Highly-Cited Ideas in System Codesign and Synthesis

Frank Vahid
Department of Computer Science and Engineering
University of California, Riverside, USA
Also with the Center for Embedded Computer Systems, UC Irvine
vahid@cs.ucr.edu

Tony Givargis
Center for Embedded Computer Systems
University of California, Irvine, USA
givargis@uci.edu

ABSTRACT

We conducted a study of citations of papers published between 1996 and 2006 in the CODES and ISSS conferences, representing the hardware/software codesign and system synthesis community. Citations, meaning non-self-citations only, were considered from all papers known to Google Scholar, as well as only from subsequent CODES/ISSS papers. We list the most-cited CODES/ISSS papers of each year, summarizing their topics, and discussing common features of those papers. For comparison purposes, we also measured citations for the computer architecture community's ISCA and MICRO conferences, and for the field-programmable gate array community's FPGA and FCCM conferences. We point out several interesting differences among the citation patterns of the three communities.

Categories and Subject Descriptors: C.0 [Computer Systems Organization] System architectures, B.0 [Hardware]. **General Terms:** Design, Performance. **Keywords:** Citations, hardware/software codesign, system synthesis.

1. INTRODUCTION

As the hardware/software codesign and system synthesis community's flagship conferences of CODES and ISSS (now merged as CODES/ISSS) proceed into their second decade, we pause to consider the ideas published in the past 10 years that have had significant impact. One measure of impact is the number of non-self-citations. This measure is clearly imperfect, but is objective, and may provide some insights despite its imperfections. Using academic paper search and citation tools recently made conveniently accessible, we thus determined the top-cited papers for each year from 1996 to 2006 of CODES/ISSS. We list those papers and discuss some of their topics and features.

For comparison purposes, we determined top-cited papers for two other communities related in some ways to the codesign/synthesis community. One community is the architecture community, whose flagship conferences are *ISCA* and *MICRO*. The other is the field-programmable gate array community, whose flagship conferences include *FPGA* and *FCCM*. Clearly, none of the three communities are solely defined by the above-listed conferences, as each community includes several other conferences, large and small. Nevertheless, comparing among the three communities as defined by the above-listed conferences provides a window into some interesting differences among the three communities, which we also discuss.

CODES+ISSS'08, October 19–24, 2008, Atlanta, Georgia, USA.

2. METHODOLOGY

For the CODES and ISSS conferences (which merged in 2003 into CODES/ISSS), we obtained tables of contents of the proceedings from 1996 to present using online ACM resources. We entered the titles from those proceedings into Google Scholar (scholar.google.com), which not only returns a link to the paper's online version, but also returns a link listing all other papers (known to Google Scholar) that cite that paper. Because self-citations are of less interest with respect to measuring impact, we processed the citing papers list further by removing all papers in the list authored by an author of the cited paper. Our studies show that self-citations typically accounted for 10% to 25% of a paper's citations. From this point, we use the term "citations" to refer exclusively to non-self-citations. The process was conducted in early 2008, representing therefore a snapshot of citations from that time.

The above process (like all automated citation methods) is not exact [1]. One reason is because not all conferences or papers are in Google Scholar's database. A second is because citing papers sometimes use an incorrect or mistyped title (Google Scholar appears to attempt to correct for such mistakes, but cannot correct for all such mistakes). A third reason is because two people may have the same name (in particular, the same first initials and last name), leading to false self-citation matches. Likewise, misspelled names in citations may also occur. A fourth reason is because citations to identically titled items (e.g., a conference paper and then journal paper with the same title and authors) seem to be grouped in Google Scholar. Our process was partially automated via the use of scripts, while certain steps required manual processing. We manually spot-checked results. While not exact, the above process is desirable over most existing impact-measuring systems that only consider a small subset of publications when counting citations (e.g., only considering journal publications), and appeared to be more comprehensive than using Citeseer (e.g., for a particular ISCA 2000 paper, Citeseer listed 224 citations, while Google Scholar listed 939). Despite the process' inexactness, the obtained citation numbers seem to still be useful to generally sort the conference papers such that highly-cited papers can be found.

The above process provides total citations to a given paper. We were also interested in the number of citations from the same community, as defined by CODES/ISSS. Thus, for each paper, we created another list containing only (non-self-)citing papers from CODES or ISSS. For example, for a paper published in CODES 2000, any citing paper published in *either* CODES 2000-2006 or

ISSS 2000-2006 (actually, the two conferences merged into CODES/ISSS in 2003), would be considered a citation from the same community.

The above process was repeated for three different years (2000, 2003, and 2006) for the architecture community as defined by MICRO/ISCA, and for the FPGA community as defined by FPGA/FCCM. Again, note that for a paper published for example in MICRO 2000, any citation from a paper in MICRO or ISCA 2000-2006 would count as a citation from within the community.

3. HIGHLY CITED CODES/ISSS PAPERS

Table 1 lists titles of the top-four most cited papers published in each year of CODES/ISSS from 1996 to 2006.

Several themes can be observed. An earlier theme seems to have been system-level (or multi-processor) partitioning and synthesis, including 96b (Year 1996, ID b), 96c, 97b, 97c, 97d, 98a, 98b, 00b, 01d, and 02d (and 06a). A related theme is scheduling, the focus of 99a, 00c, 02b, 03a. A theme that seems to have evolved through the entire time period is communication, with interface and bus papers 96a, 96d, then 98b, 02c, 03d, 04b, and continuing to network-on-chip papers 05a, 05b, and 06b. Memory issues have attracted some attention, with synthesis in 97a, scratchpad memory in 02a and 04a, and cache in 03c. The emphasis on low-power starting in the late 1990s can be seen with

Table 1: Highly-cited CODES/ISSS papers from 1996 to 2006. The first number is all (non-self) citations. The second is the subset of those that came from other CODES/ISSS papers ("Comm," for "Community").

Year	ID	Title	All	Comm
1996	a	Bus-Based Communication Synthesis on System-Level	98	13
	b	Process Partitioning for Distributed Embedded Systems	78	17
	c	PACE: A Dynamic Programming Algorithm for Hardware/Software Partitioning	54	17
	d	Grammar-based Hardware Synthesis of Data Communication Protocols	45	5
1997	a	Architectural Exploration and Optimization of Local Memory in Embedded Systems	40	7
	b	Critical Path Driven Cosynthesis for Heterogeneous Target Architectures	37	3
	c	An Evolutionary Approach to System-Level Synthesis	36	3
	d	Embedded System Synthesis by Timing Constraints Solving	35	3
1998	a	TGFF Task Graphs for Free	194	7
	b	Integrating Communication Protocol Selection with Partitioning in Hardware/Software Codesign	93	3
	c	Software Timing Analysis Using HW/SW Cosimulation and Instruction Set Simulator	56	4
	d	A Processor Description Language Supporting Retargetable Multi-Pipeline DSP Program Development Tools	49	3
1999	a	Real-Time Task Scheduling for a Variable Voltage Processor	64	10
	b	Automatic Architectural Synthesis of VLIW and EPIC Processors	55	9
	c	Optimized Rapid Prototyping for Real-Time Embedded Heterogeneous Multiprocessors	53	1
	d	Compiling Esterel into Sequential Code	45	4
2000	a	Compiler Optimization on Instruction Scheduling for Low Power	50	10
	b	Compaan: Deriving Process Networks from Matlab for Embedded Signal Processing Architectures	44	3
	c	Low-Power Task Scheduling for Multiple Devices	42	0
	d	Heterogeneous Modeling and Simulation of Embedded Systems in El Greco	40	7
2001	a	Designing Domain-Specific Processors	70	5
	b	SystemC: A Modeling Platform Supporting Multiple Design Abstractions	58	0
	c	Embedded UML: a Merger of Real-Time UML and Co-Design	49	7
	d	Hardware / Software Partitioning of Embedded System in OCAPI-xl	45	0
2002	a	Scratchpad Memory : A Design Alternative for Cache On-chip Memory in Embedded Systems	108	0
	b	Holistic Scheduling and Analysis of Mixed Time/Event-Triggered Distributed Embedded Systems	54	8
	c	An Adaptive Low-Power Transmission Scheme for On-Chip Networks	52	5
	d	Multi-Objective Design Space Exploration Using Genetic Algorithms	45	5
2003	a	Pareto Optimization Based Run-time Task Scheduling for Embedded Systems	29	0
	b	Hardware Support for Real-time Operating Systems	28	1
	c	Accurate Estimation of Cache-Related Preemption Delay	24	2
	d	A Modular Simulation Framework for Architectural Exploration of On-Chip Interconnection Networks	19	1
2004	a	Dynamic Overlay of Scratchpad Memory for Energy Minimization	35	5
	b	Design and Programming of Embedded Multiprocessors: An Interface-Centric Approach	20	1
	c	Transaction Level Modeling: Flows and Use Models	20	1
	d	Parallel Programming Models for a Multi-Processor SoC Platform Applied to High-Speed Traffic Management	18	1
2005	a	A unified approach to constrained mapping and routing on network-on-chip architectures	26	3
	b	Key research problems in NoC design: a holistic perspective	22	5
	c	An automated exploration framework for FPGA-based soft multiprocessor systems	21	0
	d	An integer linear programming approach for identifying instruction-set extensions	12	1
2006	a	Multi-processor system design with ESPAM	10	0
	b	A buffer-sizing algorithm for networks on chip using TDMA and credit-based end-to-end flow control	7	0
	c	Challenges in exploitation of loop parallelism in embedded applications	6	0
	d	Efficient computation of buffer capacities for multi-rate real-time systems with back-pressure	6	0

99a, 00a, 00c, 02a, 02c, and 04a. Application-specific instruction-set processors were addressed by 99b, 01a, and 05d. System-level modeling was addressed by 00d, 01b, 01c, 04c, and 04d, and simulation by 98c, 00d, and 03d. Note that the above themes do not necessarily describe popular topics covered by CODES/ISSS during those years, but rather they describe the papers that have later proven popular (as measured by citations) during those years.

We briefly summarize the top six most highly-cited papers appearing in CODES/ISSS from 1996-2006. This summary does not take into account the number of years since publication, and thus older papers have an advantage.

TGFF: Task Graphs for Free [9] – The most-cited paper is 98a, with 194 citations. The paper describes a tool, TGFF, for generating synthetic task graphs that can be used to model applications being input to a system synthesis or scheduling tool. The tool's synthetic task graphs have been used by numerous subsequent synthesis and scheduling approaches, accounting for many of the citations.

Scratchpad Memory: A Design Alternative for Cache On-chip Memory in Embedded Systems [5]. The next most-cited paper is 02a, with 108 citations. This number is especially interesting due to the paper being recent (2002), because there has been less time for citations to occur. The paper discussed scratchpad memory as a means of reducing energy. By using a compiler to insert instructions that move data to and from an on-chip SRAM, rather than relying solely on cache, the simpler memory design and control yields reduced energy. Of note is that all 108 citations appear to come from outside CODES/ISSS.

Bus-Based Communication Synthesis on System-Level [12] – This paper, 96a, received 98 citations, and is one of the top-three most cited by the CODES/ISSS community with 13 citations. The paper presents an algorithm for generation of low cost communication topologies for statically scheduled systems. Given a set of processes communicating via abstract send and receive methods and detailed information on communication requirements of each process, the authors' approach performs a clustering of data transfers and, for each cluster, the approach executes a bus generation algorithm that schedules bus accesses while minimizing total communication cost.

Integrating Communication Protocol Selection with Partitioning in Hardware/Software Codesign [19] – Paper 98b, with 93 citations, presents a codesign approach that incorporates communication protocol selection as a design parameter within hardware/software partitioning. The approach takes into account data transfer rates depending on communication protocol types and configurations, and different operating frequencies of system components, i.e. CPUs, ASICs, and buses. It also takes into account the timing and area influences of drivers and driver calls needed to perform the communication.

Process Partitioning for Distributed Embedded Systems [15] – Paper 96b, with 78 citations, presents a technique for partitioning processes in distributed embedded systems. The partitioning heuristic minimizes both context switch and communication overhead under real-time deadline and process size constraints, while allocating functions to processors that are well suited to those functions.

Designing Domain-Specific Processors [3] – Paper 01a, having 70 citations, presents a semi-automated method for the detection and exploitation of application domain specific instruction set extensions for embedded (VLIW) processors. It consists of three steps. The first step detects frequently occurring operation patterns. In the second step, the patterns are grouped and implemented in a number of Special Function Units. The third step incorporates the custom operations into the code generation process.

One might attribute some of the impact of the above papers as being due to the following. 98a provided a useful mechanism for generating task graphs needed to validate synthesis/scheduling approaches. It thus was in the category of providing a framework to support basic system codesign and synthesis research (another framework-type paper is the SystemC paper of 01b). 96b defined a core codesign / system synthesis problem in a fairly comprehensive manner. 96a and 98b emphasized a rather new problem involving communication synthesis, which continues today as a unique issue in system synthesis. 02a described an architecture/compiler technique (compiler-controlled scratchpad memory) that enabled substantial energy reduction in the energy-hungry memory sub-component of a system and that encouraged much subsequent work in compilation and architectures that further improved on the technique. 01a described a new direction in architecture design, application-specific instruction-set processors, yielding performance and energy improvements, and also leading to a large number of new problems and thus subsequent work.

We compared the most cited papers of each year with the least cited papers[1], and observed that the most cited papers tended to address topics more generally, while the least cited papers tended to focus on highly-specific subjects, or on subjects not central to codesign/synthesis. As an informal test of this observation, we enlisted 5 community members to attempt to rank 6 papers from each of 4 randomly selected years. 3 of those papers were the top-3 cited papers of the given year, and 3 were the least-cited, combined in random order. Each participant was asked to rank each paper 1 (likely most cited) to 6 (likely least cited), based solely on the paper title alone. We summed the ranks and sorted the papers. For each year, the summed ranking of the top two were correct; the third was correct half the time.

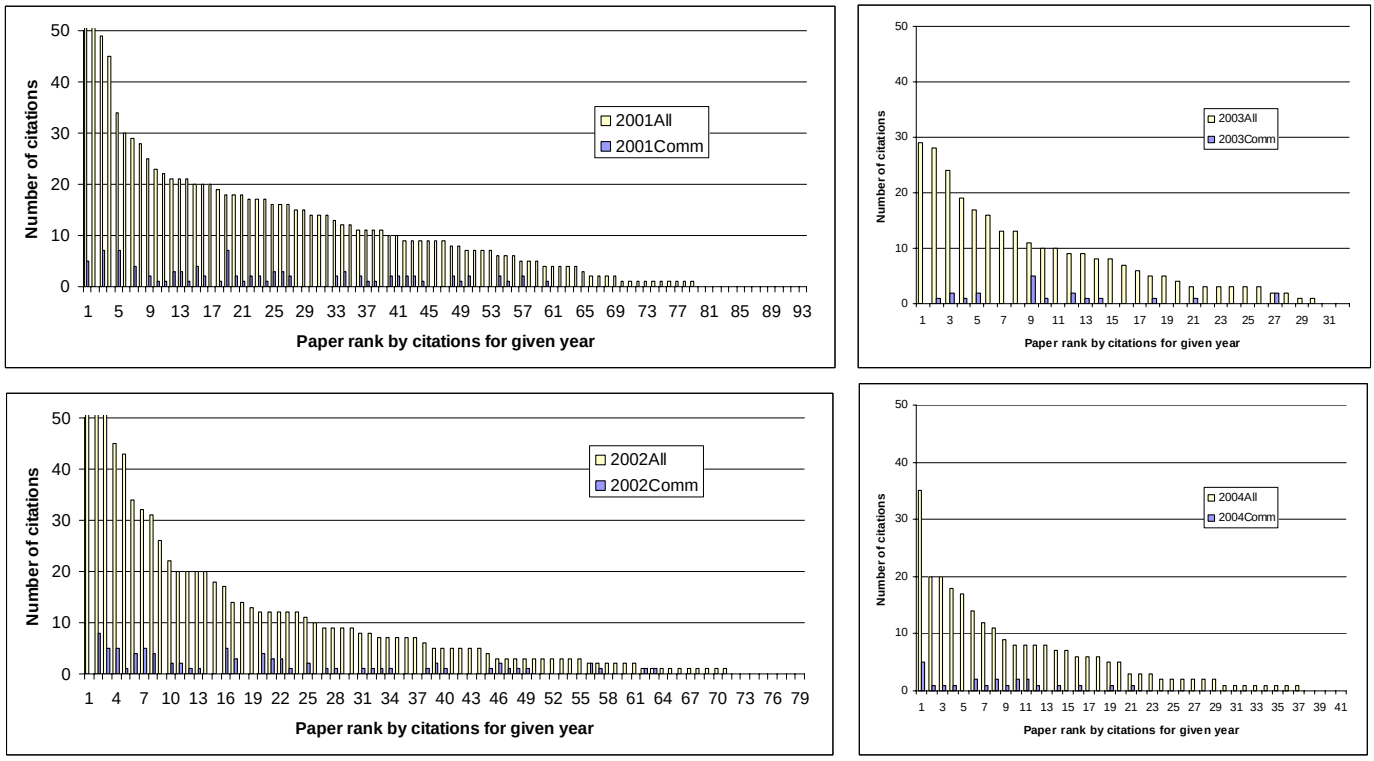
4. CITATION PATTERNS

Figure 1 shows citations per CODES/ISSS paper for the years 2001-2004, thus providing plots for two years before the merge and two years after the merge. The plots include all citations ("All"), and citations from within the CODES/ISSS community ("Comm"). The number on the x-axis represents the rank of a particular paper for the given year based on all citations.

The plots of all citations generally exhibit a shape to be expected, where some papers are highly-cited, tapering off approximately linearly to lesser-cited papers. However, two features draw attention. The first feature is the tendency for 3-5 papers per year to get many more citations than remaining papers. Thus, the top-4 papers per year in Table 1 tend to have many more citations than the next 4 papers per year. A second interesting feature is the large number of papers, on the right side of each plot, that are never cited or cited only a few times. A third feature is the relatively small number of citations from within the

[1] We do not list the least-cited papers here, for reasons of propriety. However, complete citation data is available for a limited time after initial publication of this paper via the web at http://www.cs.ucr.edu/~vahid.

Figure 1: Citations of 2001-2004 CODES/ISSS papers, showing data for all citations ("All"), and for citations from within the CODES/ISSS community ("Comm"). Note that some of the highest-cited papers extend off the top of the chart – see Table 1 for their citation counts.



CODES/ISSS community. This feature led us to compare with other communities.

Figure 2 shows citations for two other communities – FPGA/FCCM, and MICRO/ISCA, for which we obtained data for years 2000, 2003, and 2006. For space reasons, the figure shows only the year 2003. Comparing the 2003 CODES/ISSS plots in Figure 1 with the 2003 FPGA/FCCM plots in Figure 2(a), we note that the total citations per paper seem roughly equivalent. However, the FPGA/FCCM community seems to cite their own top papers at a higher rate than CODES/ISSS. Comparing instead with the 2003 MICRO/ISCA plots in Figure 2(b), we note several significant differences. First, the total citations per paper are much higher (the first 14 papers have citations between 50 and 116, which is beyond the plot's maximum y value). Second, the lowest-ranked papers still have between 5-10 references. Third, the citations from within the community is significantly higher.

The data for 2000 and 2006 showed similar comparative trends relative to CODES/ISSS. Of particular note though was a MICRO/ISCA paper from 2000 that had 939 citations, with 123 from within the community. That paper was entitled: "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations." Like the most highly-cited CODES/ISSS paper, that paper described a framework that enabled extensive subsequent research by others in the community.

The citations from within the community are of particular interest, and thus we examined this feature further. In particular,

Figure 2: Citations of year 2003 papers for: (a) FPGA/FCCM community, and (b) MICRO/ISCA community.

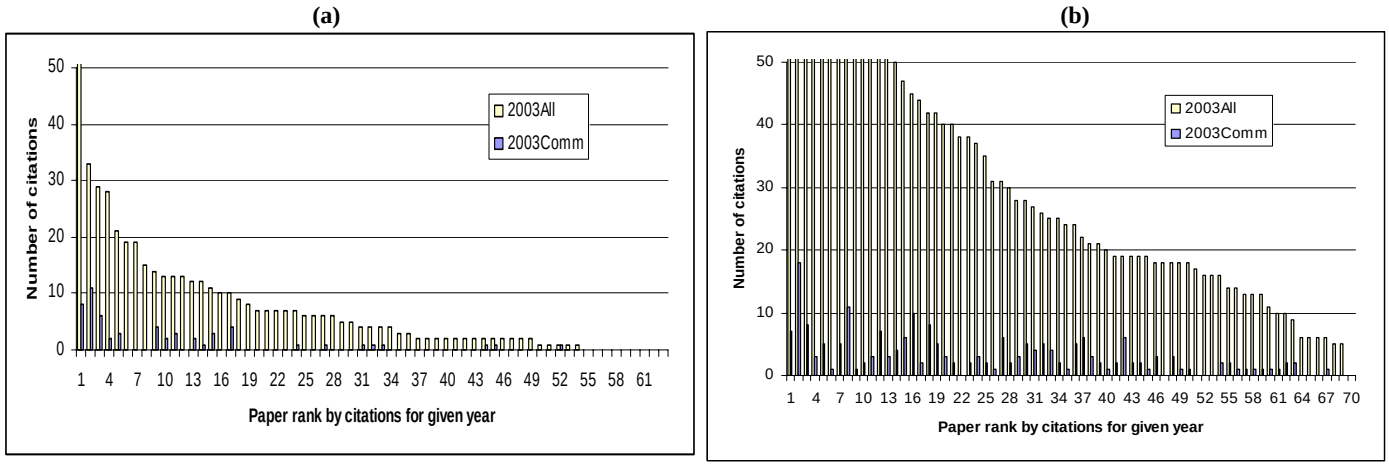
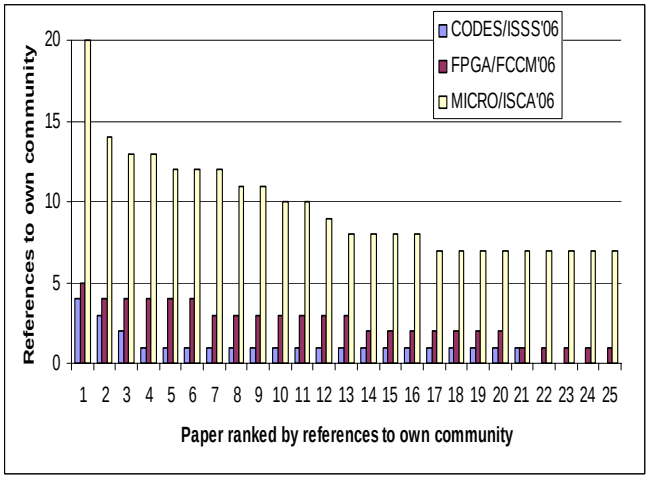


Figure 3: Non-self-references by 2006 papers to previous papers within the same community. Only the 25 top-community-referencing papers are shown for each community.



we looked at the situation from the reverse perspective. We examined all 2006 papers published in the three communities, and looked at the list of references cited by those papers. We counted the number of references that referred to papers within the same community (CODES/ISSS, FPGA/FCCM, or MICRO/ISCA). Figure 3 shows the resulting plots for the three communities, for the 25 papers having the most intra-community references for each community. The plots show that MICRO/ISCA papers reference other MICRO/ISCA papers at a much higher rate than the other two communities. The plots also show that FPGA/FCCM papers reference within the community at a higher rate than CODES/ISSS. The percentage of references may also be of interest: For all the papers of each community (not just the top 25), 18% of all MICRO/ISCA references were to papers within the community; the FPGA/FCCM rate was 6%, and the CODES/ISSS rate was 3%.

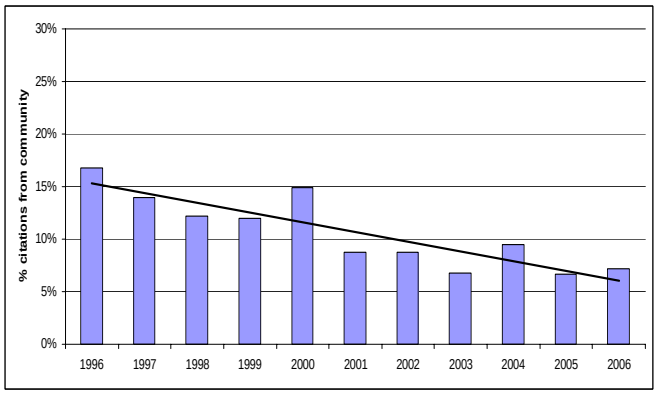
We also examined the percentage of citations from within the community versus all citations, for each year since 1996, for the top-25 cited CODES/ISSS papers in each year (data for remaining papers did not differ significantly, as the remaining papers have few citations). Figure 4 shows results. In 1996, 17% of all citations to CODES/ISSS papers were from subsequent CODES/ISSS papers. In 2006, the rate was 7%.

5. CONCLUSIONS

Examining the most highly-cited papers from CODES/ISSS can provide insight as to the most significant or influential system codesign and synthesis ideas contributed by the community. The data provided in this paper may prove useful to new researchers in the field attempting to form ideas for high-impact contributions, and to form the basis of paper selection in graduate seminars. It may also prove useful to the community in forming strategic goals for future years.

Furthermore, this paper examined citation and referencing patterns, with some comparison to two related communities. The data showed some differences among communities, and some trends. Conclusions from such data are hard to immediately draw, and instead the data may prove useful for discussion by the community. The differences among communities could suggest that the CODES/ISSS community may seek to define certain foci,

Figure 4: Percentage of all citations coming from within the CODES/ISSS community.



or to develop common benchmarks or frameworks to encourage comparative work. Alternatively, the differences may reflect a broader range of strong forums, or the diverse nature of the field. Further discussion and analysis could prove fruitful. Other measures of impact, such as commercialization, are also important to consider.

6. REFERENCES

[1] Google Scholar from Wikipedia. http://en.wikipedia.org/wiki/Google_Scholar.

[2] S. Aditya, B. Ramakrishna Rau, V. Kathail. Automatic architectural synthesis of VLIW and EPIC processors. Int. Symp. on System Synthesis (ISSS), 1999, p. 107.

[3] M. Arnold , H. Corporaal. Designing domain-specific processors. Int. Symp. on Hardware/Software Co-Design (CODES), 2001, pp. 61-66.

[4] K. Atasu, G. Dündar, C. Özturan. An integer linear programming approach for identifying instruction-set extensions. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2005, pp. 172-177.

[5] R. Banakar, S. Steinke, B. Lee, M. Balakrishnan, P. Marwedel. Scratchpad memory: design alternative for cache on-chip memory in embedded systems. Int. Symp. on Hardware/Software Co-Design (CODES), 2002, pp. 73-78.

[6] P. Bjørn-Jørgensen and J. Madsen. Critical path driven cosynthesis for heterogeneous target architectures. Int.. Workshop on Hardware/Software Codesign (CODES/CASHE), 1997, pp. 15–19.

[7] J. Buck, R. Vaidyanathan. Heterogeneous modeling and simulation of embedded systems in El Greco. Int. Workshop on Hardware/Software Co-Design (CODES), 2000, pp. 142-146.

[8] M. Coenen, S. Murali, A. Ruadulescu, K. Goossens, G. De Micheli. A buffer-sizing algorithm for networks on chip using TDMA and credit-based end-to-end flow control. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2006, pp. 130-135.

[9] R.P. Dick and W. Wolf. TGFF: Task graphs for free. Int.. Workshop on Hardware/Software Codesign (CODES/CASHE), 1998, pp. 97-101.

[10] A. Donlin. Transaction level modeling: flows and use models. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2004, pp. 75-80.

[11] S.A. Edwards. Compiling Esterel into sequential code. Int. Workshop on Hardware/Software Co-Design (CODES/CASHE), California, 1999, pp. 147-151.

[12] M. Gasteier and M. Glesner. Bus-based communication synthesis on system-level. Int. Symp. on System Synthesis (ISSS), 1996, pp. 65-70.

[13] T. Grandpierre, C. Lavarnenne, Y. Sorel. Optimized rapid prototyping for real-time embedded heterogeneous multiprocessors. Int. Workshop on Hardware/Software Co-Design (CODES/CASHE), 1999, pp. 74-78.

[14] A. Hansson, K. Goossens, A. Rădulescu. A unified approach to constrained mapping and routing on network-on-chip architectures. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2005, pp. 75-80.

[15] J. Hou and W. Wolf. Process partitioning for distributed embedded systems. Int. Workshop on Hardware/Software Co-Design (CODES/CASHE), 1996, pp. 70-76.

[16] Y. Jin, N. Satish, K. Ravindran, K. Keutzer. An automated exploration framework for FPGA-based soft multiprocessor systems. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2005, pp. 273-278.

[17] A. Kejariwal, A.V. Veidenbaum, A. Nicolau, M. Girkarmark, X. Tian, H. Saito. Challenges in exploitation of loop parallelism in embedded applications. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2006, pp. 173-180.

[18] B. Kienhuis, E. Rijpkema, E. Deprettere. Compaan: deriving process networks from Matlab for embedded signal processing architectures. Int. Workshop on Hardware/Software Co-Design (CODES), 2000, pp. 13-17.

[19] P.V. Knudsen and J. Madsen. Integrating communication protocol selection with partitioning in Hardware/Software Co-Design. Int. Symp. on System Synthesis (ISSS), 1998, pp. 111-116.

[20] P.V. Knudsen and J. Madsen. PACE: A dynamic programming algorithm for hardware/software partitioning. Int. Workshop on Hardware/Software Co-Design (CODES/CASHE), 1996, pp. 85-92.

[21] T. Kogel, M. Doerper, A. Wieferink, R. Leupers, G. Ascheid, H. Meyr, S. Goossens. A modular simulation framework for architectural exploration of on-chip interconnection networks. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2003, pp. 7-12.

[22] P. Kohout, B. Ganesh, B. Jacob. Hardware support for real-time operating systems. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2003, pp. 45-51.

[23] K. Kuchcinski. Embedded system synthesis by timing constraints solving. Int. Symp. on System Synthesis (ISSS), 1997, pp. 50.

[24] C. Lee, J. Kuen Lee, T. Hwang, S. Tsai. Compiler optimization on instruction scheduling for low power. Int. Symp. on System Synthesis (ISSS), 2000, pp. 55-60.

[25] R. Leupers, F. David. A uniform optimization technique for offset assignment problems. Int. Symp on System Synthesis (ISSS), 1998, p 3.

[26] J. Liu, M. Lajolo, A. Sangiovanni-Vincentelli. Software timing analysis using HW/SW cosimulation and instruction set simulator. Int. Workshop on Hardware/Software Co-Design (CODES/CASHE), 1998, pp. 65-69.

[27] Y. Lu, L. Benini, G. De Micheli. Low-power task scheduling for multiple devices. Int. Workshop on Hardware/Software Co-Design (CODES), 2000, pp. 39-43.

[28] G. Martin, L. Lavagno, J. Louis-Guerin. Embedded UML: A Merger of real-time UML and co-design. Int. Symp. on Hardware/Software Co-Design (CODES), 2001, pp. 23-28.

[29] H.S. Negi, T. Mitra, A. Roychoudhury. Accurate estimation of cache-related preemption delay. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2003, pp. 2-12.

[30] H. Nikolov, T. Stefanov, E. Deprettere. Multi-processor system design with ESPAM. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2006, pp. 211-216.

[31] J. Oeberg, A. Kumar and Ahmed Royal. Grammar-based hardware synthesis of data communication protocols. Int. Symp. on System Synthesis, 1996, pp. 14-19.

[32] U.Y. Ogras, J. Hu, R. Marculescu. Key research problems in NoC design: a holistic perspective. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2005, pp. 69-74.

[33] T. Okuma, T. Ishihara, H. Yasuura. Real-time task scheduling for a variable-voltage processor. Int. Symp. on System Synthesis (ISSS), 1999, pp. 25-29.

[34] M. Palesi, T. Givargis. Multi-objective design space exploration using genetic algorithms. Int. Symp. on Hardware/Software Co-Design (CODES), 2002, pp. 67-72.

[35] P. R. Panda. SystemC: A modeling platform supporting multiple design abstractions. Int. Symp. on System Synthesis (ISSS), 2001, pp. 75-80.

[36] P.R. Panda, N.D. Dutt, and A. Nicolau. Architectural exploration and optimization of local memory in embedded systems. Int. Symp. on System Synthesis (ISSS), 1997, pp. 90-97.

[37] P.G. Paulin, C. Pilkington, M. Langevin, E. Bensoudane, G. Nicolescu. Parallel programming models for a multi-processor SoC platform applied to high-speed traffic management. Int. Conference on Hardware/Software Co-Design and System Synthesis, 2004, pp. 48-53.

[38] T. Pop, P. Eles, Z. Peng. Holistic scheduling and analysis of mixed time/event-triggered distributed embedded systems. Int. Symp. on Hardware/Software Co-Design (CODES), 2002, pp. 187-192.

[39] C. Siska. A processor description language supporting retargetable mult-pipeline DSP program development tools. Int. Symp on System Synthesis (ISSS), 1998, pp. 31-36.

[40] J. Teich, T. Blickle, L. Thiele. An evolutionary approach to system-level synthesis. Int. Workshop on Hardware/Software Co-Design (CODES/CASHE), 1997, pp. 167-171.

[41] G. Vanmeerbeeck, P. Schaumont, S. Vernalde, M. Engels, I. Bolsens. Hardware/software partitioning of embedded system in OCAPI-xl. Int. Symp. on Hardware/software Co-Design (CODES), 2001, pp. 30-35.

[42] M. Verma, L. Wehmeyer, P. Marwedel. Dynamic overlay of scratchpad memory for energy minimization. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2004, pp. 104-109.

[43] M. Wiggers, M. Bekooij, P. Jansen, G. Smit. Efficient computation of buffer capacities for multi-rate real-time systems with back-pressure. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2006, pp. 10-15.

[44] P. Wolf, E. Kock, T. Henriksson, W. Kruijtzer, G. Essink. Design and programming of embedded multiprocessors: an interface-centric approach. Int. Conference on Hardware/Software Co-Design and System Synthesis (CODES/ISSS), 2004, pp. 206-217.

[45] F. Worm, P. Ieene, P. Thiran, G. De Micheli. An adaptive low-power transmission scheme for on-chip networks. Int. Symp. on System Synthesis (ISSS), 2002, pp. 92-100.

[46] P. Yang, F. Catthoor. Pareto-optimization-based run-time task scheduling for embedded systems. Int. Conference on Hardware/Software Codesign and System Synthesis (CODES/ISSS), 2003, pp. 120-125.