

References

- [1] Scott M. Nettles and James W. O'Toole. Real-time replication-based garbage collection. In Proceedings of SIGPLAN'93 Conference on Programming Languages Design and Implementation, volume 28(6) of ACM SIGPLAN Notices, Carnegie Mellon University, USA, June 1993, ACM Press.
- [2] José A. Joao, Onur Mutlu, Yale N. Patt; "Flexible Reference-Counting-Based Hardware Acceleration for Garbage Collection". *ISCA '09*, Austin, Texas, USA June 20–24, 2009
- [3] D. F. Bacon, P. Cheng, and V. T. Rajan. "A unified theory of garbage collection". In *ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*, 2004
- [4] Dan Grossman; "The Transactional Memory / Garbage Collection Analogy". OOPSLA'07, Montréal, Québec, Canada, October 21–25, 2007
- [5] J. McCarthy. History of LISP. In *History of programming languages I*, pages 173–185. ACM, 1981
- [6] K. Arnold and J. Gosling. *The Java Programming Language*. Addison Wesley, 1996
- [7] D. Syme, A. Granicz, and A. Cisternino. *Expert F#*. Apress, 2007

- [8] G. E. Collins. A method for overlapping and erasure of lists. *Commun. ACM*, 3(12):655–657, 1960
- [9] J. McCarthy. Recursive functions of symbolic expressions and their computation by machine, part I. *Commun. ACM*, 3(4):184–195, 1960
- [10] S. M. Blackburn, P. Cheng, and K. S. McKinley. Myths and realities: the performance impact of garbage collection. *SIGMETRICS Perform. Eval. Rev.*, 32(1):25–36, 2004
- [11] D. Ungar. Generation scavenging: A non-disruptive high performance storage reclamation algorithm. *SIGPLAN Not.*, 19(5):157–167, 1984
- [12] A. W. Appel. Simple generational garbage collection and fast allocation. *SPE*, 19(2):171–183, 1989
- [13] W. Chen, S. Bhansali, T. Chilimbi, X. Gao, and W. Chuang. Profile-guided proactive garbage collection for locality optimization. In *PLDI'06*
- [14] L. P. Deutsch and D. G. Bobrow. An efficient, incremental, automatic garbage collector. *Communications of the ACM*, 19(9):522–526, 1976
- [15] Y. Levanoni and E. Petrank. An on-the-fly reference-counting collector for Java. In *OOPSLA'01*
- [16] S. M. Blackburn and K. S. McKinley. Ulterior reference counting: fast garbage collection without a long wait. In *OOPSLA'03*
- [17] W. J. Schmidt and K. D. Nilsen. Performance of a hardware-assisted real-time garbage collector. In *ASPLOS'94*
- [18] D. S. Wise, B. C. Heck, C. Hess, W. Hunt, and E. Ost. Research demonstration of a hardware reference-counting heap. *Lisp and Symbolic Computation*, 10(2):159–181, 1997

- [19] M. Meyer. Novel processor architecture with exact tag-free pointers. *IEEE Micro*, 24(3):46–55, 2004
- [20] M. Meyer. An on-chip garbage collection coprocessor for embedded real-time systems. *RTCSA*, 00:517–524, 2005
- [21] M. Meyer. A true hardware read barrier. In *ISMM'06*
- [22] S. Stanchina and M. Meyer. Mark-sweep or copying? A “best of both worlds” algorithm and a hardware-supported real-time implementation. In *ISMM'07*
- [23] R. Jones and R. Lins. *Garbage Collection: Algorithms for Automatic Dynamic Memory Management*. Wiley, 1996
- [24] X. Huang, S. M. Blackburn, K. S. McKinley, J. E. B. Moss, Z. Wang, and P. Cheng. The garbage collection advantage: improving program locality. In *OOPSLA '04*
- [25] S. Z. Guyer and K. S. McKinley. Finding your cronies: static analysis for dynamic object colocation. In *OOPSLA '04*
- [26] G. M. Amdahl. Validity of the single-processor approach to achieving large-scale computing capabilities. In *AFIPS Conference Proceedings*, pages 483–485, 1967
- [27] C. Click, G. Tene, and M. Wolf. The pauseless GC algorithm. In *VEE'05*.
- [28] C. Peng and G. S. Sohi. Cache memory design considerations to support languages with dynamic heap allocation. Technical Report 860, CS Dept., University of Wisconsin-Madison, 1989
- [29] J. M. Chang and E. F. Gehringer. Evaluation of an object-caching coprocessor design for object-oriented systems. In *ICCD'93*

- [30] P. G. Joisha. Compiler optimizations for nondeferred reference-counting garbage collection. In *ISMM'06*
- [31] P. G. Joisha. Overlooking roots: a framework for making nondeferred reference-counting garbage collection fast. In *ISMM'07*
- [32] J.-D. Choi, M. Gupta, M. Serrano, V. C. Sreedhar, and S. Midkiff. Escape analysis for Java. In *OOPSLA'99*
- [33] S. Z. Guyer, K. S. McKinley, and D. Frampton. Free-Me: a static analysis for automatic individual object reclamation. In *PLDI'06*
- [34] H. G. Baker. List Processing in Real Time on a Serial Computer. *Communications of the ACM*, Vol. 21, No. 4, April, 1978
- [35] R. A. Brooks. Trading Data Space for Reduced Time and Code Space in Real-Time Garbage Collection on Stock Hardware. In *Proceedings of the 1984 Symposium on Lisp and Functional Programming*. Austin, Texas, August, 1984
- [36] M. Bengtsson. *Real-Time Garbage Collection*. Licentiate thesis, Dept. of Computer Science, Lund University, 1990
- [37] R. Virding. A Garbage Collector for the Concurrent Real-Time Language Erlang. In *Proceedings of IWMM'95*, Springer-Verlag, LNCS 986, Kinross, Scotland, September, 1995
- [38] A. W. Appel, J. R. Ellis, K. Li. Real-Time Concurrent Collection on Stock Multiprocessors. In *Proceedings of the ACM SIGPLAN'88 Conference on Programming Language Design and Implementation*, Atlanta, Georgia, 1988
- [39] H. G. Baker. The Treadmill: Real-Time Garbage Collection without Motion Sickness. *ACM SIGPLAN Notices*, Vol. 27, No. 3, March 1992

- [40] P. R. Wilson, M. S. Johnstone. *Real-Time Non-Copying Garbage Collection*. OOPSLA'93
- [41] S. Nettles, J. O'Toole. Real-Time Replication Garbage Collection. In *Proceedings of SIGPLAN'93 Conference on Programming Languages Design and Implementation, ACM Sigplan Notices*, Vol. 28, No. 6, June, 1993
- [42] K. D. Nilsen, W. J. Schmidt. A High-Performance Hardware Assisted Real Time Garbage Collection System. *Journal of Programming Languages*, Vol. 2, No. 1, March, 1994
- [43] G. L. Steele Jr. Multiprocessing Compactifying Garbage Collection. *Communications of the ACM*, Vol. 18, No. 9, September, 1975
- [44] E. W. Dijkstra, L. Lamport, A. J. Martin, C. S. Scholten, E. F. M. Steffens. On-the-Fly Garbage Collection: An Exercise in Cooperation. *Communications of the ACM*, Vol. 21, No. 11, November, 1978
- [45] A. W. Appel, J. R. Ellis, K. Li. Real-Time Concurrent Collection on Stock Multiprocessors. In *Proceedings of the ACM SIGPLAN'88 Conference on Programming Language Design and Implementation*, Atlanta, Georgia, 1988
- [46] R. E. Johnson. Reducing the Latency of a Real-Time Garbage Collector. *ACM Letters on Programming Languages and Systems*, Vol. 1, No. 1, March, 1992
- [47] H-J. Boehm, A. J. Demers, S. Shenker. Mostly Parallel Garbage Collection. In *Proceedings of the ACM SIGPLAN'91 Conference on Programming Language Design and Implementation*. Toronto, Canada, June, 1991

- [48] Olukotun, K. and Hammond, L. 2005. "The future of microprocessors". Queue 3, 7 (Sep. 2005), 26-29
- [49] Tensilica Inc. <http://www.tensilica.com>. 2001
- [50] Paul R. Wilson. Uniprocessor garbage collection techniques. Technical report, University of Texas, January 1994. Expanded version of the IWMM92 paper.
- [51] Shinichi Furusou, Satoshi Matsuoka, and Akinori Yonezawa. Parallel conservative garbage collection with fast allocation. In Paul R. Wilson and Barry Hayes, editors, *OOPSLA/ECOOP'91 Workshop on Garbage Collection in Object-Oriented Systems*, 1991
- [52] Stephen Swoyer, Impressive Growth Ahead for Application Server Market, On-line article, <http://www.adtmag.com/article.aspx?id=19970> (JAN.'07)
- [53] D. Sholler, .NET Seen Gaining Steam in DEV Projects, On-line article, <http://techupdate.zdnet.com> (April 2002)
- [54] Sun Microsystems, Five Reasons to Move to the J2SE 5 Platform, <http://java.sun.com/developer/technicalArticles/J2SE/5reasons.html> (2005)
- [55] M. Welsh, D. E. Culler, E. A. Brewer, SEDA: An Architecture for Well-Conditioned, Scalable Internet Services, in: Proceedings of the ACM Symposium on Operating Systems Principles (SOSP), Chateau Lake Louise, Banff, Canada, 2001, pp. 230–243.
- [56] Netcraft, Video iPod Launch Slows Apple Store, On-line article, http://news.netcraft.com/archives/2005/10/12/video_iPod_launch_slows_apple_store.html (2005)

- [57] Chosun Ilbo, Cyber Crime Behind College Application Server Crash, Onlinearticle,<http://english.chosun.com/w21data/html/news/200602/200602100025.html> (2006)
- [58] S. M. Blackburn, R. E. Jones, K. S. McKinley, J. E. . . B. Moss, Beltway: Getting Around Garbage Collection Gridlock, in: Proceedings of the ACM SIGPLAN Programming Languages Design and Implementation (PLDI), Berlin, Germany, 2002, pp. 153–164.
- [59] R. Jones, R. Lins, Garbage Collection: Algorithms for automatic Dynamic Memory Management, John Wiley and Sons, 1998
- [60] P. R. Wilson, Uniprocessor Garbage Collection Techniques, in: Proceedings of the International Workshop on Memory Management (IWMM), St. Malo, France, 1992, pp. 1–42.
- [61] J. L. McCarthy, Recursive Functions of Symbolic Expressions and Their Computation by Machine, Communications of the ACM 3 (4) (1960) 184–195.
- [62] D. Ungar, Generation Scavenging: A non-disruptive high performance storage reclamation algorithm, in: Proceedings of the First ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments, 1984, pp. 157–167.
- [63] J. Weizenbaum, Symmetric List Processor, Communications of the ACM 6 (9) (1963) 524–544.
- [64] S. M. Blackburn, K. S. McKinley, Ulterior Reference Counting: Fast Garbage Collection Without a Long Wait, in: Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), Anaheim, California, USA, 2003, pp. 344–358.

- [65] D. F. Bacon, C. R. Attanasio, H. Lee, V. T. Rajan, S. Smith, Java without the Coffee Breaks: A Nonintrusive Multiprocessor Garbage Collector, in: Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2001, pp. 92–103.
- [66] T. W. Christopher, Reference Count Garbage Collection, *Software Practice and Experience* 14 (6) (June 1984) 503–507.
- [67] John McCarthy. Recursive functions of symbolic expressions and their computation by machine. *Communications of the ACM*, 3(4):184–195, 1960
- [68] Shinichi Furusou, Satoshi Matsuoka, and Akinori Yonezawa. Parallel conservative garbage collection with fast allocation. In Paul R. Wilson and Barry Hayes, editors, *OOPSLA/ECCOP '91 Workshop on Garbage Collection in Object-Oriented Systems*, 1991
- [69] Taichi Yuasa. Real-time garbage collection on general-purpose machines. *Journal of Software and Systems*, 11(3):181–198, 1990
- [70] James W. O'Toole and Scott M. Nettles. Concurrent replicating garbage collection. Also LFP94 and OOPSLA93 Workshop on Memory Management and Garbage Collection.
- [71] Hans-J. Boehm, Alan J. Demers, and Scott Shenker. Mostly parallel garbage collection. In *SIGPLAN Symposium on Programming Language Design and Implementation*, pages 157–164, June 1991
- [72] Tony Printezis and David Detlefs. A Generational Mostly-Concurrent Garbage Collector. In *International Symposium on Memory Management (ISMM '00)*, volume 36(1) of ACM SIGPLAN Notices, January 2001

- [73] Yoav ossia, Ori Ben-Yitzhak, Irit Gofit, Elliot K. Kolodner, Victor Leikehman, and Avi Owshanko. A parallel, incremental, and concurrent GC for servers. In *Proceeding of the ACM SIGPLAN 2002 Conference on Programming language design and implementation*, pages 129-140, 2002
- [74] Guy L. Steele. Multiprocessing compactifying garbage collection. *Communications of the ACM*, 18(9):495-508, September 1975
- [75] Guy L. Steele. Corrigendum: Multiprocessing compactifying garbage collection. *Communications of the ACM*, 19(6):354, June 1976
- [76] Edsger W. Dijkstra, Leslie Lamport, A. J. Martin, C. S. Scholten, and E. F. M. Steffens. On-the-fly garbage collection: An exercise in cooperation. *Communications of the ACM*, 21(11):965–975, November 1978
- [77] David Gries. An exercise in proving parallel programs correct. *Communications of the ACM*, 20(12):921-930, December 1977
- [78] Mordechai Ben-Ari. On-the-fly garbage collection: New algorithms inspired by program proofs. In M. Nielsen and E. M. Schmidt, editors, *Automata, languages, and programming*. Ninth colloquium (Aarhus, Denmark), pages 14-22, New York, July 12-16 1982. Springer-Verlag.
- [79] Mordechai Ben-Ari. Algorithms for on-the-fly garbage collection. *ACM Transactions on Programming Languages and Systems*, 6(3):333-344, July 1984
- [80] H. T. Kung and S. W. Song. An efficient parallel garbage collection system and its correctness proof. In *IEEE Symposium on Foundations of Computer Science*, pages 120-131. IEEE Press, 1977

- [81] L. Lamport. Garbage collection with multiple processes: an exercise in parallelism. In *Proceedings of the 1976 International Conference on Parallel Processing*, pages 50-54, 1976
- [82] Damien Doligez and Xavier Leroy. A concurrent generational garbage collector for a multi-threaded implementation of ML. In *POPL 1993*
- [83] Damien Doligez and Georges Gonthier. Portable, unobtrusive garbage collection for multiprocessor systems. In *POPL 1994*
- [84] Tamar Domani, Elliot K. Kolodner, and Erez Petrank. A generational on-the-fly garbage collector for java. In *Proceedings of the SIGPLAN 2000 Conference on Programming Language Design and Implementation*, pages 274–284, 2000
- [85] Tamar Domani, Elliot K. Kolodner, Ethan Lewis, Elliot E. Salant, Katherine Barabash, Itai Lahan, Yossi Levroni, Erez Petrank, and Igor Yanover. Implementing an On-the-fly Garbage Collector for Java. *The 2000 International Symposium on Memory Management*, October, 2000
- [86] Richard L. Hudson and J. Eliot B. Moss. Sapphire: Copying GC Without Stopping The World, Joint *ACM Java Grande — ISCOPE 2001 Conference*.
- [87] D. Bacon, C. Attanasio, H. Lee, V. Rajan, and S. Smith. Java without the coffee breaks: A nonintrusive multiprocessor garbage collector. *The ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, Snowbird, Utah, June 20-22 2001
- [88] John McCarthy. Recursive functions of symbolic expressions and their computation by machine. *Communications of the ACM*, 3(4):184–195, 1960

- [89] K. Goossens *et al.* Interconnect and memory organization in SOCs for advanced set-top boxes and TV. Evolution, analysis, and trends. In *Interconnect-Centric Design for Advanced SoC and NoC*. Kluwer, 2004
- [90] JEDEC Solid State Technology Association. *DDR2 SDRAM Specification*, JESD79-2C edition, May 2006
- [91] L. Woltjer. Optimal DDR controller. Master's thesis, University of Twente, Jan. 2005
- [92] S. Rixner *et al.* Memory access scheduling. In *ISCA '00: Proceedings of the 27th annual international symposium on Computer architecture*, 2000
- [93] Luud Woltjer. Optimal DDR controller. University of Twente, MSc thesis, 2005
- [94] Scott Rixner, William J. Dally, Ujval J. Kapasi, Peter Mattson, and John D. Owens. Memory access scheduling. *SIGARCH Comput. Archit. News*, 28(2):128–138, 2000
- [95] W.-D. Weber. Efficient Shared DRAM Subsystems for SOCs. Sonics, Inc, 2001. White paper, 2001
- [96] ARM. PrimeCell Synchronous Static Memory Controller, 2004
- [97] Kun-Bin Lee, Tzu-Chieh Lin, and Chein-Wei Jen. An efficient quality-aware memory controller for multimedia platform SoC. In *Circuits and Systems for Video Technology*, IEEE Transactions on, volume 15, page 5pp., New York, NY, USA, May 2005. ACM Press
- [98] A. Burks, H. Goldstein, J. von Neumann: Preliminary discussion of the logical design of an electronic computing instrument; US Army Ordnance Department Report, 1946

- [99] H. Goldstein, J. von Neumann, A. Burks: Report on the mathematical and logical aspects of an electronic computing instrument; Princeton IAS, 1947
- [100] M. J. Smith: Application Specific Integrated Circuits; Addison Wesley 1997
- [101] R. Hartenstein (invited): The Microprocessor is no more General Purpose; Proc. IEEE International Symposium on Innovative Systems (ISIS), Oct.9-11 1997, Austin, Texas
- [102] L. Rizzatti: Is there any future for ASIC emulators? (Alternatively, How FPGA-Based Prototypes Can Replace ASIC Emulators); EDA Alert e-Newsletter PlanetEE, Febr. 18, 2003
- [103] T. Makimoto (keynote): The Rising Wave of Field-Programmability; FPL 2000, Villach, Austria, Aug. 27 - 30, 2000; Springer Verlag, Heidelberg/New York, 2000
- [104] J. Becker (invited tutorial): Reconfigurable Computing Systems; Proceedings Escola de Microelectronics da SBC - Sul (EMICRO 2003), Rio Grande, Brazil, September 2003
- [105] J.Becker, S. Vernalde (editors): Advances in Reconfigurable Architectures - Part 1; special issue, International Journal on Embedded Systems, 2006
- [106] J.Becker, S. Vernalde (editors): Advances in Reconfigurable Architectures - Part 2; special issue, International Journal on Embedded Systems, 2006
- [107] F. Faggin, M. Hoff, S. Mazor, M. Shima: The history of 4004; IEEE Micro, Dec. 1996

- [108] R. Hartenstein (invited talk): Reconfigurable supercomputing: What are the Problems? What are the Solutions? Seminar on Dynamically Reconfigurable Architectures, Dagstuhl Castle, Wadern, Germany, April 2 - 7, 2006
- [109] R. Hartenstein (opening keynote): From Organic Computing to Reconfigurable Computing; 8th Workshop on Parallel Systems and Algorithms (PASA 2006), March 16, 2006, Frankfurt/Main, Germany - in conjunction with the 19th International Conference on Architecture of Computing Systems (ARCS 2005), March 13 - 16, 2006, Frankfurt/Main, German
- [110] R. Hartenstein (opening keynote): Supercomputing goes Reconfigurable - About key issues and their impact on CS education; Winter International Symposium on Information and Communication Technologies (WISICT 2005), Cape Town, South Africa, Jan. 3 - 6, 2005
- [111] R. Hartenstein (plenum keynote address): Software or Configware? About the Digital Divide of Computing; 18th International Parallel and Distributed Processing Symposium (IPDPS), April 26–April 30, 2004, Santa Fe, New Mexico, USA
- [112] R. Hartenstein (invited presentation): The Digital Divide of Computing; 2004 ACM International Conference on Computing Frontiers (CF04); April 14-18, 2004. Ischia, Italy
- [113] R. Hartenstein (invited keynote address): The Impact of Morphware on Parallel Computing; 12-th Euromicro Conference on Parallel, Distributed and Network based Processing (PDP04); February, 11-13, 2004. A Coruña, Spain

- [114] D. Chinnery, K. Keutzer: Closing the Gap between ASIC & Custom; Kluwer 2002
- [115] M. Pedram and A. Abdollahi, "Low-Power RT-level synthesis techniques: a tutorial," *IEE Proceedings-Computer and Digital Techniques*, Vol. 152, No. 3, May 2005
- [116] A. Terechko, "Clustered VLIW Architectures, A Quantative Approach," PhD Thesis, Technical University Eindhoven, 2007.
- [117] A. Bechini, T. M. Conte, C. A. Prete, "Opportunities and Challenges in Embedded Systems," *Proc. of the Intl. Symposium on Microarchitecture*, August 2004
- [118] S. Haga, N. Reeves, R. Barua, D. Marculescu, "Dynamic Functional Unit Assignment for Low-Power," *Proceedings of the Design, Automation and Test in Europe Conference*, 2003
- [119] U. Ko, P. T. Balsara, W. Lee, "Low-Power Design techniques for High-Performance CMOS adders," *IEEE Transactions on VLSI Systems*, Vol. 3, No. 2, June 1995
- [120] N. Ramsey, J. W. Davidson, "Machine Description to build tools for embedded systems," *Lecture Notes in Computer Science*, Vol. 1474, 1998
- [121] J. Fischer, P. Faraboschi, G. Desoli, "Custom-Fit Processors: Letting Application Define Architectures," *Proceedings of the International Symposium on Microarchitecture*, pp. 324-335, 1996
- [122] P. Faraboschi, G. Brown, J. A. Fischer, "Lx: A Technology Platform for Customizable VLIW Embedded Processing," *Proceedings of the International Symposium on Computer Architecture*, 2000

- [123] N. Clark, et al., “OptimoDE: Programmable Accelerator Engines through Retargetable Customization,” *Hct-Chips*, 2004
- [124] Tensilica Xtensa 7(LX2),
<http://www.tensilica.com/products/xtensa/index.htm>
- [125] N. Clark, Personal Conversation, Georgia Institute of Technology
- [126] Tensilica Incorporated Website, www.tensilica.com, June 2012
- [127] “Tensilica Diamond Standard Series Product Brief”, Tensilica Incorporated, 2006
- [128] Tom R. Halfhill, “Tensilica’s Preconfigured Cores”, Microprocessor Report, March 20, 2006
- [129] Xtensa Overview, www.tensilica.com/products/overview.htm, June 2012
- [130] Create TIE, www.tensilica.com/products/create_tie.htm, September 2012
- [131] TIE Compiler, www.tensilica.com/products/tie_compiler.htm, September 2012
- [132] XPRES Compiler, <http://www.tensilica.com/products/xpres.htm>, September 2012
- [133] P. Athanas and H. Silverman. Processor reconfiguration through instruction-set metamorphosis. *IEEE Computer*, 26(3):11–18, March 1995

- [134] R. Razdan and M. Smith. High-performance microarchitecture with hardware-programmable functional units. In *Proceedings of the 27th Annual International Symposium on Microarchitecture*, November 1994
- [135] Z.A. Ye, N. Shenoy, and P. Banerjee. A C compiler for a processor with a reconfigurable functional unit. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, February 2000
- [136] R. Wittig and P. Chow. OneChip: An FPGA Processor with Reconfigurable Logic. In *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, pp.126-135, Napa Valley, California, March 1996
- [137] T. Callahan, J. Hauser, and J. Wawrzynek. The Garp architecture and C compiler. *IEEE Computer*, 33(4):62–69, April 2000
- [138] R.David,D. Chillet,S. Pillement and O. Sentieys DART: a dynamically reconfigurable architecture dealing with future mobile telecommunications constraints In *Proceedings of the IEEE Reconfigurable Architectures Workshcp*, Fort Lauderdale, Florida, April 2002
- [139] A. DeHon. DPGA-Coupled Microprocessors: Commodity ICs for the Early 21st Century. In *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines* ,pp.31-39, Napa Valley, California, April 1994
- [140] S. Trimberger, D. Carberry, A. Jhonson, and J. Wong. Configuration Caching Techniques for FPGA. In *Proceedings of the IEEE Symposium on Field- Programmable Custom Computing Machines* ,pp.34- 40, Napa Valley, California, April 2000

- [141] P. Marwedel and G. Goossens, editors. *Code generation for embedded processors*. Kluwer Academic Publishers, 1995
- [142] R. Kastner, A. Kaplan, S. Memik, and E. Bozorgzadeh, "Instruction generation for hybrid reconfigurable systems", *ACM Transactions on Design Automation of Embedded Systems (TODAES)*, October, 2002
- [143] M. Arnold and H. Corporaal, "Designing domain specific processors", *Proceedings of the 9th International Workshop on Hardware/Software CoDesign*, pages 61–66, Copenhagen, April 2001
- [144] B. Kastrup, A. Bink, and J. Hoogerbrugge, "ConCISe: A compiler-driven CPLD-based instruction set accelerator", *Proceedings of the 5th IEEE Symposium on Field-Programmable Custom Computing Machines*, Apr. 1999
- [145] Z. A. Ye, A. Moshovos, S. Hauck, and P. Banerjee, "CHIMAERA: A high-performance architecture with a tightly-coupled reconfigurable functional unit", *Proceedings of the 27th Annual International Symposium on Computer Architecture*, pp. 225–235, June 2000

PUBLICATIONS BASED ON THE THESIS WORK

International Journal

1. Sudarshan Patel, Dr. Vishvjit K. Thakar, "Investigation & Comparative Study Of Hardware Assisted Unused Memory Collector Over Software Memory Collector Approach" in *International Journal Research Journal of Computer Science Engineering and Applications* ISSN 2319-8672 (IRJCSEA) Volume 2, Issue 4 April 2013.

* Two more Journal papers are ready to communicate.