Report on MPSoC'04

Students' Summary of Lectures

Xi CHEN

TIMA Laboratory
46 Avenue Félix Viallet
38031 Grenoble CEDEX
France
Xi.Chen@imag.fr

Contents

Introduction

This is a summary of MPSoC'04 lectures (except for the business session) by TIMA-SLS students: Youssef Atat, Marius Bonaciu Aimen Bouchhima, Iuliana Bacivarov, Youngchul Cho, Arnaud Grasset, Frédéric Hunsinger, Lobna Kriaa, Lorenzo Pieralisi, Adriamo Sarmento, Wassim Youssef. The whole document was assembled and compiled by Xi Chen.

WARNING

Since the summaries of the presentations were made by different students, the style of the paragraphs may be different. The content of this document may not correspond to the whole content of MPSoC'04. The summaries reflect the students' view on the presentations.

For each lecture, in addition to the abstract provided by the speaker, a report on the work gives the context of work, the addressed problem and the proposed solution.

Summary of Lectures

SESSION 1: TUTORIALS

LECTURE 1.1: POWER MANAGEMENT IN WIRELESS SOC

Author: *Jan Rabaey* , UC Berkeley, USA

Abstract:

SOCs for wireless systems (ranging from the high-data rate LANs to the low-data rate sensor networks) combine a wide variety of concurrent functions with quite contrasting activity levels. To avoid wasting energy in non-active modules (either through unwanted activity or leakage), effective shut-down of these modules is essential. A chip-level supervisory power management function is introduced to do just that. One of the most important features of the approach is the introduction of the "power domain" concept, in analogy with the clock domains which are now common in industry.

Area of interest:

This tutorial deals with methods for power management applied to wireless SoC.

Addressed problem:

Leakage power is going to become more important than switching power. Actually most of the leakage comes from memories. Due to the increasing importance of power leakage, power predictability becomes very difficult. Leakage is a problem for the design of fast and low power SoC. The optimal point for leakage depends on the design and the activity of the circuit.

Solution:

The use of low power dedicated to logical cells enables to control the power. Power control according to the activity of the circuit can be done using calibration of the process. As SoCs combine a wide variety of concurrent functions with quite contrasting activity levels, the concept of power domain is introduced in the tutorial. A power scheduler "turn on and turn off" for each power domain allows dynamic voltage control. A session controller opens and closes sessions to control the power of the process. A connection must be established between two modules (with two different power domains) to establish a communication. The method has been used for the design of network processors. The use of power domains is interesting to help to control power in system-on-chip.

LECTURE 1.2: MEMORY ISSUES IN SOC

Author: *Doris Keitel-Schulz*, Infineon Technologies, Germany

Abstract:

Memory content in SOCs is growing steadily and has reached already more than 50% of the die area. The technical requirements for these memories however are varying dramatically. To achieve performance and cost optimized circuits, integration of all memories is not always possible with today's design methodologies and manufacturing technologies. An outlook on possible next generation memories will complement the presentation.

Area of interest:

This paper is related to different types of memories that can be used in today's MP-SoC architectures. The paper is concentrated on volatile types of memories (SRAM, DRAM) and non-volatile (Flash).

Addressed problem:

During the last years, the size of the memory which needs to be implemented into the MP-SoC increased from 20% in 1999, to 60% in 2004. By the year 2014 the memory will arrive to be almost 95% of the entire MP-SoC. (Slide 3)

Solution:

Choosing the right type of memory for an application depends on the overall requirements of power/ speed/ surface. If high-speed is the main target, choosing SRAM type of memory might be the best solution, taking into account that the read/ write processes are the fastest for this type of memory. In case the target is the surface, the best solution will be choosing the eDRAM type of memory. The Flash type of memory is the best solution in case the power consumption is the main target, because it is a non volatile memory, so refreshing processes are not required. (Slide 11)

Memory issues are more relevant in case of multimedia applications (Digital still camera (Slide 9), Security Safety Monitoring Systems (Slide 10), Video Coding/Decoding), where there are the bottlenecks of the design/functionality. The integration of the memory into chips is a very costly thing. Exploration is required. The best idea is to put chips on top of each others (multiple layers?).

Even so, the evolution of the technology might solve many of the already existing problems (Slides 12&13). This is because, by 2007, the use of forty times bigger memory will be possible (from 256MB today to 10GB in 2007, in case of the Flash memory, and from 64MB today to 2GB in 2007 for DRAM memories) (Slide 9). Also, the main future challenges for integration which need to be overcome will be functionality vs. process cost, performance vs. process features, special process vs. area utilization, reliability issues and time to market (Slide 8).

LECTURE 1.3: DESIGNING RELIABLE, POWER-EFFICIENT SYSTEMS

Author: *Mary Jane Irwin* , Pennsylvania State U., USA

Abstract:

Designing reliable systems in the nanometer era requires a radically different approach than previously. Computing systems and circuits that are designed to handle the worst-case scenario are becoming too costly in terms of power, area and performance. An alternative approach is make the system performance metrics and design techniques statistical in nature and addresses the average, not worst, case. A statistical approach permits the use of inexpensive design-time and run-time solutions where the worst-case situations are handled by the micro-architecture and/or the software. This tutorial will first survey issues that impact system reliability in the nanometer era including noise (from wire cross-talk and substrate coupling) and soft-error upsets. Next several statistical optimizations for designing reliable memory systems and reliable interconnect that span the circuit, micro-architecture, and software levels and that trade-off performance, reliability, and power consumption will be discussed.

Area of interest:

Design constraints involve making a trade-off between the time (of execution), the reliability and the power. This tutorial deals with the design of reliable power-efficient systems.

Addressed problem:

System reliability is limited by many sources of noise. The tutorial focuses on crosstalk and soft errors.

Solution:

Crosstalk is the interaction due to capacitive coupling between signals on adjacent layers. The drawback of the shielding layer is wiring area doubles. The signal coding to prevent crosstalk needs 50% extra-bit lines and an encoder and a decoder. The tutorial presents a solution based on a crosstalk aware interconnection. The transmission delay (in number of clock cycles) depends on the signal value. An analyzer computes this number.

An example has shown an average 31.5 % performance improvement over the original scheme. But the power is bigger. Error detection method could be used to improve reliability. The different error detection schemes provide different capabilities to detect errors and different energy consumptions. The presentation introduces an adaptive error protection to save energy in function of the environmental factors and operational conditions. An example has shown 8% energy saving for the same level of reliability.

Another kind of error is the soft errors which are induced by external radiations that can change the internal state of a register. Soft errors increase when technology scales down and when leakage decreases. Physical solutions are difficult. Redundancy (spatial or temporal) or SW techniques are possible solutions.

LECTURE 1.4: NETWORKS ON CHIP (NOCS)

Author: *Luca Benini* , U. of Bologna, Italy

Abstract:

Networks on chip (NoCs) are emerging as a "revolutionary" solution for addressing the on-chip communication challenges in current and future silicon platforms. In this tutorial we take an evolutionary path, which shows how the key ideas in NoC design can be inferred by tracking and extrapolating the evolution of traditional on-chip busses. We move from traditional on-chip bus standards (e.g. AMBA AHB) to advanced, split transaction buses (eg. AMBA AXI), to highly parameterized multi-stage and parallel interconnects (e.g AMBA Multilayer, STBUS Type 3). Finally we discuss and compare state-of-the art interconnect solutions available in industry today with various recent NoC research prototypes. A quantitative analysis, based on accurate functional simulation and prototype silicon data (as reported in the literature), will be used as foundation for a critical assessment of what lies ahead in NoC research and development.

Area of interest:

This presentation covered the subject of on-chip interconnects and their evolution for future SoCs.

Addressed problem:

The progresses in technology do not allow us to predict a slow-down. Transistors become smaller and smaller but wires are slower, hence global wires do not scale in the way computation units do. The complexity of design will become unmanageable if design reuse is not taken into account. Performance requirements keep going up but power budgets last the same.

Actually, semiconductor houses producing on-chip buses (ARM AMBA, STM STBus) tend to patch their interconnects in order to improve parallelism, and consequently performance. But temporal solutions like multi-layer or bridged buses do not scale well and their possibility of reuse is poor. Bridged-busses might be useful if bridges performances are not strained too much, in that case resulting in high-latency. Multi-Layer architectures shift the bottleneck to the slaves' side.

Solution:

The speaker shows by experimental results that advanced protocols pay off. He also emphasized the benefits that design reuse might allow. Nowadays, busses are not designed as transaction centric systems; designers have to know too many details about implementations, hindering the development of highly reusable components. Scalability of different bus-based systems has been proved through highly parallel benchmarking. Configuration with little cache size (high on-chip traffic) shows that even on-chip communication allowing multiple outstanding transactions (STBus), starts suffering when the number of cores in the chip increases.

MPSoC community is looking for new kind of on-chip communication, allowing a clear separation of computation and communication, pushing design towards high degree of IP reuse and high performance. NoCs seem to be a given solution for future on-chip communication. NoCs provide a packet based communication and a distributed vs. centralized management. This avoids a global control bottleneck and improves the quality of placement and routing, providing in the same time bandwidth scalability required by nowadays applications for SoC.

The speaker stated that the main power of NoC resides in the fact that scalability is supported from the ground up, not as an afterthought. Moreover, NoCs design methodology allows a clear separation at the session layer:

1. Define end-to-end transactions
2. Define QoS requirements

3. Design of layered protocol stack

NoC design achieves high modularity at the HW level using only two building blocks:

1. NI (Network Interface)
2. Router

The Network interface is going to be a key component of NoC design. It must cover a huge spectrum of complexity, through parametrizable blocks.

The group at University of Bologna has been working on SoC platforms (MPARM) allowing benchmarking of different SoC configurations and interconnects.

SESSION 2: TUTORIALS

LECTURE 2.1: XPRES: FULLY AUTOMATED PROCESSOR GENERATION FROM C

Author: *Chris Rowen* , Tensilica Inc., USA

Abstract:

This talk is the first public disclosure of Tensilica XPRES (Xtensa PRocessor EXtension Synthesis) Compiler product. The tool allows completely automated generation of application-specific instruction-set processors (ASIPs) from unmodified application source code. The compiler enables ANSI C/C++ code in a matter of hours, and to avoid the expense and risk associated with generating custom logic. Unlike C-to-gates behavior synthesis tools, the XPRES Compiler creates general-purpose reprogrammable processor architectures. The resulting architectures are tuned to run the target application suite at maximum throughput and minimum energy, and are implemented by automatically generated RTL implementations, simulators, debuggers, operating system ports and code compilers.

XPRES profiles code to identify performance critical regions, then generates a large set of candidate architectures (millions of alternatives for large applications) using long-instruction word architecture based on Tensilica Xtensa LX processor, SIMD and operation fusion methods. XPRES uses an calibrated hardware cost model to closely estimate processor gate-count vs. application performance tradeoffs, and to generate a family of ASIPs, each providing a different cost-performance combination. Application performance improvement may range from speed-ups of 10-30% for less than a thousand gates, to 20x speed up for 100,000 gates. The resulting processors remain fully reprogrammable, and generated compilers are able to use all generated instruction sets on any similar application by recognition of common computational requirements. The XPRES Compiler also includes interactive application tuning and instruction-set selection modes to help high-level algorithm architects and software developers optimized source code and implementation together.

The impact of the XPRES Compiler is dramatic reduction of effort for creation of processors that rival the efficiency of RTL designs. It enables a new SOC design style using automatically generated task engines as the basic building block, where high silicon efficiency, reduced design time and increased post-silicon programmability make multiple-processor SOC design an attractive investment.

Area of interest:

The main issue for actual SoCs is that general purpose processors are not fast enough, requiring custom RTL in order to accelerate data-intensive or compute-intensive tasks. But hand coded RTL is a very slow and tedious approach, implying high time to market, risk of re-spin during design, low programmability and slow design and verification.

Addressed problem:

Tensilica introduced a fast way to design RTL introducing a processor centric design methodology. The tutorial mainly focused on the way XPRES compiler allows automation of Processor design flow. XPRES compiler allows automatic generation of Processor Extensions for Xtensa LX processors.

Xtensa LX tackles three challenges vexing today's CPU architects: architecture limitations on compute efficiency, bottlenecks on I/O bandwidth and rising power consumption. Xtensa LX aims at providing the advantages of a customizable CPU architecture while laying the groundwork for a tool that automates the task of creating an optimized SoC design (XPRES compiler).

Solution:

Tensilica's primarily solution for boosting compute performance is well known: customers can create their own application-specific extensions to Xtensa's base ISA. XPRES compiler can automatically generate Tensilica instruction extension (TIE) from application software written in C/C++.

Xtensa LX breaks through computation and I/O bottlenecks allowing multiple parallel execution units and avoiding bus accesses for function extensions. Xtensa LX also gives the possibility to add a second load/store unit doubling the available bandwidth. Results state that LX Tensilica processor is 9x faster than other cores, more or less as dedicated hardware with a very fast design time. Xtensa LX and XPRES processor generator can automatically implement clock gating for every functional logic block at RTL level. All logic in the

independently gated clock domains is inactive unless it is actually processing. No manual RTL insertions are required.

XPRES compilers, together with Tensilica's automated processor generation technology, automatically generate RTL hardware and associated tool chain. The hardware block is delivered in the form of a pre-verified Xtensa LX processor core, enabling to proof designs due to its inherent programmability and avoid the cost and risk associated with verifying standard logic.

Additionally, the generated RTL fully rivals the performance and efficiency of hand coded RTL blocks with many concurrent operations, efficient data types and optimized wide deep pipelines.

Summing up, from ANSI C/C++ application the XPRES compiler generates an optimized set of processor extensions that is reusable over a range of similar applications of software code, resulting in a dramatic productivity enhancement.

LECTURE 2.2: VIDEO ARCHITECTURES AND NETWORKS-ON-CHIPS

Author: *Wayne Wolf* , Princeton University, USA

Abstract:

Video is a natural example for networks-on-chips---the high computational and memory requirements of video require complex architectures that take advantage of networks-on-chips. This talk will use the Princeton Smart Camera as an example of network-on-chip design for video SoCs.

Area of interest:

This presentation outlines the Princeton Smart Cameras project. Smart cameras are useful in domains like surveillance, medicine, etc. The smart camera employs:

1. Algorithms adaptation: in order to face the needs of real-time embedded video applications;
2. Distributed systems: by using several communicating cameras;
3. Embedded SW: for the code optimization and middleware;
4. SoC architecture by using heterogeneous multi-processor.

Addressed problem:

The smart camera performs real-time analysis in order to recognize gestures. Programmable processors are used, instead of inflexible HW. VLSI architectures for smart cameras are analyzed. The NoC design is useful in video processing because it offers a more structured wiring, it encourages the interconnect design reuse and in the same time it provides high level communication primitives.

The design challenges to be addressed are: what networks, and protocols to use, how they will be implemented in the physical layer, to provide a correct but still cheap interconnect, what parts of the application should be modified.

Solution:

Solution overview:

1. A human activity recognition algorithm was developed. It is based on contour following, region extractions, graphs matching and gesture classification, based on extracted markers. The algorithm use low-level analysis of body parts and hidden Markov models for the gesture moves.
2. The camera can fuse the results of multiple cameras, in order to avoid occlusion, to synthesize new views like distributed attention, to offer larger aperture, and greater zoom/resolution.
3. Architectural decisions are taken after analyzing several setups on their developed platform.
4. In order to face real-time computing requirements, the cycles per frame metric is used for processor selection: A rate of 20 frames/second is attained for gestures recognition. Also they use the branch per instruction, cache miss rates and data per each computation stage.

Result overview:

1. The test architecture is a good case for the NoC design, because it involves multiple processors IPs, aiming at high performance: 150 frames /second. Different communication architectures are explored: a bus-based architecture and a switch-based architecture. Several topologies are explored: crossbar dimension, memory ports width, and different memory architectures: double/simple port.
2. The simulation environment uses telecommunication simulator, OPNET with a cycle-accurate simulation model.

3. The methodology is a trace-driven simulation: simulation traces are recorded for each node.

LECTURE 2.3: INTRODUCTION TO ELECTRICAL ISSUES IN SOC/SOP

Author: *Hannu Tenhunen* , KTH, Sweden

Abstract:

None.

Area of interest:

The tutorial deals with electrical issues in system-on-chip and in system-on-package.

Addressed problem:

Designing a SoC/SoP isn't just put components together. Design challenges are how to deal with complexity, power, and noise. Platform based design has to answer these challenges. A platform is defined by its on-chip communication infrastructure, the provided on-chip services and the VLSI design methodology to map and implement an application into the platform. Network-on-chip platform offers a communication framework. Traditional ASIC design has a strict partitioning of logical and physical design phases.

Solution:

To deal with electrical issues, we need to take into account the global interconnect in the first step of the design with a wiring plan and a placement plan. A first issue for the interconnection is the wire plan. Repeaters must be added to deal with interconnect delay but this step is dependent on the layout. Network based systems require a large amount of wiring resources, low thermal mismatch, low cost, easy to process and environment friendly. In the future, a chip will have to be synchronized by different clock domains due to interconnection limitations. Another electrical issue is the noise and interference caused by multiple sources in mixed signal systems. Different models are available to model crosstalk effect. Shielding is a solution. Another limitation is due to the parasitic impedance of wires which makes the power distribution on the chip complicated. As this impedance increases with frequency, a solution is the use of decoupling capacitor. As the design of this capacitor is complex, the power distribution should be planned from the beginning of a chip design.

LECTURE 2.4: FORMAL ARCHITECTURE ANALYSIS IN COMMUNICATION CENTRIC MPSOC DESIGN

Author: *Rolf Ernst* , TU Braunschweig, Germany

Abstract:

MPSOC architectures are heterogeneous as a result of hardware and software component specialization and reuse. Design is increasingly an integration problem with emphasis shifting from a component centric to a communication centric design style. New tools for design space exploration and performance validation have been developed that use formal communication and execution models to rapidly analyze systems and guide designers. The tutorial gives an overview of such models and techniques that are based on work in real-time systems analysis. Application to different EDA problems in communication centric design are discussed, such as priority or time slot assignment, buffer sizing, flow control (traffic shaping), and sensitivity analysis up to first steps towards global system optimization.

Area of interest:

When independent components are integrated on the same platform, they meet inter-dependency problems due to communication: such as resource sharing. This has influence on timing and power.

The presentation tackles the communication centric design used for flexible integration of heterogeneous subsystems. Such systems have complex performance objectives and constraints. The formal approaches vs. simulation approaches are studied. The need of a global and heterogeneous validation approach is the motivation of this work.

Addressed problem:

The performance model uses a hierarchical bottom-up structure:

1. Execution timing analysis for separate processes: based on execution path and communication, both being data and architecture dependent.
2. Real-time analysis for single component: processes start to share resources, are influenced by activation patterns and are dependent of the cache memory state.
3. Modeling and analysis for the overall system: including several components that share the communication media, accesses to memory and the environment.

Two scheduling strategies and corresponding performance local analysis, are addressed: the TDMA and the static priority with arbitrary deadlines. The TDMA is applicable to both processing and communication. The predictability and independency of performance domains offer a good advantage, because individual solutions can be coupled. The disadvantage is low resource utilization and the extended induced deadlines. The static priority with arbitrary deadlines: may create complex execution sequences like bursts.

There are two global performance analysis techniques:

1. Holistic approach that takes global system knowledge into account
2. Hierarchical approach where independently scheduled systems are coupled by data flows and events.

Solution:

The problem reduces to coupling different event propagations.

1. Several analysis iterations are done until the convergence is attained. The iterations consist of: (1) the local analysis that produces the correspondent output model and (2) the adaptation between input models – output model that is realized for coupled components.
2. Abstract interfaces are used to couple local analysis. The input interface models are used from RT analysis: periodic, periodic with jitter, periodic with burst and sporadic. The component produces outputs from the same set of models. The input-output model relation is studied. The presence of process scheduling or various signal paths induces a non-determinist behavior that is visible as an output jitter. It has consequences in degrading the performance: overloads, higher memory and power requirements.
3. Traffic shaping is used to re-arrange events models and to eliminate the uncertainty. Typical data-flow models are applied to couple different subsystems and scheduling domains: OR activation, AND activation and multi-rate activation.
4. Event models interfacing is based on network calculus mixed with an intermediate mathematical formalism: arrival event curves are described by their minimum and maximum bounds and the minimum events distance.
5. Event models transformation use global inter- and intra-frames dependencies. In this way, solutions from real-time analysis are applied to iterative global analysis. Transactions are used to couple dependent activations.

Result overview:

1. The SYMTA/S (Symbolic Timing Analysis for Systems) tool is used. SYMTA/S offers an evolutionary optimization platform, with possible user interaction via a GUI.
2. The analysis benefits of fast results computation, and good accuracy, offering 80% constraints meeting.
3. The considered system parameters and constraints are: the core execution and communication time, the given event models at external inputs, and constraints related to path latency and to the output jitter.

LECTURE 2.5: NANOSENSOR ARRAYS: INTEGRATING NANO-SCALE SENSORS AND VLSI PROCESSING CIRCUITRY

Author: *Vijaykrishnan Narayanan* , Computer Science and Engineering Department, Pennsylvania State U., USA

Abstract:

Nanoscale chemical sensors deliver key attributes of low cost, low power consumption, low signature, massive redundancy, and high-sensitivity. The integration of a large array of cross-reactive nanosensors on the same chip is desirable for providing an accurate report of the concentration of analyzes in complex mixtures by

virtue of the varied response of different sensor elements. The nanosensor array integrated together with the underlying VLSI circuit fabric for processing the data on the same chip forms an emerging class of heterogeneous SoC architectures that are suitable for a wide variety of sensing applications. For a physical realization of such a system, the control and characterization of individual sensory elements must be integrated with procedures for positioning them in multi-element arrays, coupling them to conventional or nanoscale electronics, and processing data from these arrays. First, this tutorial will provide an overview of these steps. The second part of the talk will investigate factors and tradeoffs that come into play during design of the system fabric consisting of local sensor interface electronics and preprocessing circuitry that underlies the nanosensors.

Area of interest:

The market of chemical sensors is becoming interesting. For example the Gas sensor market in 2002 constitutes 754million Dollars.

This talk deals with the importance of chemical nanosensors and their use. It presents the manner that we fabricate these sensors, integrating them in the circuit (CMOS and VLSI as example) and gives the steps to use to detect Gas.

Addressed problem:

Why use nanosensors?
Which type of nanosensors to use?
How to integrate chemical nanosensors in the SoC?
What system architecture?

Solution:

Chemical sensors are used for different application (detecting hazardous gas, detecting and controlling automobile emissions, etc.). Many types of the nanosensors exist: physical, biological and chemical). The choice of the type of the nanosensors is based on the different advantages that are given. The solution, used in this talk, is a chemical one which gives better results in terms of low power and sensitivity. Another advantage is that those sensors can be integrated easier on chip. To fabricate those sensors, we use different mechanisms such us electroplating and oxidation.

The integration of nanosensors can be done by using diclecrophores (DEP) for parallel self assembly. The speaker gives two examples of the integration of these sensors in two circuits VLSI and CMOS circuit. The architecture of the system is based on the sensor interface which is a standard electronic circuit, called signal conditioning, and the gas detection mechanism. For the gas detection four steps are used:

1. Training by marking all bad sensors
2. Computing signature feature vector
3. Processing data and getting feature vector
4. Final gas detection using least square estimation

LECTURE 2.6: THE NEXT LEVEL OF PLATFORMS: NETWORKS-ON-SILICON

Author: *Albert van der Werf* , ESAS, Philips Research, The Netherlands

Abstract:

To handle the parallelism in future designs and exploit the growing compute power of silicon efficiently, we need to take a next step in platforms - as defined some years ago at the level of busses, memories, and processors. This next step is Networks-on-Silicon (NoS), where the focus is on communication. The communication technology involved here spans the area of electronic circuits to distributed systems providing the basic technology for a scalable platform. It concerns both intra and inter-chip communication since system implementations may be based on more than one chip. Here the emphasis is on a structured approach for connecting highly autonomous subsystems, consisting of processors, memories, busses (today's embedded systems) using a communication infrastructure that is layered similar to the OSI model. In such a model we distinguish between services provided by a lower layer, interfaces to interact between communication layers and protocols to communicate with peer entities. The subsystems may have their own clock system and regulate their own supply voltage. In his presentation, Albert van der Werf gives an overview of activities in Philips Research in the area of Networks-on-Silicon.

Area of interest:

This presentation focused on the definition of future generation MPSoC platforms that will be based on the Networks-on-silicon concept. They must be well-suited in order to exploit the growing computer power of silicon efficiently.

Addressed problem:

In current SoC implementation, the factors driving the implementation choices consist in choosing a trade-off between number of processors and hardwired functionality (e.g. MPEG2 systems). The growing power offered by technology improvements allows cheaper implementation with increasing performance. In order to obtain high performance with low power, systems must exploit massive parallelism through multi-processors systems and subsystems integration.

Future applications will be more and more heterogeneous, inducing a problem concerning the integration of multiple functionalities in a single die. Mask cost of silicon grows in an uncontrolled manner. Integration of different subsystems with low effort and high efficiency requires wide reuse of Intellectual properties from many IP suppliers, driving the design towards new paradigms.

Application Specific ICs are not affordable anymore, hence design is moving towards Programmable ICs that allow product differentiation through SW.

Solution:

The next level of platforms will be based on the interconnect as basic concept to build predictable systems in a programmable way: traditional interconnect as busses will be replaced by a micro-network of components that exchange information using a layered protocol stack, each layer providing well defined services to others. Network-on-silicon platforms facilitate the mapping of given applications onto networks of subsystems. Application mapping takes place in two phases:

1. System synthesis: minimal hardware required to meet time requirements defined by specification
2. System programming: given a multiprocessor network, find out a mapping of the application that satisfies time requirements

Network-on-silicon platforms facilitate HW & SW linkage through the definition of well-defined interfaces. Nowadays systems have differentiated requirements in terms of temporal constraints (hard & soft real time, best effort systems), hence the future Network-on-silicon platforms will be integrating QoS concepts like guaranteed bandwidth and predictable latency.

Networks-on-silicon implement a communication centric approach to system design. Digital architectures offer a wealth of implementation option, pushing the needs of protocols, interfaces and services standardization.

Networks-on-silicon will allow an automated system design flow with fast performance evaluation.

SESSION 3: PROGRAMMING MODELS FOR MPSOC

LECTURE 3.1: AN HDS TAXONOMY AS ONE OF THE FIRST STEPS TOWARDS CLOSING THE GAP BETWEEN HW AND SW DESIGN

Author: *Frank Pospiech* , S.E.S.A. AG, Germany

Abstract:

A standard interface between HW and SW in terms of an HdS-API is the key for increasing portability and re-use. Since a while, VSIA is working on standardizing this API. The first very important step towards this standardization is a systematic common language between all stakeholders, i.e. chip designers, SW engineers, system integrators, EDA tool providers... VSIA's HdS taxonomy that was released in 09/2003 and that will soon be available for public sets the ground for this common language. Besides a foundation on the need for the HdS-API, this talk presents the basic ideas in the HdS taxonomy, and provides an outlook on further HdS related activities in VSIA and beyond.

Area of interest:

This presentation addresses the software related design issues in modern SoC context. It highlights the challenges that face SW designers in terms of HW platform complexity, lack of portability and design reuse as well as validation and time to market pressure. In short, what the speaker qualified as productivity gap.

Addressed problem:

In order to shorten the productivity gap, the speaker introduces the concept of Hardware dependent Software (HdS) as a layer that shields hardware details for upper layer application software and enables a HW-SW co-development process.

Solution:

The proposed solution goes through the definition of the used terminology and its scope e.g. HdS on slides 17, 18 and HAL on slide 25. The speaker also focuses on the standardization efforts inside VSIA in order to come up with a clear HdS taxonomy that should be at the basis of a standard HdS API.

An important result highlighted in the presentation is the specification of a Hardware Abstraction Layer component (slide 23) in terms of what it hides or abstracts and what it adds compared to the underlying associated hardware component. Furthermore, a HAL framework is proposed (slide 26) based on a hierarchical decomposition of HAL functionalities (access shielding, register shielding and functional shielding).

LECTURE 3.2: SYSTEM-LEVEL DESIGN TOOLS AND RTOS FOR MULTIPROCESSOR SOC

Author: *Hiroyuki Tomiyama*, Department of Information Engineering, Graduate School of Information Science, Nagoya University, Japan

Abstract:

This talk describes RTOS and a suite of system-level design tools which we have been developing for embedded systems and MPSoC. First, we will present the TOPPERS project, which aims at development of production-quality open-source RTOS for embedded systems. Then, we will describe the system-level design tools which feature synthesis of software runnable on the multiprocessor RTOS from system-level specification, synthesis of hardware/software interface, and hardware/software cosimulation with RTOS supports.

Area of interest:

This presentation is logically divided into two largely distinct parts: the first one introduces an open source initiative for developing RTOSs which comply with the ITRON de facto standard in Japan; the second part describes a system level design tool for embedded systems that features automatic software generation, and HW/SW interface refinement.

Addressed problem:

In the first part, the speaker highlights the need for a common platform for embedded real-time software design to promote IP exchange and reuse among different actors in the domain.

The second part addresses mainly the problem of top down system refinement using an automated SoC design flow.

Solution:

The JSP real time kernel developed within the Toppers platform was presented as the reference implementation of the ITRON specification. The kernel features a low overhead and a small footprint and may be easily extended to cope with application requirements.

On the other hand, SystemBuilder, a system level design environment was presented which aims at providing a top down solution to SoC design. Starting from a high level system description, and given a hardware/ software partitioning, the tool automatically generates the HW/SW interface including automatic software synthesis and hardware adaptation.

Concerning the JSP RTOS, besides the interesting features of low overhead and small footprint, the kernel has the advantage of being structured as a target-dependent and target independent parts. This makes it easy to port to new target processors / systems (porting work as short as 3 days -slide 12-).

LECTURE 3.3: MODEL-INTEGRATED DESIGN OF EMBEDDED MPSOC-S

Author: *Janos Sztipanovits* , ISIS-Vanderbilt University, USA

Abstract:

Software composition for embedded MPSoC-s exhibits unique challenges. Embedded MPSoC-s, viewed from their sensor and actuator interfaces, act like physical processes with dynamics, noise, fault, size, power and other physical characteristics. Consequently, we need to go beyond conventional software technology to a model-integrated computing (MIC) technology, which addresses the design of the whole system with its many interdependent physical, functional and logical aspects. In MIC, programming models are model-based: applications are represented using platform independent models (PIM-s) and mapped onto Platform Specific Models (PSM-s) by model transformation tools. This presentation provides an overview of recent advances in Model-Integrated Computing, which offer new opportunities for MPSoC designers.

Area of interest:

This presentation addresses the model integrated design paradigm. Although this concept is not specific to a particular context, its application to the SoC design context is of great benefit according to the speaker.

Addressed problem:

Embedded systems are composed of heterogeneous components which belong to different domains having different semantics (model of computations). The speaker argued that having formal models that capture those different components in a common way is a key issue in any design flow that targets this heterogeneity.

Solution:

The proposed solution features a language-based modeling approach which relies heavily on the Model-driven Architecture as specified by the OMG standard. According to this solution, a given design environment is viewed as a set of transformations that have as input platform-independent models (PIM) and produces platform specific models (PSM). The PIMs, PSMs and transformations are all specified using a common formalism based on the meta-model paradigm. This unified and formal approach (1) allows easy design and validation of the different models in the system, (2) makes it possible to automatically generate the transformation tools from their meta-model description and (3) provides a common environment for tool integration.

The speaker mentioned briefly several case studies that applied the proposed methodology (slides 10, 13, 14). As modeling platform, he referred to the GME project inside ISIS. He concluded that domain-specific modeling languages and model transformations are key technologies for future progress in SoC design.

LECTURE 3.4: MPI AND ITS LIGHT-WEIGHT IMPLEMENTATION FOR NETWORKED/ PARALLEL EMBEDDED SYSTEMS

Author: *Alexander V. Veidenbaum*, Dept of Computer Science, University of California Irvine, USA

Abstract:

MPI is one of the most widely used parallel programming tools for communication between processors in parallel/networked systems. We will briefly review the MPI programming model and its typical implementation. Our work on a light-weight, restricted MPI implementation suitable for parallel embedded systems will then be presented.

Area of interest:

This talk introduces MPI, a standard for communication between processors (slide 7), and gives an overview of its features.

Addressed problem:

Parallel programming is wanted, but still hard to master, especially in the context of MPSoC. The speaker tried to answer "Can MPI be used efficiently in constrained parallel systems?" (Slide 4)

Solution:

The first idea presented is that parallel programming is wanted, but still hard to master, especially in the context of MPSoC. Then the existing parallel programming approaches were presented (Data-parallel, Task-parallel, MIMD, SIMD (slide3)).

MPI was presented as a good mean of communication for systems that are based on distributed memory (not suitable for shared memory) offering high level of abstraction. The presentation addressed the following problem: "Can MPI be used efficiently in constrained parallel systems?"(Slide 4).

A program is assumed as a set of communicating tasks, MPI offering communication and synchronisation primitives which forces the designer to explicitly think about application "vectorisation".

There are two versions of MPI standards, the first was published in 1994, the second is not yet fully implemented and is characterised by a parallel I/O and a dynamic task allocation. A subset of MPI primitives has been shown, i.e. mpi_init() joins a group of processors defined with a communicator. In MPI, there is no assumption on the hardware architecture, and the communicating processes are identified with a number.

An approach to how to start for adopting MPI in an embedded context is also given (slide 20). Results show that Lightweight MPI is a good candidate for embedded systems.

LECTURE 3.5: FIVE WAYS TO DESIGN FUTURE SOC

Author: *Kazuaki Murakami* , Department of Informatics, Kyushu University, Japan

Abstract:

In my talk, I will discuss the two traditional and three contemporary ways to design custom logic embedded in SoC; (1) hardwired logic, (2) processor-based, (3) configurable processor, (4) reconfigurable hardware logic, and (5) reconfigurable processor.

Area of interest:

This presentation is related to the SoC design and describes different methods in order to reduce SoC design time.

Addressed problem:

It deals with five methods to design custom logic (ex: MP3, MPEG4) embedded in SoC. These five methods are:

1. Hardwired logic or IP reuse (1);
2. Processor with software (2);
3. Configurable processor and software (3); (Slide 10)
4. Reconfigurable hardware logic (4); (Slide 11)

5. Reconfigurable processor and software (5). (Slide 10-11)

Solution:

The main concern of this presentation is configurability (2-3) and reconfigurable (4-5) of platform-based design (2-3-4-5), which for the speaker is the future way to design SoC. The benefit of using a platform-based approach is the reduction of the design time.

The second concern is the presentation of solutions that avoid the use of general-purpose processors in order to meet power consumption or performance requirements. In case of platform based design this can be achieved through configurable or reconfigurable HW and SW (3-4-5). Furthermore, for (3-4-5), the granularity of the configurability and reconfigurable is described. For hardware two levels of granularity are defined:

1. The FSM level;
2. The state level. (Slide 12-13)

For the processor two levels are also defined:

1. The ISA level;
2. The instruction level. (Slide 14-15)

Finally the design methods are compared in terms of:

1. Cost / performance;
2. And time to accommodate the platform. (Slide 17 to 20)

The presentation concludes by defining the future work that will quantify the ratio between cost and performance.

LECTURE 3.6: HW-SW INTERFACES ABSTRACTION AND DESIGN FOR MULTI-PROCESSOR SOC

Author: *Ahmed Jerraya* , TIMA, France

Abstract:

Future System on Chips (SoCs) will be highly programmable platforms. These will be organized as Multi-processor SoC (MPSoC) and may include several heterogeneous CPU and Memory subsystems to run complex Embedded Software. The lack of early coordination between Hardware and Software teams causes delay and cost overheads that are no more acceptable for the design of embedded systems. Abstract HW-SW Interfaces are extensively used to coordinate software and hardware communities for the design of classic computers. Abstract HW-SW Interfaces allow concurrent design of complex systems made of sophisticated software and hardware platforms. Examples include API at different abstraction levels, RTOS libraries, drivers, typically summarized as hardware dependent software. This abstraction smoothes the design flow and eases interaction between different teams belonging to different cultures, hardware, software and system architecture.

This presentation deals with HW-SW Interfaces abstraction for heterogeneous MPSoCs based on complex communication protocols and topologies. Different abstraction schemes and Transaction-Level-Modeling (TLM) will be explored to ease different design steps including SW and HW design, integration, debug and validation.

Area of interest:

HW/SW interface abstraction and design for heterogeneous MPSoC based on complex communication protocols and topologies.

Addressed problem:

Abstract HW-SW interfaces allow concurrent design of complex systems and fast integration.

Solution:

SoC is a chip that integrates several components that we used to put on one or several boards. Today, the majority of SoCs (90%) already contain one CPU, which means that the majority of today's SoC already contain software. The speaker remarked that the CPU is the ultimate frontier between HW and SW, and that the development of sophisticated interfaces has a high cost in terms of design time. This illustrates clearly the move from hardware to software. As an example, GSM system was a rack in van in 1986 and will be a SW component on a generic platform in 2006.

We can distinguish two approaches: there will be platform based solutions (cost advantage) and application specific solutions (performance advantage).

In the ideal world, a SoC will be a set of components (HW or SW) connected with a NoC. The problem is that the application is written on top of API(associated with a programming model), and the other SoC API (for NoC programming model or CPU subsystem programming model – slide6) are hidden by this application API and thus complicating the integration of several components. Also, designers focus only on one dimension of these programming models (slide 9) and simplify the others.

The speaker also present the SoC design as the adaptation of different program layers (slide6). The proposed solution is to fix the CPU subsystem, what is HW and what is SW, then to automatically generate the adaptation. The same approach was already used in telecommunication and is based on a component assembly process for protocol adaptation: a subsystem interface will then be a composition to adapt different programming models.

A virtual component model for MPSoC was presented as well as the ROSES environment (an implementation of virtual component model and model adaptation approach).

The automated generation of model adaptation allows us to reduce cost/time design. Also to reduce debug cost, HW simulation models (slide 47) should be designed to validate SW before prototyping platform is ready; however these models are expensive to build.

Results shown:

1. Complexity of MPSoC design is also in software design (HdS very hard to design and debug).
2. It is possible to design MPSoC using high level HW-SW interfaces, some experimental results (related to video compression at several frame resolution) were shown.

SESSION 4: HW DEPENDENT SW AND PROGRAMMING MODELS FOR MPSOC

LECTURE 4.1: THE PROGRAMMER'S VIEW OF A DYNAMICALLY RECONFIGURABLE ARCHITECTURE

Author: *Luciano Lavagno* , Politecnico di Torino, Italy

Abstract:

Reconfigurable computing is emerging as a promising means to tackle the ever-rising cost of design and masks for Application-Specific Integrated Circuits. Run-time reconfigurability even allows one to adapt the hardware to changing needs, evolving standards and multiple applications. In this presentation we will introduce a software-oriented approach to programming a dynamically reconfigurable machine that allows one to simply tag portions of the code that must be implemented as specialized instructions on the reconfigurable portion of the processor. A compiler and simulator suite then is used to evaluate the area cost and performance of the choice. No special knowledge of hardware design is required. We will illustrate the approach by describing the implementation of a UMTS turbo-decoder on one such platform, the XiRisc reconfigurable processor designed at the University of Bologna.

Area of interest:

Reconfigurable computing as a mean to reduce the design cost. The reconfigurability is the middle point in the context of HW and SW design, from power, area, and performance points of view. The reconfigurable array is used as a functional unit, because of its easier integration into ISA and compiler.

Addressed problem:

SW oriented approach for run-time reconfigurability, independent of underlying HW;
Source code statement tagging for implementing them to specialized reconfigurable processor instructions;
Evaluation using a compiler and a simulator;
Design space exploration from area, cost and performance point of view.

Solution:

The methodology is applied to the XiRisc Processor that features: 2-channel VLIW-like elaboration, shared DSP-like functional units (for complex operations like MAC), dynamic instruction sets that are scheduled by the compiler, due to their data-dependency. The reconfigurable array of the functional unit is based on a row-based architecture in order to efficiently map on it, arithmetic operations. This implementation is efficient from HW design point of view and in the same time keeps a high level of abstraction;

The design flow is as follows:

1. The intensive computing kernels are identified by profiling, and annotated with pragmas;
2. The instructions are grouped into user-defined pGA instructions.
3. Cost figures like latency, delay, area are compared for each instruction implementation for different HW/SW partitions.
4. At the last step, the cost figures are refined.

Design space exploration offers the software designer a high-level perspective: pseudo-functions calls and pragmas (language extensions). The aim of optimization for the reconfigurable arrays is to increase performance and reduce the energy.

The performance increasing is based on:

1. Increasing the concurrency: by exploiting the reconfigurable array topology. On the horizontal direction, multiple data can be processed. On the vertical direction, the pipeline is used.
2. Minimizing memory access by internal array storage,
3. Customized data-width and optimized data structures: by taking advantage of shifting and reordering, by a SIMD-like instruction coding;

The energy reduction is based on:

1. Reducing fetches for both instructions and data;
2. Reducing switching activity with respect to a general-purpose ALU and register file;

The limitations of the reconfigurable array architecture:

1.

The read/write ports are still limited, hence the memory unit is still a bottleneck;
2. The reconfigurable resources are limited;
3. The number of custom instructions is limited due to the OP-code space limitation;
4. The reconfiguration time depends on the configuration caches available;
5. Control flow is still limited (fixed and data-dependent loops; if-then-else done by multiplexing);

The Turbo-Decoder application shows:

1. The memory layout optimizations result, by reordering the data in memory in order to transfer several sub-word data in a single memory access. In this case, data reordering in order to match the needs of the next transfer is cheaper than for standard processors, thanks to the FPGA-like internal routing.
2. Several choices for instruction selection and implementation in order to better exploit the parallelism.
3. pGA instructions mapping aiming at speeding-up the computation; speculative execution is used.
4. The pipelined implementation from the latency and delay points of view; the delay is 1 clock cycle because there is no internal feedback storage.

The results show a 10x speed-up and 80% energy reduction versus a standard XiRisc. When compared to Tensilica, Tensilica achieves 50-100x speed-up because it uses pure HW unit implementation. But it uses 10x more gates and it does not provide the reconfiguration flexibility.

LECTURE 4.2: SYSTEM-LEVEL MODELLING FOR MULTI-PROCESSOR SOC

Author: *Jan Madsen* , Informatics and Mathematical Modeling, Technical University of Denmark

Abstract:

One of the challenges of designing a multiprocessor SoC is to partition the application onto the architecture such that essential requirements are met. In order to do so at an early stage in the design process, where not all parts have been implemented or even designed, a system-level model of the multiprocessor SoC design which allows for an accurate modeling of the global performance of the system, including the interrelationships among the diverse processors, software processes and physical interfaces and inter-connections, is needed. Such a model will allow the designer to explore and analyze; the network performance under different traffic and load conditions, consequences of different mappings of tasks to processors (software or hardware), and effects of RTOS selection, including scheduling, synchronization and resource allocation policies. This presentation will present such a system-level model and demonstrate how it can be used to model and analyze a multiprocessor SoC application.

Area of interest:

This talk was about the modelling of wireless sensor networks for a pig farm. This is in relation with the Hogthrob project. Those sensors will be useful for the study of behaviours of pigs and to get a real example of the use of a system on chip in reality.

Addressed problem:

How to model such sensors and how to connect them?

How to get different RTOS Strategies?

How to be effective (low cost < 1euro) and low power consumer?

Not forget that those problems are addressed to farmers (they have to be easy to use and useful for their works).

Solution:

The proposed solutions address the first problem: modelling sensors and the network. The other problems were not detailed in this talk.

The modelling of these sensors is divided into two steps. First, the modelling of sensors, second the modelling of the networks of these sensors.

(1) Modeling sensors

One sensor will be composed of three parts: sensing part, processing part and communications part which contain radios to receive information from the external environment. The processing part may contain several CPU running several tasks, a battery and an ASIC. For instance, they use a unique CPU with primary

functionality and different RTOS Strategies that give basic mechanisms. The modelling of a task is given by an FSM which contain a special part for the communication with the channel. The system is modelled by using SystemC and the master-slave library is used.

(2) Modeling the wireless sensor network

For the network, the speaker exposed the communication protocol as an FSM where it can manage the multiple accesses and the synchronisation between nodes. This protocol is named CSMA (Carrier Sense Media Access). As illustration of this protocol, three examples are presented to show message sending between nodes.

LECTURE 4.3: WILL THE SOFTWARE DINOS STEP ASIDE OR STEP ON MPSOC

Author: *Martijn de Lange*, CEO, ACE Associated Computer Experts bv, The Netherlands

Abstract:

This presentation will be about software Dinosaurs who may/will need to move (a community of 20M people) as it is seen from the embedded (MPSoC) community, but will SoC designers and financial volumes be strong enough to enforce new paradigms as needed for MPSoCs to sustain and be made effective for a longer period. I will give an overview and assessment of both the many technologies and paradigms to address MP and heterogeneity and the psychology of the software world and the software engineering challenges that face the MPSoC designer and that should make him a cautions inventor.

Area of interest:

The context of this talk is the necessity of matching high level programming environment with architecture specific functionality to have an efficient embedded application development using EDA tools (especially compilers).

Addressed problem:

How can it be possible for high level programming environment with architecture specific functionality to have an efficient embedded application development using EDA tools (especially compilers)?

What did compilers have to do?

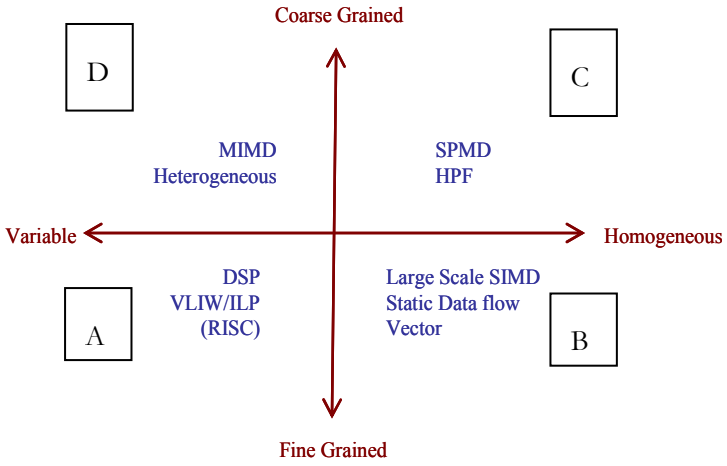
Solution:

The proposed solution is based on some characteristics:

1. Let more design money flow to software
2. Allow for 10 years of research
3. Agree on methods and paradigms (not standards)
4. Stop h/w designers from being ‘clever’

Compilers should efficiently employ specific architectural feature. They have to be able to represent characteristics such as grain size, amount of work per task, homogeneity of task size, computation, data distribution and communication. So those characteristics can be resumed as grain size and homogeneity.

The speaker presented four types of grain size and homogeneity



For each part, he indicates what the compilers have to do and specifies that the MPSoC design will be in MIMD Heterogeneous phase.

	A	B	C	D
Compilers	Can be automatically parallelized by the compiler for pipelined, DSP and VLIW/ILP architectures	- Allocation of resources - Tiling of arbitrary data-size - Streams paradigm	- Generate and optimize Global communication and control - Abstract interpretation	Have to support similar (add MPI) to Type C, but more complex because of heterogeneous architecture

For part D, the compiler can optimize message passing to exploit shared memory (avoiding copies).

But for instance, there are some realities such as C/C++ codes that are expensive to reengineer; the compilers cannot identify sufficient parallelism, the design money is drained into H/W design. So to permit an evolution of design, it will be better to drain money to EDA and SW designers.

LECTURE 4.4: A QUALITATIVE ANALYSIS OF THE BENEFITS OF LUTS, PROCESSORS, EMBEDDED MEMORY AND INTERCONNECT IN MPSOC PLATFORMS

Author: *Kees Vissers* , senior researcher Xilinx, research fellow UC Berkeley, USA

Abstract:

Historically FPGAs started out as 'glue'. However lately they have turned into 'superglue' complete MPSoC platforms. After a short introduction of FPGA basics an overview will be given of various approaches in reconfigurable computing. Today's FPGA platforms already contain multiple processors, millions of bits of embedded memory, millions of gates of reconfigurable logic and the worlds best configurable interconnect network. A good analysis of the benefits, costs and technology scaling of all these components is essential in designing future platforms. A qualitative reasoning of the economic benefits of the various components will be given, illustrated will large commercially successful implementations of video processing applications and networking applications on today's Xilinx FPGA platforms. The cost of programming/reconfiguring these platforms will be summarized.

Area of interest:

The FPGA treated in this presentation reached a rather advanced level. A good qualitative analysis becomes essential for the design of the platforms of future.

Addressed problem:

The purpose of the presentation is to illustrate the importance of different components, especially from economic point of view, when building the FPGA.

Solution:

The economic interest was presented in detail with the encoder JPEG 2000. It can encode the images completely or partially without loss, and provides good capacities of compressions. Consequently JPEG 2000 became important in the field of imagery. Then a generating system for DSP was presented, it has several options, such as the automatic generation of VHDL code. It is also possible to integrate software like Simulink and Matlab for the analysis of the data.

LECTURE 4.5: THE PRINCETON ZEBRANET PROJECT: SENSOR NETWORKS FOR WILD-LIFE TRACKING

Author: *Margaret Martonosi* , Princeton University, Dept. of Electrical Eng., USA

Abstract:

The field of wireless sensor networks offers many interesting applications involving autonomous use of compute, sensing, and wireless communication. In this talk, I discuss the design tradeoffs that arise when applying wireless peer-to-peer networking techniques in a mobile sensor network designed for wildlife tracking. The ZebraNet system includes custom tracking collars (nodes) carried by animals under study across a large, wild area; the collars operate as a peer-to-peer network to deliver logged data back to researchers. The collars include global positioning system (GPS), Flash memory, wireless transceivers, and a small CPU; essentially each node is a small, wireless, computing device. Since there is no cellular service or broadcast communication covering the region where animals are studied, ad hoc, peer-to-peer routing is needed. Overall, our goal is to use the least energy, storage, and other resources necessary to maintain a reliable system with a very high 'data homing' success rate. In January, 2004, we did a test deployment of the ZebraNet system near the Mpala Research Centre in central Kenya. More broadly, we believe that the domain-centric protocols and energy tradeoffs studied for ZebraNet will have general applicability in other wireless and sensor applications.

Area of interest:

To make biologists capable of tracking animals for a long time, over long distances in order to study their interactions within a species and between species, designers have to model new kinds of sensors, using new technologies that facilitate this work. This talk was about a new kind of sensors used for studying Zebras.

Addressed problem:

How to model such sensors and how to be connected?
How to model the software layering?
How to have mobility models in order to collect a lot of information without moving a lot?
How to have low power management?

Solution:

The proposed solutions address the problems by considering two characteristics: animals can move on a large area and the sensors have to be able to be autonomous. To be able to check animals on a long distance the sensor will contain a GPS. And to achieve low power, they used an adaptable approach.

The hardware design of such sensors contain in addition to the GPS, a Flash memory (it is so big to contain a lot of information about 4Mbit), a microcontroller (containing 2KB of RAM and 60 KB of ROM) and a radio to receive and send information.

The principle of communication between nodes is to track data. In fact, it is a collection of information between nodes (all neighbor nodes have the same information).

The processing part uses lightweight parallel threads and queues that handle data synchronization to achieve load balance across threads. They use a middleware called IMPLA to support modularity of protocol and applications. This middleware is working in a layered approach. It consists of protected services between application and hardware devices. This layer has five services (scheduling operations, adapter, updater, event filter and network supporting). The implementation of Impla was on HP/Compaq iPAQ PC and now it is implemented on ZebraNet hardware.

For the results, the implementation of the Impla, their events require less time than application events except for receiving a code packet. The whole prototype implementations and simulations demonstrate the low overhead, an efficient network reprogramming and good code updates.

LECTURE 4.6: APPLICATION SPECIFIC MEMORY NETWORK ORGANIZATIONS FOR NANOMETER TECHNOLOGIES

Author: *Rudy Lauwereins* , IMEC & K.U. Leuven, Belgium

Abstract:

The use of distributed memory organizations is the most power efficient alternative for implementing the local layer(s) of a SoC memory hierarchy. However, the energy spent in the (shared) busses, typically used for the memory communication network, can become so dominant that the complete approach stops scaling up as the number of memories starts increasing. Moreover, this situation will be worse at future technology nodes, because

interconnect scaling does not follow transistor scaling and hence long lines will cause severe delay and energy problems. As an alternative to the standard general purpose shared bus architecture we propose the utilization of application-domain specific segmented bus architecture. In order to minimize the total bus energy, shorter segments in average must be assigned to frequently activated memories, during the floor-planning step. This clearly requires a strong link between the system level exploration and the physical design phase to decide on the relative placement of the blocks within the die. This presentation illustrates how the energy associated to the resulting memory communication network can be made significantly smaller compared to that of a shared bus by a combination of these techniques. This not only allows the overall architecture to scale up, but it also enables true low power operation for the distributed memory organization which is critical in data dominated applications.

Area of interest:

This presentation addresses power efficient implementation of distributed memory organisation by means of bus segmentation and shows the up to factor 10 less power consumption through the experiments with digital audio broadcasting (DAB) and MPEG4 encoder in data dominated application domain.

Addressed problem:

The use of distributed memory organisations is the most power efficient alternative for implementing the local layers of a SoC memory hierarchy. However, the energy spent in the (shared) busses, typically used for the memory communication network, can become so dominant that the complete approach stops scaling up as the number of memories starts increasing. Moreover, this situation will be worse at future technology nodes, because interconnect scaling does not follow transistor scaling and hence long lines will cause severe delay and energy problems.

Solution:

As an alternative to the standard general purpose shared bus architecture this presentation proposes the utilisation of application-domain specific segmented bus architecture. In order to minimise the total bus energy, shorter segments in average must be assigned to frequently activated memories, during the floor-planning step. This clearly requires a strong link between the system level exploration and the physical design phase to decide on the relative placement of the blocks within the die.

The design flow is as follows

1. Optimized memory organization (small memories -> high activity and vice versa)
2. Segmented bus architecture
3. Activity-Aware Placement (AAP) following floor plan template

This presentation shows 2 kinds of experimental results, one for power consumption and the other for L1 bus critical path delay analysis. As the number of the memories increases from 3 to 25, the segmented bus scheme reduces the communication network energy consumption up to factor 10 with hierarchical video encoder example. And the results of power aware floor plan with segmented buses show the factor 3 communication network energy reduction compared to area optimized floor plan with shared buses with DAB example while the results of area optimized floor plan with segmented buses show only the factor 2 communication network energy reduction with only 5% care overhead. And thanks to using segmented bus the critical path delay of the bus is reduced by about 1/3. And the impact of delay optimized floor plan shows minor changes of critical path delay.

So we can see that using the segmented bus as memory communication network induces the tremendous reduction of communication network energy and also improves bus delay significantly.

SESSION 5: MPSOC PLATFORMS

LECTURE 5.1: LOW POWER SYSTEM'S ON A CHIP -- TODAY'S CHALLENGE

Author: *Trevor Mudge* , U. of Michigan, USA

Abstract:

Power consumption has the potential to limit integration in systems on a chip. In this talk we identify the causes of dynamic and static power consumption, and their trends as feature sizes shrink. These trends are unsustainable. Until recently, dynamic power has been the major concern. However, as feature sizes shrink, static power consumption has started to become more important. We examine this trend for both components of static power: sub-threshold leakage and gate oxide leakage. Some simplified power equations are presented and used to illustrate how dynamic and static power can be reduced. It has been shown that parallel processing can reduce dynamic power consumption. We show that pipelined processing is equally good at reduce dynamic power consumption, but that it alone shows a clear advantage if static power is included. Dynamic voltage scaling was introduced as a powerful technique for reducing dynamic power consumption. We show it also reduces static power, and in particular that "just-in-time" scheduling is an optimal policy for reducing static as well dynamic power consumption. Finally, we review some of the methods for saving dynamic power now being employed, and we also review methods for saving static power in the future.

Area of interest:

With increasing clock frequencies and silicon integration, power aware computing became a critical concern in the design of embedded processors and systems-on-chip.

As components size reduces, both dynamic and static power consumption becomes a great concern; lately, the static power started to become more important because of leakages of oxide gate and sub-threshold devices employment.

Pipelined processing and dynamic voltage scaling are important methods in reducing static power consumption.

Addressed problem:

One of the more effective and widely used methods for power-aware computing is dynamic voltage scaling (DVS). However, this approach leads to a very conservative supply voltage since such a worst-case combination of different variability will be very rare.

On-chip caches represent a sizable fraction of the total power consumption of microprocessors. As feature sizes shrink, the dominant component of this power consumption will be leakage. However, during a fixed period of time the activity in a data cache is only centered on a small subset of lines. This behavior can be exploited to cut the leakage power of large data caches by putting the cold cache lines into a state preserving, low-power sleep mode. Circuit techniques for implementing drowsy data caches are investigated.

Solution:

A new approach to DVS, called Razor, is presented. It is based on dynamic detection and correction of circuit timing errors.

The key idea of Razor is to tune the supply voltage by monitoring the error rate during circuit operation. It therefore eliminates the need for voltage margins that are necessary for correct circuit operation in traditional designs. In addition, a key feature of Razor is that operation at sub-critical supply voltages does not constitute a catastrophic failure. It instead represents a trade-off between the power penalties incurred from error correction against additional power savings obtained from operating at a lower supply voltage.

The Razor implementation (slides 24/33) uses a flip-flop that double-samples pipeline stage values, along with a fast clock and again with a time-borrowing delayed clock. A comparator validates latch values sampled with the fast clock. In the event of a timing error, a modified pipeline recovery mechanism restores correct program state.

The proposed Razor pipeline technique was implemented in a prototype 64-bit Alpha processor design (slides 27/31). A prototype implementation in 0.18 um technology was used to obtain a realistic prediction of the power overhead for in-situ error correction and detection.

Study of error-rate trends for data path components using both circuit-level simulation as well as silicon measurements of a full-custom multiplier block (slide 29/30).

LECTURE 5.2: TILES: THE HETEROGENEOUS PROCESSING ABSTRACTION FOR MP-SOC

Author: *Drew Wingard, CTO, Sonics Inc., USA*

Abstract:

Leading edge SoC's for many applications now integrate multiple processors, and this trend is accelerating. To date, no homogeneous computing fabric has shown any real promise in high-volume applications. The application specificity of MPSoC's ensures the opportunity to optimize processing, memory, and I/O resources on a per-subsystem basis to meet the application requirements. However, programming heterogeneous multiprocessors severely stretches the capabilities of embedded system designers, and re-optimizing each subsystem on a per-SoC basis only leads to huge, unpredictable design teams. In this talk, I will describe a generic MPSoC platform architecture based upon loosely-coupled *tiles*. Each tile performs a subsystem function for the SoC, and is largely independent from the other tiles in the same system. Each tile is nominally composed of the processing, memory and I/O resources necessary to perform its function, and the processing may be performed in fixed or programmable logic, or general-purpose or special-purpose programmable processors. The key elements of this abstraction include: socket-based design, decoupled interconnect architectures, communication constraint capture, and firmware packaging. Tiles enable MPSoC's where developers and users are only rarely cognizant of more than one program counter, and subsystem designs that are robust and reusable. I will illustrate this abstraction using examples from several recent MPSoC designs targeting high-volume applications.

Area of interest:

This talk describes a generic MPSoC platform architecture based upon loosely-coupled tiles. Each tile performs a subsystem function for the SoC and is largely independent of the other tiles in the same system.

Addressed problem:

For many applications, the architecture of SoC integrates multiple processors, and this trend is accelerating. No homogeneous computing fabric has shown any real promise in high-volume applications. Heterogeneous components for today’s applications make the design so complicated and so difficult to design (slide 15).

Solution:

The solution for this complicated situation is to hide the complexity of functional and interconnect cores. The functional unit is called Tile, the communication unit is Network and the connectivity is Socket.

The solution of the complexity is to abstract all details. So they define the level of abstraction and explain the anatomy of a Tile (slide 18) and the method of socket based design (slide 19). They compare the traditional partitioning to the socket-based design partitioning that is simple and flexible. Finally, they define the OCP socket (slide22). An example was proposed to illustrate their architecture.

The result of this work is a method to abstract different component to manage MPSoC complexity.

LECTURE 5.3: QUALITY IMPROVEMENT METHODS FOR SYSTEM-LEVEL STIMULI GENERATION

Author: *Roy Emek* , IBM Research Laboratory in Haifa, Israel

Abstract:

Functional verification is known to be a major part of the hardware design effort. System verification is specifically aimed at validating the integration of several previously verified components. As such, it deals with complex designs, and invariably suffers from tight schedules and scarce resources. We present a set of methods, collectively known as testing knowledge, aimed at increasing the quality of automatically generated system-level test-cases. These methods are based on the commonality of some basic architectural concepts. Testing knowledge reduces the time and effort required to achieve high coverage of the verified design. Towards the end of the talk, I

will compare the coverage achieved with and without the usage of testing knowledge, and describe related hardware bugs.

Area of interest:

This presentation deals with SoC verification at system level. System level verification is aimed at validating the integration of several previously verified components. But due to complexity of the systems, limited resources, and tight schedule verification represents still 70% of HW design effort.

Addressed problem:

The main method for system level verification today is simulation. Furthermore, to perform verification the primary need is stimuli (Test Cases). The proposed solution in this presentation is the reuse of checking mechanisms (ex: protocol checker) and especially the reuse of Testing Knowledge.

Solution:

Testing knowledge (Slide 4) enables the exploitation of verification knowledge about bug prone areas like resource contention or collision (Slide 5 to 8). Testing knowledge is a way to improve the quality of the generated test cases (X-Gen: IBM in house test case generator). The concept of testing knowledge is implemented through constraint in X-Gen test case generator. It enables the reduction of the number of generated test cases and better coverage (Slide 11).

LECTURE 5.4: APPLICATION SPECIFIC PROCESSORS IN INDUSTRIAL SOC DESIGNS

Author: *Steffen Buch* , INFINEON Technologies AG, Germany

Abstract:

More and more flexibility is required in today's SoC designs for several reasons: platform concepts, uncertainty in standardization processes, late changes, easy enhancements or bug fixes and so on. However, on the other side chip manufacturers are still under a tremendous cost pressure in consumer oriented high volume markets. Especially in portable devices and dense multi-channel applications power consumption is also crucial. This means that careful trade-offs between required flexibility on one side and area or power on the other side have to be made. This presentation first outlines the possible trade-offs. Then one area, namely the design of application specific instruction-set processors (ASIPs) is more deeply explored. Application specific processors promise to deliver higher flexibility compared to traditional hardware and better performance and lower power consumption compared to standard µCs and DSPs. The design effort to develop such ASIPs including the tool chain has however to be considered. Possible methodologies are described and application examples in different domains are given.

Area of interest:

This presentation first outlines the possible trade-offs between flexibility and area or power in SoC design and explores the design of application specific instruction-set processors (ASIPs) more deeply with application examples in two different domains (digital filter and network processing).

Addressed problem:

More and more flexibility is required in today's SoC designs for several reasons: platform concepts, uncertainty in standardisation processes, late changes, easy enhancements or bug fixes and so on. However, on the other side chip manufacturers are still under a tremendous cost pressure in consumer oriented high volume markets. Especially in portable devices and dense multi-channel applications power consumption is also crucial. This means that careful trade-offs between required flexibility on one side and area or power on the other side have to be made.

Solution:

This presentation first outlines the possible trade-offs between flexibility and area or power. Then one area, namely the design of application specific instruction-set processors (ASIPs) is more deeply explored. Application specific processors promise to deliver higher flexibility compared to traditional hardware and better performance

and lower power consumption compared to standard uCs and DSPs. The design effort to develop such ASIPs including the tool chain has however to be considered. Possible methodologies are described and application examples in different domains are given.

This presentation shows two applications of specific processor generation in the domains of digital filter and protocol processing. In case of digital filter processor (Application Specific Multi-Rate DSP – ASMD), the designer can program in assembler language and also re-configure the processor architecture. In the case of protocol processor (32bit Protocol processor – PP32), they generate full tool chain from C compiler to debugger, assembler linker, simulator. In performance point of view PP32 shows a half execution cycle of the case of MIPS M4K microprocessor. This result shows us that the design of application specific processors can be a solution to satisfy flexibility, area, and power requirements in SoC design.

LECTURE 5.5: CHALLENGES IN PROGRAMMING THE MULTIPROCESSOR PLATFORMS

Author: *John Goodacre*, ARM, UK

Abstract:

Even with the growing acceptance within the hardware community that there is more performance and better processing efficiency when using an MPSoC, for the software development process, these platforms are becoming ever more difficult to effectively utilize. Drawing from the experiences in the development of the ARM multiprocessor core, this talk considers a selection of technologies and methods that can be used by software developers to program the various permutations of homogeneous, heterogeneous, symmetric and asymmetric multiprocessor platforms.

Area of interest:

For programming SoC multiprocessor platforms, open platforms (OMAP) and ARM based multiprocessor platforms were addressed.

Addressed problem:

The increased complexity of programming multiprocessor platforms is a problem.

Solution:

The speaker starts with an “easy to understand” terminology (slide 2). Then he remarked that programmability is essential because there are many “brakes” for fast MPSoC SW development, such as an increased complexity when addressing multiple dynamic applications. At this point, the speaker pointed out that reusability could be the solution, and reusability is enabled by designing portable solution based on a layered abstraction of a solution.

The HW is reaching its physical limits, and the race of MHZ (increase in core frequency) is deserted in the favour of multiprocessor based cores. Then several multiprocessor architectures were discussed (slides 6, 7, 8, 11). We could identify:

1. “Classic” heterogeneous asymmetric (slide 6)
2. Homogeneous asymmetric (slide 7)
3. Asymmetric multiprocessor (slide 8) –general case-
    a. Multiple simultaneous applications
    b. Message based interconnect
    c. Good for statically partitioned applications but requires a good knowledge of the application.
    d. Complex to manage dynamic task allocation (an application on the host processor is dispatching work to “ready to execute” slave processors) limits this model approval.
    e. Isolated tasks effects!
4. Symmetric multiprocessor (slide 11)
    a. Multiple instruction context architecture
    b. Common memory and peripherals
    c. Coherency is highly important!
    d. Thread representation of tasks and an operating system for thread scheduling over processors
    e. Then some mechanisms to manage threads were presented (slide 14, 15, 16)
    f. Difficult isolation of tasks effect!

g. Software designers need to simplify common memory system whereas hardware designers have to ensure memory coherency and processors synchronisation.

Results shown:

1. Power performance ratio between MP and single processors (slide 5)
2. Examples of code for Asymmetric MP and thread based multitasking.
3. Parallelization of media codec tasks, when designer efforts for adding parallelization are only required when a task requires more performance than a single processor can provide.

LECTURE 5.6: CHALLENGES AHEAD IN DESIGNING EMBEDDED ANALOG CIRCUITS IN NANOMETER TECHNOLOGIES

Author: *Georges Gielen* , Katholieke Universiteit Leuven, Belgium

Abstract:

The progress in ULSI technology towards nanometer transistor sizes and the corresponding reduction in supply voltages, poses serious problems for the embedding of analog functions in fully integrated systems. This talk will focus on these difficulties. The limiting factors to the performance of analog circuits will be reviewed, and their evolution with technology will be outlined. The impact of the scaling supply voltages will be described, and the consequence on the power consumed by the analog circuits will be explained. This will be illustrated for some typical analog blocks, such as data converters. In addition, the problems due to the analog-digital co-integration such as coupling of switching noise will be highlighted. The different coupling mechanisms and their evolution with progressing technology will be explained, and an efficient analysis methodology will be described.

Area of interest:

This presentation deals with analog design and integration in SoC.

Addressed problem:

In today's mixed-signal design, analog design is becoming a bottleneck in terms of design effort. Mixed-signal design faces:

1. Technology problems: analog does not become smaller when moving to a smaller technology (65 nm);
2. The decrease of power supply decreases SNR (SNR: Signal Noise Ratio decreases);
3. Embedding problems: signal integrity.

Solution:

The presentation can be divided into two parts. The first part presents the analog design limitations and the second describes the problem of signal integrity in mixed-signal design.

In analog design the trade-off can be stated as: (Speed * Accuracy²) / power = Technology constant. Basic analog limitations are:

1. Noise;
2. Mismatch due to process fabrication variations;
3. Jitter.

Noise and Mismatch have an impact on power consumption and mismatch is the dominating factor, whereas jitter directly impacts SNR. One solution for mismatch is to build transistor as an assembly of transistors, and at layout stage to spread the transistors over the full chip area. (Slide 14)

On the other hand, mixed-signal design introduces crosstalk effects that add noise. Noise is propagating through the substrate. There are three kinds of substrate noise sources: (Slide 28)

1. Inductive noise;
2. Capacitive coupling;
3. Impact ionization.

An analysis of the impact of this noise is given in the presentation (Slides 29 to 34), in which the crosstalk effects of the digital part are modeled as a noise source. The basic effects of crosstalk are:

1. Changing the operating point;
2. Performance degradations;
3. Failures.

In conclusion, mixed-signal design needs solutions for complexity, productivity and crosstalk problems, but analog design has to deal with limitations and issues in power consumption.

SESSION 6: BREAKTHROUGH SESSION

LECTURE 6.1: SINGLE CHIP HETEROGENEOUS MULTIPROCESSOR DESIGN

Author: *JoAnn M Paul*, Electrical and Computer Engineering, Center for Silicon System Implementation, Carnegie Mellon U., USA

Abstract:

The Single Chip Heterogeneous Multiprocessors (SCHMs) of future portable and handheld, "embedded," devices represent a hybrid style of computer design. These systems are not fully custom designs traditionally targeted by the Computer Aided Design (CAD) community nor general purpose designs traditionally targeted by the Computer Architecture (CA) community nor pure embedded designs traditionally targeted by the real-time (RT) community. Because they are design hybrids they represent an opportunity to re-visit design tools, simulators, benchmarks, programs, schedulers, and communication. Only if this opportunity is taken can new concepts for design be created. In this talk, I will identify some of the fundamental differences between traditional computer system design and the design of SCHMs.

Area of interest:

This presentation identifies some of the fundamental differences between traditional computer system design and the design of SCHMs.

Addressed problem:

The Single Chip Heterogeneous Multiprocessors (SCHM) of future portable and handled (embedded) devices represent a hybrid style of computer design. These are not fully custom designs that are traditionally targeted by the computer Aided Design (CAD) community nor general purpose designs that are targeted by the Computer Architecture (CA) community nor pure embedded designs targeted by the real-time (RT) community. In fact, they are hybrid designs that represent an opportunity to re-visit design tools, simulators, benchmarks, programs, schedulers, and communication. New concepts can be created only if this opportunity is exploited.

Solution:

It is a hybrid design where they propose a tool called MESH (Modelling Environment for Software and Hardware) to implement something different from the classical tools. MESH is based on high level modelling where MESH do the partitioning and analyse the power.

The MESH Simulator is a tool for high level (above instruction set simulation) modelling and performance estimation of heterogeneous systems on-Chips (SoCs). The MESH modelling environment grew out of concepts of Frequency Interleaving. The goal of the MESH modelling environment is to facilitate design space exploration by helping the designer to answer common early design questions such as: “Do I have enough processing resources for the application at hand?”, “What kind and how many processors should I put in my system?”, “How to choose interconnect and evaluate performance of the system?”

LECTURE 6.2: ENERGY-EFFICIENT EMBEDDED SYSTEMS BY MEANS OF A CODESIGN OF APPLICATION DOMAINS

Author: *Patrick Schaumont*, UCLA, USA

Abstract:

For reasons of energy-efficiency, modern embedded systems use specialized and distributed processing, implemented using programmable but specialized components. We present a system architecture called RINGS, as well as a tool called GEZEL to support the design of such heterogeneous multiprocessors. Our system design philosophy is to perform codesign of application domains (such as networking or security) rather than codesign of implementations (such as hardware and software). The RINGS system architecture combines specialized processing units with a general-purpose control component in a reconfigurable interconnection network. RINGS designs are explored, refined and validated in GEZEL, a design language and open-source design environment for

domain-specific micro-architectures. We present demonstrator designs based on the RINGS system architecture and other sample applications of GEZEL.

Area of interest:

This talk presents how embedded systems can be designed efficiently by using a design environment called GEZEL. This environment enables fast design space exploration, thus easing the design of efficient systems.

Addressed problem:

For reasons of energy-efficiency, current embedded systems use specialized and distributed processing, implemented using programmable but specialized components. Energy-efficient systems means that not only they should provide low-power consumption, but also that component-utilization should be maximized. However, as embedded systems may change over time, the right degree of programmability and reconfigurability must be provided in order to allow systems to evolve. So the designer must face two contradictory issues: specialized components increase energy-efficiency but decrease flexibility, while programmable components decrease energy-efficiency but increase programmability. Thus, the designer must have a way of finding out how to design energy-efficient and programmable embedded systems. This work is focused on providing means to enable designers to explore many design options to find the right ratio between efficiency/programmability.

Solution:

The proposed solution is to define a generic architecture called RINGS and a design environment for this architecture named GEZEL. RINGS is an application-domain architecture (in this work the focus was in security-domain applications) that combines specialized processing units with general-purpose control components in a reconfigurable network. The RINGS architecture is illustrated in slide 9. To enable designers to explore different RINGS configurations, the GEZEL design environment was implemented. Subsystems are described using the GEZEL language, which represents the subsystem as FSM + data path, then GEZEL generates a simulation model of the system, which can be cycle-accurately simulated. One important remark is that only HW subsystems can be described in GEZEL. For simulating SW subsystems, GEZEL can be cosimulated with instruction-set simulators using memory-mapped cosimulation interfaces. The reason why the GEZEL language was developed is that first HW architects were more comfortable with it than SystemC for example, and it speeds up design iteration time. By design iteration time is meant, the consumed time to build the design and simulate it. The ThumbPod application was designed using this approach. Results that are illustrated in slide 22 show the design iteration time gain of using GEZEL in comparison to using SystemC.

LECTURE 6.3: CACHE COHERENCY AND MEMORY CONSISTENCY IN NOC BASED SHARED MEMORY MULTIPROCESSOR SOC ARCHITECTURES

Author: *Frédéric Pétrot,* Dépt ASIM, LIP6, UPMC (Paris VI), France

Abstract:

The 2003 ITRS Design report states that cutting down design costs can only be achieved by very large block reuse, and, although the same report notes that up to 80% of the design cost can be due to the embedded software, CPUs are major candidates for reuse in MPSoC. Embedded applications can usually be implemented as coarse grain parallel program, whose natural targets can be shared memory multiprocessor architectures. The presentation will review the cache coherency and memory consistency problems that arise in such architectures, and show that the increasing acceptance of NoC technology makes it difficult to use simple hardware schemes to solve them. Taking the SoC specificities into account, we define acceptable solutions to both problems.

Area of interest:

This presentation deals with MP SoC design. It describes the problem of cache coherency and memory consistency in MP SoC using shared memory for task communications.

Addressed problem:

The example used to illustrate the presentation is based on a model of the application, which is composed of a set of tasks that communicate through FIFOs. It also uses a MP SoC HW platform composed of a communication network and processing elements, on which the application is mapped.

The problems of cache coherency are due to: (Slide 7)

1. The programming model: task sharing memory;
2. The software assumptions: for a given task the order of writing in memory respects the order of processor requests and read operations return the latest value;

So, when reaching synchronization point the problem is how to insure, for a set of tasks, memory/cache coherency and memory consistency. The problem of memory/cache coherency is illustrated through two examples: task migration (Slide 8) and shared data (Slide 9).

Solution:

The cache coherency proposed solution requires: (Slide 11, Slide 11 to 13 give an illustration of the solution)

1. No task migration and one kernel per processor;
2. Cache data are local to a processor;
3. Make un-cached data shared with other processor.

The memory consistency problems are well known in pipelined processors:

1. RAW: Read After Write;
2. WAW: Write After Write;
3. WAR: Write After Read.

In the case of MP SoC the proposed solution is to make the initiators responsible of the memory order accesses and to use acknowledgement on all transfers (Slide 15-16).

LECTURE 6.4: MULTIPLE TECHNOLOGIES WILL BE INTEGRATED IN FUTURE MPSOCS

Author: *Gabriela Nicolescu*, Ecole Polytechnique de Montréal, Canada

Abstract:

Systems on chip are currently drivers for convergence of multiple technologies. In 2001 FRAM and FPGA were already integrated on chip; MEMS and chemical sensors were integrated in 2003; electro-optical is expected for 2004 and electro-biological for 2006. Thanks to the benefits that they will bring (e.g. less power consumption, less area, elimination of inter-chip communications, which can cause transmission delays, data corruption, and interference), the entrance of multi-technologies systems on chip in IC community is expected with optimism. However, their design is still a challenge and more work is required at the technological, architectural and system level. The presentation gives an overview of the challenges and possible solutions for system-level design of SoC including different technologies.

Area of interest:

The context of this talk is the modelling and validation of heterogeneous systems having different components that can be designed at different abstraction levels. To handle information between components, we use different kind of interconnections. In order to be safe and to use a big bandwidth, the optical network seems to be an efficient alternative. But the problem also exists when using different models and simulators; and how to integrate all components together.

Addressed problem:

How useful is the use of an optical network?
How to model an optical network?
How to model a heterogeneous system?
How to validate it in different steps of the design cycle?

Solution:

The requirements for MPSoC platform design are flexibility, efficiency and short time to market. For this different alternatives are used. We need clean programming models and to have simple, predictable and scalable network on chip. To have this kind of interconnect, the optical one seems to be an efficient alternative. It can offer:

1. High bandwidth and density
2. Reduction of power dissipation
3. Relief of a broad range of design problems experienced in current electronic systems (crosstalk, voltage isolation, wave reflection)
4. Routing congestion problems alleviated

However, to model systems that contain different kinds of components (especially optical one), the definition of new CAD tools is mandatory. It consists on defining new specification and execution models. The major challenge then will be how to accommodate and validate the different application domains (optical, electrical, and mechanical). The solutions proposed in this talk were:

1. For the specification of heterogeneous systems, they use the concept of virtual component model. It consists in gathering the components in a wrapper. It is a set of virtual ports (internal ports and external ones). With this model we can hide heterogeneity and can have a global specification of a system.
2. For the validation, they use the cosimulation method to have a global execution model for their virtual specification. This cosimulation is done by generating adaptors between heterogeneous specification based on the use of a special library.

To show their approach, the speaker shows an example of an optical switcher.

SESSION 7: APPLICATIONS

LECTURE 7.1: SOC-NETWORK FOR INTERLEAVING IN COMMUNICATIONS APPLICATIONS

Author: *Norbert Wehn* , University of Kaiserslautern, Germany

Abstract:

Interleaving is an important step on the physical layer in communication systems. It is used to minimize burst errors. Interleaving scrambles the data processing order to yield timing diversity. Sophisticated channel coding schemes such as Turbo- or LDPC Codes heavily rely on it. The interleaving process represents the bottleneck in high throughput architectures. In this talk we will present a NoC approach to break open this bottleneck. Random and regular topologies with optimized deadlock-free packet routing schemes are investigated. The different architectures are compared w.r.t. performance and VLSI implementation complexity.

Area of interest:

This paper is related to the wireless communication based on MP-SoC. For its purpose, different "wireless base band algorithms" were developed, each of them requiring different architectural solutions.

Addressed problem:

The "wireless base band algorithms" are for inner modems, outer modems and channel coding (Slide 5). For the last category, different channel coding techniques exist: Turbo-Codes (based on parity check matrix) and LDPC Codes (based on interleaving), which are the most efficient (Slide 6). To implement them into MP-SoC, many challenges need to be taken into account: flexibility and scalability, low power/low energy, parallelism (Slide 4). Another big issue in wireless communications that needs to be taken into account is that "the data distribution is the actual bottleneck and NOT the data computations".

Solution:

Experiments regarding some comparisons between serial implementations and parallel implementation were done for the LDPC Codes (Slides 9&11). The full parallelism is inefficient for this algorithm, in case of large block-sizes. In case of partially parallelism, the interleaving network becomes a bottleneck and conflicts in standard and communication might appear.

In order to cope with these conflicts at design times, dedicated permutation networks (Slide 15) and using suited schedule/assignments are required. To cope with the conflicts at run-time, specific topologies (mesh topologies, Slide 19), routing algorithms and flow controls (Slides 17&18) are required. Slide 20 presents a summary of conflict handling.

LECTURE 7.2: HOMOGENEOUS PARALLEL PROCESSING ARCHITECTURE FOR VIDEO PROCESSING AND SOFTWARE

Author: *Takao Nishitani*, Kochi University of Technology (NEC Media & Information Labs), Japan

Abstract:

A consideration on video signal processor architecture, having parallel processing elements is described for future video SOC applications. The parallel processing merits are first described, and then our first trial on video signal processors with linear parallel processor approach is given. Finally, we will describe the improved version of video signal processor of a single chip SIMP machine which contains 128 PEs. The improved machine reduces capacity of input storage by introducing inter-communications to adjacent two PEs and has suitable functions to a newly introduce one dimensional "C" language by a global control RISC. The resulting chip has maximum processing speed of 50GOP with maximum power dissipation of 4W. This is 5 times faster and 1/10 lower power dissipation. It will be used for mobile safety control systems in near future.

Area of interest:

Complex future applications embedded on cell phones design. These applications (object recognition including faces and letters and words translation) require high processing performance (Tetra Operation per Second TOPS).

Addressed problem:

Cell phone has to offer high processing speed, at a low power and ease of programmability.
The Homogeneous parallel processing architecture could provide such capabilities.

Solution:

First the context of this work was discussed, then the difficulties for face recognition application were presented: the illumination constraint is not taken into account for conventional approaches, thus conventional result is only about 50% of face identification. They propose a new approach based on pre-processing the data base of faces with the same illumination condition of the face to identify, before comparing them. This approach gives 96.1% of faces identification. But this approach also requires very high computation power (all faces in data base, which is supposed to contain a huge number, are processed!) estimated to 1 TOPS for only 100 registered people in database. However, this approach is easily supported by homogeneous multiprocessor (video/image domain specific) architecture (same processing to be done on all data base faces).

Then two approaches for video/image domain specific multiprocessor were identified:

1. Video signal processor (VSP)
    a. Overlap technique (requires double memory size!)
    b. Loosely coupled multiprocessor
    c. C language programming for standard RISC chips
    d. Homogeneous MP with video rate busses
2. Integrated Memory Array Processor (IMAP)
    a. Tightly coupled multiprocessor
    b. Homogeneous PE in a linear lines (SIMD linear array processor)
    c. Expanded C language (One Dimensional)

They used the IMAP and introduced a communication between adjacent PE, added index addressing for an arbitrary processing area and enabled background I/O operation via shift registers (details on slides 14, 15, 16, 17).

At the end of this talk, TOPS DSP were presented (slides 23, 24, 25, 26) as a future component of future cell phones (once power problems solved)

Results shown:

1. Chip specification (slide22).
2. IMAP as the best architecture for programmability, high speed processing and low power capabilities.

LECTURE 7.3: GPU ARCHITECTURES FOR GLOBAL ILLUMINATION AND BEYOND

Author: *Donald S. Fussell* , Department of Computer Sciences, The University of Texas at Austin

Abstract:

Commodity graphics processing units (GPUs) are probably the most common high-performance specialized parallel processing units available today. A modern GPU is a SoC optimized to implement a z-buffer pipeline renderer similar to that of OpenGL. GPU architectures have been evolving toward greater programmability and hence more general utility. Recent work has demonstrated the feasibility of implementing more general and powerful rendering techniques such as ray tracing and radiosity on these more flexible GPUs for very limited classes of scenes. Our work aims at supporting real time global illumination for dynamic scenes as complex as those supported by today's z-buffer systems. This will allow a quantum leap in the image quality produced by commodity GPUs and catalyze their evolution toward more general-purpose programmable parallel machines. We see next generation GPU SoCs as comprising a number of multi-threaded cores optimized to support irregular computations involving spatial data structures in addition to more conventional streaming operations characteristic of today's GPUs. In this talk, we will motivate such architecture in terms of its suitability in supporting improvements in today's z-buffer rendering algorithms and in supporting real time ray tracing.

Area of interest:

This presentation approaches the graphics processing units (GPUs). It motivates architecture in terms of suitability in supporting improvements in today's z-buffer rendering algorithms and in supporting real time ray tracing.

Addressed problem:

The GPUs are probably the most common high-performance specialised parallel processing units available today. The idea is to design a modern GPU as a SoC optimised to implement a z-buffer pipeline rendered. Recent work has demonstrated the feasibility of implementing more general and powerful rendering techniques such as ray tracing and radiosity on these more flexible for very limited classes of scenes.

Solution:

The work aims at supporting real time global illumination for dynamic scenes as complex as those supported by today's z-buffer systems. This will allow a quantum leap in the image quality produced by commodity GPUs and catalyze their evolution toward more general –purpose programmable parallel machine.

The next generation GPU SoCs will comprise a number of multi-threads cores optimised to support irregular computations involving spatial data structures in addition to more conventional streaming operations characteristic of today's GPUs.

LECTURE 7.4: PARALLEL PROGRAMMING MODELS AND PLATFORMS: APPLICATION TO NETWORKING AND MULTIMEDIA

Author: *Pierre Paulin* , STM, Canada

Abstract:

We present the mapping of networking and multimedia applications to the StepNP flexible MP-SoC platform, using the MultiFlex MP programming environment. The MultiFlex environment supports two parallel programming models: a distributed system object component (DSOC) message passing model, and a symmetrical multi-processing (SMP) model using shared memory. We demonstrate the combined use of the MultiFlex MP Compilation tools, supported by high-speed hardware-assisted messaging, context-switching and dynamic task allocation in the StepNP platform. Four applications are presented, making use of both programming models:

1. IPv4 packet forwarding at 10 Gb/s
2. Traffic management at 2,5 Gb/s
3. MPEG4 video codec
4. 3G base station layer1, layer2 processing

Area of interest:

This talk deals with the mapping of network and multimedia applications to the StepNP platform using the Multiflex MP programming environment. Both platform and programming environment were developed at ST Microelectronics-Canada.

Addressed problem:

This work addresses one important MPSoC conception problem that is the mapping of a high-level programming model into an efficient architecture. Two important issues must be considered in order to provide a reasonable solution:

1. The employed programming model should make very few assumptions about the final architecture in order to be generic enough to be employed in almost any architecture;
2. The system architecture should be efficient enough to run multimedia and networks application which demand high performances.

So, the problem lies in how to satisfy these two conditions in a way that this mapping process could be semi-automated and eases design space exploration.

Solution:

The proposed solution is: (1) providing a flexible SoC platform called StepNP in which applications using high-level programming models could be mapped onto ; (2) providing the MultiFlex programming environment which helps the application to platform mapping by offering many facilities such as: SW drivers generation, profiling, HW interfaces generation, etc.

In order to achieve the easy application to platform mapping some issues were considered when defining the StepNP platform:

1. The platform should be simple, regular and predictable
2. It should use industry standards for processing elements, interconnects and I/O.
3. It should simplify the use of legacy architectures.
4. It must provide means for providing high-performance architectures

StepNP, then, tries to satisfy all of these issues by being a multi-processor and multi-thread platform. It consists of several processing elements (PEs) and I/O units interconnected by a NoC. Processing elements can be DSPs, MCUs, FPGAs, etc. Additionally, processors options are: MIPS, ARM, PowerPCs, Tensilica Xtensa, etc. The key benefit of this platform for achieving high performances is provided by the HW multithreading implementation.

The MultiFlex programming environment is in charge of easing the mapping process. For this, applications must be specified using two proposed parallel programming models: DSOC (Distributed System Object Component) and SMP (Symmetric Multiprocessing). Specifications must also have user-defined parallelism, i.e, users must define task partitioning.

While in the talk DSOC is presented as Message Passing model, it can be viewed more like a RMI (Remote Method Invocation) model. It is based on the concepts of distributed objects provided by CORBA, DCOM, etc (see slide 9). Inter-object communication is done through standard interfaces. And these interfaces are specified by an IDL. Data going from one object to another are converted to a platform-independent format in order to make no mapping assumptions. The Message Passing engine and the Object Request Broker are implemented in HW in the StepNP platform, which speed up object communication and scheduling respectively (see slide 11). The other programming model is SMP, which is based on the shared memory communication model (see slide 12). DSOC objects can optionally have internal SMP programming models. This allows for interoperable message passing and SMP models. One important reason to use this model is that it is more suitable to multimedia applications, as we do not want objects sending frequently large chunks of data to each other. The MultiFlex environment then has several tools that synthesize from the IDL: drivers between different OS, drivers between SW PEs and NoC, HW interfaces between HW PEs and NoC. Additionally, it has some analysis tools to guide the designer's mapping decisions.

For performance issues, in the StepNP one part of the OS was moved to HW. It was the scheduling mechanism. Results showed that moving this mechanism to HW improved the performance of the applications designed in the environment. In one of the examples showed in the talk, which was the MPEG4 codec a priority-less scheduling mechanism was implemented.

The results they got showed that they were able to design very efficient architectures using StepNP. Moreover the MultiFlex environment enabled them to analyze in detail the trade-off of moving some parts from SW to HW, and also some mapping decisions (see slides 17-22).

LECTURE 7.5: MP SYSTEM-ON-CHIP: EMBEDDED TEST, DIAGNOSIS AND REPAIR IN PRACTICE

Author: *Yervant Zorian*, Virage Logic, USA

Abstract:

Spurred by technology leading to the availability of millions of gates per chip, system-level integration is evolving as a common paradigm, allowing entire systems to be built on a single chip. This tutorial presents the state-of-the-art in system-level integration and addresses the strategies and current industrial practices in the test and diagnosis of system-on-chip. It discusses the requirements for test reuse in hierarchical design, such as embedded test strategies for individual cores, diagnosis and repair mechanisms, test interface standardization, and embedded test management and integration at the complex System-on-Chip level. Industrial experiences will be shared with the audience.

Area of interest:

This tutorial presents the state-of-the-art in system-level integration and addresses the strategies and current industrial practices in the test and diagnosis of system-on-chip.

Addressed problem:

This presentation discusses the requirements for test reuse in hierarchical design, such as embedded test strategies for individual cores, diagnosis and repair mechanisms, test interface standardization, and embedded test management and integration at the complex System-on-Chip level.

Solution:

In this presentation for the yield optimization Infrastructure IP (I-IP) is introduced. I-IP includes IP for process monitoring, IP for testing, IP for diagnosis and debug, IP for repair, IP for characterization & measurement, and IP for robustness and fault tolerance. Also distributed infrastructure IP design is introduced.

The presentation proposes leveraging I-IPs for higher yield and reliability and for time-to-volume acceleration. And also yield optimized loops, leveraged at different product realization steps during design, fabrication, test and in-field, are introduced.

LECTURE 7.6: SYSTEM ENGINEERING OF MPSOC SOLUTION FOR DIGITAL AM-FM CHIPSET

Author: *Michel Sarlotte*, Thales Communications, France

Abstract:

The new standard DRM for digital radio broadcast in AM band requires integrated devices. At low cost and very low power consumption. A chipset has therefore been designed based upon an ARM9 multi-cores architecture. The talk introduces the application itself and the HW architecture of the SoC. Then the issues for the best tradeoff of the DSP treatments in floating point versus fixed point are presented. The embedded SW - including physical layer, receiver management, the applications layer and the global sequencer upon a real-time OS - is covered, issuing both data flow and memory architectures. The HW/SW codesign process is addressed, focusing on real time and debug aspects. A roadmap is presented. Future improvements of the MPSOC solution are outlined: low power, memory optimization, introduction of AFS mode, reconfigurability for multi-service multiple-standards.

Area of interest:

The radio communication is based mainly on AM and FM based communication.

Addressed problem:

Taking into account, that the FM provides very high sound quality close to the source but very small area coverage, and the AM provides low sound quality but very big area coverage, moving the sound quality of FM to the AM is a challenge. (Slide 3)

Solution:

The solution proposed is Digital AM implemented in MP-SoC. To achieve this, different architectural challenges need to be explored. The choice of the processing elements (DSPs, RISCs or Dedicated Processors) was investigated. (Slides 9&10).

Finally, the selected architecture is based on 2 ARM processors + 2 co-processors. The main reason why this configuration was selected is to cover the gap between the fixed/floating point, power consumption, addressing capabilities and SW development environment (Slide 11). Also, strategies of verification/validation methodologies were presented. (Slides 16…20)

SESSION 8: APPLICATION SPECIFIC INTERCONNECT

LECTURE 8.1: SYNTHESIS OF RELIABLE NOCS

Author: *Giovanni De Micheli* , Stanford U., USA

Abstract:

This presentation will address two issues. The first one is the synthesis of NoCs from high-level structural models. The second is how to improve communication reliability by using encoding and redundancy.

Area of interest:

This presentation addresses two issues. The first one is the introduction of dependable design by using redundancy and power management. The second is to support reliability in micro network (network on chip) by using encoding and decoding redundant blocks.

Addressed problem:

In system on chip design, on the one hand, we should consider performance, energy consumption and cost. On the other hand we should also consider correctness, reliability, safety and robustness. We can see the importance of the dependable design in application domains from the traditional applications like aircraft engine control, or nuclear industry to newer critical-computation applications like health industry, banking, etc.. And as the system becomes bigger the reliability problem becomes more important.

Solution:

In the presentation providing component redundancy and power management are introduced as a solution to improve reliability. For example, in a system we can add the stand by cores such that this core can be replaced when the operating core fails. And also using power management can decrease failure rates by providing for load and stress sharing. For interconnect design for MP SoC multiple source/destination paths implementation, and redundant logic for error detection and correction in each layer of physical links, switches and routers, network interfaces, processor cores are introduced to increase reliability.

In this presentation one experimental result shows that under the same mean time to failure (MTTF) the average energy of the communication link is decreased by using redundant encoder/decoder blocks in the communication link, especially AMBA bus compared to that of the original AMBA bus. This result shows that adding redundant blocks for error detection and correction is a good approach to improve reliability and power efficiency.

LECTURE 8.2: PREDICTABLE SYSTEMS: REALITY, OR JUST AN ILLUSION?

Author: *Kees Goossens* , Philips Research Laboratories, The Netherlands

Abstract:

Consumer electronics products are often embedded, real time, and to some extent safety critical. It makes sense to build systems on a chip that provide these characteristics by having architectures that are somehow predictable. In this talk we discuss what predictability means, if we should strive for it, if it can be achieved at all, and if so, how. We pay particular attention to the interconnection, which is crucial in communication-centric design.

Area of interest:

This presentation deals with MP SoC design for consumer electronic products.

Addressed problem:

The design method described in this presentation is communication centric. Communication becomes important in case of large system composed of many IPs. The problem is to ensure their interconnection and the

system predictability. An illustration of unpredictability is given in slide 7 and slides 9-10 define the sources of the problem.

Solution:

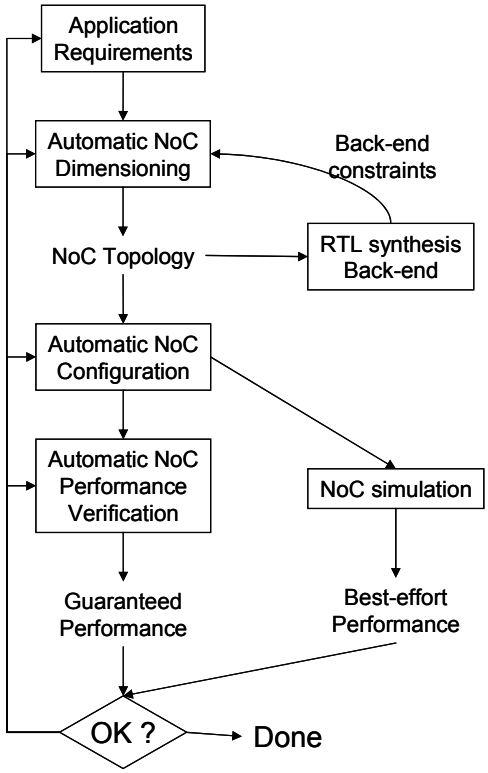
The solution is based on a NoC centric approach using Quality of Service and best effort (slide 11). The concept of Quality of Service is defined as: (slide 12 to 16)

1. Services;
2. Guarantees;
3. Quality of service.

The proposed solution can be divided into four points:

1. For unpredictable resource usage use QoS to characterise resource usage;
2. For unpredictable resources use calibration and predictable design;
3. For resources that are shared by multiple users use resource management and services;
4. For users using multiple resources this is separated through guaranteed services.

Moreover, unpredictability can be simplified through simple choices (Slides 17 to 20), for instance, using local memory data can solve cache problems.



The example used to illustrate the solution is based on this Aethereal NoC (Slides 31 to 23), which characteristics are:

1. To decouple IP implementations through the separation of computation and communication;
2. To focus on guaranteed communication services;
3. To offer best effort for high resource utilisation;
4. To perform verification of communication;
5. To decouple interconnect and IP verifications.

The presentation then describes the proposed flow for NoC generation. This flow is shown in the above figure.

LECTURE 8.3: SPIDERGON A NOC FOR FUTURE SMP ARCHITECTURES

Author: *Marcello Coppola* , STM, France

Abstract:

My presentation will give a overview about a NoC for future SMP architectures that will be developed in AST. Such NoC, called Spidergon, is a packet-switching NoC based on a scalable topology, which includes the OCTAGON NoC as the 8-node member of the family.

Area of interest:

The motivation of this work is the growing system complexity, and the more and more demanding multimedia applications. The network evolution has to follow the same direction, because it is the key point for system integration.

The SW is a new challenging MPSoC design factor, because it becomes more and more complex and it takes most of the design area. This induces the need of adapted programming models.

Addressed problem:

Spidergon is a packet-switching NoC, based on a scalable topology of the Octagon NoC and the STBus experience. A NoC is composed of a set of on-chip routers and network interfaces that transport packets between different sub-systems.

The Spidergon aims at the best trade-off between complexity and performance. A fixed topology like Octagon is suited for this approach, because it leads to a low-cost and still high performance implementation.

Solution:

The Spidergon interconnects a generic even number of nodes, with fixed properties concerning the diameter of the network and the number of existing links.

The comparison with a Mesh implementation is presented. The Mesh network has intermediary nodes increasing the number of links, and reducing the maximum delay.

The routing is realized in two steps: (1) at packet generation – in the Network Interface – and (2) during packet traveling – in the router. It does not have a routing table, and it is based only on a logic function for simple decision making. This simple logic differentiates Spidergon from Octagon and makes it efficient: because it does not have routing tables and consequently no latency induced by routing.

For the moment, it does not support QoS. Previous architectures (like STBus) offer this feature, and that's why, the application demands should be studied in order to evaluate if Spidergon should also support QoS.

The Spidergon communication is implemented at three levels of abstraction: message, packet and FLIT (FLow control digIT).

The router design is based on:

1. virtual and fast wormhole,
2. simple and high abstraction send/receive protocol, with adaptive packet lengths,
3. for the system reliability the routing is not adaptive,
4. for area costs it does not contain routing tables,
5. the best performance is obtained with the output queuing mechanism,
6. the QoS is not yet implemented.

The router scheme is presented in slide 21.

1. A question to be answered by exploring different application features is: is it worthwhile to replace the NoC with a bus.

LECTURE 8.4: IP REUSE AND INTEGRATION IN MPSOC: HIGHLY CONFIGURABLE PROCESSORS

Author: *Grant Martin* , Chief Scientist, Tensilica Inc., USA

Abstract:

It is quite likely that your MPSoC will not contain all your own blocks, especially your own programmable processors. Although sourcing and integrating 3rd. party IP blocks have made some progress in recent years, there is still more to be done to make the integration flow easier and more comprehensive. This tutorial will outline design and integration flows for re-using IP blocks, using a major commercial extendible and configurable processor as the key example. Highly configurable processors, which will form the heart of MPSoC, have some

unique characteristics which pose integration challenges beyond those of static hardware and software IP blocks. The talk will also touch on some of the new trends and requirements for more effective IP integration, including emerging standards and system-level integration concerns.

Area of interest:

This presentation covers the design and integration flows for re-using IP blocks, and some of the new requirements for more effective IP integration, including emerging standards and system-level integration concerns.

Addressed problem:

In his talk, the speaker treats the creation and integration of "Static IP" and "Highly configurable IP". He introduces the SoC design flow using "Static IP"(Slide number 4) and "Highly configurable IP" (Slide number 7). He presents the IP reuse as the main challenge for the SoC design. The first issue is the quality of the IP, and the major question is how it can be evaluated?

Solution:

The Quality of the IP depends on the application architecture where there is a huge variety of coarse-grained configuration options; users need basic feedback on the likely outcome of the generation process. The integration of the IP reuse in the MPSoC requires:

1. An estimator of power, performance and area (cost) such that the estimators must decide the parameter values according to the technology choices.
2. An additional configuration to adapt configurable processors much more closely to applications.
3. Deliverable output from processor configuration like SW development environment (compliers, SW libraries, OS. Support…) and System level modeling and debugging (ISS, HW-SW Co-simulation).
4. Additional required deliverables- implementation and verification flows.

The business of IP integration is based on companies and organizations that produce standards for defining IPs, Among these standards are the VSIA, VCX, FSA, OCP-IP, SPIRIT.

In his analysis to the situation of the IP reuse, Grant Martin concluded the following:

1. The quality has to be defined by integrators.
2. Reputation is possibly the best guide today.
3. Static IP needs adaptation to cope with highly configurable IP.
4. The standards have an ambitious role in the integration of the IP reuse domain.
5. The standards organizations seem to be bogging. The politics and economics of EDA seem to trump the interests of the IP creators and users. These problems make it unclear if the situation will change.

This presentation invokes the integration of the IP reuse. It is one of the big issues in the domain of MPSoC. In the beginning it introduces the different IP types and the design flow. After this, it outlines the problem of the IP qualification, and adaptation and proposes some solutions like estimators and an additional configuration. Before concluding it evokes the state of the standards and the organizations.

LECTURE 8.5: MULTI-LEVEL COMPUTING ARCHITECTURE FOR EMBEDDED SYSTEMS

Author: *Faraydon Karim*, STM, La Jolla, USA

Abstract:

Multi-level computing architecture is architecture for Heterogeneous MPSoC. The architecture is a two-level hierarchy, which consists at the bottom level of several processing elements (PE s), controlled at the top level by a control processor. The main characteristic of this architecture is that it exploits in hardware parallelism among tasks executing on different PE in the same way a superscalar processor exploits instruction level parallelism. This hardware support for task level parallelism gives rise to a natural programming model that relieves programmers from explicitly synchronizing tasks and communicating data.

Area of interest:

The goals of this work are to provide a highly parallel architectural framework called Multi-Level Computing Architecture (MLCA), define a programming model suitable for this framework and finally define tools and methodologies to map the proposed programming model to the MLCA framework.

Addressed problem:

The main problem discussed in this work is how to map an application onto an efficient SoC architecture. As we know, SoC architectures present heterogeneous components, and applications that run in these architectures should be paralleled as much as possible in order to be efficient. However, writing parallel programs is hard. So, there is a need to fill the implementation gap between the application and the final architecture by means of a programming model that abstracts the architecture and facilitates mapping of the application model. Another important issue to be considered is that as SoCs should be efficient, there must be tools that enable fast design space exploration.

Solution:

The proposed solution was to provide an architectural framework called MLCA, define a programming model suitable for this framework and implement a programming environment to ease design space exploration.

The key idea of MLCA is making an analogy between superscalar and multi-processor SoC architectures. In other words, employing techniques used in superscalar computing to multi-processor SoCs. This can be seen in slide 12. So MLCA consists basically of a control processor that commands several processing elements (PEs), which exchange data through Unit Register Files (URF). The main benefit of this approach is giving the system an image of a single processor. This simplifies not only design, but also programming. It makes multiprocessing easier.

This work proposes a layered programming model, which is basically a top-level Control-Data-Flow-Graph (CDFG) with scheduling mechanisms (see slide 13,14 and 24). Another important aspect of this programming model is that it does not assume synchronous communications (it actually adopts GALS model) and it is independent of memory architecture. A new language called Hyperprocessor ASM was developed in order to program applications using this programming model.

Although it is not implemented yet, the idea is to have in the future a tool called HyperCompiler, which has as input the application model and produces as outputs the control task (executed in the control processor) and the tasks executed in the PEs.

Finally, some tools were developed to profile applications and ease design space exploration.

A MP3 application was designed employing MLCA and was coded using HyperASM. Thanks to the profile tools, many architecture decisions were explored and an efficient implementation of the decoder was possible.

LECTURE 8.6: THE EMBEDDED PROCESSORS IN FPGA'S

Author: *Yankin Tanurhan* , Sr. Director, Applications and IP Solutions, Actel Corporation, USA

Abstract:

This talk will address challenges and solutions of implementing multiple processors on FPGA including I/O limitations, interface synthesis and periphery adaptation, task balancing, on-chip communication and clock limitations.

Area of interest:

The context of this talk is implementing multiprocessors on FPGA, challenges and solutions.

Addressed problem:

FPGAs are winning because they reduce development cycle (reconfigurable) and cost (increased density). Also, designers need custom parts on demand which is the real added value of FPGA.

Solution:

The speaker started by presenting the advantage of using FPGA in embedded system: reduction of cost and design cycle by increasing density and using a reconfigurable approach. He remarked that reconfigurable

computing is an efficient solution for long running or streaming computations. There are several RC (reconfigurable computing) fabric options:

1. Coarse-Grained
2. FPGA based
3. FPGA + CPU

Platform based FPGA is presented as the solution for MPSoC design constraints, but still finding the right MPU (microprocessor unit) architecture. MPU were preferred to MCU (microcontroller unit) because they offer higher programmability (soft IP).

Actel's products such as the platform8051 with its design tools are introduced and the flexibility of the platform (several possible architectures) is presented.