components such as FIFO buffers, links and routers in isolation. The energy models are developed at an transaction level. Validation of the individual components is carried through detailed power analysis.

The power model of FIFO depicts a linear behavior of power consumption with respect to the number of places in the FIFO buffer. Additionally, it is also observed that the leakage power increased almost linearly with the increase in clock frequencies. Regression analysis of the FIFO model with the numbers obtained through experiments is carried out and it is seen that a mean square error of 13% is obtained which is acceptable.

In the energy measurements regarding the transfer of one flit through a router, it was observed that in the worst case (with continuous bit flips) the crossbar was a bottleneck consuming 60% of the total energy consumption. However, arbiters and the crossbar consumed almost the same amount of energy for a more realistic case with 25% bit switching activity. A regression analysis is performed on the energy model with the energy numbers obtained from the experiments, and again a mean square error of 13% is obtained which is acceptable.

When the individual components are integrated to form the complete NoC architecture (with two routers connected via a link), it is seen that the energy consumed by the link is proportional to its length, as expected. Further, it is observed that the routers placed far apart consumed more energy than placed closer to each other.

Thus the energy models developed in this thesis capture all the parameters influencing energy consumption at the low implementation levels in terms of constants and express the models at an intermediate level with parameters recognizable at this level. Some of these parameters are application dependent while some are architecture dependent. It is also observed that a reasonably good accuracy is maintained in the energy models. These models can therefore be used to estimate the energy consumption of a complete NoC. The models suggest that the energy consumption for the transfer of one flit through a NoC is, as expected, linearly dependent on the number of hops in the network.

As a continuation of this project, it is recommended that in the future energy models for the computation elements such as processors and memory elements also be developed and integrated together with the communication elements to derive an energy model for the whole SoC.

References

[1] Nilanjan Banerjee, Praveen Vellanki and Karam S. Chatha: "*A Power and Performance Model for Network-on-Chip Architectures",* Department of CSE, Arizona State University, Tempe, AZ, Proceedings of DATE, 2004. http://www.eas.asu.edu/~csrl/publications/nocpowerperf.pdf

[2] Seven Layer Model.http://www2.rad.com/networks/1994/osi/layers.htm

[3] Josias Dezeme and Renan Laurore: "Networks on Chip" : http://www.cvsi.fau.edu/codesign/Presentations%5COriginal%20Presentations%5CNoC.ppt

[4] Sun Lingao, "Interconnect-centric Design for Advanced SoC and NoC: Communication based design for Networks-on-chip", TUT, 2004. http://www.tkt.cs.tut.fi/kurssit/8404941/S04/chapter1.ppt#32

[5] A.G.Wassal and M.A. Hasan: "Low-power system-level design of VLSI packet switching fabrics", IEEE Transactions on CAD, 20:723–738, June 2001. http://www.ece.mcmaster.ca/faculty/teds/COURSES/725-folder/SWITCHING/Powerswitching.pdf

[6] Terry Tao Ye: "On-chip Multiprocessor Communication Network Design and Analysis", PhD thesis, Stanford University, Dec 2003.

[7] Jingcao Hu, Radu Marculescu, "Energy *and Performance Aware Mapping for Regular NoC Architectures*", IEEE transaction on Computer Aided Design of Integrated Circuits and Design, Vol 24, No 4, April 2005. http://www.ece.cmu.edu/~sld/pubs/papers/T-CAD_April05.pdf

[8] Scott Rixner, William J. Dally, Brucek Khailany, Peter Mattson, Ujval J. Kapasi, and John D. Owens: *"Register Organization for Media Processing"*, Computer Systems Laboratory Dept. of Electrical Engineering and Computer Science, Appeared in HPCA6 (2000).

[9] Ozgur Celebican, Tajana Simunic Rosing, Vincent J. Mooney III: *"Energy Estimation of Peripheral Devices in Embedded Systems"*, Center for Research on Embedded Systems and Technology, Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0250, USA. http://codesign.ece.gatech.edu/publications/celebi/paper/p191-celebican.pdf

[10] Steven Derrien, Sanjay Rajopadhye : "*Energy/Power Estimation of Regular Processor Arrays*", Colorado State University, Appeared in ISSS 2002. http://www.sigda.org/Archives/ProceedingArchives/Compendiums/papers/2002/isss02/htmfiles/sun_sgi/issstoc.htm

[11] Alexander Chatzigeorgiou, Stamatiki Kougia, and Spiridon Nikolaidis: "*Evaluating the Effect of Data-Reuse Transformations on Processor Power Consumption*", 2001, Electronics and Computers Division, Department of Physics, Aristotle University of Thessaloniki, 54006 Thessaloniki, Greece. http://patmos2001.eivd.ch/program/Repro%5CArt_3_5.pdf

[12] Tiwari V., Malik S. and Wolfe A: "*Power Analysis of Embedded Software*: *A First Step Towards Software Power Minimization",*. IEEE Transactions on VLSI Systems, Vol. 2.(1994) 437-445. http://www.sigda.org/Archives/ProceedingArchives/Iccad/Last20/Papers/1994/ICCAD94_P0384.PDF

[13] Tiwari V., Malik S. and Wolfe A.: "*Instruction Level Power Analysis and Optimization of Software*", Journal of VLSI Signal Processing. Kluwer Academic Press. (1996) 1-18.

[14] Sinevriotis G. and Stouraitis Th: "*Power Analysis of the ARM 7 Embedded Microprocessor*", Proc. of 9th Int. Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), (1999) 261-270

[15] Noel Eisley and Li-Shiuan Peh: "*High level Power Analysis for on-chip Networks*", Princeton University, CASES 2004. http://www.princeton.edu/~eisley/powertheory.pdf

[16] Dirk Postma, *"From RTL to Layout using Blast chip",* Master thesis, Tu/e, Aug 2002.

[17] Enrico Macii, Massimo Poncino, Sabino Salerno: "*Combining Wire Swapping and Spacing for Low-Power Deep-Submicron buses*", Apr 2003. http://easy.intranet.gr/paper_33.pdf

[18] Evgeny Bolotin, Israel Cidon, Ran Ginosar and Avinoam Kolodmy: *"Cost considerations in Networks on chip"* Electrical Engineering Department, Technion-Israel Institute of Technology, Haifa 32000, Israel 19th March 2004 http://www-ee.technion.ac.il/~ran/papers%5CNoC_cost.pdf

[19] Lance Pickup and Scott Tyson: *"Efficient Power Management in the 90-nanometer foundry reference flow": Aug-Sep 2004 issue of Chip Design Magazine.* http://www.chipdesignmag.com/display.php?articleId=76&issueId=6%09

[20] *Nuntachai Poobuapheun: Department of Electrical Engineering and computer sciences, University of California, Assignment Sep 2003,* http://bwrc.eecs.berkeley.edu/Classes/ICDesign/EE141_f03/Homeworks/ee141_Fa03_Hw4_Sol.doc

[21] Ron Ho, Kenneth W. Mai, and Mark A. Horowitz: "*The Future of Wire"* Stanford University, Invited Paper for Defense Advanced Research Project Agency and MARCO Interconnect Focus Centre, July 2000. http://mos.stanford.edu/papers/rh_ieeeproc_01.pdf

[22] Taku Uchino and Jason Cong: *"An Interconnect Energy Model considering coupling effects",* IEEE transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol 21, No 7, July 2002. http://ballade.cs.ucla.edu/~cong/papers/tcad02_power.pdf

[23] Telikepalli Anil: "Power vs Performance-The 90nm deflection point", April 2005. http://www.xilinx.com/bvdocs/whitepapers/WP223.pdf

[24] Lecture 18 of Course CMOS VLSI Design: Design of low power at Electrical Engineering dept of Texas University, 2004 http://www.ece.utexas.edu/~adnan/vlsi-04/lec18LowPower.ppt#12

[25] IBM journal of Research and development: Low Power Circuits and technology for wireless digital systems, March/May 2003. http://www.findarticles.com/p/articles/mi_qa3751/is_200303/ai_n9180119

[26] David Culler, *"Components and design techniques of digital systems"*, Lec-16, Storage: DRAM, SDRAM, University of California, Berkeley. http://www-inst.eecs.berkeley.edu/~cs150/fa04/Lecture/lec16.ppt#25

[27] Hang Sheng Wang, Xinping Zu, Li Shiuan Peh, Sharad Malik, *"Orion: A Power Performance Simulator for Interconnection networks"*, department of electrical university, Princeton University.
http://www.princeton.edu/~peh/publications/orion.pdf

[28] Jingcao Hu, Radu Marculescu, "Energy and Performance Aware Mapping for Regular NoC Architectures", IEEE transaction on Computer Aided Design of Integrated Circuits and Design, Vol 24, No 4, April 2005.
http://www.ece.cmu.edu/~sld/pubs/papers/T-CAD_April05.pdf

[29] Kia Bazargan: Course VLSI Design-II: Memory Design- Part 5, University of Minnesota.
http://mountains.ece.umn.edu/~kia/Courses/EE5324/05-Mem/EE5324-Mem.ppt#259

[30] Iakovos Mavroidis, Alessandra Nardi, Engling Yeo, *"Comparisons of Synchronous and Asynchronous Designs For a maximum A-Posterior ProbabilityDecoder Memory Block"*.
http://www-cad.eecs.berkeley.edu/~nardi/EE241/mavro-nardi-yeo-mid-report.pdf

[31] Poletti Francesco, Poggiali Antonio, Paul Marshal: *"Flexible hardware/software support for message passing on a distributed shared memory architecture"*, University of Bologna, Italy, appeared in DATE 2005.
http://www.hwswworld.com/downloads/f1/06E_2.PDF

[32] Sambuddhi Hettiaratchi, Peter Y K Cheung, Thomas J W Clarke, *"Performance Area trade-off of Address Generators for Address Decoder-Decoupled memory",* Department of Electrical and Engineering, Imperial College, London, appeared in DATE 2002.
http://www.hwswworld.com/downloads/f6/09B_2.PDF

[33] Sudarshan Banerjee, Nikhil Dutt: *"FIFO based Power Organization for on-chip Networks",* Technical report #03-40, Dec 2003-department of Information and Computer Science, University of California, Irvine, CA, USA.
http://www.cecs.uci.edu/technical_report/TR03-40.pdf

[34] Ulf Nordquist and Dake Liu: *"Power optimized packet buffering in a protocol processor"*, department of electrical engineering, Linkoping University, Sweden, proceedings of 2002 IEEE ICECS conference.
http://www.da.isy.liu.se/pubs/ulfnor/ulfnor-icecs2003.pdf

[35] Jan. M. Rabaey and Massoud Pedram: *"Low power design methodologies",* Kluwer Academic publishers, 1996.

[36] Mathematica in Engineering: *"Least square multiple regression*"
http://www.efunda.com/math/leastsquares/lstsqrzrwtxy1d.cfm

[37] Jack Horgan: "*Low power Soc Design*", EDA cafe, Weekly issue of May 24, 2004.
http://www10.edacafe.com/nbc/articles/view_weekly.php?newsletter=1&run_date=24-May-2004

[38] Hangsheng Wang, Li-Shiuan Peh, Sharad Malik: "*Power-driven Design of Router Microarchitectures in on-chip networks*", Proceedings of the 36th annual IEEE/ACM International Symposium on microarchitecture, 2003.

[39] Stergios Stergiou, Federico Angiolini, Salvatore Carta, Luigi Raffo, David Bertozzi and Giovanni De Micheli, "*xPipes Lite, A Synthesis Oriented Design Library for Networks on Chip*". Appeared in DATE 2005.

[40] Fernando Moraes, Ney Calazans, Aline Mello, Leandro Möller, Luciano Ost : "*HERMES: An infrastructure for low area Overhead of Packet-Switching Networks-on-Chip", Jan 2005.*

[41] L. Benini and G. de Micheli, "*Networks-on-Chip: A new soc paradigm,"* IEEE computer, vol. 35, no. 1, pp. 70-78, January 2002.

[42] G. Cai and C.H Lim, "*Architectural level power/performance optimization and dynamic power estimation",* Cool Chips, Tutorial colocated with MICRO32, November 1999.

[43] Pascal T Wolkotte, Gerard J M Smit, Nikolay Kavaldijev, Jens E Becker, Jurgen Becker, "*Energy Model of Networks-on-Chip and a Bus"* Accepted for the International Symposium on System-on-Chip (SoC 2005) Tampere, Finland, November, 2005.

[44] Bolotin, E.; Cidon, I; Ginosar, R.; Kolodny, A. "*QNoC: QoS architecture and design process for Network on Chip"*. The Journal of Systems Architecture, Special Issue on Networks on Chip,2004.

[45] Matteo Monchiero, Gianluca Palermo, Cristina Silvano, Oreste Villa: "*An Efficient Synchronization Technique for Multiprocessor Systems on-Chip"*: In Proceedings of MEDEA-05, International Workshop on Memory performance: Dealing with Applications, systems and architecture held in conjunction with PACT'05, Saint Louis, Missouri. September 17, 2005.

[46] ARM cores specification : http://www.arm.com/products/CPUs/ARM926EJ-S.html