Reconfigurable Platform for Content Science Research*

Chi-Sheng Shih Chia-Lin Yang Mong-Kai Ku Tei-Wei Kuo Shao-Yi Chien Yao-Wen Chang Liang-Gee Chen[1]
Graduate Institute of Networking and Multimedia
Dept. of Computer Science and Information Engineering
Graduate Institute of Electronic Engineering
Dept. of Electrical Engineering
National Taiwan University
Taipei, Taiwan 106, R.O.C.
{cshih, yangc, mku, ktw}@csie.ntu.edu.tw {shaoyi, ywchang, lgchen}@cc.ee.ntu.edu.tw

[1]Electronics Research and Service Organization
Industrial Technology Research Institute (ITRI)
Hsin Chu, Taiwan 310, R.O.C.

Abstract

The College of Electrical Engineering and Computer Science at the National Taiwan University has identied the area of content science for media-rich life, broadly construed, as one of core areas for the college's future directions. One major aspect of this project is to develop the enabling technology for reconfigurable platforms for multimedia applications. Specifically, the goal is to develop reconfigurable platforms of system-on-a-chip (SoC) components for the applications to support the needs of multi-modal multimedia contents and to provide rapid system prototyping. The faculties in the College of Electrical Engineering and Computer Science have formed a multi-discipline team to develop such technology. Our team includes seven faculties and more than thirty students from the college. This short report describes the reconfigurable platform for content science research activities currently underway by our team. Our current activities include to develop the technology to analyze the critical path for avoiding hardware contention, to minimize the use of logic components, to design the multimedia IPs, to optimally route the bus and place the logic units, to design energy efficient cache, to evaluate the performance and power consumption, to design the algorithm for temporal floor-planning/placement.

The College of Electrical Engineering and Computer Science at the National Taiwan University has identied the area of content science for media-rich life, broadly construed, as one of core areas for the college's future directions. Content science herein is the scientific foundations including rationales, theories, methodologies and technologies that support the entire value chain of content production and utilization in a media-rich life: creation, analysis, storage, search, manipulation, organization, management, delivery, presentation and interaction.

One major aspect of this project is to develop the enabling technology for reconfigurable platforms for multimedia applications. Specifically, the goal is to develop reconfigurable platforms of system-on-a-chip (SoC) components for the applications to support the needs of multimodal multimedia contents and to provide rapid system prototyping. A multimedia SoC consists of large-scale logic/functional blocks (e.g., JPEG, MPEG, microprocessors, embedded memory, etc.) and interconnections for connecting those blocks to provide enough capability and flexibility to handle different computing needs of structured media contents and in supporting human perception. The faculty in the College of Electrical Engineering and Computer Science at National Taiwan University have formed a research team to develop such technology. Our team is lead by Prof. Liang-Gee Chen, who joined the Department of Electrical Engineering in 1988 and is now the executive vice president and general director of Electronics Research and Service Organization of the Industrial Technology Research Institute (ITRI). Other team members include Tei-Wei Kuo and Chia-Lin Yang, who joined the Department of Computer Science and Information Engineering in 2000 and 2002, respectively; Yao-Wen Chang, who joined the Graduate Institue of Electronics Engineering in 2001; Mong-Kai Ku and Chi-Sheng Shih, who joined the Department of Computer Science and Information Engineering in 2003 and 2004, respectively; and Shao-Yi Chien, who joined the Department of Electrical Engineering in 2004. There are more than thirty students in the team.

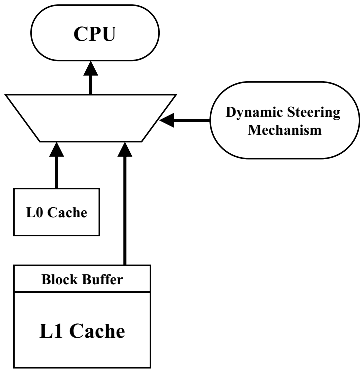
This short report describes the reconfigurable platform for content science research activities currently underway in our college. More information on these activities can be found through our web site, `http://www.csie.ntu.edu.tw/~excellence`.

1 Energy-Efficient Cache Architecture

Multi-modal interactive access is highly data-dominated. The efficiency on the processing of data-dominated applications depends on the design of the memory subsystem. Since the memory subsystem contributes a significant por-

*This work is supported in part by a grant from the NSC program NSC 93-2752-E-002-008-PAE.

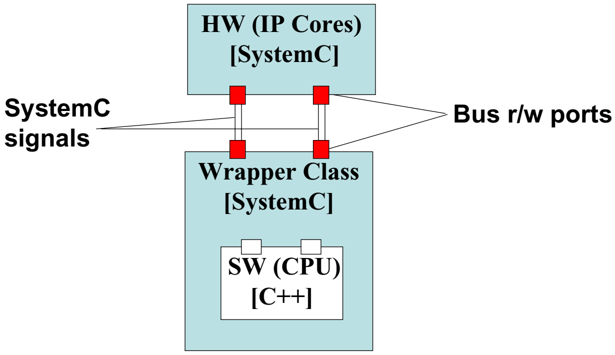
tion of overall performance and energy consumption for this type of applications, an energy-efficient cache architecture that meets the needs of performance and energy consumption of selected applications is an important design issue. We have developed several energy-aware cache management schemes for multimedia applications. The Software-Controlled Cache [1] proposes to control data allocation on the cache in the granularity of data types in an application since most multimedia applications contain several major data types that contribute to most of the memory accesses. The Hotspot cache architecture [2] as shown in Figure 1 dynamically identifies frequently accessed basic blocks and stores them in the L0 cache, while others are placed only in the L1 cache. The L1 cache is augmented with a block buffer to exploit the spatial locality of non-hot basic blocks. We also propose a phase-aware I-cache size synthesis algorithm for cache reconfiguration considering the QoS (Quality of Service) demand in multimedia applications [3]. We identify phases and the required instruction cache size through static program analysis, and associate phases with long-running loops, which are commonly seen in multimedia applications. We are currently investigating techniques for the cache leakage management with the real-time consideration.



2 System-Level Performance/Power Evaluation Framework

To facilitate early design space exploration for a platform SoC, we are currently developing a system-level performance/power evaluation framework. A platform-based SoC contains a microprocessor, memory hierarchy, interconnected buses, and a set of IP cores. For a platform SoC designer, it is challenging to determine platform components in view of performance and power at the early stage of the design flow. The proposed system-level simulation framework provides cycle-accurate performance and power evaluation at the system-level through efficient hardware-software co-simulation.

In traditional hardware-software co-simulation environments, hardware and software are usually described in different languages. This kind of approach requires careful control and synchronization among multiple simulation engines. In our work, we use SystemC [4] to integrate hardware (IP cores) and software (CPU) components into one single engine. SystemC is an open source C/C++ simulation environment that provides several class packages for specifying hardware blocks and communication channels. The CPU simulator is embedded within the SystemC simulator as a C++ class as shown in Figure 2. To provide fast simulation, we adopt timed Transaction-Level Modeling (TLM) [5]. Communication is modelled by channels which model data transfer through message passing without implementing protocol details. IP cores are modelled at the register-transfer level and the CPU is modelled at the micro-architecture level that provides cycle-accurate performance information. For the system-level power evaluation, the system components are decomposed into three types: CPU, IP cores and bus. The power model for the processor is based on the approach proposed in the Wattch tool set [6] which provides parameterized power models for four major structures: array structure, content-addressable memories, combinational logic/wire, and clocking. To model the power consumption of the bus, we decompose the bus architecture into three components: the arbiter, decoder and multiplexors. Similar to Caldari et al. [7], we consider four types of bus activities: idle, read, write, and idle with bus handover. The power characterizations of IP cores are given by the IP providers. In our current implementations, we model the ARM processor and the Advanced Microcontroller Bus Architecture (AMBA). We are currently integrating RTOS into our framework to handle multitasking workloads.



3 Multimedia IP Design

The line of works is to design efficient hardware IPs for multimedia applications. These IPs are designed to make the content-science based applications feasible and practical. That is, with these IPs, an SoC can be developed easily for different applications, the realtime requirements of some applications can be achieved, and the strong computing power of these IPs can be integrated into small embedded portable devices.

Our earlier works for IP design can be classified into to two categories. The first is about video and image coding systems. Advanced hardware architectures and chip implementations of MPEG-4, JPEG2000 [8], and H.264 [9] have been successfully developed. The hardware architectures of the key components of these systems are also carefully designed. On the other hand, video analysis is also an important issue in a multimedia system. Therefore, in the second research direction, some on-going research topics are about hardware supported video analysis systems, and some results about video segmentation are published in [10] and [11].

In order to deal with various applications in content-rich lives, reconfigurable hardware may be a good solution for hardware accelerators because of its efficiency and flexibility. Based on the research results about video coding and analysis, the reconfigurable key components of multimedia systems will be developed in this project as hardware IPs, including hardware IPs for image and video coding and for intelligent video analysis.

Hardware IP for Image and Video Coding Some research works about hardware IP design for image and video coding systems are introduced in this subsection. Motion estimator can be said as the most important module in a video coding system. One of our research works is about reconfigurable motion estimator for power-aware video coding systems. A power-aware video coding system can dynamically change its configuration to achieve the maximum quality under different power constraints. In this project, we will develop a reconfigurable motion estimator for power-aware video coding systems. The reconfigurable hardware is designed to support our previous work, multi-mode content-aware motion estimation algorithm. The hardware architecture is based on parallel-tree architecture, and the reconfigurability can be achieved by changing the routing between the search range data buffer and the parallel-tree. We have implemented a prototype chip with UMC 0.18 μm 1P6M CMOS process. Experiments show that for video sequences in CIF 30 fps with search range [-16, 15], the power consumption can vary from 51.27 mW, 24.88 mW, 13.76 mW, to 8.46 mW. It is also shown that the proposed motion estimator can achieve better quality under each power constraint. The proposed architecture is the first motion estimator that has power-aware ability in the literature. It can be integrated into portable embedded devices to enable power-aware video coding ability.

On the other hand, for image coding systems, we develop a reconfigurable discrete wavelet transform (DWT) architecture for advanced multimedia systems, where a novel reconfigurable DWT architecture is proposed to meet the diverse computing requirements of advanced multimedia systems. The proposed architecture mainly consists of a reconfigurable processing element array and a reconfigurable address generator, featuring a dynamically reconfigurable capability where the wavelet filter kernels and wavelet decomposition structures can be reconfigured at run-time with little overhead. The lifting-based reconfigurable processing element array possesses better computational efficiency than a convolution-based architecture, and a systematic design method is provided to generate the hardware configurations of different wavelet filter kernels for it. The reconfigurable address generator handles flexible address generation for data I/O access in different wavelet decomposition structures. A prototyping chip has been fabricated by the TSMC 0.35 μm 1P4M CMOS process. At 50 MHz, it can achieve at most 100 Mpixel/sec transform throughput.

Hardware IP for Intelligent Video Analysis Other than coding, intelligent video analysis is also an important research issue in content-rich live. It can be used for two purposes. If the intelligent video analysis of the video content is done before coding, more interactivity of content can be accomplished. Users can easily search and manipulate the contents by themselves. On the other hand, if the intelligent video analysis system is places on the living environments of users, it can be used to analyze the behavior of users to let machine interact with users as a new human-machine-interface.

Video segmentation is a key operation of intelligent video analysis, which can segment out significant objects in a video content. For real-time applications, hardware implementation of video segmentation is inevitable. In this project, we will develop a reconfigurable architecture for real-time moving object segmentation systems, where various segmentation algorithms, including our previous work, an effective moving object segmentation algorithm, can be supported for different applications.

4 Hardware Contention Avoidance for Multi-Tasking SoC

Most embedded systems are now synthesized by a number of hardware processing elements and customized microprocessors, such as system on a chip (SoC). Vendors have to face different pressure on the time to market, user requirement, timing specification of applications and cost control on the end products. Given those stringent requirements and harsh challenges, there is never replacement for proper and intelligent methodology in system designs and implementations for successful product development. To reduce the cost of production, they might share the same hardware processing elements for the same functionality such as IDCT, DeQ for H.264 decoding and MP3 decoding in video phone. We are concerned with analyzing the cause of hardware contention and developing a resource management mechanism to avoid hardware contention. As the complexity of SoC increasing, the system analysis becoming more difficult. We explore the system behavior in coarse-grain level, which is task level.

Researchers have primarily focused their interests in last decade on hardware-software co-design with static scheduling of hardware resources and software tasks in the design time (e.g, [12].) Such an approach has good performance predictability but often results in less flexibility in run-time resource optimization. When dynamic scheduling is applied, most works discuss the conflicting behavior analysis among tasks on microprocessors. However, the scheduling issues for tasks of complicated functionality in hardware resources (and software tasks) is different from those for tasks running on microprocessors. The interference behavior of tasks may cause unpredictable performance. If a task is interrupted on a processing element, the task fails and its results lost. We call this phenomenon as *hardware contention*. To avoid such interference, we investigate the causes and properly control the times for accessing certain hardware components.

Intuitively adding local buffers for each hardware component may resolve the hardware contention problem. However, it increases not only manufacture cost but also design complexity for optimizing buffer size and queuing mechanism. Specifically, when non-preemptible resources such as processing elements are shared among tasks, scheduling anomaly may occur [13]. Alternatively, in [14], we propose a non-buffered approach: The time instance at which a task departs the microprocessor is properly controlled, so that hardware contentions can be eliminated. The approach is called Departure Time Guarder, referred as DTG.

Assuming that the release time for tasks are arbitrary, departure time guarder approach is designed to analyze hardware contention in all possible cases. To consider the dynamic system workload and reduce run-time overhead, DTG approach consists of two phases: off-line feasible interval determination phase and on-line control phase. In the off-line phase, we propose an algorithm to conduct all possibly hardware contention range among any two tasks in pessimistic manner. The results computed by off-line phase are used as the inputs for on-line control phase. During online phase, DTG adjusts the departure time for tasks which execute on microprocessor to signal the succeeding tasks executing on the processing elements. The rationale is to insert delay time such that the departure time of all tasks are not in hardware contention ranges which are computed in off-line phase. The on-line adjustment could be done by revising the invocation call to update the control register. With run-time release time information, this method could also shorten the response times of tasks.

Although adding local buffers for hardware components may cause several drawbacks, it usually results in better performance. To resolve the performance anomaly problem, we shall study the buffer supporting problem for processing elements and then explore the optimal combination of buffer support and system performance in SoC designs. We shall show cases in which buffer support is not necessary and propose scheduling algorithms for such cases. Furthermore, resource replication is another approach to resolve hardware contention. The tradeoff between resource replication and performance improvement should be explored. Although problems in this direction are usually NP-hard, our goal is to propose systematic and engineering solutions for hardware-software co-design. Specifically, we shall propose methodology to evaluate the interference behaviors among multiple task flows with a given set of replicated resources. The proposed methodology shall provide an engineering base for engineers in determining the best system configuration, given a set of functions to be implemented and their performance requirements.

5 Minimizing Reconfigurable Logics Usage

Dynamically reconfigurable logic (DRL) devices provide system engineers/vendors great flexibility in reconfiguring hardware for proper functionalities. With reconfigurability technology, the time to prototype and even the time to market could be significantly shortened. We target one essential issue in the reconfigurability study: minimizing the hardware usage in reconfigurable devices when a schedule and a processing element (PE) mapping are derived from a hardware/software co-design process.

This line of works propose scheduling algorithms for the derivation of *reconfiguration plans* to minimize the required FPGA configuration contexts, where a configuration context might operate independently while another one is under loading. We are interested in the derivation of a plan to reconfigure the configuration contexts of an FPGA based on a given schedule and a given allocation map [15]. Note that a schedule and an allocation map could be results derived by a CAD tool for hardware/software co-design, such as Ptolemy and Synopsys CCSS. A reconfiguration plan should be derived to determine the time and the configuration context ID for each PE to satisfy the needs of a given schedule and a given allocation map, with an objective to minimize the number of required configuration contexts.

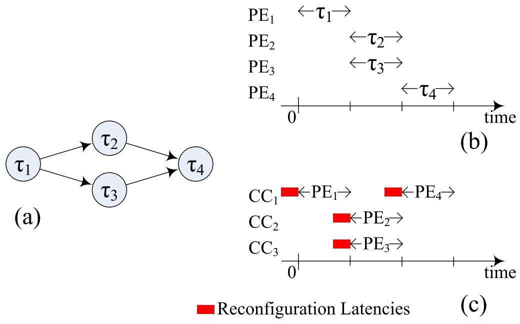


Figure 3 shows an example. Figure 3(a) is an application represented by a directed acyclic graph. The application consists of tasks with dependencies and a task is a set of operations. With a CAD tool, a schedule and a task-PE allocation map are given as shown in Figure 3(b). Our proposed algorithms could derive a reconfiguration plan with the consideration of reconfiguration latencies as shown in Figure 3(c).

To solve this problem, we first assume that tasks in allocation maps do not share PEs. We consider two cases: (1) PEs have the same reconfiguration latency and (2) PEs have different reconfiguration latencies. Both cases are solved in this work by proposed scheduling algorithms with time complexity $O(n \log n)$, where n is the number of tasks. We also prove the optimality that the derived reconfiguration plans use minimal configuration contexts. After that, a more general instance in which tasks in allocation maps share PEs is taken into consideration. The problem becomes more complicated since the reconfiguration latency of a PE can be avoided if the PE has been configured in contexts by the request of an executed task. We use the methodology to merge tasks sharing the same PE with successive requests into a new task, and then schedule them with the proposed scheduling algorithm for tasks without sharing PEs. This algorithm is also with time complexity $O(n \log n)$, however, the derived reconfiguration plans are not optimal. Several experiments are done in this work for the comparison between optimal solutions and our plans. We must emphasize that the results of this research could be applied to the co-designs of applications with general task graphs or the hardware-size optimization of specific functionalities.

We will continue to develop the scheduling algorithms for hardware/software co-design, especially in SoC domain. The first direction is on the minimization of memory usage. We found that the biggest portion of today's chips, no matter ASICs or DSPs, is occupied by memory blocks. However, these memory blocks are separated into several parts and dedicated for specified hardware components in traditional design. We argue that these memory blocks can be shared by different hardware components with a proper schedule to reuse the memory blocks. By doing so, we can decrease memory usage and, thus, decrease die size to saves the manufacture cost of chip production. We are also developing a scheduling algorithm for the minimization of memory allocation. The other direction is the allocation of on-chip bus. Since applications are usually a set of dependent tasks, the tasks may need to transfer data among each other. These communications are accomplished by buses. A bus can only transmit several bits of data at a time. If several tasks would transmit data, there are bus contentions. Hence, we are also working on an approach to find the proper number of buses so that the performance is improved.

6 Automatic Device Interface Generation

Our work for focuses on the tools that automate the reconfigurable embedded system design process. In embedded system design, proper design of interface between software and hardware is crucial in determining the functionality and performance of the overall system. Dynamically reconfigurable hardware system promises further integration between software and hardware. Current design methodology for device driver and hardware bus are error prone and requires hand tweaking by engineers that is knowledgeable in both software and hardware design. The goal of our project is to produce tools that automate and accelerate the generation of interface for embedded systems

Embedded system device driver development is one of the most critical tasks in a SoC design. We propose a novel technology called "Expert Template"[16] to automate the device driver synthesis process. Device driver specifications written in extensible markup language (XML) inside C/C++ source code comments are translated into native device drivers for any specified OS. Mapping between different OS is achieved by a pure conceptual layer instead of a physically existing platform independent layer. The device driver synthesized by our methodology will have the same quality of an expertly written one. There will be no additional mapping overheads with our approach.

On the hardware side, our bus wrapper generation tools will enable existing hardware IP using different bus standard to work seamlessly together. It will also allow new IP to easily adapt to different system buses. An IP Library for dynamically reconfigurable system is built to support the system development. This library contains both hardware and software views for the IP to ease the design and verification of embedded system.

Finally, we developed a co-simulation technique in SystemC that incorporates the modeling of real-time operating systems (RTOS). We build an RTOS simulation layer on top of the desktop OS to support accurate RTOS modeling in our co-simulation. This add-on RTOS layer can be later completely removed from the application once the application is running on the target embedded system. A hardware accelerated JPEG2000 codec with an RTOS (eCOS) running on a Altera Excalibur board is developed to validate our tool chain design.

7 Temporal Floorplanning/Placement for Reconfigurable Computing

Improving logic capacity by time-sharing, dynamically reconfigurable FPGAs are employed to handle designs of high complexity and functionality. We model each task as a 3D-box and deal with the temporal floorplanning/placement problem for dynamically reconfigurable FPGA architectures. Previously, we present a topological data structures, called 3D-subTCG's (3-Dimensional sub-Transitive Closure Graphs) [17], to represent the spatial and temporal relations among tasks to deal with the 3-dimensional (temporal) floorplanning/placement problem, arising from dynamically reconfigurable FPGAs. The 3D-subTCG uses three transitive closure graphs to model the temporal and spatial relations between modules, which was the most effective and efficient representation for 3-dimensional (temporal) floorplanning/placement. Because the geometric relationship is transparent to 3D-subTCG and its induced operations, we can easily detect any violation of temporal precedence constraints on 3D-subTCG.

We proposed the first tree-based representation, called T-trees, for temporal floorplanning/placement [18]. Given a compacted placement that cannot move toward the origin, we can construct its corresponding T-tree in linear time. Each node in a T-tree has at most three children which represent the dimensional relationship among tasks. In comparison with 3D-subTCG, T-trees have the following advantages:

- Since the operations on a 3D-subTCG are performed on edges, its time complexity is up to $O(n^2)$, where n is the number of nodes/modules. In contrast, the operations on a T-tree are performed on nodes; therefore, the time complexity is only $O(n)$.
- Based on the T-tree representation, we can derive a more efficient packing method than using 3D-subTCG.
- There are only $O(n!3^{3n}/2^{2n}n^{1.5})$ combinations of a T-tree while a 3D-subTCG has $O((n!)^3)$ combinations.

To handle the precedence constraints among tasks, we derive an effective and efficient method to examine the feasibility of a T-tree. The structure of T-tree presents a nice property that enables easy feasibility detection; that is, if node n_j is in the left subtree of node n_i, task v_j must be executed after task v_i. Based on this property, we can perform feasibility detection in $O(h)$ time, where h is the height of a T-tree (note that we treat the number of precedence constraints as a constant). If a T-tree results in an infeasible placement, the T-tree is re-constructed to remove the violation conditions. To reduce the probability that a T-tree results in an infeasible placement after an operation, we filter out a set of operations that will definitely introduce precedence violations. We also derive in this paper the solution space of T-tree and prove the reachability of the solution space. The study provides a solid theoretical foundation for the effectiveness and efficiency of the simulated annealing (SA) based optimization process used in our temporal floorplanner. Experimental results show that our T-tree based simulated annealing (SA) scheme consistently obtains much better results in shorter running time than the 3D-subTCG approach. For a large circuit of 300 tasks and 120 precedence constraints, for example, the T-tree based SA scheme obtains a solution of 13.7% deadspace in less than 1.15 hours, while the 3D-subTCG method needs about 7.51 hours and results in a solution of 34.2% deadspace.

8 summary

Our research team has focused on developing the enabling technology to automate or shorten the design cycle for reconfigurable platforms for media-rich life. Thus far, we have developed the technology for reducing the usage of memory and DRT, avoiding the hardware contention, designing multimedia IPs, and conducting temporal floorplanning/placement. In the following years, the faculty of this team will continue to collaborate and develop the technology for reconfigurable platforms for media-rich life. More information on our latest activities can be found through our web site, `http://www.csie.ntu.edu.tw/~excellence`.

References

[1] C.-L. Yang, H.-W. Tseng, C.-C. Ho, and J.-L. Wu. Software-controlled cache architecture for energy efficiency. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(5), 2005.

[2] C.-L. Yang and C.-H. Lee. Hotspot cache: Joint temporal and spatial locality exploitation for i-cache energy reduction. In *Proceedings of IEEE/ACM Internatioanl Symposium on Low Power Electronics and Design*, 2004.

[3] Y.-J. Chen, C.-L. Yang, and E.-K. Lin. Phase-aware i-cache size synthesis with qos consideration. Technical Report 05-03, National Taiwan University, 2005.

[4] L. Benini, D. Bertozzi, D. Bruni, N. Drago, F. Fummi, and M. Poncino. Systemc co-simulation and emulation of multiprocessor soc designs. *IEEE Computer*, 36(4), 2003.

[5] L. Cai and D. Gajski. Transaction level modeling: An overview. *CODES+ISSS03*, 2003.

[6] D. Brooks, V. Tiwari, and M Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. *ISCA2000*, 2000.

[7] M. Caldari, M. Conti, M. Coppola, P. Crippa, S. Orcioni, L. Pieralisi, and C. Turchetti. System-level power analysis methodology applied to the amba ahb bus. *DATE'03 Designers' Forum*, 3(5), 2003.

[8] H.-C. Fang, C.-T. Huang, Y.-W. Chang, T.-C. Wang, P.-C. Tseng, C.-J. Lian, and L.-G. Chen. 81MS/s JPEG 2000 single-chip encoder with rate-distortion optimization. In *Digests of 2004 IEEE International Solid-State Circuits Conference (ISSCC 2004)*, 2004.

[9] Y.-W. Huang, T.-C. Chen, C.-H. Tsai, C.-Y. Chen, T.-W. Chen, C.-S. Chen, C.-F. Shen, S.-Y. Ma, T.-C. Wang, B.-Y. Hsieh, H.-C. Fang, and L.-G. Chen. A 1.3TOPS H.264/AVC single-chip encoder for HDTV applications. In *Digests of 2005 IEEE International Solid-State Circuits Conference (ISSCC 2005)*, 2005.

[10] S.-Y. Chien, Y.-W. Huang, B.-Y. Hsieh, S.-Y. Ma, and L.-G. Chen. Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques. *IEEE Transactions on Multimedia*, 6(5):732–748, October 2004.

[11] S.-Y. Chien, Y.-W. Huang, and L.-G. Chen. A hardware accelerator for video segmentation using programmable morphology pe array. In *Proceedings of International Symposium on Circuits and Systems 2002 (ISCAS 2002)*, 2002.

[12] Y. Xie and W. Wolf. Co-synthesis with custom ASICs. In *Asia and South Pacific Design Automation Conference*, pages 129–134, 2000.

[13] Y.-S. Chen, L.-P. Chang, T.-W. Kuo, and A. K. Mok. Real-time task scheduling anomaly:observations and prevention. In *The 20th ACM Symposium on Applied Computing*, 2005.

[14] Y.-S. Chen, N.-C. Perng, C.-S. Shih, and T.-W. Kuo. Interference analysis of multiple critical paths. In *Asia and South Pacific International Conference on Embedded SoCs (ASPICES)*, 2005.

[15] N.-C. Perng, J.-J. Chen, T.-W. Kuo, and C.-S. Shih. On the minimization of fpga configuration contexts for given schedules and hardware designs. Under preparation for submission.

[16] L. Y.-X. Lee and M. Ku. Hardware / software interface synthesis for a co-design framework. In *Asia and South Pacific International Conference on Embedded SoCs (ASPICES)*, 2005.

[17] P.-H. Yuh, C.-L. Yang, Y.-W. Chang, and H.-L. Chang. Temporal floorplanning using 3D-subTCG. In *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference*, pages 725 – 730, Yokohama, Japan, January 2004.

[18] P.-H. Yuh, C.-L.Yang, and Y.-W. Chang. Temporal floorplanning using the t-tree formulation. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pages 300 – 305, San Jose, CA, U.S.A., November 2004.