PRO3D, Programming for Future 3D Manycore Architectures: Project Interim Status

Christian Fabre, Iuliana Bacivarov, Ananda Basu, Martino Ruggiero, David Atienza, Éric Flamand, Jean-Pierre Krimm, Julien Mottin, Lars Schor, Pratyush Kumar, et al.

PRO3D, Programming for Future 3D Manycore Architectures: Project Interim Status

Christian Fabre[1,7], Iuliana Bacivarov[3], Ananda Basu[2,7], Martino Ruggiero[4], David Atienza[6], Éric Flamand[5], Jean-Pierre Krimm[1,7], Julien Mottin[1,7], Lars Schor[3], Pratyush Kumar[3], Hoeseok Yang[3], Devesh B. Chokshi[3], Lothar Thiele[3], Saddek Bensalem[2,7], Marius Bozga[2,7], Luca Benini[4], Mohamed M. Sabry[6], Yusuf Leblebici[6], Giovanni De Micheli[6], and Diego Melpignano[5]

[1] CEA, LETI, Campus Minatec, Grenoble, France.
{christian.fabre1, jean-pierre.krimm, julien.mottin}@cea.fr
[2] VERIMAG, Centre Équation, 2 av. de vignate, 38610 Gières, France.
{ananda.basu, saddek.bensalem, marius.bozga}@imag.fr
[3] ETHZ, Computer Engineering and Networks Laboratory, 8092 Zürich, Switzerland.
{bacivarov, lschor, kumarpr, hyang, dchokshi, thiele}@tik.ee.ethz.ch
[4] Università di Bologna, Bologna, Italy.
{martino.ruggiero, luca.benini}@unibo.it
[5] STMicroelectronics, Grenoble, France.
{eric.flamand, diego.melpignano}@st.com
[6] EPFL, Lausanne, Switzerland.
{david.atienza, mohamed.sabry, yusuf.leblebici, giovanni.demicheli}@epfl.ch
[7] CRI – Centre de recherche intégrative, 7 al. de palestine, 38610 Gières, France.
http://www.cri-grenoble.fr

Abstract. PRO3D tackles two important 3D technologies, that are Through Silicon Via (TSV) and liquid cooling, and investigates their consequences on stacked architectures and entire software development. In particular, memory hierarchies are being revisited and the thermal impact of software on the 3D stack is explored. As a key result, a software design flow based on the rigorous assembly of software components and monitoring of the thermal integrity of the 3D stack has been developed. After 30 months of research, PRO3D proposes a complete tool-chain for 3D manycore, that integrates state-of-the-art tools ranging from system-level formal specification and 3D exploration, to actual programming and runtime control on the 3D system. Current efforts are directed towards extensive experiments on an industrial embedded manycore platform.

1 Introduction

Three dimensional stacked integrated circuits (3D ICs) are extremely attractive for overcoming the barriers in interconnect scaling, offering an opportunity to continue CMOS performance trends for the next decade. With the ever increasing demand for higher data rates and performance as well as multi-functional capabilities in circuits, vertical integration of IC dies using through-silicon vias is envisioned to be one of the most viable solutions for the development of new

generation of electronic products. 3D integration of multi-core processors offers massive bandwidth improvements while reducing the effective chip footprint. However 3D integration introduces several challenges, mostly related to the following factors:

- increasing amount of logic that can be placed onto a single 3D IC,
- related thermal dissipation problem,
- a necessary shift in programming models towards more parallelism.

The manycore revolution and the ever-increasing complexity of 3D ICs is dramatically changing system design, analysis and programming of computing platforms. Future 3D architectures will feature hundreds of cores and on-chip memories connected through complex 3D communication architectures. Moreover, the third dimension leads to a tremendous increase in heat dissipation per unit area of the chip. This in turn results in higher chip temperatures and thermal stress, hence, (a) limiting the performance and reliability of the chip and (b) requiring software development tools and runtime to address thermal concerns.

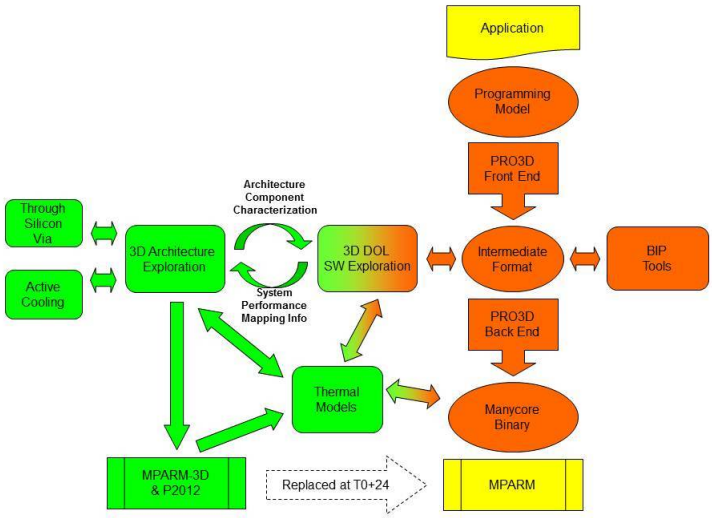


Fig. 1. PRO3D Exploration & Design Flow for 3D Manycore

PRO3D addresses the above mentioned challenges and proposes Fig. 1 a software and exploration design flow based on the rigorous assembly of software components and monitoring of the thermal integrity of the 3D stack: Section 2 investigates memory hierarchies and thermal-aware architectural exploration (corresponding to *TSV, Active Cooling, MPARM, MPARM-3D & Architecture Exploration* in Fig. 1 above); Section 3 details the active cooling strategy for the

proposed 3D stacks (correspponding to *TSV, Active Cooling & Thermal Models*); Section 4 investigates system-level formal solutions that guarantee thermal properties during mapping of application tasks on the 3D architecture (corresponding to *3D DOL SW Exploration & Thermal Models*); Section 5 describes formal verification methods for PRO3D systems (From *Programming Model* to *Manycore Binary*, plus *3D DOL & BIP Tools* in Fig. 1); Section 6 provides an overview of STHORM, the PRO3D target platform (corresponding to *STHORM*). Finally, our current achievements are summarized in Section 7.

2 3D Architectural Exploration

With three dimensional stacked integrated circuits (3D ICs), accurate 3D thermal-aware system-level architectural exploration plays a fundamental role in system design. System-level architectural explorations and thermal issues have so far been addressed independently at different levels of the system design. Hence, new methodologies that address the heat removal problem concurrently at all stages and levels of the 3D chip design need to be developed and to be exploited by high-level software programming frameworks. Designers of upcoming 3D chip will need new distinctive tools for thermal-aware 3D architectural exploration, enabling a cooling-aware design of 3D ICs.

PRO3D has developed a flexible virtual platform infrastructure (VPI) for modelling and analysis of 3D integrated architectures and memory systems, as well as accurate thermal models for calculating the costs of operating the cooling, determining the overall energy budget and performing run-time thermal management.

MPARM [28] has been used as main VPI tool for design space explorations. It is a virtual SoC platform based on the SystemC simulation kernel, which could be used to model both HW & SW of complex systems. The system architecture simulated by the default MPARM distribution is represented by a homogeneous multicore system based on shared bus communication. During PRO3D, MPARM has been enhanced with several HW parametric models of the main micro-architectural components of a 3D integrated interconnect and memory hierarchy [6, 31], and a support of modular plug-ins for thermal models interfacing. The new models are highly parametric, flexible and customizable.

2.1 Functional Modelling of 3D Memory Hierarchy

To keep the pace of Moore's law, future 3D-IC platforms will be embracing the many-core paradigm, where a large number of simple cores are integrated onto the same die. Current examples of many-cores include GP-GPUs such as NVIDIA *Fermi* [23], the *HyperCore Architecture Line (HAL)* [24] processors from Plurality, or ST Microelectronics *Platform 2012* [5, 20, 36].

While there is renewed interest in Single Instruction Multiple Data (SIMD) computing, thanks to the success of GP-GPU architectures, strict instruction scheduling policies enforced in current GP-GPUs are being relaxed in the most

recent many-core designs to exploit data parallelism in a flexible way. Single Program Multiple Data (SPMD) parallelism can thus be efficiently implemented in these designs, where processors are not bound to execute the same instruction stream in parallel to achieve peak performance.

All of the cited architectures share a few common traits: their fundamental computing tile is a tightly coupled cluster with a shared multibanked L1 memory for fast data access and a fairly large number of simple cores, with ≈ 1 Instruction Per Cycle (IPC) per core. Key to providing I-fetch bandwidth for cluster-based CMP is an effective instruction cache architecture design, therefore a detailed design space exploration and analysis have been conducted to evaluate how microarchitectural differences in L1 instruction cache architectures may affect the overall system behavior and IPC.

We analyzed and compared the two most promising architectures for instruction caching targeting tightly coupled CMP clusters, namely private instruction caches per core and shared instruction cache per cluster.

Experimental results showed that private cache performance can be significantly affected by the higher miss cost; on the other hand the shared cache has better performance, with speedup up to almost 60%. However, it is very sensitive to execution misalignment, which can lead to cache access conflicts and high hit cost [6].

2.2 Enabling Thermal-Aware System-Level Architectural Exploration

PRO3D has also produced 3D-ICE, a compact transient thermal model (CTTM) for liquid cooling that provides fast and accurate thermal simulations of 3D ICs with inter-tier microchannel cooling [42]. 3D-ICE can accurately predict the temporal evolution of chip temperatures when system parameters (heat dissipation, coolant flow rate, etc.) change during dynamic thermal management. We have validated the accuracy of the model with a commercial computational fluid dynamics simulation tool as well as measurement results from a 3D test IC and have foudn a maximum error of 3.4 % in temperature.

PRO3D has also defined and characterized (electrically and thermally) a 3D integration process flow [21, 35, 45] that combines TSV and microchannels fabrication for liquid cooling of multiple tiers and has developed 3D-ICE [34], a complete transient thermal simulation tool that can be used to validate 3D integration stacks of multi-core designs in a very early stage of the design flow, thus enabling much more thermally-balanced and controlled 3D multi-core designs. These high-level technology models of complete 3D stacks have been successfully used to validate the effects of the cooling methods while executing benchmarks in the VPI [18].

3 Thermal Management

Inter-tier liquid cooling is a recently proposed and a promising thermal packaging solution to counter the aggravated thermal issues arising from vertical stacking

in 3D-multiprocessor ICs [8]. With this packaging solution, inter-tier thermal resistances are reduced considerably, enabling the 3D ICs to operate at much lower temperatures than those with conventional heat sinks [30, 26].

However, inter-tier liquid cooling also brings with it new design-time and run-time challenges for the designers. For instance, a serious challenge that single-phase liquid cooling brings is the increased thermal gradient. The sensible heat absorption that occurs as the coolant flows along the microchannels raises its temperature [34]. This results in an increase of coolant temperature from inlet to the outlet, which in turn, results in an undesirably augmented thermal gradient on the IC surface [30]. These gradients cause uneven thermally-induced stresses on different parts of the IC, significantly undermining overall system reliability [10].

In this respect, we propose a novel design-time thermal balancing technique by modulating the microchannel width from inlet to outlet, without adding to the existing fabrication costs. This technique, referred to as *channel modulation*, relies on the well-known observation that the thermal resistance of microchannel heat sinks reduces with increasing aspect ratio of the channel cross-section [41]. Our proposed work provides an optimal solution for thermal balancing and hotspot minimization. This work contributes to providing an additional dimension of design-space exploration, in the form of channel modulation, to IC designers for the purpose of thermal balancing.

3.1 Thermal Model and Problem Formulation

The goal of our optimization is to find a sequence of channel widths, as a function of the distance from the inlet, which minimizes the intended cost function: the temperature gradient. Hence, the **steady-state temperatures** of the 3D IC must be written as a function of this distance in the analytical formulation, with the channel widths as an input parameter. In other words, if the distance from the inlet is measured along the coordinate axis z, then we need to find an equation of the form:

$$\frac{d}{dz}\mathbf{T}(z) = \Phi(z, \mathbf{w_C}(z), \mathbf{T}(z)), \tag{1}$$

where $\mathbf{T}(z)$ is the steady-state temperatures vector on the IC and $\mathbf{w_C}(z)$ is a vector of width functions of different microchannels written as a function of z. Our goal, then, is to find $\mathbf{w_C}(z)$ that minimizes the gradients in $\mathbf{T}(z)$. It is important to mention that there are five heat transfers occurring along the channel that must be taken into account in the thermal model [34, 35]:

1. Longitudinal heat conduction inside the two active silicon layers, parallel to the microchannel.
2. Vertical heat conduction from the active silicon layers to surface of the top and bottom microchannel walls.
3. Vertical heat conduction between the active silicon layers through the microchannel silicon side walls.

4. Convective heat transfer from the surface of the microchannel walls into the bulk of the coolant.
5. Convective heat transport downstream along the channel due to the mass transfer (flow) of the coolant.

In our optimization, we define our cost function as the square of the Euclidean norm of the thermal gradient ($\mathbf{T}'$). Our optimal control design problem can be formulated as:

$$\min_{\mathbf{w_C}(z)} J = \int_0^d \|\mathbf{T}'\|^2 dz \tag{2}$$

$$\text{Subject to :} \quad \begin{array}{l} 1.\ \text{System state-variable equations} \\ 2.\ \text{Design constraints} \end{array}$$

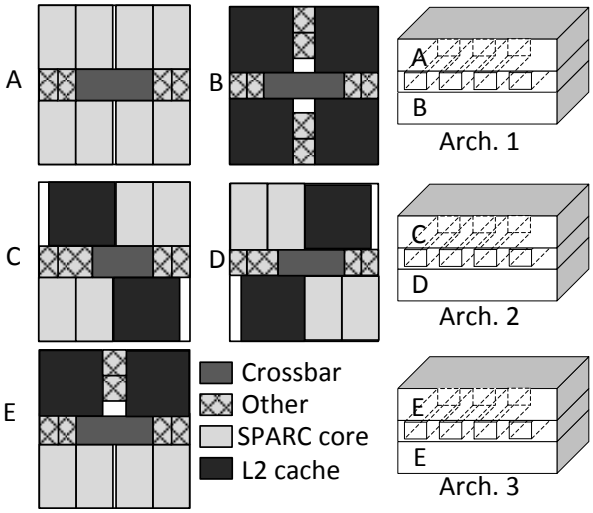


Fig. 2. Layout of the 3D-MPSoCs Used in our Experiments

3.2 Experimental Results

We apply the optimal channel modulation design to different liquid-cooled 3D-MPSoC architectures to demonstrate how the optimal channel modulation technique can be used with the conventional floorplan exploration to obtain the desired thermal behavior during the IC design. We use different configurations of the 90 *nm* UltraSPARC T1 (Niagara-1) processor [16] architecture. Fig. 2 shows the layout of the 3D-MPSoCs used in this experiment [30, 16]. The dies are of size 1 cm $\times$ 1.1 cm and the heat flux densities range from 8 W/cm^2 to 64 W/cm^2.

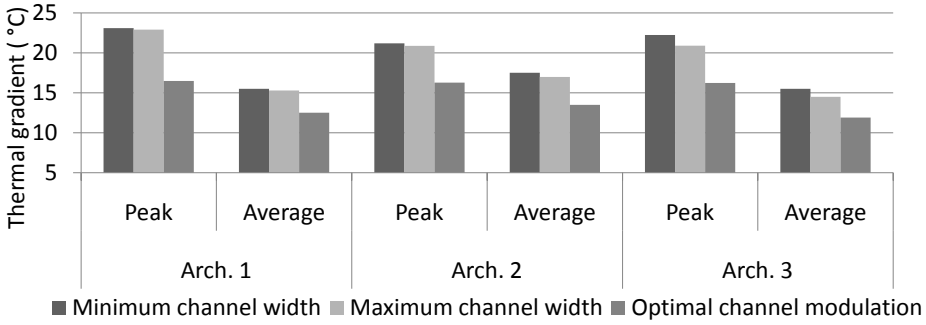
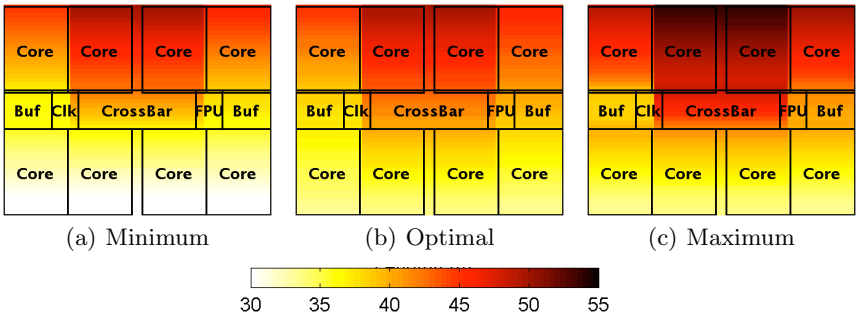


Fig. 3. Thermal Gradients Observed in the Different 3D-MPSoC Architectures Dissipating Peak and Average Level Heat Fluxes, Using Maximum, Minimum and Optimally Modulated Channel Widths

In our optimization technique, we are using the worst-case (peak) power dissipation of the 3D-MPSoC functional elements [30, 16]. Our proposed method achieves a thermal gradient reduction of 31% (23°C to 16°C). When the peak heat flux levels were replaced by average values, this same optimal channel modulation configuration manages to reduce the thermal gradient by 21 % compared to the uniform channel width case. In addition, we observe that the peak temperature in the optimally modulated channel case equals to the peak temperature of the minimum channel width case. Thus, our proposal implicitly minimizes the peak temperature to the lowest value achievable within a given channel width range. The thermal gradients obtained for the different cases and for various channel types are plotted in Fig. 3. Sample thermal maps of the *Arch. 1* top-die, for the case of peak heat flux are also plotted in Fig. 4 to illustrate the ameliorating effect our proposed method has on the thermal gradients. The direction of coolant flow is from bottom to top of the figures.



(a) Minimum (b) Optimal (c) Maximum

Fig. 4. Thermal Maps of Arch. 1 (Fig. 2) Top Die with Peak Heat Flux Levels, when Minimum, Maximum and Optimally Modulated Channel Widths are Applied. All the Thermal Maps are Drawn With Identical Temperature Scale ($[30 - 55]^{\circ}$C)

4 Thermal-Aware Application Mapping on 3D Platforms

Distributing tasks optimally on a parallel platform is known to be NP-hard [9, 40], but approximate methods exist. 3D platforms add new aspects to the problem and require rethinking the methods for system-level analysis, optimization, and exploration of the design space. Although considering thermodynamics of 3D stacks at system-level is crucial, none of the existing system-level mapping frameworks is thermally aware. Considering thermal management at system-level is important not only because of high cooling costs or the potential reliability problems if the circuit is not correctly designed, but also because latencies and other performance metrics might depend on temperature. In particular, if temperature variations are ignored, unpredictable runtime overheads or unexpected performance degradations might occur, e.g., due to reactive thermal mechanisms such as dynamic voltage and frequency scaling.

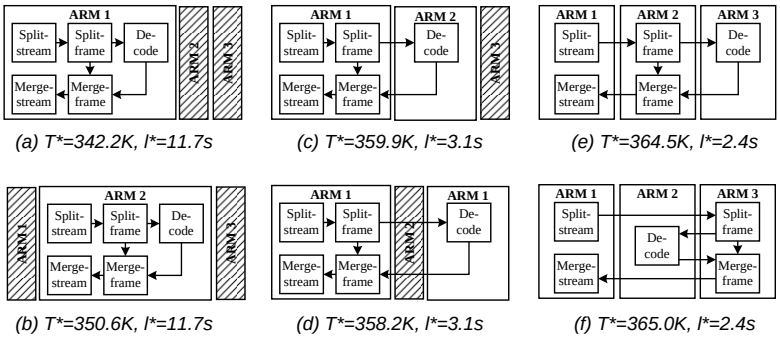


Fig. 5. Worst-case latency versus worst-case peak temperature for similar bindings but different placements, of an MJPEG decoder evaluated on MPARM platform [4].

Let us consider the diagram in Figure 5 that has been first introduced in [19] and [33]. Solution pairs where only the placement of processing components is different are illustrated and indicate that physical placement cannot be ignored in temperature analysis, e.g., mappings (a) and (b) have the same latency, but their peak temperatures differ by more than 8 K. Therefore, even if the mapping is already predefined, the system designer might still reduce the temperature by selecting a different placement. The same is true for the opposite case, when designs might violate temperature thresholds if the physical placement has not been properly included in the system-level analysis. These experiments show that temperature distributions and temperature peaks are not easy to infer at system level, since they are governed by complex dependencies on the actual topology of the chip, its physical parameters, heat transfer rules, and accumulated bursts of jobs in applications' workloads that actually produce worst-case temperatures [27]. In fact, for any manycore design, without accurate worst-case chip temperature analysis tools included into system performance analysis, no

guarantees can be given and mappings cannot be ruled out at system-level. To answer all these challenges, we have extended the distributed operation layer (DOL) [39] to consider system-level thermal-aware task to processors mapping.

4.1 Mapping Optimization Framework

The mapping optimization cycle implemented in the distributed operation layer [39] is illustrated in Fig. 6. In PRO3D, DOL considers parallel streaming applications represented as synchronous dataflow graphs (SDF) [15] and specified independently from the given PRO3D architecture. After the analysis of different design alternatives, a set of optimized mappings are provided. In PRO3D, each mapping is individually analyzed in terms of performance and (worst-case) thermal behavior. Finally, the chosen mapping specification will be further synthesized and implemented on the final system or can be simulated on the virtual platform. Typically, we use this low level simulation in a feedback loop for automatically calibrating the time and thermal analysis models [11, 12, 38].

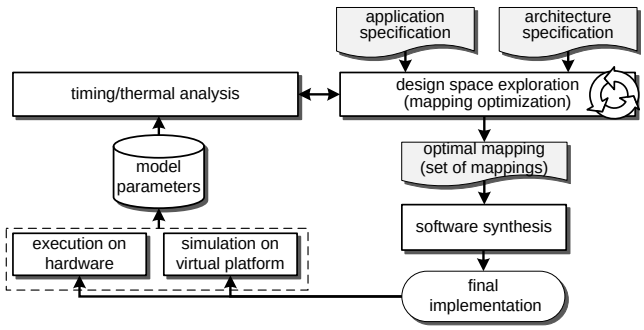


Fig. 6. Real-time and thermal-aware mapping optimization loop in distributed operation layer for PRO3D (DOL3D).

4.2 Thermal Models and Analysis in DOL3D

Several system-level analysis models are included in DOL [39], ranging from very simple, static models to more complex, dynamic analytic models such as modular performance analysis (MPA). MPA [43] is an analytic approach targeting real-time systems and based on real-time calculus (RTC) [37]. From elementary knowledge about the best-case and worst-case behavior of system components in all operating conditions, MPA provides hard upper and lower bounds for various performance criteria of the system, such as end-to-end delays, buffer requirements, or resource utilizations. The system is abstractly modeled by bounded timing properties of event streams traversing the system, bounded capabilities of architectural units, and bounded execution requirements of event streams

on individual components. Abstract components define the semantics of task executions and resource sharing mechanisms. Based on these abstractions, in classical timing analysis, the critical instant of task releases is used to guarantee the system worst-case execution time. Inspired from this time critical instant, we determine the temperature critical instant guaranteeing the worst-case peak temperature in the system in [27]. Similar to timing analysis, this critical temperature trace is identified among infinitely many traces that comply with the event stream specification in MPA and then the temperature of the system is simulated for the identified critical temperature trace. To apply the proposed method in [27] to a multi-core system such as PRO3D, in [33] we have extended the analysis to also consider the heat transfer among neighboring components. Therefore, in [33] we provide a tight upper bound on the worst-case peak temperature of the entire multi-core system.

However, the method proposed in [33] uses linear search to calculate a tight bound on the worst-case peak temperature, and therefore exhibits a too long execution time for the design space exploration of a multi-core system with tens of processing components. An approximate method with a lower time complexity and that is three orders of magnitude faster has been determined in [32] to calculate an upper bound on the maximum temperature of a multi-core system. To extend the search options in the design space, in a thermal-aware task assignment is currently investigated such that the worst-case chip temperature is minimized and all real-time deadlines are met. This is possible due to individual static frequency selection for all cores in the system. An illustrative example is shown in Figure 7, where two identical tasks are mapped on three homogeneous processing components. When assigning the maximum operation frequency on all cores, the worst-case chip temperature is obtained when the tasks are assigned to different processing components. This is because both processing components process in parallel in the thermal critical scenario. When the operation frequency of every processing component is the minimum frequency such that all deadlines are just met, the lowest peak temperature is found when both tasks are mapped to different, non-adjoined processing components. This is because the individual operation frequencies can drastically be reduced when tasks are mapped onto different processing components.

The techniques described so far can be applied at design time, having the advantage of thermal-aware performance estimations and early thermal optimizations. However, in spite of thermal-aware design-time choices, there may be the need to respond to run-time thermal emergencies. In this case, specific thermal management actions might be applied as those described in section . However, to benefit of pre-calculated and still predictable performance, these dynamics have to be a-priori considered and included in the design strategy. One option is to select a set of optimized mappings after the design space exploration, instead of just one mapping. Each such mapping is having different guaranteed performance and temperature characteristics that can be exploited at run-time. The alternative is to apply control-theory to control the speed of processors in a loop receiving feedback from temperature sensors as described

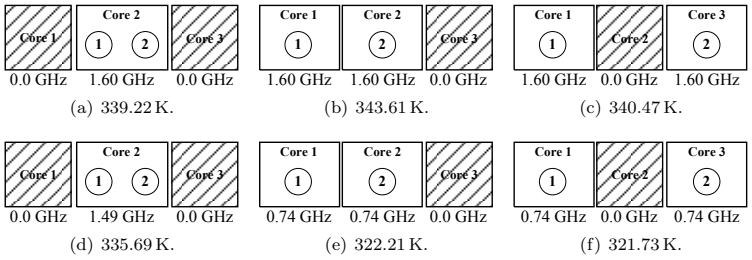


Fig. 7. Worst-Case Chip Temperature for Different Task Assignments and Clock Frequencies

in [14]. The solution in [14] is designed to meet thermal constraints and simultaneously provide safe bounds on worst-case delays suffered by all jobs in the system.

5 Generation & Simulation of the System-Model

The PRO3D system construction method [7] starts from a DOL [39, 12] specification and is both rigorous and allows fine-grain analysis of system dynamics. It is rigorous because it is based on formal BIP models [3] with precise semantics that can be analyzed by using formal techniques. A system model in BIP is derived by progressively integrating constraints induced on an application software by the underlying hardware. It is obtained, in a compositional and incremental manner, from BIP models of the application software and respectively, the hardware platform, by application of source-to-source transformations that are proven correct-by-construction [7]. The system model describes the behavior of the mixed HW/SW system and can be simulated and formally verified using the BIP toolset.

The method for the construction of mixed HW/SW system models is illustrated in Fig. 8. It takes as inputs: (i) the (untimed) application software, (ii) the (timed) hardware architecture and (iii) the mapping between them described in DOL. It proceeds in two main steps. The first step is the construction of the *abstract system model*. This model represents the behavior of the application software running on the hardware platform according to the mapping, but without taking into account execution times for the software actions. In the second step, the (bounds for) execution times are obtained by running every software process in isolation on the platform. These bounds are injected into the abstract system model and lead to the *system model*. This final model allows for the accurate estimation through simulation of real-time characteristics (response times, delays, latencies, throughput, etc.) and indicators regarding resource usage (bus conflicts, memory conflicts, etc.).

System models are furthermore used for platform-dependent code generation. As illustrated in the Fig. 8, the generated code consists mainly of two parts:

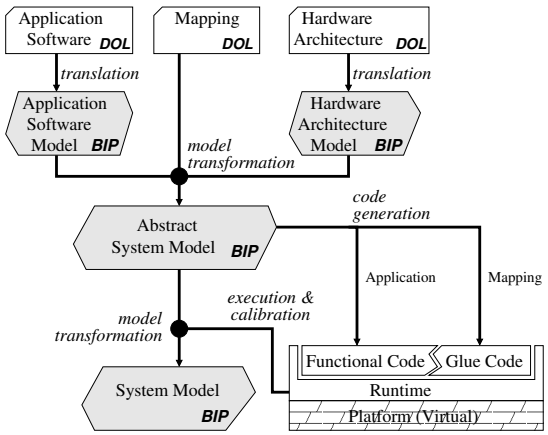


Fig. 8. System Model Construction & Code Generation

the *functional code*, which implements the different application tasks and their communication and the *glue code*, which implements the deployment of the application onto the platform according to the mapping and manages its execution lifecycle. This code is built on top of platform *runtimes*, that is, available APIs and libraries for thread management, memory allocation, communication and synchronization, etc. Once generated, the code is compiled by the native platform compiler and linked with the runtime libraries to produce the binary image for execution on the platform. This approach has been implemented and validated on *mpsim* (MPARM cycle-accurate simulator), *Gepop* (STHORM Posix simulator), STHORM TLM simulator and will be tested on the real STHORM silicon during Fall 2012. As for the target runtimes, we originally started using the Native Programming Layer (NPL), a common runtime implemented for both MPARM and STHORM; since mid-2012 we developed an implementation of the MCAPI standard for the STHORM platform [22].

6 STHORM, a Manycore Platform

Formerly known as P2012 [5, 20, 36] the STHORM modular architecture is shown Fig. 9. At the fabric level, an asynchronous NoC (Network-on-Chip) is organized in a 2D-mesh topology of clock-less routers. Each router has a NI (Network Interface) that connects to a cluster made of up to 16 cores in SMP and a number of communication engines to connect user defined HW IPs. This archtecture is a natural Globally Asynchronous Locally Synchronous (GALS) scheme and isolates logically the clusters. The NI gives access to the cluster main Clock, Variability and Power (CVP) controller, to control a power management harness. Within PRO3D we investigate 3-tier stacking for STHORM: a bottom SoC carrier for the general purpose host and IOs, a STHORM computing die, and a memory die. The experiments will include a number of VPI thermal modeling extensions to exercise the whole PRO3D SW development flow.

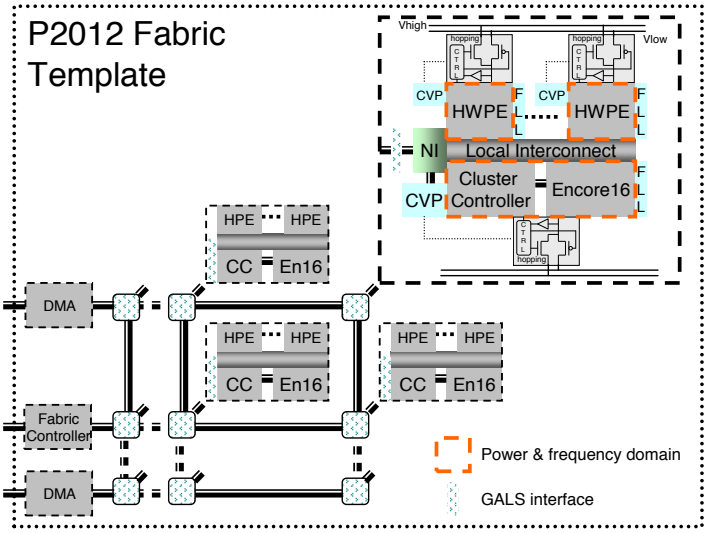


Fig. 9. The STHORM Computing Fabric Template.

7 Conclusion after 24 Months into PRO3D

Two years and a half into its workplan, PRO3D has developed a number of tools and already assembled them into a consistent 3D exploration and programming workflow: A compact transient thermal model for simulation of 3D ICs with liquid cooling and its corresponding monitoring runtimes, a flexible virtual platform infrastructure for modelling and analysis of 3D architecture, a high level mapping optimisation tool focusing on peformance and preliminary support for temperature analyses, a rigorous transformation toolset for components that allow for the construction and assembly of system models and the generation of distributed intermediate format for deployment on the target platform. The last year of PRO3D will be focused on experiments with an actual industrial embedded manycore platform STHORM. Experiments have started on virtual platforms, and will move to real STHORM silicon during Fall of 2012.

Challenges for 3D and Programming

Besides these practical results, we think that the main challenges raised by 3D are a related to a retrofit of characteristics of the architecture into compilation flows and runtimes. Somehow, this is very similar to the issues encountered in HPC with distributed machines in the early '90. The problem is difficult, but a wide body of literature exists for purely topological issues. The new issues introduced by 3D stacking are mostly related to thermal aspects. Theses issues have two main origins:

1. *Thermal cross-coupling of execution units.* The relative position of processing units as a whole, or computing units from therein (operators, instructions

decoders, register files, caches, etc.) and memory defines how heat from one element impacts another one. If two processors are too close to each other, we may have to offload both of them in situations where a single one could have run without harm. So not only the topoplogy of the manycore will have to be know from the compilation flow and the runtime, but also the geometry and thermal characteristics of the hardware [38];

2. *Different time scale for thermal propagation and computation forecast.* Manycore architecture are in the GHz range, while the evolution of the temperature is in the Hz range. This means several order of magnitude between the cause of heating –computations– and heating itself [34]. This gap in dynamic magnitude is reinforced by the fact that even at constant frequency, energy consumption increases with temperature. All this makes it difficult to reverse temperature variations. Any decision related to thermal management will probably have to use predictive thermal models [1].

We thing that this will bring a number of consequence on programming models, compilation and runtimes:

- *The fading of pure static compilation.* Due to the huge gap of time scale between computation and thermal effects, it seems very difficult, if doable at all, to build full-static compilation schemes where the compiler will decide of the mapping off-line, before execution, once and for all. At least to ensure platform's thermal integrity, some level of responsibility w.r.t. mapping must be left to the runtime [44]. To ensure this integrity the runtime will have to deal with tasks scheduling and resource allocation while taking into account not only the architecture's topology and the computation load [18], but also the actual geometry and thermal characteristics of the material involved in the architecture [29]. This will require programming models that can provide enough flexibility at execution whereas essential properties can be guaranteed at compile-time [3].
- *The fading of von Neumann as a programming model.* As for programming models, we should move away from von Neumann –only as programming model, not as computing architecture– and consider other kinds of programming models naturally parallel, like process network and message passing already discussed [2, 13, 17]. Even these parallel programming models must be checked to be amendable to analyses that can predict the amount of computing load, if not to an absolute time reference, at least towards a moving horizon. This is necessary to provide computation forecasts to a runtime scheduler that can efficiently use the stacked architecture while preserving its thermal integrity.

Acknowledgments & Consortium

PRO3D is funded by the EU under FP7 GA n° 248776. It brings together **CEA**, Commissariat à l'énergie atomique et aux énergies alternatives (coord.), Fr.; **VERIMAG**, represented by Université Joseph Fourier Grenoble 1, Fr.;

ETHZ, Eidgenössische Technische Hochschule Zürich, CH; **UNIBO**, Università di Bologna, It.; **STM**, STMicroelectronics, Fr.; **EPFL**, École polytechnique fédérale de Lausanne, CH. PRO3D Started in Jan. 2010 for an original duration of 30 months. It has been granted a six months extension to experiment with actual STHORM silicon, and will now end in Dec. 2012

References

1. Aly, S., Mostafa, M., Coskun, A.K., Atienza Alonso, D.: Fuzzy Control for Enforcing Energy Efficiency in High-Performance 3D Systems. In: Proceedings of the 2010 International Conference on Computer-Aided Design (ICCAD 2010. New York (2010)
2. Basu, A., Bozga, M., Sifakis, J.: Modeling Heterogeneous Real-time Systems in BIP. In: Software Engineering and Formal Methods SEFM'06 Proceedings. pp. 3–12. IEEE Computer Society Press (2006)
3. Basu, A., Bensalem, S., Bozga, M., Combaz, J., Jaber, M., Nguyen, T.H., Sifakis, J.: Rigorous component-based design using the BIP framework. IEEE Software, Special Edition – Software Components: Beyond Programming 28(3), 41–48 (Jun 2011)
4. Benini, L., Bertozzi, D., Bogliolo, A., Menichelli, F., Olivieri, M.: MPARM: Exploring the Multi-Processor SoC Design Space with SystemC. J. VLSI Signal Process 41(2), 169–182 (2005)
5. Benini, L., Flamand, E., Fuin, D., Melpignano, D.: P2012: Building an ecosystem for a scalable, modular and high-efficiency embedded computing accelerator. In: Rosenstiel, W., Thiele, L. (eds.) DATE 2012. pp. 983–987. IEEE (Mar 2012)
6. Bortolotti, D., Paterna, F., Pinto, C., Marongiu, A., Ruggiero, M., Benini, L.: Exploring instruction caching strategies for tightly-coupled shared-memory clusters. In: In Int. Symp. on Systems-on-Chip (2011)
7. Bourgos, P., Basu, A., Bozga, M., Bensalem, S., Sifakis, J., Huang, K.: Rigorous system level modeling and analysis of mixed HW/SW systems. In: Proceedings of MEMOCODE. pp. 11–20. IEEE/ACM (2011)
8. Brunschwiler, T., et al.: Interlayer cooling potential in vertically integrated packages. Microsyst. Technol. 15(1), 57 – 74 (2009)
9. Burns, A.: Scheduling hard real-time systems: a review. Softw. Eng. J. 6, 116–128 (May 1991)
10. Coskun, A.K., et al.: Utilizing predictors for efficient thermal management in multiprocessor socs. IEEE Transactions on CAD 28(10), 1503–1516 (2009)
11. Haid, W., Keller, M., Huang, K., Bacivarov, I., Thiele, L.: Generation and calibration of compositional performance analysis models for multi-processor systems. In: Proc. Intl Conference on Systems, Architectures, Modeling and Simulation (SAMOS). pp. 92–99. IEEE, Samos, Greece (2009)
12. Huang, K., Haid, W., Bacivarov, I., Keller, M., Thiele, L.: Embedding formal performance analysis into the design cycle of mpsocs for real-time streaming applications. ACM Trans. Embed. Comput. Syst. 11(1), 8:1–8:23 (Apr 2012), http://doi.acm.org/10.1145/2146417.2146425
13. Joven, J., Marongiu, A., Angiolini, F., Benini, L., De Micheli, G.: Exploring programming model-driven qos support for noc-based platforms. In: Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2010 IEEE/ACM/IFIP International Conference on. pp. 65–74 (Oct 2010)

14. Kumar, P., Thiele, L.: Timing analysis on a processor with temperature-controlled speed scaling. In: Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). IEEE Computer, Beijing, China (2012)
15. Lee, E., Messerschmitt, D.: Synchronous data flow. Proceedings of the IEEE 75(9), 1235 – 1245 (Sep 1987)
16. Leon, A., et al.: A power-efficient high-throughput 32-thread SPARC processor. ISSCC 42(1), 7 – 16 (2007)
17. Marongiu, A., Benini, L.: An OpenMP compiler for efficient use of distributed scratchpad memory in MPSoCs. Computers, IEEE Transactions on PP(99), 1 (Oct 2010)
18. Marongiu, A., Burgio, P., Benini, L.: Vertical stealing: robust, locality-aware do-all workload distribution for 3D MPSoCs. In: Kathail, V., Tatge, R., Barua, R. (eds.) CASES. pp. 207–216. ACM (2010)
19. Marwedel, P., Teich, J., Kouveli, G., Bacivarov, J., Thiele, L., Ha, S., Lee, C., Xu, Q., Huang, L.: Mapping of applications to MPSoCs. In: Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2011 Proceedings of the 9th International Conference on. pp. 109–118 (Oct 2011)
20. Melpignano, D., Benini, L., Flamand, E., Jego, B., Lepley, T., Haugou, G., Clermidy, F., Dutoit, D.: Platform 2012, a many-core computing accelerator for embedded SoCs: performance evaluation of visual analytics applications. In: Groeneveld, P., Sciuto, D., Hassoun, S. (eds.) DAC. pp. 1137–1142. ACM (Jun 2012)
21. Micheli, G.D., Pavlidis, V., Alonso, D.A., Leblebici, Y.: Design methods and tools for 3d integration. In: Proceedings of the Symposium on VLSI Technology. pp. 182–183. Kyoto, Japan (Jun 2011)
22. The Multicore Association: The Multicore Communications API (MCAPI™) v2.015 (2011), http://www.multicore-association.org
23. NVIDIA: Next Generation CUDA Compute Architecture: Fermi (2010), http://www.nvidia.com, whitepaper
24. Plurality: The HyperCore Processor (2010), http://www.plurality.com, Plurality Ltd.
25. PRO3D – Programming for Future 3D Multicore Architectures (2010), http://pro3d.eu
26. Qian, H., et al.: Cyber-physical thermal management of 3D multi-core cache-processor system with microfluidic cooling. ASP Journal of Low Power Electronics 7(1), 1–12 (2011)
27. Rai, D., Yang, H., Bacivarov, I., Chen, J.J., Thiele, L.: Worst-case temperature analysis for real-time systems. In: Design, Automation Test in Europe Conference Exhibition (DATE), 2011. pp. 1–6 (Mar 2011)
28. Ruggiero, M., Angiolini, F., Poletti, F., Bertozzi, D., Benini, L., Zafalon, R.: Scalability analysis of evolving SoC interconnect protocols. In: Int. Symp. on Systems-on-Chip. pp. 169–172 (2004)
29. Sabry, M., Atienza, D., Coskun, A.K.: Thermal Analysis and Active Cooling Management for 3D MPSoCs. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS'11) (2011)
30. Sabry, M.M., et al.: Energy-Efficient Multi-Objective Thermal Control for Liquid-Cooled 3D Stacked Architectures. IEEE Transactions On CAD 30(12), 1883–1896 (2011)
31. Sabry, M.M., Ruggiero, M., Del Valle, P.G.: Performance and energy trade-offs analysis of L2 on-chip cache architectures for embedded MPSoCs. In: Proceedings of the 20th symposium on Great lakes symposium on VLSI. pp. 305–310. GLSVLSI '10, ACM, New York, NY, USA (2010)

32. Schor, L., Bacivarov, I., Yang, H., Thiele, L.: Fast worst-case peak temperature evaluation for real-time applications on multi-core systems. In: Proc. IEEE Latin American Test Workshop (LATW). IEEE, Quito, Ecuador (2012)
33. Schor, L., Bacivarov, I., Yang, H., Thiele, L.: Worst-case temperature guarantees for real-time applications on multi-core systems. In: Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). IEEE Computer, Beijing, China (2012)
34. Sridhar, A., Vincenzi, A., Ruggiero, M., Brunschwiler, T., Atienza, D.: 3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling. In: Computer-Aided Design (ICCAD), 2010 IEEE/ACM International Conference on. pp. 463–470 (2010)
35. Sridhar, A., Vincenzi, A., Ruggiero, M., Brunschwiler, T., Atienza, D.: Compact transient thermal model for 3D ICs with liquid cooling via enhanced heat transfer cavity geometries. In: Thermal Investigations of ICs and Systems (THERMINIC), 2010 16th International Workshop on. pp. 1–6 (2010)
36. STMicroelectronics, CEA: Platform 2012 – A Manycore Programmable Accelerator for Ultra-Efficient Embedded Computing in Nanometer Technology (Nov 2010), whitepaper
37. Thiele, L., Chakraborty, S., Naedele, M.: Real-Time Calculus for Scheduling Hard Real-Time Systems. In: Proc. IEEE Int'l Symposium on Circuits and Systems (ISCAS). vol. 4, pp. 101–104 (2000)
38. Thiele, L., Schor, L., Yang, H., Bacivarov, I.: Thermal-aware system analysis and software synthesis for embedded multi-processors. In: Proc. Design Automation Conference (DAC). pp. 268 – 273. ACM, San Diego, California, USA (2011)
39. Thiele, L., Bacivarov, I., Haid, W., Huang, K.: Mapping Applications to Tiled Multiprocessor Embedded Systems. In: Proc. Int'l Conf. on Application of Concurrency to System Design (ACSD). pp. 29–40 (2007)
40. Tindell, K.W., Burns, A., Wellings, A.J.: Allocating hard real-time tasks: an np-hard problem made easy. Real-Time Syst. 4, 145–165 (May 1992)
41. Tuckerman, D.B., Pease, R.F.W.: High-performance heat sinking for VLSI. IEEE Electron Device Letters 5, 126–129 (1981)
42. Vincenzi, A., Sridhar, A., Ruggiero, M., Atienza, D.: Fast thermal simulation of 2D/3D integrated circuits exploiting neural networks and GPUs. In: Proceedings of the 17th IEEE/ACM international symposium on low-power electronics and design. pp. 151–156. ISLPED '11, IEEE Press, Piscataway, NJ, USA (2011)
43. Wandeler, E., Thiele, L., Verhoef, M., Lieverse, P.: System architecture evaluation using modular performance analysis - a case study. Software Tools for Technology Transfer (STTT) 8(6), 649 – 667 (2006)
44. Zanini, F., Atienza, D., Benini, L., de Micheli, G.: Thermal-Aware System-Level Modeling and Management for Multi-Processor Systems-on-Chip. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS'11) (2011)
45. Zervas, M., Temiz, Y., Leblebici, Y.: Fabrication and characterization of wafer-level deep tsv arrays. In: Proceedings of 2012 Electronic Components and Technology Conference. San Diego, CA (2012)