Universiteit Leiden
The Netherlands

Transformations for polyhedral process networks

Meijer, S.

Citation
Meijer, S. (2010, December 8). *Transformations for polyhedral process networks*. Retrieved from https://hdl.handle.net/1887/16221

Version:	Corrected Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/16221

Note: To cite this publication please use the final published version (if applicable).

Bibliography

[1] M. Adiletta, M. Rosenbluth, and D. Bernstein. The next generation of Intel IXP network processors. *Intel Technology Journal*, 06(03), 15 aug 2002.

[2] A. V. Aho, R. Sethi, and J. D. Ullman. *Compilers: principles, techniques, and tools*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986.

[3] T. W. Ainsworth and T. M. Pinkston. On characterizing performance of the Cell Broadband Engine Element Interconnect Bus. In *NOCS '07: Proceedings of the First International Symposium on Networks-on-Chip*, pages 18–29, 2007.

[4] R. Bagnara, P. M. Hill, and E. Zaffanella. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Sci. Comput. Program.*, 72(1-2):3–21, 2008.

[5] F. Balarin, Y. Watanabe, H. Hsieh, L. Lavagno, C. Passerone, and A. Sangiovanni-Vincentelli. Metropolis: An integrated electronic system design environment. *Computer*, 36(4):45–52, 2003.

[6] U. K. Banerjee. *Dependence Analysis for Supercomputing*. Kluwer Academic Publishers, Norwell, MA, USA, 1988.

[7] C. Bastoul. Code generation in the polyhedral model is easier than you think. In *PACT '04: Proceedings of the 13th International Conference on Parallel Architectures and Compilation Techniques*, pages 7–16, 2004.

[8] S. Bhattacharrya, R. Leupers, J. Takala, and E. Deprettere, editors. *Handbook on signal processing systems*, chapter by Verdoolaege, S., Polyhedral process networks. Springer, 2010.

[9] S. S. Bhattacharyya and E. A. Lee. Scheduling synchronous dataflow graphs for efficient looping. *J. VLSI Signal Process. Syst.*, 6(3):271–288, 1993.

[10] A. Bik, M. Girkar, P. Grey, and X. Tian. Efficient exploitation of parallelism on Pentium III and Pentium 4 processor-based systems. *Intel Technology Journal Q1 (March) (2001) 1-9*, 2001.

[11] G. Bilsen, M. Engels, R. Lauwereins, and J. Peperstraete. Cyclo-static dataflow. *IEEE Transactions on signal processing*, 44(2):397–408, 1996.

[12] W. Blume, R. Eigenmann, K. Faigin, J. Grout, J. Hoeflinger, D. A. Padua, P. Petersen, W. M. Pottenger, L. Rauchwerger, P. Tu, and S. Weatherford. Polaris: Improving the effectiveness of parallelizing compilers. In *LCPC '94: Proceedings of the 7th International Workshop on Languages and Compilers for Parallel Computing*, pages 141–154, 1995.

[13] U. Bondhugula, A. Hartono, J. Ramanujam, and P. Sadayappan. A practical automatic polyhedral parallelizer and locality optimizer. *SIGPLAN Not.*, 43(6):101–113, 2008.

[14] P. M. Carpenter, A. Ramirez, and E. Ayguade. Mapping stream programs onto heterogeneous multiprocessor systems. In *CASES '09: Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems*, pages 57–66, 2009.

[15] J. Ceng, J. Castrillon, W. Sheng, H. Scharwächter, R. Leupers, G. Ascheid, H. Meyr, T. Isshiki, and H. Kunieda. MAPS: an integrated framework for MPSoC application parallelization. In *DAC '08: Proceedings of the 45th annual Design Automation Conference*, pages 754–759, 2008.

[16] S. Chakraborty, S. Kunzli, and L. Thiele. A general framework for analysing system properties in platform-based embedded system designs. In *DATE '03: Proceedings of the conference on Design, Automation and Test in Europe*, page 10190, Washington, DC, USA, 2003. IEEE Computer Society.

[17] E. Cheung, H. Hsieh, and F. Balarin. Automatic buffer sizing for rate-constrained KPN applications on multiprocessor system-on-chip. In *Proc. of HLDVT*, pages 37–44, 2007.

[18] P. Clauss. Counting solutions to linear and nonlinear constraints through Ehrhart polynomials: applications to analyze and transform scientific programs. In *ICS '96: Proceedings of the 10th international conference on Supercomputing*, pages 278–285, 1996.

[19] P. Clauss. Handling memory cache policy with integer points counting. In *Euro-Par '97: Proceedings of the Third International Euro-Par Conference on Parallel Processing*, pages 285–293, 1997.

[20] P. Clauss, V. Loechner, and D. Wilde. Deriving formulae to count solutions to parameterized linear systems using Ehrhart polynomials: Applications to the analysis of nested-loop programs, 1997.

[21] R. Dömer, A. Gerstlauer, J. Peng, D. Shin, L. Cai, H. Yu, S. Abdi, and D. D. Gajski. System-on-chip environment: a SpecC-based framework for heterogeneous MPSoC design. *EURASIP J. Embedded Syst.*, 2008:1–13, 2008.

[22] B. K. Dwivedi, A. Kumar, and M. Balakrishnan. Automatic synthesis of system on chip multiprocessor architectures for process networks. In *Proc. of CODES+ISSS*, pages 60–65, 2004.

[23] J. Falk, J. Keinert, C. Haubelt, J. Teich, and S. S. Bhattacharyya. A generalized static data flow clustering algorithm for MPSoC scheduling of multimedia applications. In *Proc. of EMSOFT*, pages 189–198, 2008.

[24] P. Feautrier. Parametric integer programming. *RAIRO Recherche Operationnelle*, 22, 1988.

[25] P. Feautrier. Dataflow analysis of array and scalar references. *International Journal of Parallel Programming*, 20, 1991.

[26] J. A. Fisher. Very long instruction word architectures and the ELI-512. In *ISCA '83: Proceedings of the 10th annual international symposium on Computer architecture*, pages 140–150, 1983.

[27] M. P. Forum. MPI: A Message-Passing Interface standard. Technical report, 1994.

[28] D. D. Gajski, F. Vahid, S. Narayan, and J. Gong. *Specification and design of embedded systems*. Prentice-Hall, Inc., 1994.

[29] L. George and M. Blume. Taming the IXP network processor. In *PLDI '03: Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation*, pages 26–37, 2003.

[30] A. H. Ghamarian, M. C. W. Geilen, T. Basten, and S. Stuijk. Parametric throughput analysis of synchronous data flow graphs. In *Proc. of DATE*, pages 116–121, 2008.

[31] M. I. Gordon, W. Thies, and S. Amarasinghe. Exploiting coarse-grained task, data, and pipeline parallelism in stream programs. In *ASPLOS-XII: Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*, pages 151–162, 2006.

[32] M. I. Gordon, W. Thies, M. Karczmarek, J. Lin, A. S. Meli, A. A. Lamb, C. Leger, J. Wong, H. Hoffmann, D. Maze, and S. Amarasinghe. A stream compiler for communication-exposed architectures. In *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, pages 291–303, 2002.

[33] K. Grüttner and W. Nebel. Modelling program-state machines in SystemC. In *FDL*, pages 7–12, 2008.

[34] M. Gschwind, H. P. Hofstee, B. Flachs, M. Hopkins, Y. Watanabe, and T. Yamazaki. Synergistic processing in cell's multicore architecture. *IEEE Micro*, 26(2):10–24, 2006.

[35] S. Ha, C. Lee, Y. Yi, S. Kwon, and Y.-P. Joo. Hardware-software codesign of multimedia embedded systems: the PeaCE. In *RTCSA '06: Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 207–214, 2006.

[36] M. W. Hall, J. M. Anderson, S. P. Amarasinghe, B. R. Murphy, S.-W. Liao, E. Bugnion, and M. S. Lam. Maximizing multiprocessor performance with the SUIF compiler. *Computer*, 29(12):84–89, 1996.

[37] C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, 1978.

[38] J. Y. Hur, S. Wong, and T. Stefanov. Design trade-offs in customized on-chip crossbar schedulers. *J. Signal Process. Syst.*, 58(1):69–85, 2010.

[39] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer, and D. Shippy. Introduction to the Cell multiprocessor. *IBM J. Res. Dev.*, 49(4/5):589–604, 2005.

[40] G. Kahn. The Semantics of a Simple Language for Parallel Programming. In *Proc. of the IFIP Congress 74*. North-Holland Publishing Co., 1974.

[41] T. Kangas, P. Kukkala, H. Orsila, E. Salminen, M. Hännikäinen, T. D. Hämäläinen, J. Riihimäki, and K. Kuusilinna. UML-based multiprocessor SoC design framework. *ACM Trans. Embed. Comput. Syst.*, 5(2):281–320, 2006.

[42] J. Keinert, M. Streubuehr, T. Schlichter, J. Falk, J. Gladigau, C. Haubelt, J. Teich, and M. Meredith. SystemCoDesigner—an automatic ESL synthesis approach by design space exploration and behavioral synthesis for streaming applications. *ACM Trans. Des. Autom. Electron. Syst.*, 14(1):1–23, 2009.

[43] K. Kennedy and J. R. Allen. *Optimizing compilers for modern architectures: a dependence-based approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[44] B. Kienhuis, E. F. Deprettere, P. v. d. Wolf, and K. A. Vissers. A methodology to design programmable embedded systems - the y-chart approach. In *Embedded Processor Design Challenges: Systems, Architectures, Modeling, and Simulation - SAMOS*, pages 18–37, 2002.

[45] A. Kumar, S. Fernando, Y. Ha, B. Mesman, and H. Corporaal. Multiprocessor systems synthesis for multiple use-cases of multiple applications on FPGA. *ACM Trans. Des. Autom. Electron. Syst.*, 13(3):1–27, 2008.

[46] J.-Y. Le Boudec and P. Thiran. *Network calculus: a theory of deterministic queuing systems for the internet*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[47] E. A. Lee and D. G. Messerschmitt. Synchronous data flow. In *Proceedings of the IEEE*, volume 75, pages 1235–1245, September 1987.

[48] C. Lemke. The dual method of solving the linear programming problem. *Naval Research Logistics Quarterly*, 1:36 – 47, 1954.

[49] L. Li, B. Huang, J. Dai, and L. Harrison. Automatic multithreading and multiprocessing of C programs for IXP. In *PPoPP '05: Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming*, pages 132–141, 2005.

[50] S. Meijer, B. Kienhuis, J. Walters, and D. Snuijf. Automatic partitioning and mapping of stream-based applications onto the Intel IXP network processor. In *SCOPES '07: Proceedingsof the 10th international workshop on Software & compilers for embedded systems*, pages 23–30, 2007.

[51] S. Meijer, H. Nikolov, and T. Stefanov. On compile-time evaluation of process partitioning transformations for Kahn process networks. In *CODES+ISSS '09: Proceedings of the 7th IEEE/ACM international conference on Hardware/software codesign and system synthesis*, pages 31–40, 2009.

[52] S. Meijer, H. Nikolov, and T. Stefanov. Combining process splitting and merging transformations for polyhedral process networks. In *Proc. of the 8th Int. IEEE Workshop on Embedded Systems for Real-Time Multimedia (ESTIMedia)*, pages 97–106, 2010.

[53] S. Meijer, H. Nikolov, and T. Stefanov. Throughput modeling to evaluate process merging transformations in polyhedral process networks. In *Proceedings of the conference on Design, automation and test in Europe (DATE'10)*, pages 747–752, 2010.

[54] B. Meister, A. Leung, N. Vasilache, D. Wohlford, C. Bastoul, and R. Lethin. Productivity via automatic code generation for PGAS platforms with the R-Stream compiler. In *APGAS'09 Workshop on Asynchrony in the PGAS Programming Model*, June 2009.

[55] A. Moonen, M. Bekooij, R. v. d. Berg, and J. v. Meerbergen. Practical and accurate throughput analysis with the cyclo static dataflow model. In *Proc. of MASCOTS*, pages 238–245, 2007.

[56] G. E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8):114–117, April 1965.

[57] S. Muchnick. *Advanced Compiler Design and Implementation*. Morgan Kaufmann Publishers, Inc., 1997.

[58] D. Nadezhkin, S. Meijer, T. Stefanov, and E. Deprettere. Realizing FIFO communication when mapping Kahn process networks onto Cell. In *SAMOS IX: International Symposium on Systems, Architectures, MOdeling and Simulation*, 2009.

[59] J. Nickolls, I. Buck, M. Garland, and K. Skadron. Scalable parallel programming with CUDA. *Queue*, 6(2):40–53, 2008.

[60] H. Nikolov, T. Stefanov, and E. Deprettere. Multi-processor system design with ESPAM. In *Proc. of CODES+ISSS*, pages 211–216, 2006.

[61] H. Nikolov, T. Stefanov, and E. Deprettere. Systematic and automated multiprocessor system design, programming, and implementation. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(3):542–555, 2008.

[62] H. Nikolov, M. Thompson, T. Stefanov, A. Pimentel, S. Polstra, R. Bose, C. Zissulescu, and E. Deprettere. Daedalus: toward composable multimedia MP-SoC

design. In *DAC '08: Proceedings of the 45th annual conference on Design automation*, pages 574–579, 2008.

[63] L. noel Pouchet, C. Bastoul, A. Cohen, and N. Vasilache. Iterative optimization in the polyhedral model: Part I, one-dimensional time. In *International Symposium on Code Generation and Optimization (CGO)*, pages 144 – 156, 2007.

[64] "OpenCL". "the open standard for parallel programming of heterogeneous systems http://www.khronos.org/opencl/", 2009.

[65] S. Pakin. Receiver-initiated message passing over RDMA networks. *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pages 1–12, April 2008.

[66] K. K. Parhi and D. G. Messerschmitt. Static Rate-Optimal Scheduling of Iterative Data-Flow Programs via Optimum Unfolding. *IEEE Transaction on Computers*, 40(2):178–195, Feb. 1991.

[67] A. D. Pimentel, C. Erbas, and S. Polstra. A systematic approach to exploring embedded system architectures at multiple abstraction levels. *IEEE Trans. Comput.*, 55(2):99–112, 2006.

[68] J. Pino, S. Bhattacharyya, and E. A. Lee. A hierarchical multiprocessor scheduling framework for Synchronous Dataflow Graphs. Technical Report UCB/ERL M95/36, EECS Department, University of California, Berkeley, 1995.

[69] S. Pop, A. Cohen, C. Bastoul, S. Girbal, G. A. Silber, and N. Vasilache. Graphite: Loop optimizations based on the polyhedral model for gcc. In *Proc. of the 4th GCC Developper's Summit*, pages 179–198, June 2006.

[70] L.-N. Pouchet, C. Bastoul, A. Cohen, and J. Cavazos. Iterative optimization in the polyhedral model: part ii, multidimensional time. In *PLDI '08: Proceedings of the 2008 ACM SIGPLAN conference on Programming language design and implementation*, pages 90–100, 2008.

[71] W. Pugh and D. Wonnacott. An exact method for analysis of value-based array data dependences. In *Proceedings of the 6th International Workshop on Languages and Compilers for Parallel Computing*, pages 546–566, 1994.

[72] K. H. Rosen. *Discrete mathematics and its applications (2nd ed.)*. McGraw-Hill, Inc., New York, NY, USA, 1991.

[73] M. S. Schlansker and B. R. Rau. EPIC: Explicitly parallel instruction computing. *Computer*, 33(2):37–45, 2000.

[74] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[75] N. Shah, W. Plishker, K. Ravindran, and K. Keutzer. Np-click: A productive software development approach for network processors. *IEEE Micro*, 24(5):45–54, 2004.

[76] J. Sjodin, S. Pop, H. Jagasia, T. Grosser, and A. Pop. Design of graphite and the polyhedral compilation package. 2009.

[77] S. Sriram and S. Bhattacharyya. *Embedded Multiprocessors: Scheduling and Synchronization*. Marcel Dekker, Inc., 2000.

[78] T. Stefanov. Converting weakly dynamic programs to equivalent process network specifications, 2004. PhD thesis, Leiden University.

[79] T. Stefanov, B. Kienhuis, and E. Deprettere. Algorithmic transformation techniques for efficient exploration of alternative application instances. In *Proc. of CODES*, pages 7–12, 2002.

[80] S. Stuijk, M. Geilen, and T. Basten. Exploring trade-offs in buffer requirements and throughput constraints for synchronous dataflow graphs. In *DAC '06: Proceedings of the 43rd annual Design Automation Conference*, pages 899–904, 2006.

[81] S. Stuijk, M. Geilen, and T. Basten. Throughput-buffering trade-off exploration for Cyclo-Static and Synchronous Dataflow Graphs. *IEEE Trans. Comput.*, 57(10):1331–1345, 2008.

[82] N. N. S. Technologies. http://www.network-speed.com.

[83] J. Teich and L. Thiele. Exact Partitioning of Affine Dependence Algorithms. *Lecture Notes in Computer Science (LNCS), Springer*, 2268:133–151, 2002.

[84] L. Thiele, I. Bacivarov, W. Haid, and K. Huang. Mapping applications to tiled multiprocessor embedded systems. In *ACSD '07: Proceedings of the Seventh International Conference on Application of Concurrency to System Design*, pages 29–40, 2007.

[85] L. Thiele, S. Chakraborty, and M. Naedele. Real-time calculus for scheduling hard real-time systems. In *ISCAS*, pages 101–104, 2000.

[86] L. Thiele and N. Stoimenov. Modular performance analysis of cyclic dataflow graphs. In *EMSOFT 09: Proceedings of the 9th ACM international conference on Embedded software*, pages 127–136, Grenoble, France, 2009.

[87] W. Thies, M. Karczmarek, and S. P. Amarasinghe. StreamIt: A language for streaming applications. In *CC '02: Proceedings of the 11th International Conference on Compiler Construction*, pages 179–196, 2002.

[88] M. Thompson and A. D. Pimentel. Towards multi-application workload modeling in Sesame for system-level design space exploration. In *SAMOS*, pages 222–232, 2007.

[89] A. Turjan. Compiling nested loop programs to process networks, 2007. PhD thesis, Leiden University, The Netherlands.

[90] A. Turjan, B. Kienhuis, and E. Deprettere. Translating affine nested-loop programs to process networks. In *CASES '04: Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems*, pages 220–229, 2004.

[91] S. van Haastregt and B. Kienhuis. Automated synthesis of streaming C applications to process networks in hardware. In *DATE*, pages 890–893, 2009.

[92] S. Verdoolaege. *Incremental Loop Transformations and Enumeration of Parametric Sets*. PhD thesis, Katholieke Universiteit Leuven, 2005.

[93] S. Verdoolaege. An integer set library for program analysis. *ACES symposium, Edegem, 7-8 september*, 2009.

[94] S. Verdoolaege, M. Bruynooghe, G. Janssens, and F. Catthoor. Multi-dimensional incremental loop fusion for data locality. In *In Proceedings of the IEEE International Conference on Application Specific Systems, Architectures, and Processors*, pages 17–27, 2003.

[95] S. Verdoolaege, H. Nikolov, and T. Stefanov. pn: a tool for improved derivation of process networks. *EURASIP J. Embedded Syst.*, 2007(1):19–19, 2007.

[96] S. Verdoolaege and K. Woods. Counting with rational generating functions. *J. Symb. Comput.*, 43(2):75–91, 2008.

[97] S. Verdoolaege, K. M. Woods, M. Bruynooghe, and R. Cools. Computation and manipulation of enumerators of integer projections of parametric polytopes. CW Reports CW392, K.U.Leuven, Department of Computer Science, Mar. 2005.

[98] D. K. Wilde. A library for doing polyhedral operations. Technical Report RR-2157.

[99] X. D. Zhang, Q. J. Li, R. Rabbah, and S. Amarasinghe. A lightweight streaming layer for multicore execution, http://cag.lcs.mit.edu/commit/papers/07/zhang-dascmp07.pdf.

Index

affine hyperplane, 17
aggregated FIFO throughput, 76
average production period, 41

Cell platform, 112
communication costs, 38
compound process, 65
computation costs, 38
control overhead, 42

Daedalus, 3
data transfer, 42

execution time of a transformation, 43

FIFO channel throughput, 74
FIFO pull strategy, 116

hyperplane, 17

initial delay, 39
input port domain, 27
Intel IXP network processor, 113
isolated process throughput, 72

lexicographical maximum, 19
lexicographical minimum, 19
lexicographical order, 19

mapping, 28
modulo unfolding, 32

output port domain, 28

parametric integer linear programming, 19
partitioning metrics, 38
plane-cutting, 32
pn compiler, 3
polyhedral model, 21
Polyhedral Process Network (PPN), 24
polytope, 18
process function, 25
process iteration, 27
process iteration domain, 27
process iteration domain size, 27
process merging, 65
process splitting, 33
process throughput, 70
process workload, 26
production period, 40

rank, 20
rational polyhedron, 18

SANLP, 21
scalar product, 17
self-edge, 35
sink process, 25
source process, 25
static affine nested loop program, 21

static control parts (SCoPs), 22
system throuhgput, 67

throughput propagation, 71

Y-chart, 5

Acknowledgments

This dissertation would not have been written without the help, assistance, and advise of many people. First of all, I would like to thank Alexandru Turjan for introducing me to the topics of compilation techniques and program analysis. Inviting me to write my master's thesis in Philips Research was really the kickstart for my research work later as a PhD-student. Alex, I learned a lot from your research mentality, interest in reading literature, and problem solving skills. It was therefore my pleasure to work briefly together again when you invited me for a PhD internship at NXP Semiconductors.

From the LERC group in LIACS, I am most thankful to Todor Stefanov and Hristo Nikolov. In the "second half" of my PhD time, when results needed to be produced, you pushed me and I pushed you. We had many interesting and challenging discussions that led to the fine results that we produced in such a short amount of time. While I was sometimes rushing, you were always checking and double-checking things and I really enjoyed working together.

From ACE Associated Compiler Experts B.V., I would like to thank Marcel Beemster, Marius Schoorel, Joseph van Vlijmen, and Martijn de Lange for giving me the right advise during my PhD, which really contributed to the successful second half my PhD.

The work presented in this dissertation has been supported by the MEDEA+ NEVA project 2A703. I would like to thank the NEVA project for financially supporting my research, and I am thankful to Sven Verdoolaege for proof reading this dissertation.

Finally, I would like to thank all my other friends, family, parents for their support. Wouter Meuleman in particular, since we finished the same bachelor and master studies, both continued as Phd-students, and thus shared many experiences. And last but not least, I would like to thank Senny for her understanding and support during my PhD time, and for her love!