Chapter Two

Related work review and NoC

2.1 Introduction:

The purpose of this chapter is to give a literature overview of power estimation models on SoC, and thin give a general description for NoC, and NoC architecture.

2.2 Power Estimation Models and Schemes:

When examining previous works in SoC power estimation, attention must be given to in what level the system was analyzed.

Kanishka Lahiri and Anand Raghunathan 2006 [15] developed 1) a systematic evaluation and analysis of the power consumption of a commercial AMBA chip and gave 2) a power management discussion, their work was one of the main references for this thesis, the power estimation model used is illustrated in Figure 2.1, by using tools and libraries to give real system description to be implemented to the simulator, thin presented the power consumed in each device inside the AMBA chip, the power dissipated in the bus wire, and application traffic characteristics influence communication architecture power consumption. In the following subsections:

1- Spatial distribution of communication transactions
2- Transaction pipelining.
3- Different transaction types, on communication architecture power.

Then presented studies that analyze the sources of power consumption in a communication architecture by considering a suite of techniques for optimizing communication architecture power like bus encoding, segmented bus design, interface power management and traffic sequencing. The purpose of their discussion is to obtain an understanding of which portions of the communication architecture are addressed by each technique, and to quantify the impact of these techniques on communication architecture power.

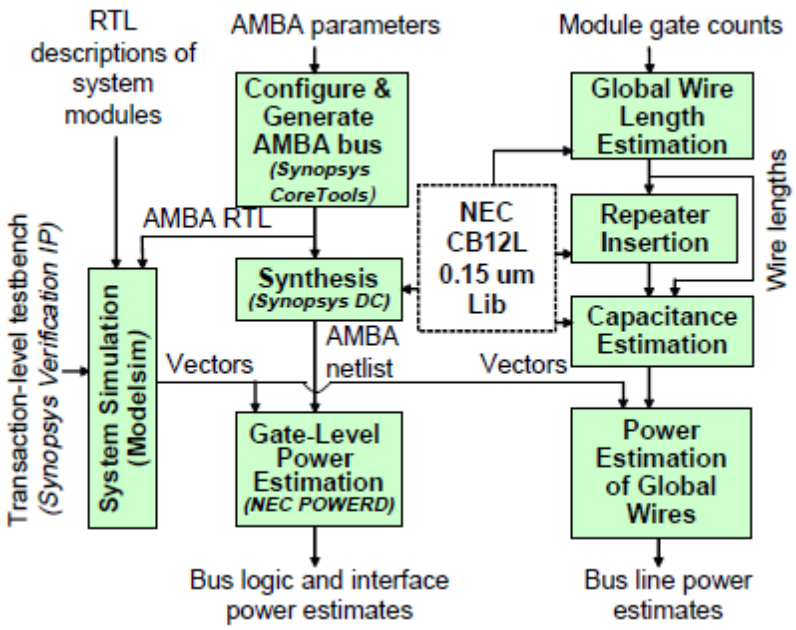

Figure 2.1: Methodology for power estimation source K.Lahiri and A.Raghunathan 2006 [32]

Hong-Hui Chen, Tung-Chien Chen et al 2011 [16] introduced a power estimation scheme and generated results of SoC fabricated with different process nodes extending to very deep submicron technology, they used different power modeling strategies to estimate power for analog and digital circuits, stating according to their analysis results that ultra-low power analog components are key to a successful biomedical SoC design if more advanced fabrication technology is utilized, and stating that digital part should be designed barely enough to serve the target application.

Saying that by integrating more dedicated digital hardware accelerators and by lowering the working frequency of system processor the total power consumption can be reduced furthermore and stating that by using the reached result a suitable technology could be selected to manufacture the SoC for biomedical usage.

Sumit Ahuja 2010 [17] stated that some factors have necessitated the abstraction level of design-entry of hardware systems to be raised beyond the Register Transfer-Level (RTL) to Electronic System Level (ESL). However, power envelope on the designs due to packaging and other thermal limitations, and the energy envelope due to battery life-time considerations have also created a need for power/energy efficient design. The confluence of these two technological issues created an urgent need for solving two problems: (i) How to enable a power aware design flow

with a design entry point at the Electronic System Level? (ii) How to enable power aware High Level Synthesis to automatically synthesize RTL implementation from ESL? This dissertation distinguishes itself by addressing the following two issues: (i) since power/energy consumption of electronic systems largely depends on implementation details, and high-level models abstract away from such details, power/energy estimation at such levels has not been addressed thoroughly. (Ii) a lot of work has been done in applying various techniques on control-data-flow graphs (CDFG) to find power/area/latency points during behavioral synthesis. However, high level C-based functional models of various compute-intensive components, which could be easily synthesized as co-processors, have many opportunities to reduce power. Some of these savings opportunities are traditional such as clock-gating, operand-isolation etc. The exploration of alternate granularities of these techniques with target applications in mind, opens the door for traditional power reduction opportunities at the high-level, this work concentrates on the aforementioned two areas of inadequacy of hardware design methodologies.

The proposed solutions include utilizing ESL simulation traces and mapping those to lower abstraction levels for power estimation, derivation of statistical power models using regression based learning for power estimation at early design stages, etc. On the HLS front, techniques that insert the power saving features during the synthesis process using exploration of granularity and scope of clock gating, sequential clock-gating are proposed. Finally, this work shows how to marry two domains, that is estimation and reduction, in regards for all of that a power model shown in Figure 2.2 is proposed, which helps in predicting power savings obtained using clock-gating and further guiding HLS to selectively insert clock-gating.

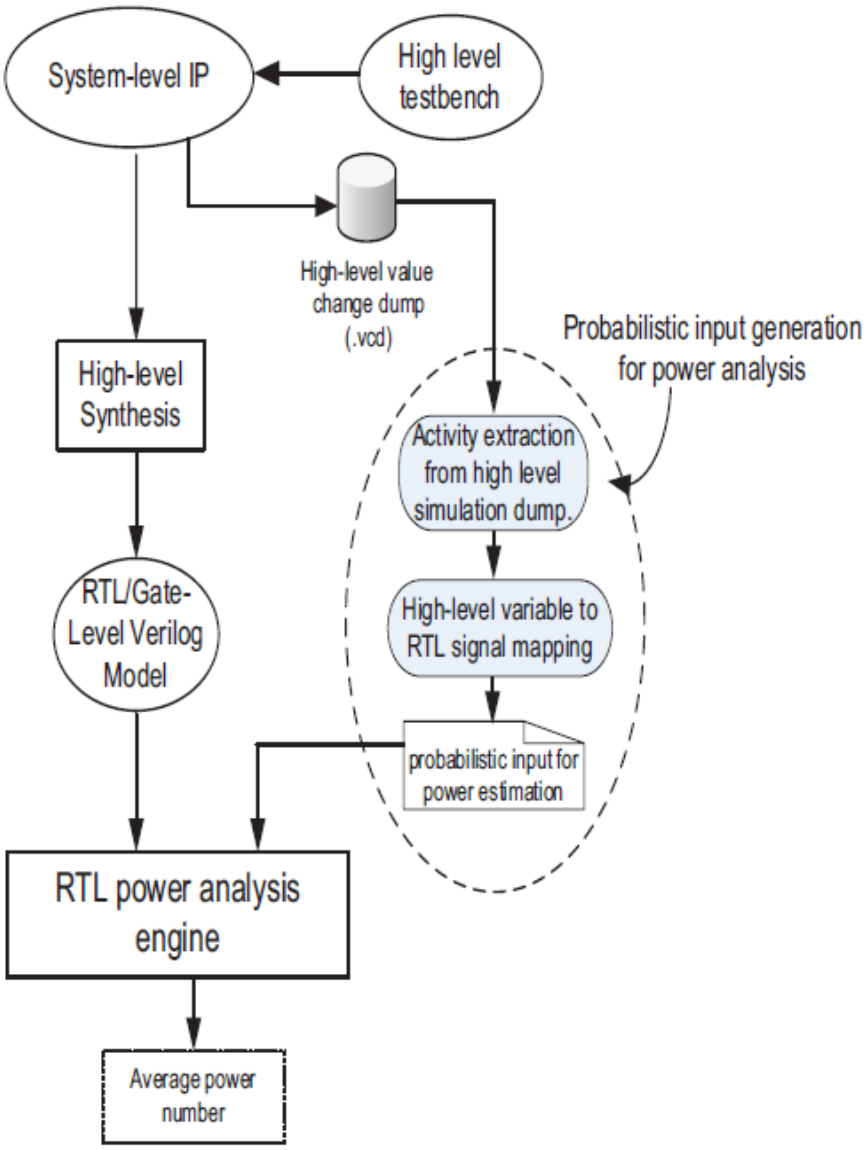

Figure 2.2: Power estimation model source: sumit Ahuja 2010 [14]

David Y. Feinstein and Mitchell et al [18] stated that accurate power consumption estimation of a System-on-Chip (SoC) using modeling techniques is difficult due to the diverse mixture of processes with radically different current consumption. It is very important that these estimations will be fine-tuned to the specific SoC with accurate current measurement during the design and prototyping phase, thin introduced an accurate method to measure power consumption using a single measurement point and a dynamic logging algorithm, presented a demonstration

tool for continuous logging of the instantaneous power consumption with an identification of the running process within the SoC, their approach can also be used to steer the dynamic power management (DPM) of a SoC.

Liang-Bi Chen and Tsung-Yu Ho et al [19] stated that in designing the portable embedded system, power consumption would be an important issue. Numerous prior issues of power dissipation have been discussed, and there are fewer functions (such as power measurement, power analysis, and power management) supporting for a SoC embedded system development learning board. In order to help designer to get power information in the beginning of design, they proposed a development platform for power analysis and build an analysis system for power analysis. The system described in this work has three important features – power measurement, power analysis, and power management. By using this system, it can help the designers to develop a high performance embedded system designs with low power consumption. This system has been implemented on the real embedded systems to demonstrate the functionalities which was proposed, Figure 2.3 shows the proposed system model.

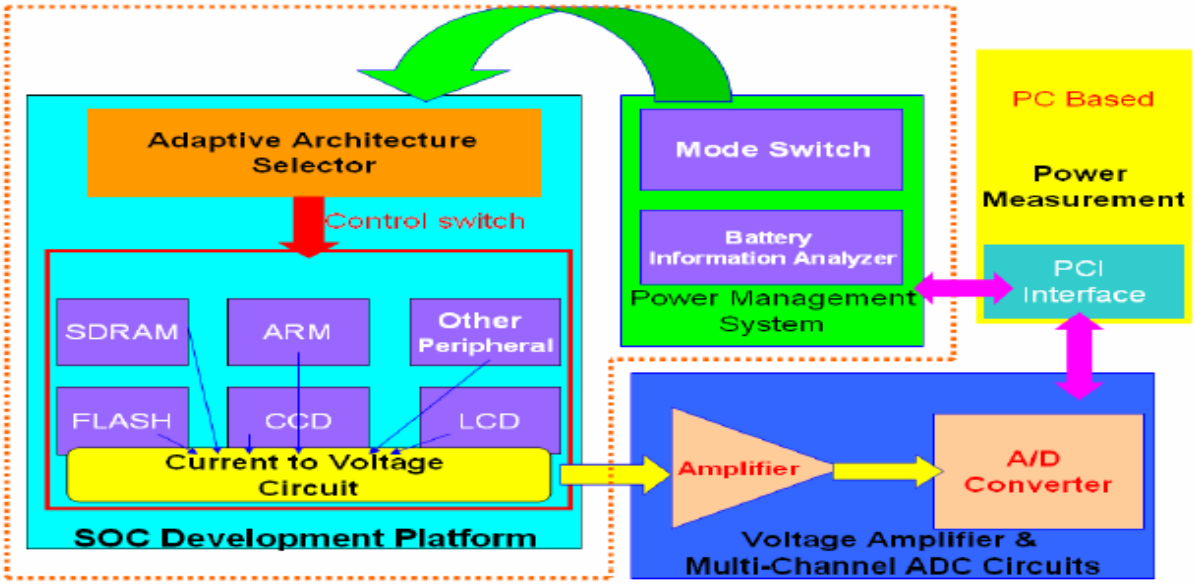

Figure 2.3: Proposed system model source: Liang-Bi Chen and Tsung-Yu Ho et al [16]

Luca Benini, Robin Hodgson and Polly Siegel [20] stated that most work to date on power reduction has focused at the component level, not at the system level, in this work, they proposed a framework for describing the power behavior of system-level designs.

The model used consists of a set of resources, an environmental workload specification, and a power management policy, which serves as the heart of the system model, they mapped this model to a simulation-based framework to obtain an estimate of the system's power dissipation. Accompanying this, they proposed an algorithm to optimize power management policies, the optimization algorithm can be used in a tight loop with the estimation engine to derive new power-management policy algorithms for a given system-level description, thin tested this approach by applying it to a real-life low power portable design, achieving a power estimation accuracy of ~10%, and a 23% reduction in power after policy optimization.

Masafumi Onouchi, Tetsuya Yamada et al [21] they have developed a specialized rapid power-estimation methodology for multimedia applications.

This methodology has adequate accuracy for the first design of a complicated SoC. For a multimedia application, they developed three new methodologies: an IP-level modeling, a power-level adjustment methodology, and a power accumulation methodology.

With these methodologies, the system-level power estimation on a SoC executing a practical application becomes so precise and easy that and designer can revise the SoC design to reduce its power.

According to a comparison of the system-level power estimated with these methodologies to board-measured power, the error between the two powers is less than 5.6%.

Marcello Lajolo and Anand et al 2000 [2] presented efficient power estimation techniques for HW/SW SoC designs. the techniques used are based on concurrent and synchronized execution of multiple power estimators that analyze different parts of the SoC (by refer to this as co-estimation), driven by a system-level simulation master.

By motivating the need for power co estimation, and demonstrating that by performing independent power estimation for the various system components can lead to significant errors in the power estimates, especially for control-intensive and reactive embedded systems.

They intended to prove that the computation time for performing power co-estimation is dominated by: (i) the requirement to analyze/simulate some parts of the system at lower levels of abstraction in order to obtain accurate estimates of timing and switching activity information, and (ii) the need to communicate between and synchronize the various simulators. Thus, a naive

implementation of power co-estimation may be too inefficient to be used in an iterative design exploration framework.

To address this issue, they presented several acceleration (speedup) techniques for power co-estimation, the acceleration techniques are energy caching, software power macro modeling, and statistical sampling, the speedup techniques reduces the workload of the power estimators for the individual SOC components, as well as their communication/synchronization overhead.

Experimental results indicate that the use of the proposed acceleration techniques results in significant (8X to 87X) speedups in SOC power estimation time, with minimal impact on accuracy.

R. Kar and V. Maheshwari et al 2010 [1] Power is increasingly becoming the bottleneck for the design of high performance VLSI circuits. It is essential to analyze how the various components of power are likely to scale in the future, thereby identifying the key problematic areas, while most analyses focus on the timing aspects of interconnects, power consumption is also important, in this paper, the power distribution estimation of interconnects is studied using a reduced order model.

The relation between power consumption and the poles and residues of a transfer function is derived, and an appropriate driver model is developed, allowing power consumption to be computed efficiently.

2.3 Development from on chip communication to NoC:

Multi-processor systems-on-chips (MPSoCs) have been widely used in today's high performance embedded systems, such as network processors (NP) and parallel media processors (PMP). They combine the advantages of data processing parallelism of multi-processors and the high level integration of systems-on-chip (SoCs). Driven by the advances of semiconductor technology, future SoCs will continue to accelerate in system's complexity and capacity. SoCs in the next decade are expected to integrate hundreds, or even more of, processing elements (PEs) and/or storage elements (SEs) on a single chip. SoC designs at this scale cannot be started from scratch, instead, it is common believe that SoCs will be designed using pre-existing components, such as processors, controllers and memory arrays. Future SoC design methodologies have to

support component re-use in a plug-and-play fashion in order to meet time-to-market requirement. In such a plug-and play system integration approach, the most critical factor will be related to the communication scheme among components. The design of reliable, low-energy and high-performance on-chip communication architectures for future SoCs will pose unprecedented challenges. Interconnect technology will become the limiting factor for achieving the operational goals.

Traditional on-chip communication structures have already encountered many limitations in today's VLSI designs. Many of these limitations will become even more problematic as the semiconductor technology advances into newer generations. These limitations are either associated with the scaling-down of the device feature size, or they are inevitable with the scaling-up of design complexity [21]. Particularly, the following issues will become the bottleneck in the future communication-centric SoC design scheme:

- **Throughput Limitation**– Traditional on-chip communication structures (i.e. the buses) cannot scale up as the number of components increases. When multiple data flows are transmitted concurrently, they will compete for the same communication resources.

- **Energy Consumption**– as the VLSI device features are continuously shrinking down, interconnect wires have been one of the major contributors of the system energy consumption. The buses used in many of today's SoC designs are notoriously not energy-efficient, because every bit transmitted is propagated throughout the bus to every terminal.

- **Signal Integrity**– Energy considerations will impose small logic swings and power supplies, most likely below 1 Volt. Smaller device feature sizes will also produce denser wires (i.e., 7 layers or more of routing wires) connecting highly compacted transistors. Therefore, future VLSI systems will become more vulnerable to various forms of electrical noise, such as cross-talk, electro-magnetic interference (EMI) and radiation-induced charge injection (soft errors). An additional source of errors

is contention in shared-medium networks. Contention resolution is fundamentally a non-deterministic process, because it requires synchronization of a distributed system, and for this reason it can be seen as an additional noise source. Because of these effects, the mere transmission of digital values on wires will be inherently unreliable.

- **Signal Latency**– The propagation delay on wires will gradually dominate the signal latency as the wire feature size shrinks. In fact, wire delay has already become a big challenge in today's VLSI systems, because the delay is determined by the physical distribution of the components, which is hard to predict in the early stages of the design flow. A more predictable communication scheme is of great importance in the future SoC designs.

- **Global Synchronization**– Propagation delay on global wires - spanning a significant fraction of the chip size - will pose another challenge on future SoCs. As the wire size continues to shrink, the signal propagation delay will eventually exceed the clock period. Thus signals on global wires will be pipelined. Hence the need for latency insensitive design is critical. The most likely synchronization paradigm for future chips is globally-asynchronous locally-synchronous (GALS), with many different clocks.

The network design technology is used to analyze and design future SoCs. In other words, view SoC as a micro-network of components, where the PEs and SEs are interconnected as node components, or simply referred to as nodes. Here can be seen that SoC interconnect design analysis and synthesis can be done by using the micro-network stack paradigm, which is an adaptation of the protocol stack [22] Figure 2.4.

Thus the electrical, logic, and functional properties of the interconnection scheme can be abstracted.

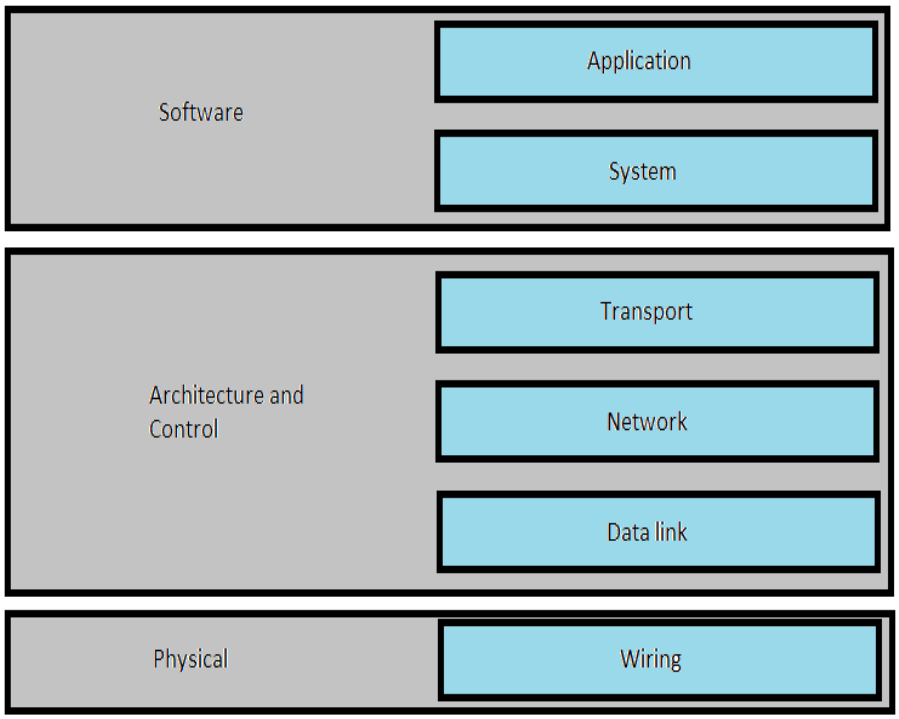

Figure 2.4: Micro Network Stack

2.4 NoC Architecture and Protocol:

In on-chip network architecture, or networks-on-chip (NoC), each PE or SE is abstracted as a node, and the nodes are interconnected by the micro-network that can provide scalable and concurrent point-to-point (P2P) or point-to-many (P2M) connection. As a new SoC design paradigm, NoCs will support novel solutions to many of above mentioned SoC interconnect problems. For example, multiple data flows can be supported concurrently by the same communication resources, data integrity can be enhanced by error correction and data restoration, and components are more modularized for IP reuse.

On-chip network architectures may adopt design concepts and methodologies from computer networks, namely from system-area networks (SAN) and parallel computer clusters (PCC). Communication in on-chip network architecture is also regulated by protocols, which are designed in layers. The layer stack in NoCs may differ from traditional networks because of local proximity and because they exhibit much less non-determinism. Although we may borrow some concepts and approaches from computer network architectures, we do not need to follow the OSI

seven-layer scheme to setup a communication transaction. Instead, on-chip networks may exploit tailor made protocols to satisfy application specific requirements.

Next will be an analysis to specific issues related to the different layers of abstraction outlined in the micro network stack in a bottom-up way.

2.4.1 Physical Layer:

Global wires are the physical implementation of the communication channels. While the VLSI technology trends lead us to use smaller voltage swings and capacitances, the signal integrity problem will get worse. Thus the trend toward faster and lower power communication may decrease reliability as an unfortunate side effect.

Current design styles consider wiring-related effects as undesirable parasitic, and try to reduce or cancel them by specific and detailed physical design techniques. It is important to realize that a well-balanced design should not over-design wires so that their behavior approaches an ideal one, because the corresponding cost in performance, energy efficiency and modularity may be too high. Physical layer design should find a compromise between competing quality metrics and provide a clean and complete abstraction of channel characteristics to micro network layers above.

2.4.2 Data link, Network and Transport Layers:

The data-link layer abstracts the physical layer as an unreliable digital link, where the probability of bit upsets is not negligible (and increasing as technology scales down). An effective way to deal with errors in communication is to packetize data. If data is sent on an unreliable channel in packets, error containment and recovery is easier, because the effect of errors is contained by packet boundaries, and error recovery can be carried out on a packet-by-packet basis.

At the data link layer, error correction can be achieved by using standard error correcting codes (ECC) that add redundancy to the transferred information.

At the network layer, packetized data transmission can be customized by the choice of switching and routing algorithms. The former establishes the type of connection while the latter determines the path followed by a message through the network to its final destination.

At the transport layer, algorithms deal with the decomposition of messages into packets at the source and their assembly at destination. Packetization granularity is a critical design decision,

because the behavior of most network control algorithms is very sensitive to packet size. Packet size can be application-specific in SoCs, as opposed to general networks.

2.4.3 Software and Application Layer:

Software layers comprise system and application software. The system software provides us with an abstraction of the underlying hardware platform, which can be leveraged by the application developer to safely and effectively exploit the hardware's capabilities.

From a high level application viewpoint, multiprocessor SoC platforms can be viewed as networks of computing nodes equipped with local storage. Software layers are critical for the NoC paradigm shift, especially when energy efficiency is a requirement. Software programming abstractions, development tools and system software need to help programmers understanding communication-related costs and coping with them.

2.5 Challenges in NoC:

The above analysis shows that on-chip networks differ from traditional computer networks, many assumptions and solutions have to be adapted to the on-chip implementation. NoC architectures and protocols have to deal with the advantages and limitations of the silicon fabric. In particular, chip-level communication is localized between nodes (PEs and SEs). On-chip networks do not need to follow the standard schemes for communication since they can use lighter and faster protocol layers. NoCs will require novel methodologies for both on-chip switch designs as well as routing algorithm designs.

2.6 Why the Need for NoC:

A basic building block of most on-chip architectures in MPSoC designs is the single shared bus. This is the simplest on-chip communication architecture, consisting of a set of shared, parallel wires to which various components are connected. Only one component on the bus can have control of the shared wires at any given time to perform data transfers. This limits the parallelism and achievable performance in the system, which makes it unsuitable for most MPSoC applications that can have tens to hundreds of components. Consequently, the single shared bus architecture is not scalable to meet the demands of MPSoC applications.

Figure 2.5 shows various kinds of on-chip communication architectures that are used in MPSoC designs. Many contemporary MPSoC designs mostly use shared bus-based communication architectures. Figure 2.5(a) shows a hierarchical shared bus architecture, which consists of a hierarchy of buses interconnected using bridge components. Shared buses higher up in the hierarchy are typically operated at higher clock frequencies, and are used to connect high speed, high performance components. On the other hand, shared buses lower down in the hierarchy are operated at lower frequencies to save power, and connect high latency, low performance components. Figure 2.5(b) shows a ring type bus, similar to that used in the IBM Cell MPSoC. The ring bus is actually a set of unidirectional, concentric and pipelined buses which allow high frequency operation and high bandwidth transfers between components on the bus. Figure 2.5(c) shows an ad-hoc bus architecture, where buses are operated at different frequencies and components can have point-to-point links with each other, as needed. Finally, Figure 2.5(d) shows the bus matrix (or crossbar bus) where a crossbar type architecture connects processors (and their local bus components) on the left to memories and peripherals on the right. This kind of architecture is a combination of shared bus and point-to-point interconnections.

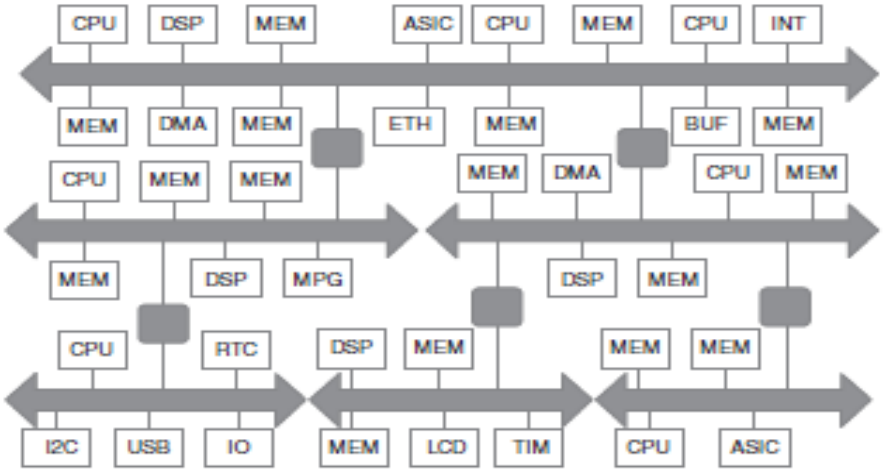

Figure 2.5: MPSoC bus-based on-chip communication architectures: (a) hierarchical bus.

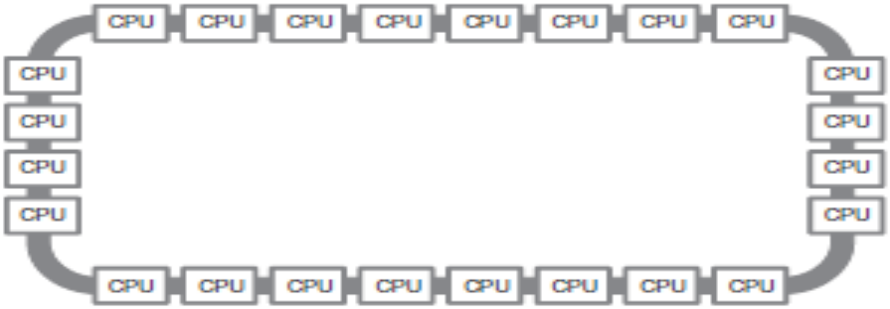

Figure2.5 (b): Ring bus

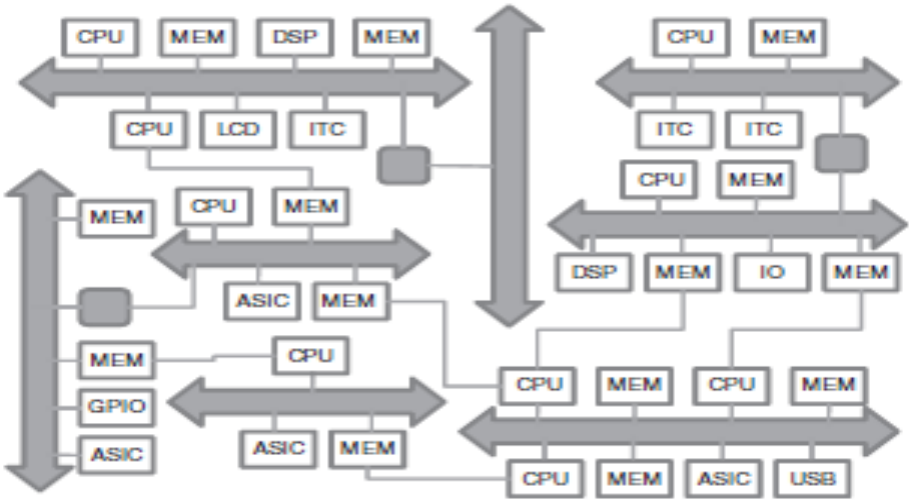

Figure 2.5(c): Ad-hoc bus

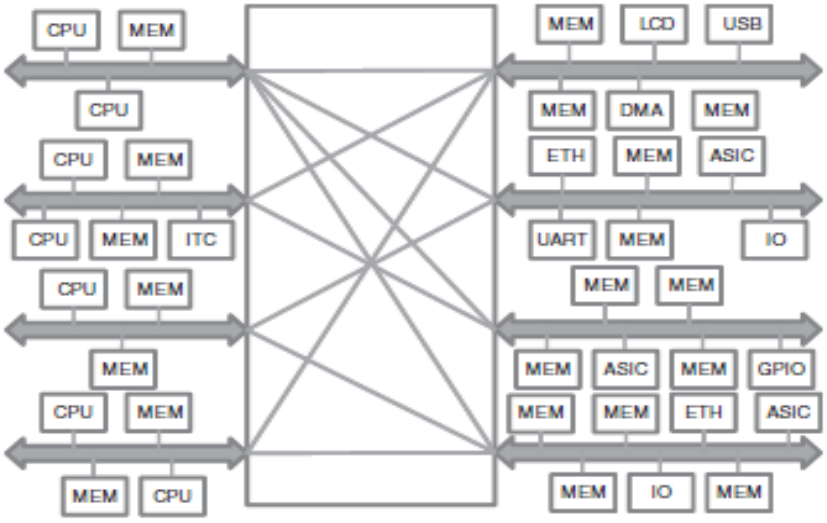

Figure 2.5(d): Bus matrix or crossbar bus

Each of the above bus-based on-chip communication architectures is defined by its two major constituents: topology and protocol parameters. The topology of a communication architecture refers to how the buses are interconnected together, and how the various components are mapped to each of the buses. The protocol parameters refer to such parameters as arbitration schemes, bus widths, bus clock frequencies, buffer sizes, and burst transfer sizes, which are specific to the protocol used by the communication architecture. Designing a communication architecture thus implies determining both its topology and protocol parameter values.

It has been projected that bus-based architectures cannot scale up with an increasing number of components, and also given the increasing amount of on chip wire delays (resulting in timing unpredictability). Thus future MPSoC designs with hundreds of components will make use of network-on-chip (NoC) communication fabrics, where instead of shared buses, packet switched network fabrics with routers are used to transfer data between on-chip components. However, NoCs are still in their early phase of research and development, and concrete implementations of NoC-based MPSoCs are only now beginning to appear.

2.7 Chapter Summary:

In this chapter, we reviewed some related literature that approached the topic of power estimation for SoC, some of this work approached the topic by analyzing the SoC in deferent level (system level, submicron level), other work suggested accelerating techniques that give accurate efficient estimation and some studied the power dissipated in the bus wire or bus interconnect all of this work is done to give a good understanding to the designer of the embedded chip to the power consumed in the deferent parts of the chip this information helps him developing a power reduction scheme that optimizes the power usage in the chip.

NoC was discussed in this chapter by defining the NoC and explaining the significance and the development in the on-chip communication, and presenting the deferent NoC architecture.