Enhancement of Smart Grid Performance through Logic based Fault Tolerant MPSoC

D.Vijayakumar* and V.Malathi**
*Assistant Professor, Dept. of Electronics and Communication Engineering, Latha Mathavan Engineering College, Alagarkoil, TamilNadu, India. Email: vkkumarin@gmail.com
**Professor, Dept. of Electrical and Electronics Engineering, Anna University Regional campus Madurai, TamilNadu, India

Abstract: Sustainable energy is the energy production without compromising the energy production for the future generations. This paper presents the picture of today’s power system structure. It also portrays a reasonable picture about the different challenges that are confronted by the present day smart grid archi tectures used in transmission framework. This paper presents a unique vision for the development of smart power grids. This paper addresses major issues in smart grid. A Multiprocessor System on Chip (MPSoC) is designed to specifically meet the niche requirements of modern power system processing elements. The existing power grid model does not provide real-time information of transmission devices during emergency events. In this paper the significance of restructuring the existing smart grid architecture using MPSoC with power system components. An embedded intelligence is inserted into the power-electronics to facilitate the reconfiguration of the system, and thereby ensuring security. As the system is designed with MPSoC modified smart grid architecture, the computational complexity of the proposed system architecture significantly improving the performance of the smart grid. This paper reveals the fault tolerant methodology using MPSoC with self-diagnosis, which is essential for enhancing the proposed architecture for smart grid functionalities. Using application-specific instructions for Heterogeneous MPSoC allows finding a good performance/energy tradeoff. The functions will enhance the general execution regarding execution to accomplish the framework a reconfigurable and thusly the execution may be enhanced by upgrading which means edges in gadget system readiness and practicality and at last, worth reduction. This leads to reduce computational complexity of the existing architecture, latency and improves performance tradeoff.

Keywords: Smart grid, MPSoC, Embedded system, Reconfigurable Smart grid architecture, Fault tolerance.

Introduction

A Multiprocessor System on Chip (MPSoC) is a complete computational system integrated in a single chip, combining multiple Processing Elements (PEs) as main components. In addition, the MPSoC is built by interconnecting Intellectual Property (IP) cores with local ports of Network on Chip (NoC) routers. Messages are transmitted between IP cores wrapped in NoC packets. The growing interest of MPSoCs lies in its ability to combine high performance, flexibility and its reconfigurable feature. In this work, MPSoC uses multiple Remote Terminal Unit (RTUs) along with other Distributed State Estimator (DSEs) to implement a system. The wide ranges of MPSoC architectures have been developed over the conventional methods to progress real time management system.

Literature Survey

The literature survey has been conducted on architectures of the power grid by using different means and approaches. This section reviews the relevant literature of the existing power grid model and infrastructure basics. Based on the different literature, the smart architecture is proposed and an low data latency and low energy consumption for communication topology are implemented for smart grid. From [8] it is proposed that a modern electrical grid infrastructure for higher efficiency and reliability via automated control, modern communications, sensing, high power converters, metering technologies, and energy management schemes. In this approach, smart grid infrastructure is highly complex. From [18] an approach about the recent smart grid management and protection systems. This approach focusses the performance of protection in smart grid. From [20] an efficient and privacy-preserving aggregation methodology for securing the smart grid communications is proposed. The performance of this method is more securable and flexible in terms of computation for consumer communication overhead. A distributed interval integration algorithm [9] is explained. In this method, the generator was chosen to collect the device output to develop an overlap function. This proposed algorithm guaranteed that the mere variations in the input intervals resulted only in mere variations of the integrated result. From [6] a method is proposed for comparative analysis for fault detection and classification using Functional Analysis and Computational Intelligence (FACI).This method lags with the computation time to detect the fault. A smart grid controller [1] for optimizing energy consumption is proposed. A framework for high level power estimation of MPSoCs architecture on FPGA is proposed in

[19]. The approach uses different interconnects method and memory hierarchies. This method does not discuss the power model of real time system-level design space exploration. From [21] proposes a algorithm of task scheduling issue to avoid the inter-core communication overhead in on-chip MPSoC architecture. This algorithm is obtained with minimum optimal solution for memory usage with reduced schedule length.

From [13] it describes the changing requirements due to privatization and the deregulations have created needs for analyzing information from different sources within DW. These needs require new high performance solutions represented by the new data warehouse of Supervision Control And Data Acquisition (SCADA). The system security and memory protection service in a wide range of NoC-based MPSoC platforms is discussed in [7]. The protection mechanism is to prevent the malicious data throughout system with the help of self-contained NoC at Network Interface (NI) layer.

A technique to eliminate inter core communication overhead on Multiprocessor System-on-Chips (MPSoCs) with combined computation and communication task scheduling of delivering virtualized applications is proposed in [12]. From [14] it is exposed that the emergence of MPSoC processor to efficiently exploit the low latency and high bandwidth of modules in the hierarchy in terms of programming framework. An MPSoC architecture which has thermal effect was presented by [5]. It leads to cause violation in timing constraints. Therefore, MPSoC have high power density and temperature which will degrade the reliability and increase the cost. The MILP solver provides the improvement of performance and reduces peak power. Hence, scheduling and assignment technique are used to overcome peak temperature.

State Estimator (SE) in the electric power grids makes technological alterations and is being proposed as a part of the smart grid development. SE turns out to be a key operation in supervisory manipulation and planning of electric power grids. It serves to observe the state of the grid and enables Energy Management Systems (EMS) to participate in quite a lot of foremost control and planning tasks corresponding to open near real-time community units for the grid, optimizing power flows, and bad information detection/evaluation as acknowledged by way [22] and [16].

In the past decade, Multiprocessor system on-chips (MPSoCs) have been developed as a significant class of Very Large Scale Integration (VLSI) systems. In VLSI system, an MPSoC is a system on-chip that includes most or the entire components essential for an application and it utilizes the multiple programmable processors as system components. MPSoCs are widely used in networking, communications, signal processing, multimedia and other applications. An method used to [15] analyze the design challenges faced by MPSoC designers at all levels. At the application level, there is an necessity for programming models and communications APIs that permit applications to be simply re-configured for much different possible architecture without rewriting, while at the same time guaranteeing efficient production code. From [13] it discusses about the communication architecture analysis for multi-processor Systems-on-Chips (MPSoCs) and it leverages a SystemC-based platform to simulate a whole multi-processor system at the cycle-accurate and accurate signal level.

The packet data flow affects the MPSoC performance and power consumption that are flowing in the network. It was [25] discussed about the packetized on-chip communication power model and quantified the effect of packet size deviation on the performance and energy consumption. For assisting MPSoC programmers with C application parallelization, MPSoC Application Programming Studio (MAPS) is developed by [4] which is an integrated framework. To extract coarse coarse-grained parallelism in C, a novel granularity level has been proposed on MAPS partitioning tools. A temperature-aware task allocation and scheduling algorithm are proposed by [23] for MPSoC embedded systems. In that method, both power-aware and thermal-aware schemes are investigated to the task allocation and scheduling.

To raise the lifetime reliability of the platform-based MPSoC embedded systems, new solutions have been proposed by [11]. An analytical model is presented to evaluate the lifetime reliability of platform-based MPSoC, when executing periodical tasks. From [24] it proposed a multi-task mapping/ scheduling heuristic based on the QEA technique considering data and temporal parallelisms as well as task parallelism for MPSoC. Compared with an ILP (Integer Linear Programming) approach, experiments with real-life examples show the feasibility and the efficiency of the proposed technique. From [3] a methodology was introduced for power modeling of global framework for power/energy evaluation and optimization of different MPSoC. In order to attain more accurate power evaluation and some improvement of power model, it must be realized by focusing on more complex heterogeneous platforms. A innovative [11] technique for task share and scheduling algorithm that are added to the processors with aging effects, based on the annealing technique. This technique exploits the life time of MPSoC design and limits the various parameter performances. A method [2] proposes a system level flexible methodology for Heterogeneous multi-processor system-on-chip (HtMPSoCs), which addresses the fault tolerance and performance trade-off such, that an economic utilization of resources and it will minimize the overall cost of the application.

Requirements of a Smart Power Grid

The power grid must be designed securely at system level and device level. The following are the requirements for a secure power grid.

Real-Time Behavior

The high frequency (per 0.1 – 10s) of event occurrence in the power grid makes the control devices incapable for real-time control and reconfiguration. The conventional data collection and contingency analysis are performed only for every few seconds and the State Estimator (SE) executes only every 5 minutes. It cannot fulfill the “real-time condition”. The factors which hinder the implementation of a real-time management scheme in smart grid are communication delay, and large amount of raw data. Real-time scheduling are required to guarantee end-to-end real-time behavior. The fault detection and mitigation processes can be accelerated via distributed processing. And the other characteristics to be considered are

- *Grid Sensitivity*
- *Fault Resiliency*
- *Communication technologies and networks*

Smart hybrid architectures for Smart Grid

Data Processing Architectures

The current topology of the smart grid network is arranged according to the demand of the consumers. The distribution network is organized into multiple subgrids and consequently, forms a hierarchical topology. The key elements of architectural model are: (i) State Estimator (SE) - the state estimator is an integral part of the overall monitoring and control systems of transmission networks. It is used for the local measurements in a substation by the local estimator to estimate the states and then, it is transmitted to a central coordinator for further estimation. Remote Terminal Unit (RTU) acts as an intermediate node between substations and power grid network. RTU receives information from the substation within the neighborhood and multiple RTUs are utilized to cover several substations (iii) Grid network represents the central power grid to control the entity of a utility provider. This node is responsible for data processing and maintenance of all substations.

Decentralized Architecture

In this architecture, only RTUs have data processing and storage capabilities which are explained. SEs transmits data periodically to the respective RTU, but instead of forwarding the data, the RTU stores and processes this data locally. In the decentralized architectures, since complete data are available at the RTUs, data aggregation is possible. For instance, RTUs can aggregate data while reporting to the grid network. The grid generates queries to retrieve information from RTUs only when required. Thus, RTUs act as central entities in this architecture.

Distributed Architecture

In distributed architectures, all SEs have data processing and storage capabilities. SEs periodically sense and store the data values locally. Also, they process the data locally by the processor. Grid initiates a query to fetch the state of the substation, which is forwarded to the RTU and in turn to the SEs. RTUs process the query and send the reply to the grid. Thus, making the architecture distributed is made as Distributed State Estimations (DSE).

Signal Processing and the Smart Grid

While some researchers could consider the power grid SE research quite mature, new tactics for SE must be developed as the power grid turns into more intricate, extra interconnected, and more shrewd. Any development in the Signal Processing (SP) can largely facilitate and benefit the progress of the smart grid. In return, study of SE inside the framework of one of the most problematic man-made systems can invigorate the signal processing research community. In distributed estimation, several nodes estimate a usual parameter vector through neighborhood collaborations. Within the case of Multi Area State Estimation (MASE), the measurement of each area relates to a small part of the whole state vector. Hence, ensuing computational and communication expenditures of an allotted estimation procedure rely upon whether local knowledge of the entire state vector is required or not.

The vector equation as

$$[\sum_{p-1}^{P} H_p^T(j) W_p^{-1} H_p(\mathrm{j})]\Delta w(j) = \sum_{p-1}^{P} H_p^T(j)[G_p - h_p(w_p(j))]$$

$$\widehat{W}(j+1) = \widehat{W}(j) + \Delta w(j) \qquad .....(1)$$

Where H_p(j) is the measurement Jacobian of area p obtained with the local state estimate $\widehat{Wp}$ (j). The signal processing community can contribute to the research on MASE for the future grid by building upon recent advances in distributed estimation.

Remote Terminal Unit (RTU) in Smart Grid

RTU is an acronym for Remote Terminal Unit. An RTU is an electronic device that is controlled by a microprocessor. The device interfaces with physical objects to a Distributed Control System (DCS) or Supervisory Control and Data Acquisition (SCADA) system by transmitting telemetry data to the system. A remote terminal unit is a control device that helps the master device (SCADA), monitors and communicates with grid devices placed across different substation locations. They can be used in a wide variety of electrical and process automation devices where SCADA is required to monitor and control equipment. RTUs are used to collect data from the equipment in the field and to transmit commands back from the master device. This can be done through wired connections (telephone lines, cables, Ethernet) or wireless connections. The data are sent to the operator in real time so that, they are aware of the situation and can carry out any necessary changes to the grid.

Restructuring of architecture for Smart Grid

MPSoC architecture for smart grid

A Multiprocessor systems-on-chip (MPSoC) is a single chip to meet strict requirements of embedded applications, such as real time, high performance and high reliability. Technology scaling has enabled the integration of an increasing number of Processor Elements (PEs) on a single chip (MPSoC). The increasing computational power in turn supports the ability of permitting higher number of applications to execute simultaneously on the chip. The increasing complexity and the demanding workload of these applications, coupled with short time-to-design and time-to-market of portable devices has given rise to the application specific Multiprocessor System on-Chip (MPSoC) platforms for multimedia. The increase in demand of the NoC-based MPSoCs in terms of processing elements (PEs), new applications can run simultaneously on such systems, which require management techniques to meet the application constraints.

In this architecture configuration, it is possible to design either Symmetric Ht-MPSoC or a AHt-MPSoC. In the case of SHt-MPSoC, the number of private processors and hardware accelerators, which are shared, are equal in number. Whereas in the case of AHt-MPSoC, the hardware accelerators are interfaced with the different processors and each processor differs with one another. The proposed Ht-MPSoC architecture shares the hardware accelerators among processors. The result is based on the MIP formulation, which is used to explore the very huge space of feasible configurations in reasonable time. In the AHt-MPSoC architecture, for improving the performance of the processors, application-specific instructions are effectively used. In these processors, the run time of the critical computations is degraded by the usage of newly added instructions implemented in HW accelerators. This HW accelerator utilization of each processor can be either connected by the system bus or memory controller to the instruction pipeline. The MILP model space explorations used to connect the computational patterns are represented by existing method on the various applications, which degrade the overall area utilization with respect to application-performance restraints. The necessary limit to be taken for producing the execution time of each processor is maintained. Thus, the MIP model is able to determine an ideal AHt-MPSoC configuration that attains desired area and performance. The proposed model, in order to prove the increasing performance and energy consumption of the system, is reduced compared to the conventional method. The modified MILP architecture is shown in Fig.1.

The conventional work focuses only on area reduction without considering the performance improvement. Whereas, the proposed work, on examining the entire feasible sharing configuration reduces the area usage and the execution time is desired based on the performance constraints. In all possible ways of sharing configuration, the usage of area diminishes and satisfies the performance in terms of its execution time.

Modified Integer Linear Programming for MPSoC configuration

An MPSoC is a complete computational system integrated in a single chip, combining multiple processing elements (PEs) as main components. In addition, the MPSoC is built by interconnecting Intellectual Property (IP) cores to local ports of NoC routers. Messages are transmitted between IP cores wrapped in NoC packets. The growing interest in MPSoCs lies in its ability to combine high performance and flexibility and its reconfigurable feature. In this work, the MPSoC uses multiple Remote Terminal Unit (RTUs) along with other Distributed State Estimator (DSEs) to implement a system. By sharing hardware accelerators among the cores, the conventional FPGA-based MPSoC architectures has been extended. In these architectures, cores on the FPGA may have different resources to share in different manners.

The application needs a fast and accurate exploration tool for exploring the enormous space of possible configurations of MPSoC on FPGA. Taking this reason, a Mixed Integer Linear Programming (MILP) model is proposed further in order to determine the MPSoC configuration. It consumes the least HW resources and also values the application execution time constraints. With the help of this MIP model, the design space of several hundreds of private and shared HW accelerators can be explored in a sensible time with high accuracy. Figure 1.1 shows the proposed MILP MPSoC architecture. As the MPSoC is designed with MILP, the computational complexity of the proposed system is significantly reduced.

Fault Indication and Propagation Infrastructure (FIPI)

An essential part of the demonstrator is the Fault Indication and Propagation Infrastructure (FIPI). When the faults are detected in the instruments, the operation of the MPSoC is threatened, unless proper action is taken. The FIPI employs an interrupt-like function that propagates fault indications into fault detection instrument and a system-level fault indication instrument, which controls a global fault indication. The fault origin is identified by the fault detection instrument, but this can take a particular time to access. The FIPI applies a noticeable track back to the instrument for easy indication of the fault origin.

Fault Tolerant MPSoC

The fault tolerance enables the correct operation of a device even in the presence of faults (errors) and it has been established along with the rise of the very first devices used in critical applications. To enable correct operation in the presence of errors, fault tolerance provides techniques that are capable of error-detection, i.e. to detect the presence of errors, and error-recovery, i.e. recover the system from errors. Usually, this is achieved by introducing a hardware and time redundancy. The methodology accepted in the scheme is defined as follows. In a NoC based homogeneous MPSoC, three different redundancy schemes are induced and the obtained results are compared. The first scheme is Dual Modular Redundancy (DMR), which is a hardware redundant technique. In DMR, two processes are performed simultaneously. By using comparator, the outputs are compared to check whether the fault has occurred or not.

To execute critical tasks, the hardware is needed additionally in this scheme. The Temporal Redundancy (TR) is the second technique in which two processes are performed sequentially and then, the comparator is used to compare the fault detection. The third technique is parity Error Correcting Code (ECC) which compares the input and the produced output to detect faults. It seems to have intermediate hardware compared to conventional schemes.

Logic Based Real time Multiprocessor Architecture (LRMA)

The Logic based Real time Multiprocessor Architecture (LRMA) is designed with the combination of DSE and RTU based MPSoC. The power consumption of this system is significantly minimized. An MPSoC is a system on-chip that includes most of the entire components essential for an application and it utilizes the multiple programmable processors as system components. The transmission parameters are transmitted to the corresponding RTU, which performs as an intermediate between all substations. In the proposed method, to minimize the execution time, the DSE is used. The proposed LRMA is attained with multiple factors that are low power consumption and low execution time.

Logic Based Fault Tolerance MPSoC (LFT-MPSoC)

The proposed Logic based Fault Tolerance MPSoC (LFT-MPSoC) is designed with the combination of DSE and RTU based FT-MPSoC method. The power consumption and fault tolerance of the proposed method are significantly achieved. Fault tolerant MPSoC has a good performance and low hardware overhead. It provides the reliability of the system under real-time constraints. In the event of the failure, Fault tolerance property enables a system to continue operating properly. The RTU acts as an intermediate between all substations. The DSE is utilized to reduce the execution time of the substation and to deliver the feedback on local faults. The numerous factors achieved by the proposed LFT-MPSoC are fault tolerant, reliability, low power consumption and low execution time.

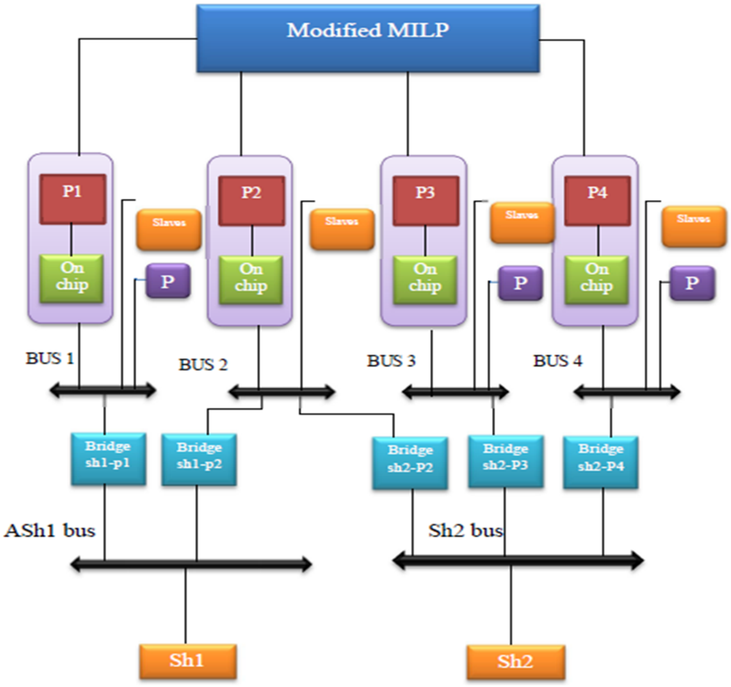

Figure 1. MILP MPSoC architecture

Testing Analysis Strategies

When there is increasing needs of performance and scalability, the commercial reconfigurable architectures such as available COTS FPGAs are used. Additionally, the usage of dynamically reconfigurable architectures can deliver a higher degree of flexibility, scalability and simultaneously maintains the computing power. The major challenge in MpSoC design is system integration. Complex hardware and software component interactions pose a serious threat to all kinds of performance pitfalls, including transient overloads, data loss and missed deadlines.

Static State Estimation (SSE)

For the last four decades, more researches on SE have been conducted about SSE. Peculiarly, for the reason that the ordinary monitoring technologies, comparable to those carried out in the SCADA method, can be best taken non synchronized measurements in each two to four seconds. Additionally, to reduce the computational complexity required in implementing SE, also it estimates and large up-to-date for simplest once in every few minutes. Thus, the usefulness of SSE as a means to furnish actual-time monitoring of the power grid is quite restrained to apply.

As specified by Yih-Fang Huang et al (2012), in an N-bus system, the (2N-1) x 1 state vector has the form $w=[\Phi_2, \Phi_{3,\ldots\ldots}\Phi_{N,} |E_N|]^T$ where Φ_i denotes the phase angles and |Ei| refers the magnitudes of the voltages at the *i*th bus. The phase angle Φ_1 at the reference bus is assumed and is normally set to zero radians. The measurements are typically obtained within SCADA systems, and are specifically related to the state vector by an over determined system of nonlinear equation given in equation (2).

$$G=h(w)+n \qquad \ldots\ldots(2)$$

where h(w) is a set of M nonlinear functions of the state vector (determined by Kirchhoff's laws and the power network admittance matrix) and n is a zero-mean Gaussian measurement noise vector with covariance matrix $b_n \epsilon R^{MxM}$.

In the traditional SSE approach, the state vector is estimated from the measurement Equation in (2) using the Weighted Least Squares (WLS) method as mentioned by schweppe et al (1990). In particular, the SSE problem is solved by finding

$$\widehat{W}= \arg\min[G-h(w)]^T D^{-1}[G-h(w)] \qquad \ldots\ldots (3)$$

where weighting matrix D is commonly taken as diagonal with elements related to background noise covariance as D=Cn. The solution for $\widehat{W}$ is obtained in an iterative fashion by linearizing equation (2) around the available estimation (at iteration j), and applying the Gauss-Newton algorithm to improve the estimate, and by using the following equations:

$$N(j)\Delta w(j)=H^T(j)D^{-1}[G-h(w(j))] \qquad \ldots\ldots(4)$$

$$\widehat{W}(j+1) = \widehat{W}(j)+ \Delta w(j) \qquad \ldots\ldots(5)$$

where $N(j)=H^T(j)\, D^{-1} H(j)$ is the gain matrix at iteration j. Equation (4) is usually referred to as the normal equation. A more recent process for reducing the computational price is to use a nested, or multilevel, system of the nonlinear dimension model as mentioned by Gomez et al 2011. This method can sustain the development in size, complexity, and data information. It is designed to operate at extraordinary phases of the modelling hierarchy to accomplish very big-scale interconnection-vast monitoring. This method uses the same over determined set of measurement equations as in (2). The equations are then unfolded into L sequential WLS problems by introducing a set of intermediate variables $X=\{x_{1,}x_{2,}\ldots.x_L\}$ with the following nested structure,

$$\begin{aligned} &G=f_1(x_1)+n \\ &x_{1=}\ f_2(x_2)+ n_1 \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ &x_{L-1=}\ f_L(x_L)+ n_{L-1} \\ &x_{L=}\ f_{L+1}(w)+ n_L \qquad \ldots(6) \end{aligned}$$

The set X is chosen such that the solution of the nested system of Equations (6) offers some desired advantage over solving (2), e.g., reduction of the computational complexity or the amount of information exchanged between different levels. This is particularly an appealing solution, when the measurement model can be factorized into separate linear and nonlinear parts.

Latency

Latency is distinct at the time that intervenes the injection start of the its data into the network of the transmitter section and its arrival at the receiver section. The latency is used as the performance metric. It is assumed that the packet latency time is the instant when the packet is created, to the time when the packet is delivered to the receiver section. Also,it is assumed that the packets are consumed immediately once they reach their receiver section. For a flit to reach the receiver section, it must travel through a path consisting of a set of links and other components.

Using network calculus, the latency D_{nocQi} in each node Q_i is expressed as follows.

$$D_{nocQi} = \frac{B_{wij}}{U_{ii}} + Llat \qquad \ldots\ldots(7)$$

where B_{wij} is the service bandwidth and L_{lat} is the latency. Thus, the latency can be intended based on the equation. The latency for the application can also be intended after evaluating the latency D_{bi} of each switch b_i. In the simulator, each process is linked with a traffic generator that injects flits according to the real time traffic model at a deterministic rate which is varied from low traffic (data) to heavy traffic (video). In this evaluation, data flows are considered dependent on each other and the interaction causes congestion in some switches. It increases the latency at a high injection rate.

Power

Link power

From the power models, for a NoC router and the power model considering the cross-coupling effect for N-wire interconnect, we may determine the total power for an N-wire link per unit length as follows: gate leak wire bias short.

$$P_{link} = \frac{1}{2} N_{wire} V_{sv}^{2} (C_{self}\alpha_{saw} + (C_{ou}\alpha C_{ou})f + N\tau\alpha_{saw} VI_{short}.f + N.(VI_{bias\text{-}wire} + VI_{leak,gate}) \qquad .....(8)$$

where N_{wire} is the total number of wires in the link, C_{self} and C_{coupl} are the self and coupling capacitance of a wire and neighbouring nodes, respectively. α_{saw} is the switching activity on a wire and α_{Cou} is the switching activity with respect to the adjacent wires, τ is the short circuit period, V_{sv} is the supply voltage and $I_{short,bias}$, wire and $I_{leak,gate}$ are currents

Static power consumption

Static power is the power immoral by a gate or a wire, after it is inactive or in an active state. The static power is mostly inclined by the structure of the circuit. The static power dissipation can be more precise by the equation

$$E_{Static} = VI_{bias,wire} + VI_{leak,gate} \qquad .....(9)$$

Energy

At upper levels, energy spent due to the scattering of one bit of data from one router (R_1) to another (R_2) via the links is a utility of the number of routers and the number of links. The full energy can be talked into the energy spent on the replacements and energy spent per wire or link coldness traveled. The total energy(t) can be intended as follows

$$\text{Energy}(t) = \sum_{i=1}^{Ni} \Phi_{consumed@time}(t) + \sum_{i=1}^{Ni} \Phi_{consumed@switch,link}(t) \qquad .....(10)$$

where $\Phi_{consumed@time}(t)$ is the energy spent, at time t, on the link l_i, $\Phi_{consumed@consumed@time)}(t)$ is the energy consumed inside the switch sw and N_{link} and N_{switch} are the number of links and switches, respectively involved in transporting the application flows. Using network calculus arrival curves, the total energy consumption can be calculated.

Results and Discussion

From the proposed modified MPSoC architecture, the performance parameter model has attained the optimal results for the following parameters. They are delay, energy consumption and power consumption. In the development section of this proposed scheme, the work is done under the combination of packages. This computes further improvement, which is done in the previous conventional works to improve the performance of the smart grid architectures. As mentioned above, the performance measurements are compared with the existing system.

Table 1. Energy Consumption comparison of Proposed AHt-MPSoC Architecture

Energy (j)					
Method	Barnes	Ocean	Radiosity	Raytrace	Avg
SHT	0.275	0.312	0.298	0.278	0.29
AHT	0.215	0.278	0.266	0.233	0.248
VRMA	0.118	0.185	0.166	0.185	0.163

Table 2. Delay comparison of Proposed AHt-MPSoC Architecture

Delay(Ps)					
Method	Barnes	Ocean	Radiosity	Raytrace	Avg
SHT	78	96	86	77	84
AHT	66	74	66	55	65
VRMA	45	48	47	49	47

Table 1 and Figure 2 depicts that the comparison of energy consumption of different architectures. The proposed AHt-MPSoC Architecture is achieved to reduce the energy consumption of different benchmark which gives better performance compared with SHT and AHT-MPSoC architecture. In Barnes, the energy consumptions of the SHT, AHT and the proposed architecture are 0.275, 0.215 and 0.118 respectively. It shows that the proposed AHt-MPSoC architecture gives better improvement than the conventional architecture.

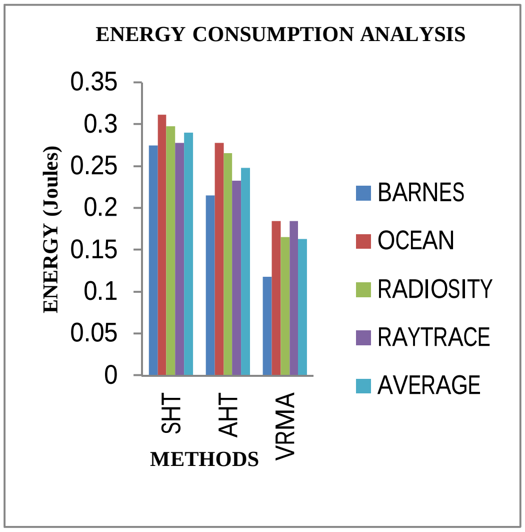

Figure 2. Energy Consumption comparison of Proposed AHt-MPSoC Architecture

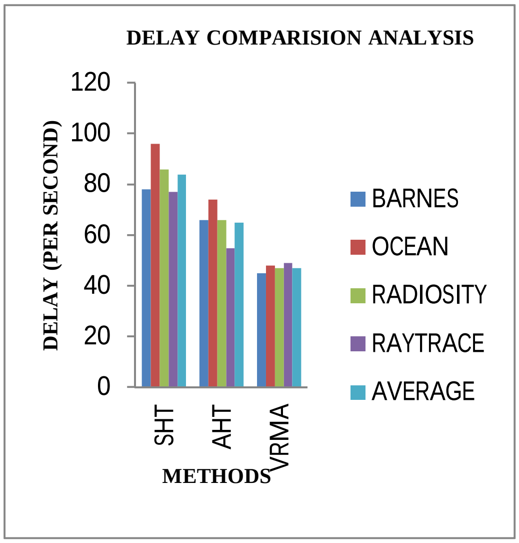

Figure 3. Delay Comparison of Proposed AHt-MPSoC Architecture

The proposed AHt-MPSoC Architecture is attained to minimize the delay of different benchmark which gives better enhancement compared with SHT and AHT-MPSoC architecture. Delay comparison of different architecture is shown in Table 2.In Barnes, the delay of the SHT, AHT and the proposed architecture are 78, 66 and 45 respectively, which shows that the proposed AHt-MPSoC architecture is better than the conventional architecture and it is shown in Figure 3.

Table 3, 4 and Figure 4,5 presents the comparison of static power of different architectures with FT-MPSoC. Static power minimization of different benchmark is achieved by the proposed FT-MPSoC Architecture and it gives better attainment compared with the base MPSoC architecture. It depicts that the proposed FT-MPSOC architecture provides better improvement than the conventional architecture.

Table 3. Static Power Comparison of Proposed AHt-MPSoC Architecture

Static Power(mw)					
Method	Barnes	Ocean	Radiosity	Raytrace	Avg
SHT	220	240	255	233	237
AHT	180	190	196	210	194
VRMA	120	110	121	123	118

Table 4. Static Power Comparison of Proposed MPSoC with FT unit Architecture

Static Power(mw)					
Method	Barnes	Ocean	Radiosity	Raytrace	Avg
MPSoC	180	190	196	210	194
FT-MPSoC	150	140	135	138	140

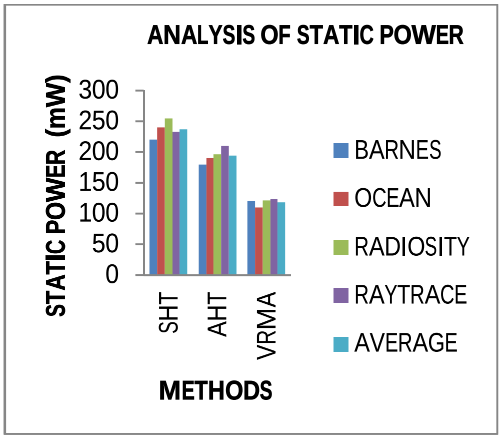

Figure 4. Static Power Comparison of Proposed AHt-MPSoC Architecture

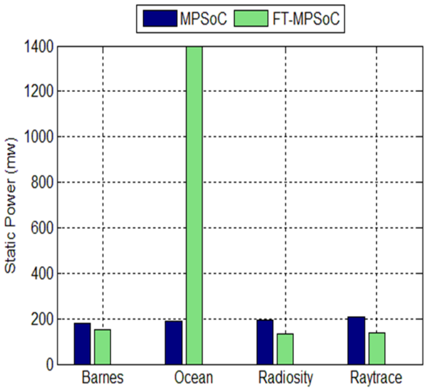

Figure 5.Static power Comparision of MPSoC with proposed Fault tolerant MPSoC

Conclusion

In this paper, the significance of restructuring the existing smart grid architecture using MPSoC is designed. The significance of the fault tolerant unit FT-MPSoC with existing smart grid architecture is designed The MPSoC is designed with modified MILP algorithm and the computational complexity of the proposed system is significantly reduced. Hence, the MPSoC with the fault tolerant unit is designed and the architecture performs better than the existing architectures. As the proposed MPSoC is designed with modified architecture the delay, energy and Power consumption are calculated and they compared with the existing architecture which gives better performance than the existing architectures.

References

[1] Al-Ali AR , “Smart grid controller for optimizing HVAC energy consumption”, International conference on Computer Systems and Industrial Informatics (ICCSII), 2012, pp.1-4.

[2] Amin, Mohsin, Mihkel Tagel, Gert Jervan & Thomas Hollstein , “Design methodology for fault-tolerant heterogeneous MPSoC under real-time constraints”, IEEE 7th International Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2012, pp.1-6.

[3] Ben Atitallah, Rabie, Eric Senn, Daniel Chillet, MickaelLanoe & Dominique Blouin , “An efficient framework for power-aware design of heterogeneous MPSoC”, IEEE Transactions on Industrial Informatics, vol 9, no.1, 2013, pp.487-501.

[4] Ceng, Jianjiang, Jeronimo Castrillon, Weihua Sheng, Hanno Scharwachter, Rainer Leupers, Gerd Ascheid, Heinrich Meyr, Tsuyoshi Isshiki & Hiroaki Kunieda , “MAPS: an integrated framework for MPSoC application parallelization”, ACM Proceedings of the Design Automation Conference, 2008, pp. 754-759.

[5] Chantem, Thidapat, Sharon Hu & Robert P Dick , “Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol 19, no.10, 2011, pp. 1884-1897.

[6] De Souza Gomes A , “Detection and classification of faults in power transmission lines using functional analysis and computational intelligence”, IEEE Transactions on Power Delivery, vol.28, 2013, pp.1402-1413.

[7] Grammatakakis, Miltos, Kyprianos Papadimitriou, Polydoros Petrakis, AntonisPapagrigoriou, George Kornaros, IoannisChristoforakis, OthonTomoutzoglou, George Tsamis & Marcello Coppola , “Security in MPSoCs: A NoC Firewall and an Evaluation Framework”, IEEE Transcations on Computer Aided Design of Integrated Circuits and Systems,vol.34, 2015, pp.1344-1357.

[8] Gungor, VC , “Smart grid technologies: communication technologies and Standards”, IEEE Transactions on Industrial Infrastructure, vol.7, 2011, pp.529-539.

[9] Hairong, Q , “A resilient real-time system design for a secure and reconfigurable power grid”, IEEE Transactions on Smart Grid, vol.2, 2011, pp.770-781.

[10] Huang, Lin, Feng Yuan & Qiang Xu , “Lifetime reliability-aware task allocation and scheduling for MPSoC platforms”, Conference on Design, Automation and Test, 2009, pp. 51-56.

[11] Huang, Lin, Feng Yuan & Qiang Xu , “On task allocation and scheduling for lifetime extension of platform-based MPSoC designs”, IEEE Transactions on Parallel and Distributed Systems, vol.22, no.12, 2011, pp. 2088-2099.

[12] Liu Weichen, Yu Wang, Xuan Wang, Jiang Xu & Huazhong Yang, “On-Chip Sensor Network for Efficient Management of Power Gating-Induced Power/Ground Noise in Multiprocessor System on Chip”, IEEE Transactions on Parallel and Distributed Systems, vol.24, no.4, 2013, pp.767-777.

[13] Loghi, Mirko, Federico Angiolini, Davide Bertozzi, Luca Benini & Roberto Zafalon, “Analyzing on-chip communication in a MPSoC environment”, IEEE Proceedings on Design, Automation and Test, vol.2, 2004, pp.752-757.

[14] Marongiu, Andrea & Luca Benini, “An OpenMP compiler for efficient use of distributed scratchpad memory in MPSoCs”, IEEE Transactions on Computers, vol.61, no.2, 2012, pp.222-236.

[15] Martin & Grant, “Overview of the MPSoC design challenge”, ACM Proceedings of the 43rd Annual Design Automation Conference, 2006, pp. 274-279.

[16] Monticelli, “Electric power system state estimation”, IEEE Proceedings, vol.88, no.2, 2000, pp.262–282.

[17] Motra, AP, Bora & Chakrabarti,I, “An efficient hardware implementation of DWT and IDWT”, International conference on Convergent Technologies for Asia-Pacific Region TENCON, vol.1, 2003, pp.95–99.

[18] Pagani, GA & Aiello, M, “The power grid as a complex network: a survey”, International Journal on Physica A:Statistical Mechanics and its Applications, vol.392, 2013, pp.2688-2700.

[19] Roberta Piscitelli & Andy D Pimentel, “A High-Level Power Model for MPSoC on FPGA”, IEEE Transactions on Parallel and Distributed Process, vol.11, no.1, 2012, pp.16-20

[20] Rongxing,L, “EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications”, IEEE Transactions on Parallel Distributed Syststem,vol.23, 2012, pp.1621-1631.

[21] Wang, Xuan, Jiang Xu, Wei Zhang, Xiaowen Wu, Yaoyao Ye, Zhehui Wang, Mahdi Nikdast & Zhe Wang5, “Actively Alleviate Power Gating-Induced Power/Ground Noise Using Parasitic Capacitance of On-Chip Memories in MPSoC”, IEEE Transactions on Very Large Scale Integration Systems, vol.23, 2015, pp. 266-279.

[22] Wu, FF, “Power system state estimation: A survey”, International Journal of Electrical Power Energy System, vol.12, no. 2, 1990, pp.80–87.

[23] Xie, Yuan & Wei-Lun Hung, “Temperature-aware task allocation and scheduling for embedded multiprocessor systems-on-chip (MPSoC) design”, Journal of VLSI signal processing systems for Signal, Image and Video Technology, vol.45, no.3, 2006, pp.177-189.

[24] Yang, Hoeseok & Soonhoi Ha, “Pipelined data parallel task mapping/scheduling technique for MPSoC”, IEEE International Conference on Design, Automation and Test, 2009, pp.69-74.

[25] Ye, Terry Tao, Luca Benini & Giovanni De Micheli, “Packetized on-chip interconnect communication analysis for MPSoC”, IEEE Conference on Design, Automation and Test, 2003, pp.344-349.