Proceedings of "Center-of-Excellence" workshop on System LSI Design Methodology

https://doi.org/10.15017/2994

出版情報：SLRC Discussion Paper Series. 3 (1), 2006-09. 九州大学システムLSI研究センター
バージョン：
権利関係：

"Center-of-Excellence" workshop on System LSI Design Methodology

Sponsored by Kyushu University
Cooperated with IPSJ Kyushu Chapter
Cooperated with IEEE Fukuoka Chapter

Place: System LSI Research Center, Kyushu University
2nd Floor Institute of System LSI Design Industry, Fukuoka
3-8-33 Momochihama, Sawara-ku, Fukuoka 814-0001 JAPAN

Date: September 9 (Sat.), 2006

Schedule: 10:00-17:20 Workshop

Aim of the Workshop

The steady progress in process technology and the expanding applications of information processing are placing enormous demands on the high performance, yet affordable design and automation of integrated circuits and systems. This workshop, contributed by representative researchers from all over the world, offers a forum for all those people working on integrated circus and systems to meet and exchange ideas about the challenges and solutions for the future of system LSI Design Technology and Automation.

List of Presenters

Yao-Wen Chang,	National Taiwan University
Charlie Chen,	National Taiwan University
Eui-Young Chung,	Yonsei University
Maziar Goudarzi,	Kyushu University
Ing-Jer Huang,	National Sun Yat-Sen University
Shinji Kimura,	Waseda University
Tei-Wei Kuo,	National Taiwan University
Jing-Jia Liou,	National Tsing Hua University
John Moondanos,	Intel
Hamid Noori,	Kyushu University
Yunheung Paek,	Seoul National University
David Z. Pan,	The University of Texas at Austin
Sri Parameswaran,	University of New South Wales
Sachin Sapatnekar,	University of Minnesota
Kazunori Shimizu,	Waseda University
Sheldon Tan,	University of California, Riverside
Ren-Song Tsay,	National Tsing Hua University
Ting-Chi Wang,	National Tsing Hua University
Shigeru Yamashita,	Nara Institute of Science and Technology
Wenjian Yu,	Tsinghua University

Workshop Program

10:00 - 10:15
Opening

Hiroto Yasuura, Workshop Co-Chair

10:15-11:15
Session-1. Research Activities in an ASP region

10:15-10:35
"Taiwan's ESL R&D Status"
Ren-Song Tsay, National Tsing Hua University

10:35-10:55
"Recent research at Design Technology Lab., Yonsei University"
Eui-Young Chung, Yonsei University

10:55-11:15
"Recent Research Activities at System LSI Field of IPS in Waseda University"
Shinji Kimura, Waseda University

11:15-11:30
Coffee Break

11:30-12:10
Session-2. DFM Issues

11:30-11:50
"Recent Research on Physical CAD and DFM at UT Design Automation Lab"
David Z. Pan, The University of Texas at Austin

11:50-12:10
"On-Chip Variability in Nanometer-Scale Circuits"
Sachin Sapatnekar, University of Minnesota

12:10-14:15
Session-3. Poster Presentations with Lunch

P1 "Quantum Circuit Design with Decision Diagram Data Structure"
Shigeru Yamashita, Nara Institute of Science and Technology

P2 "Fast BEM algorithms for 3D interconnect capacitance and resistance extraction"
Wenjian Yu, Tsinghua University

P3 "A Diagnosis Framework for Defects on Path Delay Faults"
Jing-Jia Liou, National Tsing Hua University

P4 "The ODYSSEY Methodology: ASIP-based Design of Embedded Systems from Object-Oriented System-Level Models"
Maziar Goudarzi, Kyushu University

P5 "VLSI Implementation of QC-LDPC Code Decoder Accelerating Message-Passing Schedule"
Kazunori Shimizu, Waseda University

P6 "A Reconfigurable Functional Unit for an Adaptive Extensible Processor"
Hamid Noori, Kyushu University

P7 "Soargen: a Retargetable Compiler for ASIP Design"
Yunheung Paek, Seoul National University

P8 "Statistical Modeling, Timing Analysis, and Parasitics Extraction"
Charlie Chung-Ping Chen, National Taiwan University

14:15-15:15
Session-4. Place & Route

14:15-14:35
"NTUplace 3: A High-Quality Large-Scale Mixed-Size Placer"
Yao-Wen Chang, National Taiwan University

14:35-14:55
"Recent Advance in Terminal and Model Order Reduction for Interconnect Circuits"
Sheldon Tan, University of California, Riverside

14:55-15:15
"Via Doubling under Density Control"
Ting-Chi Wang, National Tsing Hua University

15:15-15:30
Coffee Break

15:30-16:10
Session-5. Embedded System Design

15:30-15:50
"Implementation and Challenging Issues of Flash-Memory Storage Systems of Embedded Systems"
Tei-Wei Kuo, National Taiwan University

15:50-16:20
"Hardware / Software support for Reliability and Security in Embedded Processors"
Sri Parameswaran, University of New South Wales

16:10-16:30
Coffee Break

16:25-17:05
Session-6. Leading Edge Design

16:25-16:45
"SoC Enabling Technologies"
Ing-Jer Huang, National Sun Yat-Sen University

16:45-17:05
"From error to Error: Verification & Debugging in the Multi-Core Era".
John Moondanos, Intel

17:05-17:20
Closing

Yusuke Matsunaga, Workshop Co-Chair

Recent research at Design Technology Lab., Yonsei University

Eui-Young Chung

School of Electrical & Electronic Eng.
Yonsei University

Outline

- Introduction
- Research topic overview
- On-chip communication architecture
- System-level low power design
- Flash memory from system perspective
- Summary

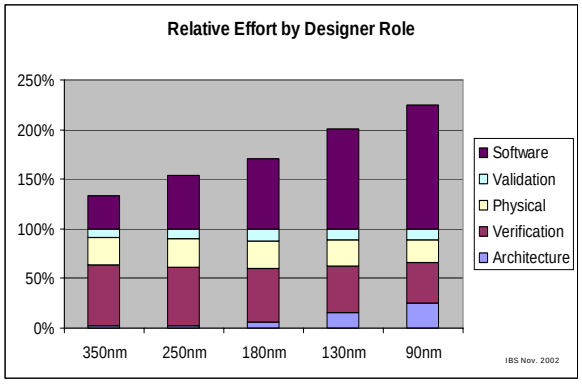
Introduction

- Design Technology Lab. (DTL)
  - Established in fall, 2005
  - # of students: 3 MS students
    - Collaboration is the viable solution
- Lab. mission
  - Develop the technology for the next generation SoC
    - Design methodology especially for system-level and architecture-level design
    - Architecture / RTL design
    - Integration of HW and SW
- Collabration partners
  - YU professors
  - Naehyuck Chang (SNU)
  - Sung Woo Chung (Korea University)
  - ...

Research topic overview

- Design technologies for portable devices
  - Low power
  - TTM
  - High performance
- To support these issues
  - System-level low power
  - System/architecture level design methodology
  - MP-SoC
  - Communication arch.
  - Memory hierarchy

Good reasoning for our topics

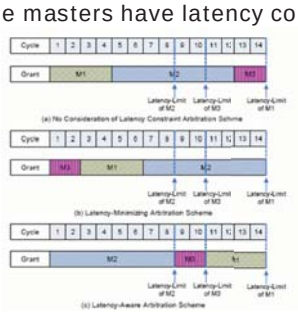


On-chip communication architecture (I)

- Bus
  - Improve the current practical solution
  - Focus on the arbitration scheme and automation
    - How to guaranttee the latency requirement as well as bandwidth requirement?
    - Automatic generation of RTL code
  - Architecture exploration
    - TLM using SystemC
- NoC
  - Prepare the solution for the next generation
  - AXI based switch design
  - Topology synthesis
    - Clock speed / hop-count trade-off

On-chip communication architecture (II)

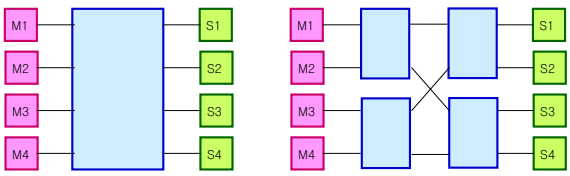
- When multiple masters have latency constraints



- Use the concept of “slack”
- Integration with bandwidth-conscious arbiter

On-chip communication architecture (III)

- Clock speed vs. hop count



- Which one is better?
  - Bandwidth utilization
  - Latency
  - Power

System level low power design

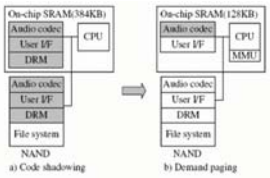
- Dynamic Power Management (DPM) and Dynamic Voltage Scaling (DVS) are still popular research topics
  - Adaptive DVS based on stochasitc approach
    - Algorithm development
    - Board-level implementation
  - Video format extension for precise execution time estimation
    - Decoding cost is included in video stream
  - DPM / DVS for MP-SoC
    - Considering interplay effect of processors
    - Feasibility study of DPM + DVS
    - Task partitioning with the consideration of DPM/DVS

Flash memory from system perspective (I)

- Flash memory is getting popular
  - NAND / NOR / OneNAND
  - MMC, SD card, ...
  - Hybrid-HDD
  - SSD
  - Flash + DRAM
- How can we utilize them efficiently?
  - NAND vs. OneNAND
    - Page access vs. random access
    - Random access is possible by XIP (eXecute-In-Place)
  - Transfer speed improvement for MMC, SD, ...
    - Timing issue due to asynchronous feature of NAND
    - PAD delay, PVT variations

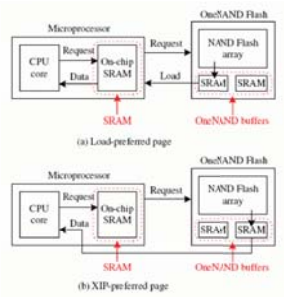
Flash memory from system perspective (II)

- Two ways to utilize NAND Flash
  - Code shadowing
    - Copy all pages to SRAM
  - Demand paging
    - Place necessary pages on SRAM
    - Trade-off between SRAM size and performance



Flash memory from system perspective (III)

- SRAM buuffers of OneNAND are treated as cache lines
- How to determine to put a page in a buffer or on-chip SRAM?



Summary

- Focusing on SoC for portable devices
  - On-chip communication architecture
  - Low power
  - Flash memory application
- Collaboration is welcome
  - Similar research topics
  - Strength in RTL and physical design
  - Strength in OS / Compiler

System LSI Design Workshop, 9/9/06

Recent Research on Physical CAD and DFM at UT Design Automation (UTDA) Lab

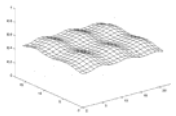
David Z. Pan
ECE Department, UT Austin
dpan@ece.utexas.edu
http://www.cerc.utexas.edu/utda

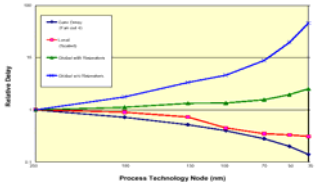
Support from SRC, IBM, Fujitsu, Sun, Intel, KLA-Tencor and Sigma-C

Overview of UTDA Lab

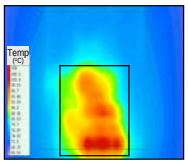
- UT VLSI Design Automation Lab
  - Founded in Sept. 2003 after DP joined UT from IBM T. J. Watson Research
  - Currently over a dozen highly motivated PhD students including part-timers from industry
- Key research objectives:
  - We aim at holistic modeling, characterization, and optimization of DFM
  - with other leading nanometer physical effects (timing, leakage, reliability, thermal, …)
  - while looking at emerging issues (up and down)…

Technology Trend/Challenges





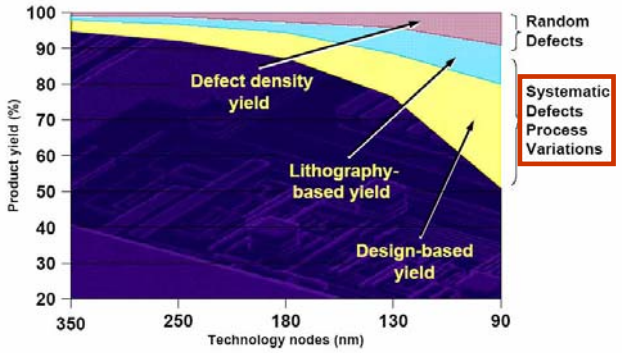
(source: ITRS)



(source: Intel)

- Manufacturability: WYS != WYG
- Interconnect determines the overall performance
- Power/leakage/thermal issues
- Other physical related issues: noise, reliability, …

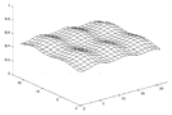
Yield Loss Projection Scary



[Courtesy IBS]

Litho Challenges are REAL

- 193nm lithography will continue as the main chip manufacturing workhorse for at least 5-7 years (thanks to RET and immersion litho…)
  - 45nm and even 32nm nodes
  - IBM news (02/06) of 29.9nm pattern
  - Nanolithography still many challenges
    - EUVL, E-Beam, nano-imprint…
- Live in deep sub-wavelength era
  - On top of DSM challenges
  - Has to be considered altogether
- Other DFM effects: CMP, VIA failure, …

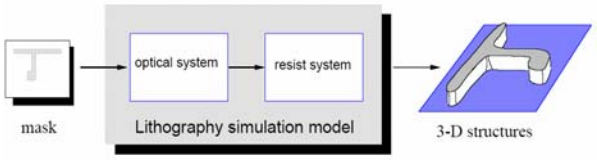


Recent Results I: DFM-Modeling

- Variational lithography modeling and process-variation aware OPC [Yu+, DAC'05, DAC'06, SPIE'06]
  - Ultra-fast litho-modeling with kernel decomposition and effective table lookup techniques
  - Analytical VLIM for two key variations: focus/dosage
  - Introduce a new concept for EPE: variational EPE for OPC
- Post-litho non-uniform gate modeling [Shi+, ICCAD'06]
  - Existing equivalent gate length method has fundamental flaw: two different EGLs (on and off) hard to assign *a priori*
  - Propose a new and more accurate, *current-based* method for unified modeling of timing and leakage.
- Accurate modeling/characterization => more accurate static/statistical timing/power analysis
  - A new sparse-matrix formulation for SSTA which can handle arbitrary correlations [Ramalingam+, ICCAD'06]

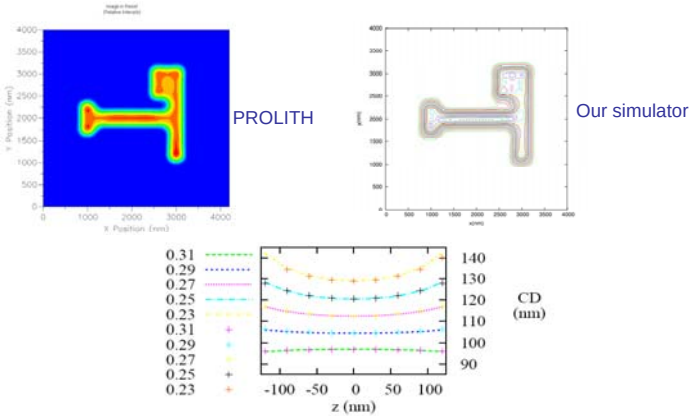
Lithography Modeling

- *Fast* yet *high-fidelity* lithography modeling essential
  - Our approach: design-oriented (vs. process-oriented) [Mitra et al, DAC'05]
- Process variations will affect printed image
  - Dosage, focus, mask, …
- Variational lithography modeling [Yu et al., DAC'06]
  - Our approach: variational kernel decomposition with moment expansion (vs. process window sampling)



Model Validation

- Validated with PROLITH (orders of 10^6 faster)



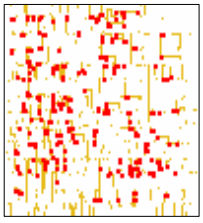
Recent Results II: DFM-Routing

- RET-aware routing with fast litho-modeling [Mitra+, DAC'05]
  - First introduced the litho-hotspot map concept
  - Ultra-fast litho-modeling
  - 40% reduction of litho-hotspot
- BoxRouter [Cho+, DAC'06]
  - A brand new global router that significantly pushes state-of-the-art (up to 16x faster, and 80% better in terms of routability)
  - Key ideas: Prerouting, box expansion, and progressive ILP
- CMP-aware routing [Cho+, ICCAD'06]
  - Predictive copper CMP modeling (with dummy insertion)
  - A unified wire-density driven global routing for CMP, timing …
  - Based on BoxRouter framework
- Redundant-via aware routing [Xu+, ASPDAC'05]
  - First work on redundant-via enhanced routing
  - Model redundant via cost/constraint during maze routing

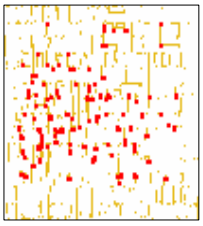
RADAR: RET-Aware Detailed Routing [Mitra et al, DAC'05]

- Raise lithography modeling up to design implementation level
  - Model-based vs. rule-based
- Conventional approaches to "separate" design from manufacturing – RULES
- Rules are starting running out of steam from 65nm
  - Exploding number of rules
  - VERY complicated rules (have you seen a Law book?)
  - Not accurate any more…
- Use our design-oriented lithography simulation to generate litho-hotspots and guide routing

Litho-aware Routing on a 65nm Industry Design



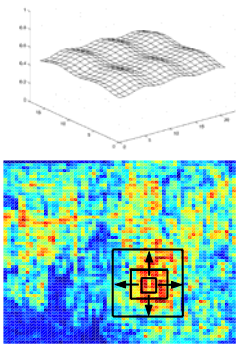
Initial routing (after design closure)



40% litho hotspot reduction

CMP-Aware Routing [ICCAD'06]

- Lithography interact with CMP
  - CMP => defocus
- CMP-Aware Routing [ICCAD'06]
  - Density-driven with predictive CMP model (validated with industry data)
  - Enhance the state-of-the-art BoxRouter [DAC'06]
  - 7.5-10% reduction in thickness var.
  - 7-10% improvement in timing



Recent Results III: Variation-Tolerant Design

- Variation reduction => variation-tolerant
- Clock network is the most sensitive to variations
- First work on temperature aware clock optimization (TACO) [Cho+, ICCAD'05]
  - Introduced a new concept of merging diamond (instead of merging segment as used in DME)
  - Variation balancing
- Clock tree link insertion and optimization – an affordable alternative to clock mesh [Rajaram+, ISPD'05, ISQED'06, ISPD'06]
  - Incremental algorithm [ISQED'06]
  - Buffered clock tree [ISPD'06]

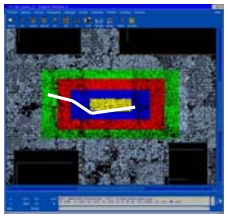
Recent Results IV: Design Closure

- DFM must be in context of DSM (design closure)
- Diffusion: a new paradigm for placement [DAC'05]
  - Nice property: good "relative orders" maintained
  - Great results and adopted in industry (IBM)
- Another stable approach for placement migration – computational geometry view [ICCAD'05]
  - Delaunay-triangulation to model neighborhood structure
- A new LP-based incremental TDP [DAC'06]
  - Hybrid path- and net-based approach
  - Optimize the critical adjacency network (not just critical paths)
  - Average 20ps worst slack improvement on multi-gHz uP designs
- Sensitivity-based netweighting [ISPD'04, TCAD'05]
- Crosstalk noise map driven placement [ICCAD'05]

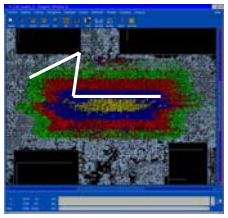
Recent Results IV: Design Closure

- Low power and thermal issues
  - Voltage islands [Puri+, DAC'03]
  - Power gating (sleep transistor) [ASPDAC'05]
  - A new analytical gate delay model for low-Vdd [ASPDAC'06]
  - Thermal modeling for non-uniform conductivity [ISQED'06]
  - Temperature aware clock optimization (TACO) [ICCAD'05]
  - Temperature aware scan optimization [VTS'06]
- Mixed signal-SOC floorplanning [ASPDAC'06]
  - We introduced a novel concept of Block Preference Directed Graph (BPDG) to model substrate noise in a VERY compact manner during floorplan evaluation
  - High fidelity with much more sophisticated model
  - 60x speedup

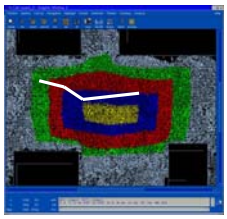
Diffusion-Based Placement



Initial placement (illegal)



Legalization from greedy alg.

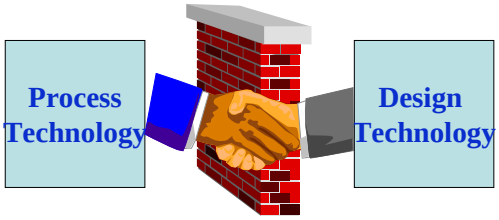


Diffusion Based Placement [Ren+, DAC'05]

Diffusion equation: $\frac{\partial d_{x,y}(t)}{\partial t} = D \nabla^2 d_{x,y}(t)$

Conclusion

- Holistic nanometer design + manufacturing closure
- Much more closer collaborations to break the red-brick wall
  - Between different "camps": designer, CAD, process
  - Between academia and industry
  - CMOS and beyond-CMOS



Quantum Circuit Design with Decision Diagram Data Structure

Shigeru Yamashita
Graduate School of Information Science,
Nara Institute of Science and Technology
ger@is.naist.jp

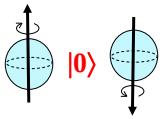
Contents
- What is Quantum Computation?
- What is Quantum Circuit Design?
  - Matrix Functions (MF)
  - Decision Diagram for MF (DDMF)
  - Quantum Circuit Design by DDMF

What is Q.C. in short?

To perform computation by using "quantum states"

Quantum States – 2 level physical system

- light phase (0° and 90°)
- spins of electron (↑ and ↓)
- energy level of quantum dots (ground and exited)
- NMR(average spin of 6×10^{23} molecules)



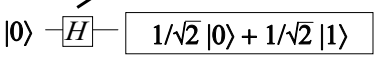
We can control quantum states by adding magnetic fields, irradiate an electromagnetic waves, ...

Quantum Bit $\alpha|0\rangle + \beta|1\rangle$ $\quad \sqrt{0.7}|0\rangle + \sqrt{0.3}|1\rangle$

Any superposition of two distinguishable quantum states

Quantum Bit & Quantum Gate

Quantum Gate = Certain Physical Operation



Superposition of |0⟩ and |1⟩ w.p. ½ each

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \quad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix} \quad H = 1/\sqrt{2}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$$

$$H\begin{bmatrix}1\\0\end{bmatrix} = 1/\sqrt{2}\begin{bmatrix}1\\0\end{bmatrix} + 1/\sqrt{2}\begin{bmatrix}0\\1\end{bmatrix}$$

- Quantum Bit: 1x2 column vector
- Quantum Gate: 2x2 Matrix

Logical Operations in Quantum Domain

$$N = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$$

Classical Operation

$|0\rangle \underset{NOT}{\Leftrightarrow} |1\rangle$ $\quad \begin{pmatrix}0&1\\1&0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$ $\quad \begin{pmatrix}0&1\\1&0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$

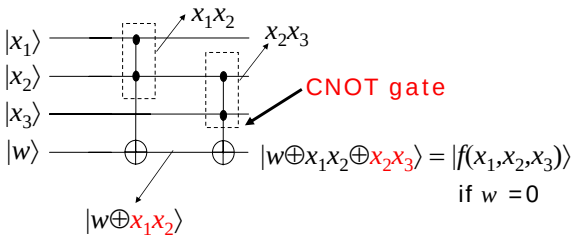
Design Boolean Circuit by Controlled-N can be considered in similar way as AND-EXOR synth.

Quantum Specific Operation

$$V = \frac{1+i}{2}\begin{pmatrix}1 & -i\\-i & 1\end{pmatrix} \quad V^{-1} = V^{\dagger} = \frac{1+i}{2}\begin{pmatrix}1 & -i\\-i & 1\end{pmatrix} \quad V^2 = N$$

We can also use Controlled-V in Q. C. Design

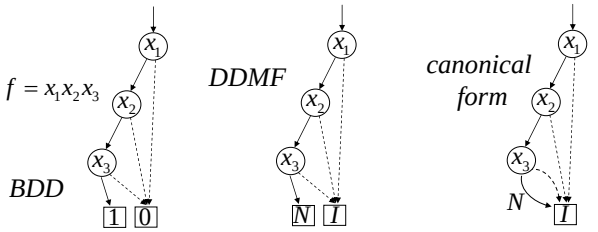
Quantum Circuit Design



- Any Boolean Function can be realized
- Classical Logic Design can be applied

We can also use Controlled-V in Q. C. Design

DDMF: Decision Diagram for Matrix Function



	f	MF
000	0	I
001	0	I
010	0	I
011	0	I
100	0	I
101	0	I
110	0	I
111	1	N

Matrix Function

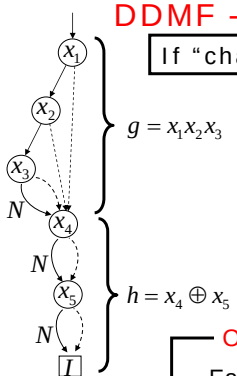
Mapping: $\{0,1\}^n \rightarrow$ 2x2 matrix

e.g.

$$MF(x_1,x_2,x_3)|0\rangle = |f(x_1,x_2,x_3)\rangle$$
$$MF(0,0,0)|0\rangle = I|0\rangle = |0\rangle = |f(0,0,0)\rangle$$
$$MF(1,1,1)|0\rangle = N|0\rangle = |1\rangle = |f(1,1,1)\rangle$$

DDMF -> Quantum Circuit

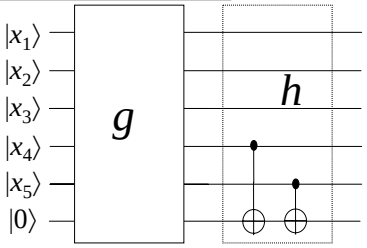
If "chain structure" exists



$g = x_1x_2x_3$

$h = x_4 \oplus x_5$

Easy



Observation 1

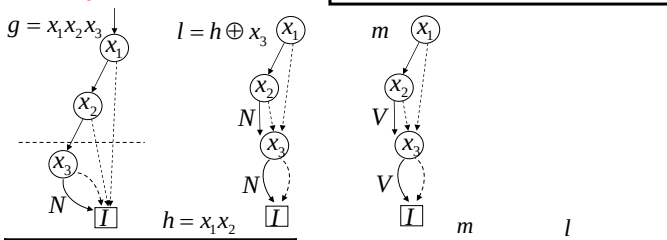
Easy to implement "Chain structure"

Observation 2

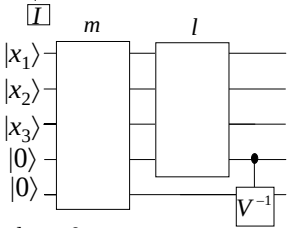
$f = g \oplus h \longrightarrow$ QC for f = (QC for g) + (QC for h)

Complicated Case

If "chain structure" exists



$x_1x_2x_3$	g	MF_g	h	l	MF_l	MF_m
000	0	I	0	0	I	I
001	0	I	0	1	N	V
010	0	I	0	0	I	I
011	0	I	0	1	N	V
100	0	I	0	0	I	I
101	0	I	0	1	N	V
110	0	I	1	1	N	V
111	1	N	1	0	I	N



$g = 1$ and $x_3 = 0$

$g = 1$ and $x_3 = 1$

Tsinghua University

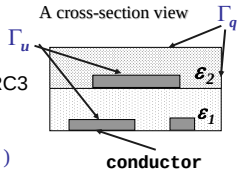
Fast BEM Algorithms for 3D Interconnect Capacitance and Resistance Extraction

Wenjian Yu

EDA Lab, Dept. Computer Science
& Technology, Tsinghua University
yu-wj@tsinghua.edu.cn

Direct BEM to solve Laplace Equ.

- Physical equations
  - Laplace equation within each subregion
  - Same boundary assumption as Raphael RC3
  - Bias voltages set on conductors



$$\begin{cases} \nabla^2 u = \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} + \dfrac{\partial^2 u}{\partial z^2} = 0, & \text{In } \Omega_i (i = 1, \cdots, M) \\ u = u_0, & \text{On } \Gamma_u \\ q = \partial u / \partial \boldsymbol{n} = q_0 = 0, & \text{On } \Gamma_q \end{cases}$$

(u is potential)

(q is normal electric field intensity)

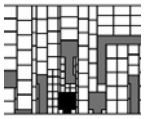
- Direct boundary element method
  - Green's Identity: $\int_\Omega (u\nabla^2 v - v\nabla^2 u)d\Omega = \oint_\Gamma (u\dfrac{\partial v}{\partial \boldsymbol{n}} - v\dfrac{\partial u}{\partial \boldsymbol{n}})d\Gamma$
  - Freespace Green's function as weighting function
  - Laplace equation is transformed into BIE:

$$c_s u_s + \int_{\partial\Omega_i} q_s^* u \, d\Gamma = \int_{\partial\Omega_i} u_s^* q \, d\Gamma$$ **s is a collocation point**

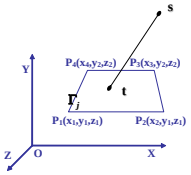
Discretization and integral calculation

- Discretize domain boundary
  - Partition quadrilateral elements with constant interpolation
  - Non-uniform element partition
  - Integrals (of kernel 1/r and 1/r³) in discretized BIE:

A portion of dielectric interface:



$$c_s u_s + \sum_{j=1}^{N} \left(\int_j q_s^* d\Gamma \right) u_j = \sum_{j=1}^{N} \left(\int_j u_s^* d\Gamma \right) q_j$$



- Singular integration
- Non-singular integration
  - Dynamic Gauss point selection
  - Semi-analytical approach improves computational speed and accuracy for near singular integration

Locality property of direct BEM

- Write the discretized BIEs as:

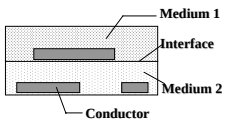
$$\boldsymbol{H}^i \cdot \boldsymbol{u}^i = \boldsymbol{G}^i \cdot \boldsymbol{q}^i, \quad (i=1, \ldots, M)$$

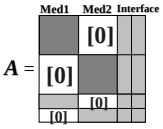
Compatibility equations along the interface

$$\begin{cases} \varepsilon_a \cdot \partial u_a / \partial \boldsymbol{n}_a = -\varepsilon_b \cdot \partial u_b / \partial \boldsymbol{n}_b \\ u_a = u_b \end{cases}$$

$$\boldsymbol{Ax} = \boldsymbol{f}$$

- **Non-symmetric large-scale matrix *A***
- **Use GMRES to solve the equation**
- **Charge on conductor is the sum of *q***

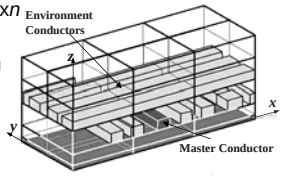




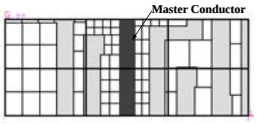
For problem involving multiple regions, matrix *A* exhibits sparsity!

Quasi-multiple medium method

- Quasi-multiple medium (QMM) method
  - Cutting the original dielectric into *mxn* fictitious subregions, to enlarge the matrix sparsity in BEM computation
  - With iterative equation solver, sparsity brings actual benefit



A 3-D multi-dielectric case within finite domain, applied 3×2 QMM cutting



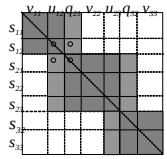
Non-uniform element partition on a medium interface

Strategy of QMM-cutting:

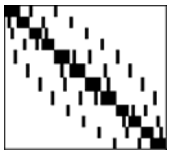
- ✓ **Uniform spacing**
- ✓ **Empirical formula to determine (m, n)**
- ✓ **Optimal selection of (m, n)**

Efficient equation organization

- Too many subregions produce complexity of equation organizing and storing
- Bad scheme makes non-zero entries dispersed, and worsens the efficiency of matrix-vector multiplication in iterative solution
- We order unknowns and collocation points correspondingly; suitable for multi-region problems with arbitrary topology
- Example of matrix population



Three stratified medium

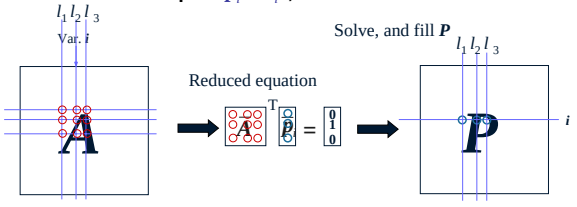


12 subregions after applying 2×2 QMM

This ensures a near linear relationship between computing time and non-zero entries

Efficient GMRES preconditioning

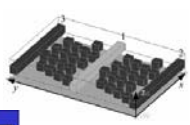
- Construct MN preconditioner [Vavasis, ***SIAM J. Matrix*,1992]**
  - $\boldsymbol{PA} = \boldsymbol{I} \Leftrightarrow \boldsymbol{A}^T \boldsymbol{P}^T = \boldsymbol{I} \Leftrightarrow \boldsymbol{A}^T \boldsymbol{p}_i = \boldsymbol{e}_i, \; i = 1, \ldots, N$
  - **Neighbor set of variable *i*:** $L = \{l_1, l_2, \ldots, l_n\} \subset \{1, 2, \ldots, N\}$
  - **Solve reduced eq.** $\bar{\boldsymbol{A}}^T \bar{\boldsymbol{p}}_i = \bar{\boldsymbol{e}}_i$ **, fill back to *i*th row of *P***

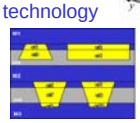


- Our work: for multi-region BEA, propose an approach to get the neighbors, making solution faster for 30% than original Jacobi preconditioner

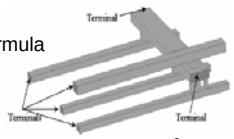
A practical field solver - QBEM

- Handling of complex structures
  - Bevel conductor line; conformal dielectric
  - Structure with floating dummy fill
  - Multi-plane dielectric in copper technology
  - Metal with trapezoidal cross section





- 3-D resistance extraction
  - Complex 3-D structure with multiple vias
  - Improved BEM coupled with analytical formula
  - Extract DC resistance network
  - Hundreds/thousands times fast than Raphael, while maximum error <3%



THE ODYSSEY METHODOLOGY: ASIP-BASED DESIGN OF EMBEDDED SYSTEMS FROM OBJECT-ORIENTED SYSTEM-LEVEL MODELS[1]

Maziar Goudarzi
System LSI Research Center
Kyushu University, Fukuoka, Japan
gudarzi@slrc.kyushu-u.ac.jp

Abstract

The object-oriented design methodology has long been used by the software community. Although it is believed that the OO methodology has been inspired by the traditional style of hardware design (interconnecting several integrated circuits as black boxes each having a well-defined function and pin configuration for easy and independent replacement or upgrade), it has not been widely adapted in VLSI design. Many researchers have tried to simulate and synthesise hardware from object-oriented models, but the resulting languages and tools have remained in limited use. We believe that this has been due to *(i)* thinking of objects mainly as structural components, resulting in hardware-incompatible interpretation of essential OO features such as polymorphism and dynamic (de)allocation; *(ii)* too verbose languages and cumbersome modelling styles, negatively affecting practicality of the approach; *(iii)* unacceptable area/power overhead, questioning suitability of the OO methodology for VLSI design; *(iv)* nonexistent or unclear path toward system (i.e. mixed hardware and software) implementations, limiting the approach only to hardware where the primary issue is performance not the extendibility, flexibility and maintainability that OO offers; *(v)* designers' reluctance to learn and use the OO modelling paradigm.

In this work, we introduce the ODYSSEY methodology for the design of embedded hardware-software systems. ODYSSEY directly addresses the above first four issues and leaves the fifth to the technology trend: we believe (and contemporary evidence confirms) that the VLSI design community shall move to system-level design where the design team will consist of "system-designers", not two collaborating teams of "hardware designers" and "software designers"; such system designers shall be closer to today software designers and more sympathetic to OO techniques.

ODYSSEY targets embedded "systems", not merely "hardware", and gives special attention to "software" that nowadays accounts for 80% of embedded systems development cost. Traditionally, the "hardware" has been the primary element in embedded system development; the "software" has been later adapted to the available hardware components and interface. ODYSSEY suggests the other way around: "software" should be the primary issue and the "hardware" must be designed such that the software can easily and effectively use it. To accomplish this, we introduce "Object-Oriented Application-Specific Instruction Processor", or OO-ASIP [2], which abstractly represents a processor whose instruction-set consists of methods of the class library which is used to develop the software; this identifies a class library with an OO-ASIP while a larger class library can be partitioned to an OO-ASIP and the program for it, and consequently, when the class library is augmented for future related applications, the same OO-ASIP can still serve the new ones. Moreover, the OO-ASIP instructions can even be "virtual methods" that may despatch to different implementations (including software implementations) according to the run-time class of the called object. This feature *(i)* allows the instruction-set to be extended without introducing new opcodes and hence without modifying the instruction-decoding circuitry; *(ii)* allows faulty method implementations to be overridden by software patches.

In this work we investigate *(i)* how the OO-ASIP conceptually provides a close coupling between hardware- and software-design; *(ii)* practical effects of this linkage on embedded system development time and cost, especially for a family of similar systems; and finally *(iii)* various platform architectures improving hardware features of the OO-ASIP such as area, power, and performance; this specifically includes a Network-on-Chip platform that realises virtual-method dispatch for no extra hardware and hence

[1] This work [1] was done when the author was with Department of Computer Engineering, Sharif University of Technology, Tehran, I.R.Iran.

imposes no overhead [3]; this addresses the most critical shortcoming, i.e. polymorphism overhead, that all previous approaches to OO VLSI design have suffered from.

We implement a prototype tool chain for our ODYSSEY methodology [4] and present experimental results arising from implementing some real systems using this methodology [5-7]. As typically expected in hardware-software co-designs, for a *single* given application we achieve area/power/performance figures somewhere in between full-hardware and full-software implementations but the design-effort and design-time may be closer to the full-hardware one; however, for a *family of related applications* (such as a generation of incrementally enhancing products typically seen in consumer electronics), the methodology shows its advantage and achieves near-to-full-software total cost and time-to-market, in addition to providing the ease of development and maintenance offered by the OO techniques; this is achieved since the OO-ASIP designed for the first member of the family is reusable for the rest of them quite in the same way that the class library designed for that first member is reusable to model the rest.

Furthermore, this work opens up the way to revisit computer organization components so as to propose customisations enabled by the fact that solely OO software is to be run in this system. One such proposal is a customised prefetching policy for data cache to increase hit ratio [8]. Since the processor is aware of the class method to be run next, and the sets of data elements that the method accesses is also known, these data can be prefetched by hardware [9,10] without the overhead associated with software-directed prefetching policies.

Research on incorporating IP cores in the OO-ASIP design flow, on automatically generating efficient co-simulation models at various and mixed abstraction levels, on investigating theory and implementation for multi-OO-ASIP and multi-processor-core implementations, and on test and fault-tolerance techniques [11] for OO-ASIP-based systems is going on in Department of Computer Engineering at Sharif University of Technology.

References

[1] M. Goudarzi, "The ODYSSEY Methodology: ASIP-Based Design of Embedded Systems from Object-Oriented System-Level Models," PhD Thesis, Sharif University of Technology, June 2005.

[2] M. Goudarzi, S. Hessabi, A. Mycroft, "Object-Oriented Embedded System Development Based on Synthesis and Reuse of OO-ASIPs," *Journal of Universal Computer Science*, vol. 10, no. 9, pp. 1123-1155.

[3] M. Goudarzi, S. Hessabi, A. Mycroft, "No-Overhead Polymorphism in Network-on-Chip Implementation of Object-Oriented Models," Proc. Of *Design Automation and Test in Europe (DATE'04*), Paris, February 2004.

[4] M. Goudarzi, S. Hessabi, "The ODYSSEY Tool-Set for System-Level Synthesis of Object-Oriented Models," in Proc. of *SAMOS V: Embedded Computer Systems: Architectures, MOdeling, and Simulation*, Greece, Springer-Verlag LNCS 3553, pp. 394-403, July 2005.

[5] M. Najafvand, "Design and Implementation of an OO-ASIP for JPEG Decompression," MSc Thesis, Sharif University of Technology, 2006.

[6] N. MohammadZadeh, "Extending a JPEG OO-ASIP to Implement MPEG2 Decoding," MSc Thesis, Sharif University of Technology, 2006.

[7] N. MohammadZadeh, S. Hessabi, M. Goudarzi, "Software Implementation of MPEG2 Decoder on an ASIP JPEG Processor," Proc. of *International Conference on Microelectronics (ICM'05)*, Islamabad, Pakistan, December 2005.

[8] M. Goudarzi, S. Hessabi, A. Mycroft, "Object-aware Cache: Higher Cache Hit-ratio in an Object-Oriented ASIP", Proc. of *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'04*), Ontario, Canada, May 2004.

[9] M. Modarressi, M. Goudarzi, S. Hessabi, "Application-Specific Hardware-Driven Prefetching To Improve Data Cache Performance," Proc. of *Asia-Pacific Computer System Architecture Conference (ACSAC'05)*, Singapore, Springer-Verlag LNCS, October 2005.

[10] M. Modarressi, M. Goudarzi, S. Hessabi, "A Table-Based Application-Specific Prefetch Engine for Object-Oriented Embedded Systems," Proc. of *IEEE IC-SAMOS conference*, Greece, 2006.

[11] S. Hessabi, A.M. Gharehbaghi, B. Hamdin Yaran, M. Goudarzi, "Integrating Assertion-Based Verification into System-Level Synthesis Methodology," Proc. of *International Conference on Microelectronics (ICM'04)*, pp. 232-235, Tunisia, December 2004.

VLSI Implementation of QC-LDPC Code Decoder Accelerating Message-Passing Schedule

Kazunori Shimizu†

† Graduate School of IPS, Waseda University

Waseda University

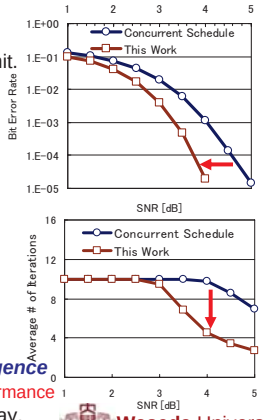
Research Background

Low-Density Parity-Check Code:
is an error correcting code which achieves information rates very close to the Shanon limit.

Message-Passing Algorithm:
is an iterative algorithm for decoding LDPC codes, is composed of row operation and column operation.
Inherent parallelism of the algorithm makes it suitable for hardware design.

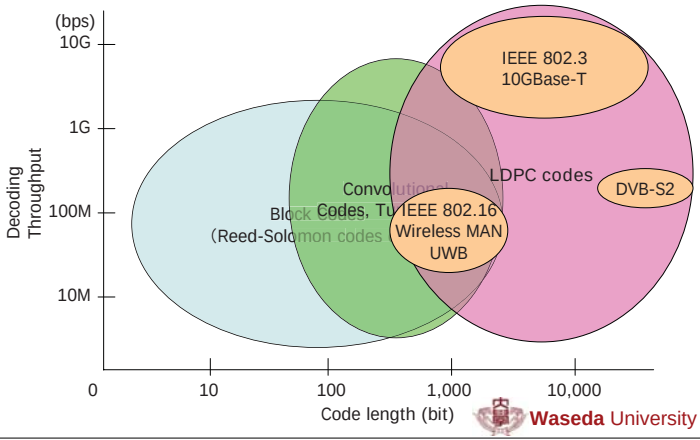
Motivation:
Requirement is performance gain (bit error performance and data rate) with a small hardware overhead.

This Work: Accelerating decoding convergence
enables the decoder to improve Bit Error Performance and Decoding Throughput within a limited delay.



Waseda University

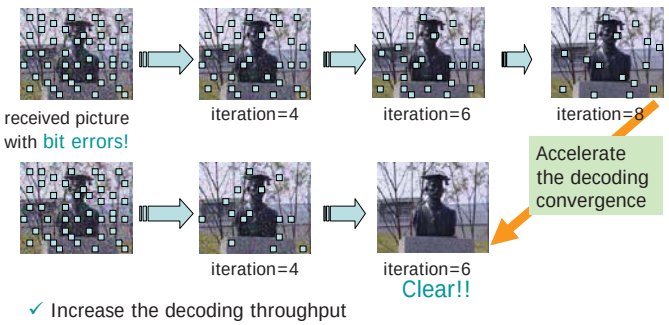
LDPC Codes



Waseda University

Our approach

- Improve the message-passing schedule

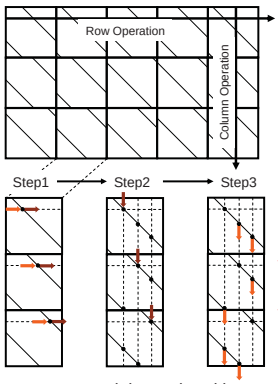


✓ Increase the decoding throughput
✓ Improve the bit error performance within a limited decoding delay time.

Waseda University

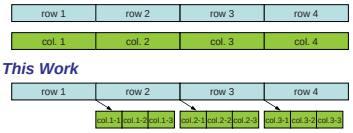
Accelerating Message-Passing Schedule

Parity check matrix for hardware sharing



message alpha
Row Operation ⇄ Column Operation
message beta

Concurrent Schedule



This Work

✓ The updated message by the row operations are fed into the column operation immediately after the row operations.

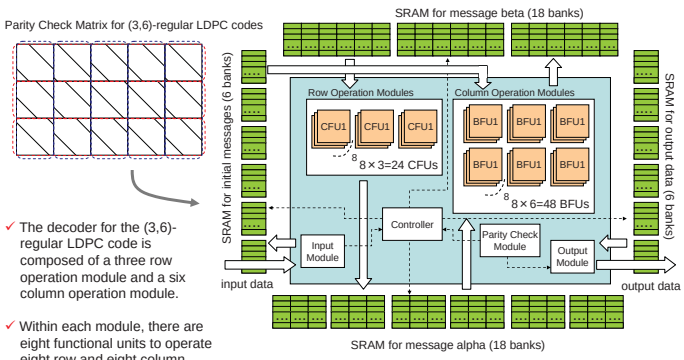
✓ The number of column operations increases three times compared to the concurrent schedule.

➡ message alpha updated by row operation
➡ message beta updated by column operation.

Waseda University

Block Diagram of LDPC Decoder

LDPC Decoder



✓ The decoder for the (3,6)-regular LDPC code is composed of a three row operation module and a six column operation module.

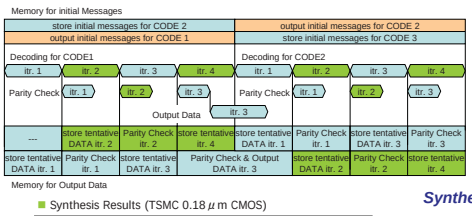
✓ Within each module, there are eight functional units to operate eight row and eight column operations in parallel.

BFU : Bit Functional Unit
CFU : Check Functional Unit

Waseda University

Experimental Result

Timing diagram of the LDPC decoder



✓ SRAM for initial message and output data stores two LDPC codewords so that the decoding hardware can be fully used.

✓ Parity check in each iteration.

✓ Switches the decoding codeword when all bit errors in each codeword are decoded correctly.

■ Synthesis Results (TSMC 0.18 μm CMOS)

	Logic[μm²]	SRAM[μm²] (# of Banks)	Total[μm²]
Concurrent Schedule	3,448,747	7,113,932 (48)	10,562,680
Proposed Schedule	4,124,134	7,113,932 (48)	11,238,066

■ Experimental Results (SNR=4.5, Iteration limit=10, @120[MHz])

	Power [mW]	Average #of Iterations	Throughput [Mbps]	Power/Thr. [mW/Mbps]	Bit Error Rate
Concurrent Schedule	327.4	8.536	48	6.8	0.000141
Proposed Schedule	528.9	3.391	122	4.3	0.000000

Synthesis & Experimental Results

✓ Decoder core based on the proposed schedule increases about 6% in the total area.

✓ Decoding throughput is about 2.54 times faster compared to the concurrent schedule.

✓ Power efficiency [mW/Mbps] can be improved up to 37%.

Waseda University

LDPC Decoder Chip

Summary of the LDPC Decoder Chip

Code length	3,072 [bits]
Code rate	0.5, (3,6)-regular
Design process	0.18um, 6Metal,CMOS
Chip size	5.0mm* 5.0mm
Gate count	96,945 gates (Decoder Core)
# of CFU	3 * 8 = 24
# of BFU	6 * 8 = 48
Total SRAM Area	7,113,932 [μm²]
PLL Area	266,136 [μm²]
Chip Density	49%
Clock Frequency	120 [MHz] (Max)
Throughput	122 [Mbps] (@120MHz, SNR=4.5)
Power Consumption	529 [mW] (@120MHz, 1.62V)

Chip Micrograph

DAC/ISSCC Student Design Contest
1st place (conceptual category)

Waseda University

A RECONFIGURABLE FUNCTIONAL UNIT FOR AN ADAPTIVE DYNAMIC EXTENSIBLE PROCESSOR

Hamid Noori, Farhad Mehdipour, Kazuaki Murakami and Koji Inoue

Department of Informatics, Graduate School of Information of Science and Electrical Engineering, Kyushu University

1. INTRODUCTION

One method for providing enhanced performance is application-specific instruction set extension. In this method, the critical portions of an application's dataflow graph (DFG) can be accelerated by mapping them to custom functional units. Instruction set extension improves performance and also maintains a degree of system programmability, which enables them to be utilized with more flexibility. The main problem with this method is that there are significant non-recurring engineering costs associated with their implementation.

In our approach, an **A**daptive dyna**M**ic extensi**B**l**E** processo**R** (AMBER) is presented in which the custom instructions (CIs) are adapted to the target applications and generated after chip-fabrication, fully transparently and automatically. This approach reduces the design time and cost drastically. Our CIs are generated by exploiting the HBBs. An HBB is a basic block that is executed more than a given threshold. We propose an RFU to support a wide range of generated CIs. Our 8-input, 6-output RFU is a coarse grain accelerator based on a matrix of functional units (FUs). It is tightly coupled with the base processor. In this method, there is no need to add extra opcodes for CIs, develop a new compiler, change the source code and recompile it.

2. GENERAL OVERVIEW OF AMBER ARCHITECTURE

AMBER has been developed by integrating a base processor (4-issue in-order RISC processor) with a reconfigurable functional unit (RFU) (Fig.1). AMBER has two operation modes: *training mode* and *normal mode*.

3. TOOL FLOW

We followed a quantitative approach by applying the flow in Fig. 2 for designing RFU, using 22 applications of Mibench [12]. Simplescalar [13] was utilized as our simulator.

4. PROPOSED ARCHITECTURE

Fig. 3 depicts our proposed architecture for the RFU. It has eight inputs and 6 outputs.

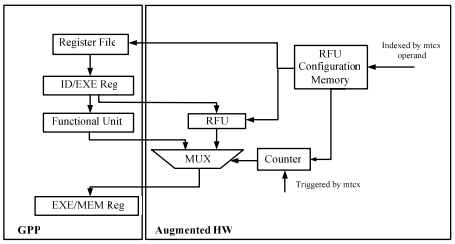


Fig. 1. Integrating RFU with the base processor

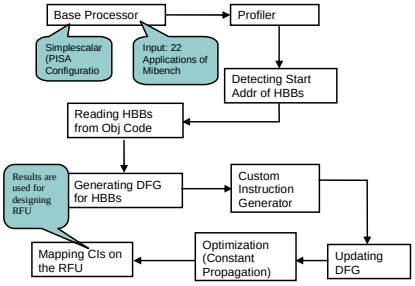


Fig. 2. Tool Flow

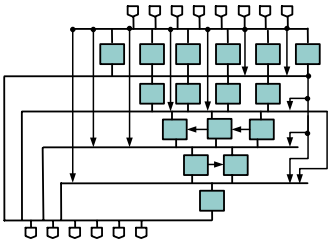


Fig. 3. Optimized RFU architecture

5. EXPERIMENTAL RESULTS

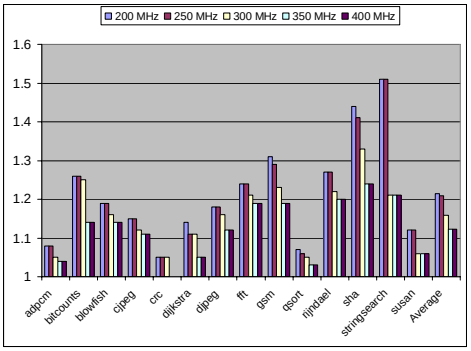


Fig. 4. Speedup for some of Mibench applications