Customization of Power Performance in Interconnected MPSOC for NOC

S.Tejaswi Lalitha *, M.Venkateswara Rao ** Dr.K.Babulu***
* (Student, Department of Electronics and Computer Engineering, KL University, Guntur)
**(Assistant Professor, Department of Electronics and Computer Engineering, KL University, Guntur)
***(Professor, Department of Electronics and Communication Engineering, JNTU, Kakinada)

ABSTRACT

A shift in the system-scope occurs on day-to-day developments in the technology, in the similar manner SoC design has occurred from the evolution of the ULSI technology. SoC provides a solution to various challenges in complex applications by its adaptability, feasibility and flexibility. In order to meet the growing requirements of the market, a new concept of multi processors on system on chip evolved. On employing such technology many challenges were unveiled like the communication b/w the components, the power consumption, the area on chip, efficiency and many more. Here we put-forth some low-power consumption techniques and performance of the communication architectures that are involved in the design.

Keywords – **Interconnections, MpSoc, NoC, Partial activation Technique, Soc**

I. INTRODUCTION

The architecture in design of SoC communication architecture is a central task which involves the design to be generic, adaptable to any topology/application, synchronized data transfer, quality of service aspects and facilitating required communication services. The communication can be done in conventional manner through buses or by providing the network. The solutions for SoC communication structures have generally been characterized by custom designed ad hoc mixes of buses and point-to-point links[1]. Both the bus architecture and the network architecture have some pros & cons. The pros for the bus architecture over network architecture include bus latency is wire-speed once arbiter has granted control, any bus is almost directly compatible with most available IPs, including software running on CPUs where the absence of degradation in performance, possibility in pipe-lining, the flexibility in making the routing decisions , use of same router for different network sizes ,good test coverage by locally dedicated BIST, aggregated bandwidth scales with the network size are the advantages provided by the later over the former technique .

The performance of SoCs will be limited by the ability to efficiently interconnect predefined and pre-verifiedIPs and to accommodate their communication requirements, i.e. it will be communication – rather than computation – dominated. Moreover the power consumption on communications becomes significant portion of overall system power budget [2].

Recently, Networks-on-Chip (NoC) architectures are emerging as a scalable, reliable, and highly modular on-chip communication infrastructure [3][4][5]. The NoCarchitecture involves the on-chip routers, network interfaces and protocols over a pre defined topology. The main function of NoC is to route packets from source to destination.

II. NOC TOPOLOGIES AND PROTOCOL

NoCs rely on specific topological connectivity, such as octagon, ring, bus, star, mesh to simplify the control logic, while others allow for arbitrary connectivity, providing more flexible matching to the target application. Hierarchical star(H-star) topologies based on the power consumption, which is our main concern.

2.1 Hierarchical star topology

The first phase for NoC architecture design is choosing the most suitable NoC topology. According to an analytical calculation shown in Fig.2.1, Mesh and H-star topologies show the lowest power consumption under not only uniform traffic and but also localized traffic condition. We chose the H-star topology for our NoC platform because it has more flexible structure, occupies only a area of 10-15% of overall chip area [1] [6] and has less switching hops than Mesh topology does[6]. The integrated on-chip network of a hierarchical star topology provides 11.2 GB/s aggregate bandwidth and consumes 51mW when the integrated IPs executed load/store operations without idle states, which is the maximum traffic condition in the system.

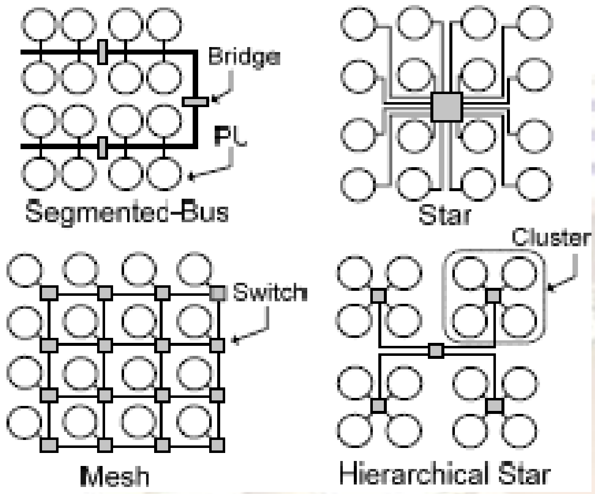


Fig 2.1.1 Topologies

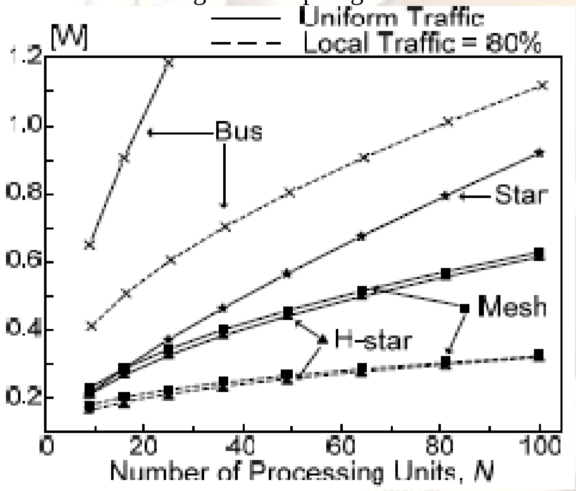


Fig 2.1: Power Analysis of Various Topologies

2.2 NoC Protocol

NoCs can be based on circuit or packet switching, or a mix of both; the former is aimed at providing hard QoS guarantees, while the latter optimizes the efficiency for the average case. When packet switching is chosen, switches provide buffering resources to lower congestion and improve performance. They also handle flow control issues, and resolve conflicts among packets when they overlap in requesting access to the same physical links. Two of the most usual flow control protocols involve switch-to-switch communication and are retransmission based (i.e., packets are optimistically sent but a copy of them is also stored by the sender, and, if the receiver is busy, a feedback wire to request retransmission is raised) or credit-based (i.e., the receiver constantly informs the sender about its ability to accept data, and data are only sent when resources are certainly available) [1] [7].

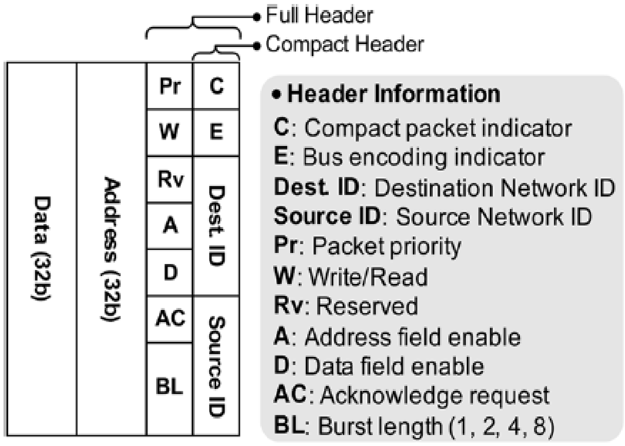


Fig 2.2.1 NoC Protocol: Packet format

Fig. 2.2.1 shows the Basic On-chip Network (BONE) protocol used in packet transactions [8]. The NoC protocol supports burst packet transactions for large data transmissions with length of 2, 4, and 8 packets.

In the implemented protocol, the packet format has 3 bits of source ID and 3 bits of destination ID; therefore, it supports 8masters and 8 slaves in maximum. To scale up the network size, you should increase the ID fields in the packet format before the chip design [6].

III. Low Power Techniques

3.1 Low-Swing Signaling Technique

The global link that connects two clusters is usually a few millimeters long in a large SoC and consumes higher power than a local link does. Low-swing signaling can alleviate its energy consumption significantly [9]. Fig. 3.1.1 shows the differential low-swing signaling scheme and its transceiver circuits used in this implementation. Global wires are laid out in zigzags to emulate a long link as long as 5.2mm without repeaters. To find out the optimum voltage swing, we conducted post-layout simulations with a precise capacitance and resistance wire model [5]. The sizes of the input gates and their bias currents are chosen to amplify the differential input of as low as 200-mV swing to 1.6-V full-logic swing with small delay [6]. A 5.2-mm metal2 wire of 0.5- m width and 1.1- m space has 330-fF parasitic and 100-fF coupling capacitance values. scans from 0.25 to 1.1 V with 50-mV step when signaling rates are 400 Mb/s, 800 Mb/s, and 1.6 Gb/s as shown Fig 3.1.2.

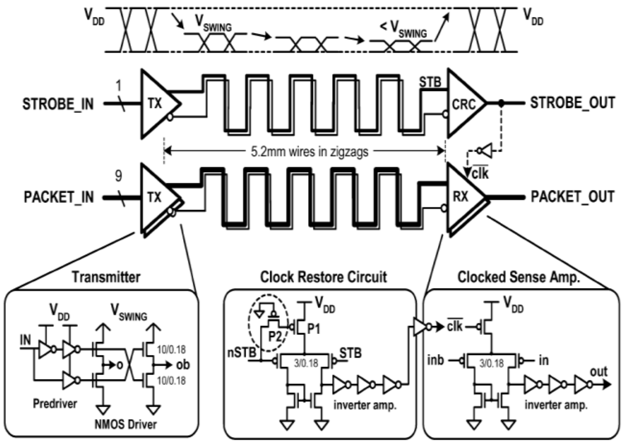


Fig 3.1.1 Low-swing signaling and its transceiver circuits

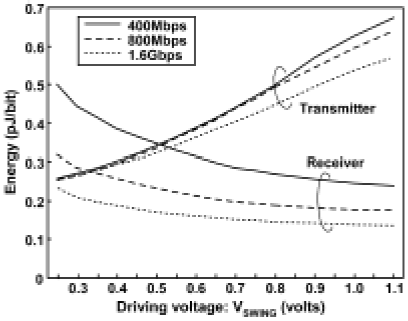


Fig 3.1.2 Energy consumption Vs. Driving Voltage

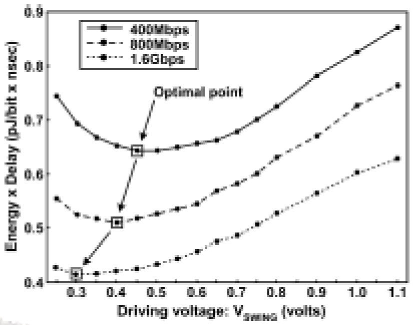


Fig 3.1.3 Optimization of V_{swing}

Due to the low-swing signaling, the power dissipation on the global link is reduced to 1/3 of that on a full swing repeated link and no repeaters are used on the wires to avoid area overhead [6].

3.2 Mux-Tree Based Round Robin Scheduler

A scheduler (or arbiter) is needed in a crossbar switch when more than two input packets from different input ports are destined for the same output port at the same time. Among a number of scheduling algorithms, a round-robin algorithm is most widely used in asynchronous transfer mode (ATM) switches and on-chip networks due to its fairness and lightness [10]. There are many ways on how to implement the round-robin algorithm [10] [11]. A mux-tree based implementation is proposed as in Fig 3.2.1 with scheduling latency is O (log n) and required resources are O (n), where n is the number of input ports in a crossbar switch. The proposed implementation, Mux-Tree, performs the minimum power and delay product; 136 and 1.05-ns delay at 100-MHz clock frequency with offered load of 50% [6].

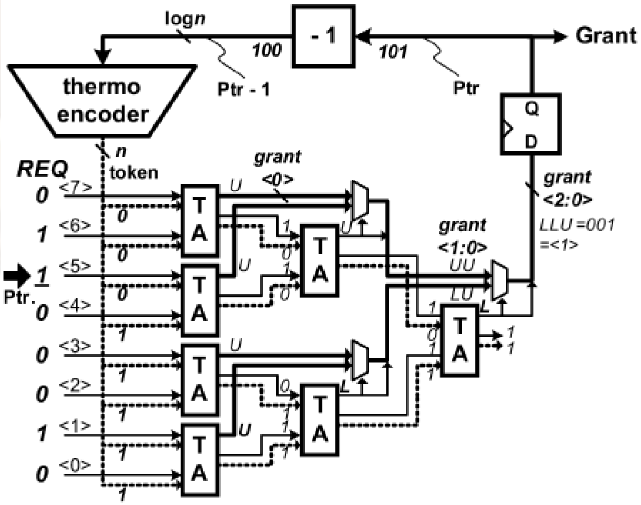


Fig 3.2.1 Mux-based tree Implementation

3.3 Partial Activation Cross Bar Technique

A conventional crossbar fabric comprises n x n crossing junctions which contain n^2 NMOS pass-transistors as in Fig 3.3.1. Each input driver wastes its power to charge and discharge two long wires row-bar (RB) and column-bar (CB) and transistor–junction capacitors. The RB and CB should be laid out with lower metal layers, M1 or M2, in order to reduce the fabric area and to minimize the number of resistive vias. Therefore, the loading on the driver becomes significant as the number of ports increases [6].

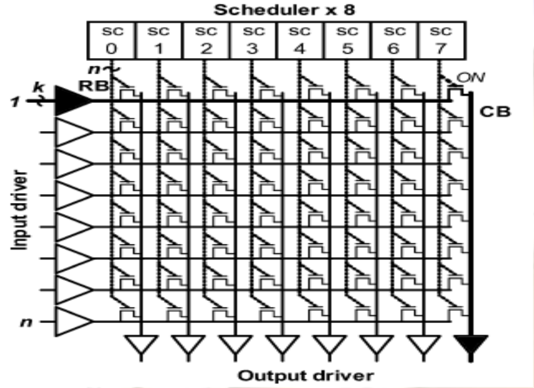


Fig 3.3.1 An 8x8 conventional Cross-bar

In order to reduce the power consumption, we proposed a crossbar switch with crossbar partial activation technique (CPAT) as illustrated in Fig. 3.1.2 [5]. By splitting the fabric into 4x4 fabrics (or tiles), the activated capacitive loading is reduce by n/4.

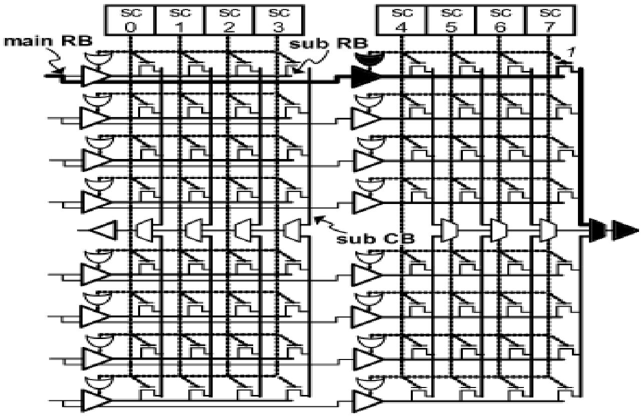


Fig 3.3.2 Proposed Partial Activation Cross-Bar

An 8x8 crossbar fabric with CPAT is analyzed in comparison with the conventional scheme. The area of the fabric is about 240x240µm2. Fig.3.3.2 shows the power comparison as a function of the offered loads. At 90% offered load, 22% power saving is obtained. This is possible because it omits the unnecessary activation of tri-state buffers and multiplexers.

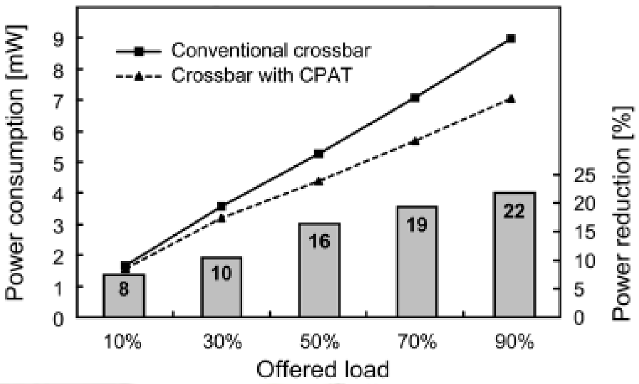


Fig 3.3.2 Power comparison of an 8x8 crossbar fabric with and without the crossbar partial activation technique [6]

3.4 Low-Energy Coding On-chip Serial Link

In serial communications, the switching activity factor of a serial wire is different from that of parallel wires. The difference in activity factor strongly depends on the transacted data patterns [6]. On-chip source-synchronous serial communication has many advantages over multi-bit parallel communication in the aspects of skew, crosstalk, area cost, wiring difficulty, and clock synchronization. However, the serial wire tends to dissipate more energy than parallel bus due to the bit multiplexing. In this work, we proposed a novel coding method, SILENT, to reduce the transmission energy of the serial communication by minimizing the number of transitions on the serial wire. The coding method saves significant amount of the communication energy for multimedia applications. It reduces maximum 77% of energy for instruction memory access, and 40~50% of energy for data memory access in a 3D graphics application [2].

3.5 Operating Frequency Scaling

PLL generates internal clocks such as a 100MHz clock for main cluster PUs, a 50MHz clock for peripheral cluster units, and a1.6GHz network clock for switches and network interfaces. The clock frequencies are scalable for power management modes, i.e.100/50/1600MHz for FAST mode, 50/25/800MHz for NORMAL mode, and 25/12.5/400MHz for SLOW mode [2].

IV. PERFORMANCE EXPLORATION

There are three major types of major performance models for communication architecture. Static estimation models provide a fast estimation of communication architecture by assuming static delays for various events these static estimation approaches assume that computation and communication in an SoC design can be statically scheduled, which is not always true. Static approaches are also unable to predict dynamic component delays as well as dynamic delays and the effect of advanced bus features. A more accurate (but slower) approach for performance estimation requires creating a model of the application that can be simulated. This allows a more accurate estimation of the dynamic data traffic behavior on the bus and the corresponding delays can be more reliably assessed. the different classes of dynamic performance estimation models that fall under four major categories in order of increasing simulation speed and decreasing accuracy: Cycle accurate (CA) models, PA-BCA models, T-BCA models, and TLM increasing levels of component integration in SoCs and the rising complexity of inter-component interactions means that simulation-based methods need to adapt to simulate only the necessary details required, to avoid a performance penalty. The other approach is development of hybrid estimation techniques provide the speed of the static technique and the efficiency of the dynamic simulation [10].

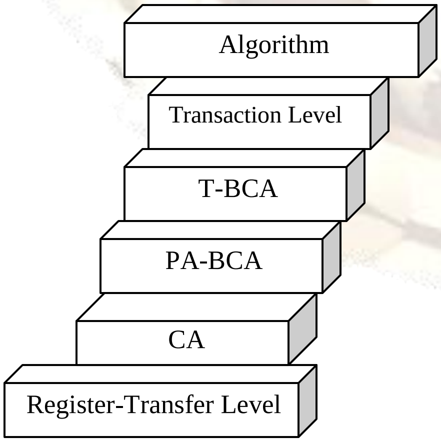
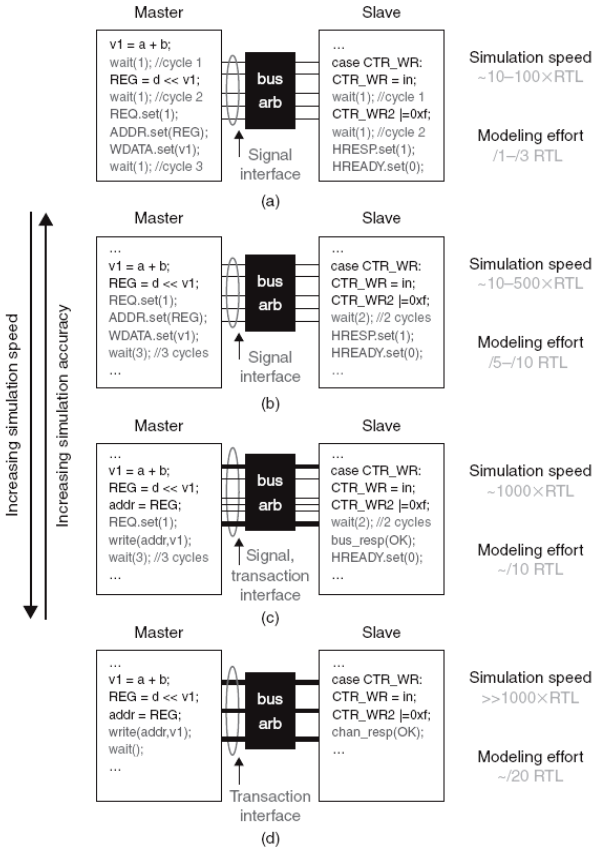


Fig 4.1: Modeling abstractions for communication architecture performance exploration



```
v1 = a + b;
wait(1); //cycle 1
REG = d << v1;
wait(1); //cycle 2
REQ.set(1);
ADDR.set(REG);
WDATA.set(v1);
wait(1); //cycle 3
```

```
...
case CTR_WR:
CTR_WR = in;
wait(1); //cycle 1
CTR_WR2 |=0xf;
wait(1); //cycle 2
HRESP.set(1);
HREADY.set(0);
```

```
...
v1 = a + b;
REG = d << v1;
REQ.set(1);
ADDR.set(REG);
WDATA.set(v1);
wait(3); //3 cycles
...
```

```
...
case CTR_WR:
CTR_WR = in;
CTR_WR2 |=0xf;
wait(2); //2 cycles
HRESP.set(1);
HREADY.set(0);
...
```

```
...
v1 = a + b;
REG = d << v1;
addr = REG;
REQ.set(1);
write(addr,v1);
wait(3); //3 cycles
...
```

```
...
case CTR_WR:
CTR_WR = in;
CTR_WR2 |=0xf;
wait(2); //2 cycles
bus_resp(OK);
HREADY.set(0);
...
```

```
...
v1 = a + b;
REG = d << v1;
addr = REG;
write(addr,v1);
wait();
...
```

```
...
case CTR_WR:
CTR_WR = in;
CTR_WR2 |=0xf;
chan_resp(OK);
...
```

Fig 4.2: Trade-offs between different modeling abstractions: (a) CA, (b) PA-BCA, (c) T-BCA, (d) TLM

V. CONCLUSION

To overcome the problems of scalability and complexity, Networks-On-Chip (NoCs) have been proposed as a promising replacement to eliminate many of the overheads of buses and MPSoCs connected by means of general-purpose communication architectures. To apply the prevailing mobile environment, it should be low-powered and efficient in its performance .Hence the technique of the partial activation cross-bar can be implemented in 16x16 matrix method for better power conservation and for better performance the hybrid approach that has both speed and perfection can be implemented as per the requirements of application.

REFERENCES

[1] **"A Survey of Research and Practices of Network-on-Chip",** Tobias Bjerregaard & Shankar

Mahadevan, Technical University of Denmark**,** ACM Computing Surveys, Vol. 38, March 2006.

[2] "**Networks-on-chip and Networks-in-Package for High-Performance SoC Platforms",** Kangmin Lee, Se-Joong Lee, Donghyun Kim, Kwanho Kim, Gawon Kim, Joungho Kim, and Hoi-Jun Yoo,Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea**.**

[3] D. Bertozzi et al., **" NoC Synthesis Flow for Customized Domain Specific Multiprocessor Systems-on-Chip,"** IEEE Transactions on Parallel and Distributed Systems, vol. 16, no. 2, pp113-129, 2005.

[4] S.-J. Lee et al., **"An 800MHz Star-Connected On-Chip Network for Application to Systems on a Chip,"** IEEE Int. Solid-State Circuits Conf., Feb. 2003, pp. 468-469.

[5]] K. Lee et al., **"A 51mW 1.6GHz On-Chip Network for Low-Power Heterogeneous SoC Platform,"** IEEE Int. Solid-State Circuits Conf., Feb. 2004, pp. 152-153.

[6] **"Low-Power Network-on-Chip for High-Performance SoC Design"**, Kangmin Lee, Student Member, IEEE, Se-Joong Lee, Member, IEEE, and Hoi-Jun Yoo, Senior Member, IEEE, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 14, NO. 2, FEBRUARY 2006.

[7] **VLSI-SoC-From systems to chip"**(published by IFIP - The International Federation for Information Processing).

[8] "**BONE: Network-on-Chip Protocol [Online]**". Available:http://ssl.kaist.ac.kr/ocn

[9] R. Ho et al., "**Efficient on-chip global interconnects**," in IEEE Symp., VLSI Circuits Dig. Tech. Papers, Jun. 2003, pp. 271–274.

[10] P. Gupta et al., "**Design and implementing a fast crossbar scheduler**," IEEE Micro, vol. 19, no. 1, pp. 20–28, Jan./Feb. 1999

[11] Shin et al., "**Round-robin arbiter design and generation,"** in Proc.IEEE Int. Symp. Syst. Synthesis, Oct. 2002, pp. 243–248.

[12]**"On-chip communication Architectures**".Pg No.120-161.

IJERA