Reconfigurable Computing for System on a Chip

Hiroto Yasuura
Kazuaki Murakami
System LSI Research Center (SLRC)
Kyushu University
E-mail: yasuura@slrc.kyushu-u.ac.jp

Outline

- Background and Requirements
- Platforms for Reconfigurable Computing
  - DRP
  - DAP/DNA
- How to use Reconfigurable Computing in SoC
  - SysteMorph
- Conclusion

Why Reconfigurable?

- Cost of Production
  - Drastic increase of design and mask cost is requesting new system architectures, especially for small scale production less than 1M.
- Customer Satisfaction
  - Various kinds of customers, each of which has different requirement and knowledge. A customized system for each user is attractive.
- Market Oriented SoC Design
  - The direction of the market changes quickly and various new services are introduced.
- Reliability and Security
  - Repairs and debugging on customer site.
  - Changing system configuration for security. (cryptography etc.)
- Global Environment Problem
  - Grow out of the throwaway society.
- Views of System Designers and Users

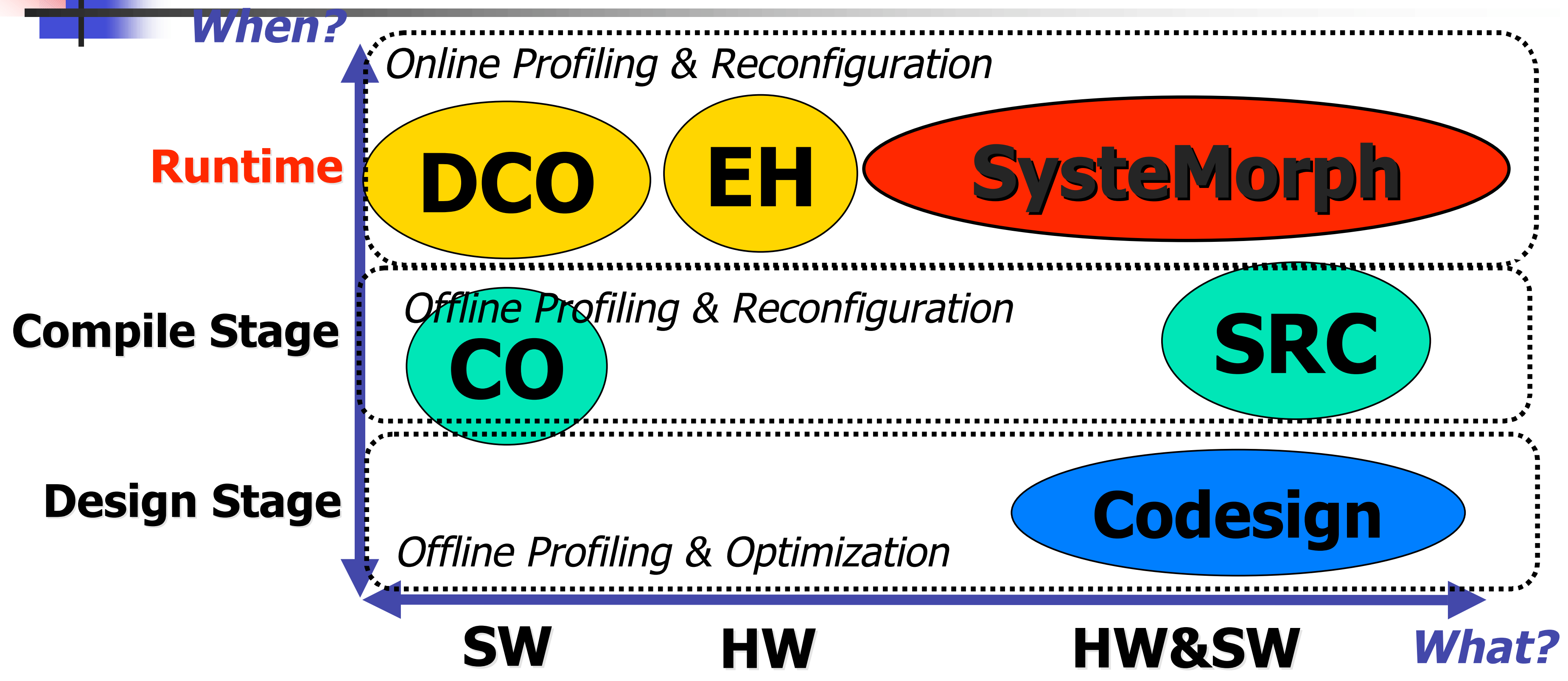
Example: Mobile Phone

- New Services
  - I-mode (Internet Access: e-mail and WWW)
  - Built-in Digital Still Camera
  - Video Phone Service (MPEG-4 in NTT Foma)
  - Melody Calling
  - Music Down Load Service(MP3)
  - Electric Ticketing
  - Electric Money for Vending Machines
  - Simple Interface for Old People
  - Car Navigation Service
- Needs for a new system architecture solution
  - Reconfigurable computing is a possible solution.

System Level Optimization

- What
  - Parameters for optimization
  - Goals of optimization　QoS (Function, Performance, Energy, Reliability, Security,…)
- When
  - Design Stage, Compilation Stage, and Runtime
- Who
  - Designers, Service Providers, and also Users
- How
  - Reconfigurable Hardware Platforms
  - Software
  - Profiling and Design Optimization Techniques

Reconfigurable Computing


When?
Online Profiling & Reconfiguration
Runtime
DCO
EH
SysteMorph
Compile Stage
Offline Profiling & Reconfiguration
CO
SRC
Design Stage
Codesign
Offline Profiling & Optimization
SW
HW
HW&SW
What?

DCO: Dynamic Compilation/Optimization
CO: Compiler Optimization
EH: Evolvable Hardware
SRC: Static Reconfigurable Computing

By K. Murakami

Runtime Reconfiguration

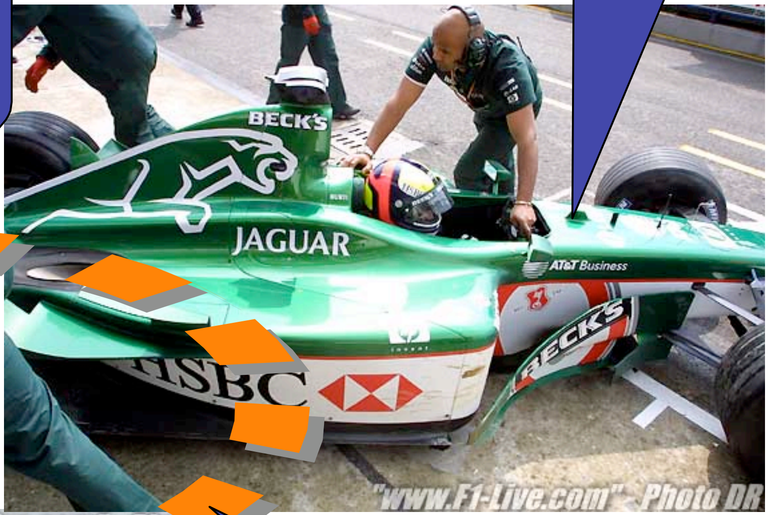
- Dynamic: Optimization is performed…
  - After SoC is shipped to the market
  - While SoC is used in the field
- Online: Profiling and optimization are performed…
  - In parallel with the execution of application programs
  - During idle or sleeping time
- Adaptive: Optimization is repeated…
  - In the form of a feedback loop

Analogy: Formula 1

The car is running in the course.

After the reconfiguration, the car returns to the course.

The car is now under reconfiguration

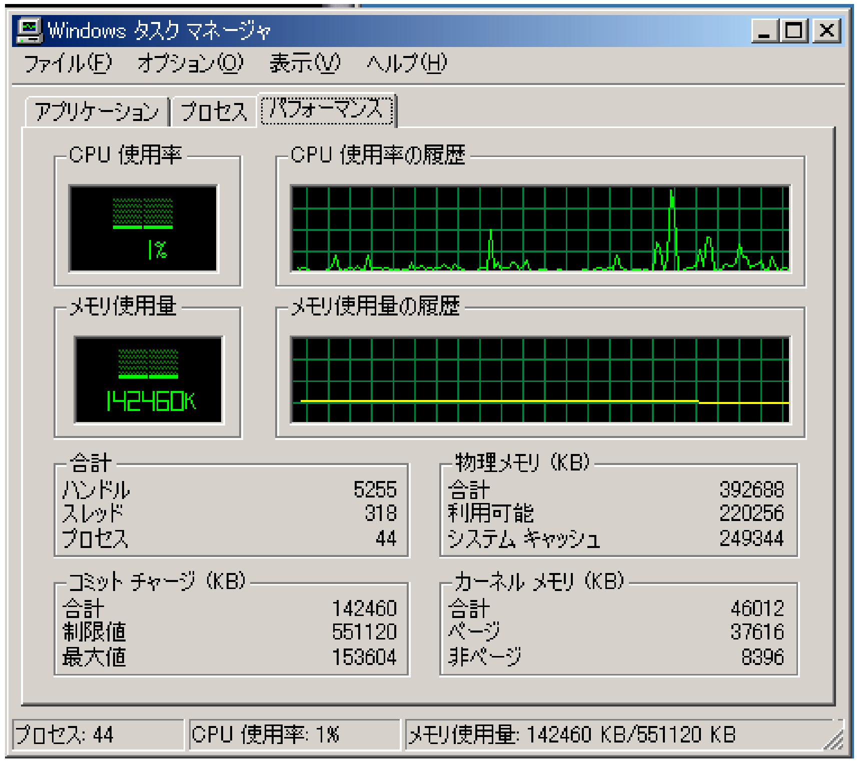


Once the pit crew finds any hints for reconfiguration, the car pits in

The pit crew is monitoring the behavior of the car

By K. Murakami

When reconfiguration is done?


Windows タスク マネージャ

ファイル(F) オプション(O) 表示(V) ヘルプ(H)

アプリケーション | プロセス | パフォーマンス

CPU 使用率: 1%

CPU 使用率の履歴

メモリ使用量: 142460K

メモリ使用量の履歴

合計	
ハンドル	5255
スレッド	318
プロセス	44

物理メモリ (KB)	
合計	392688
利用可能	220256
システム キャッシュ	249344

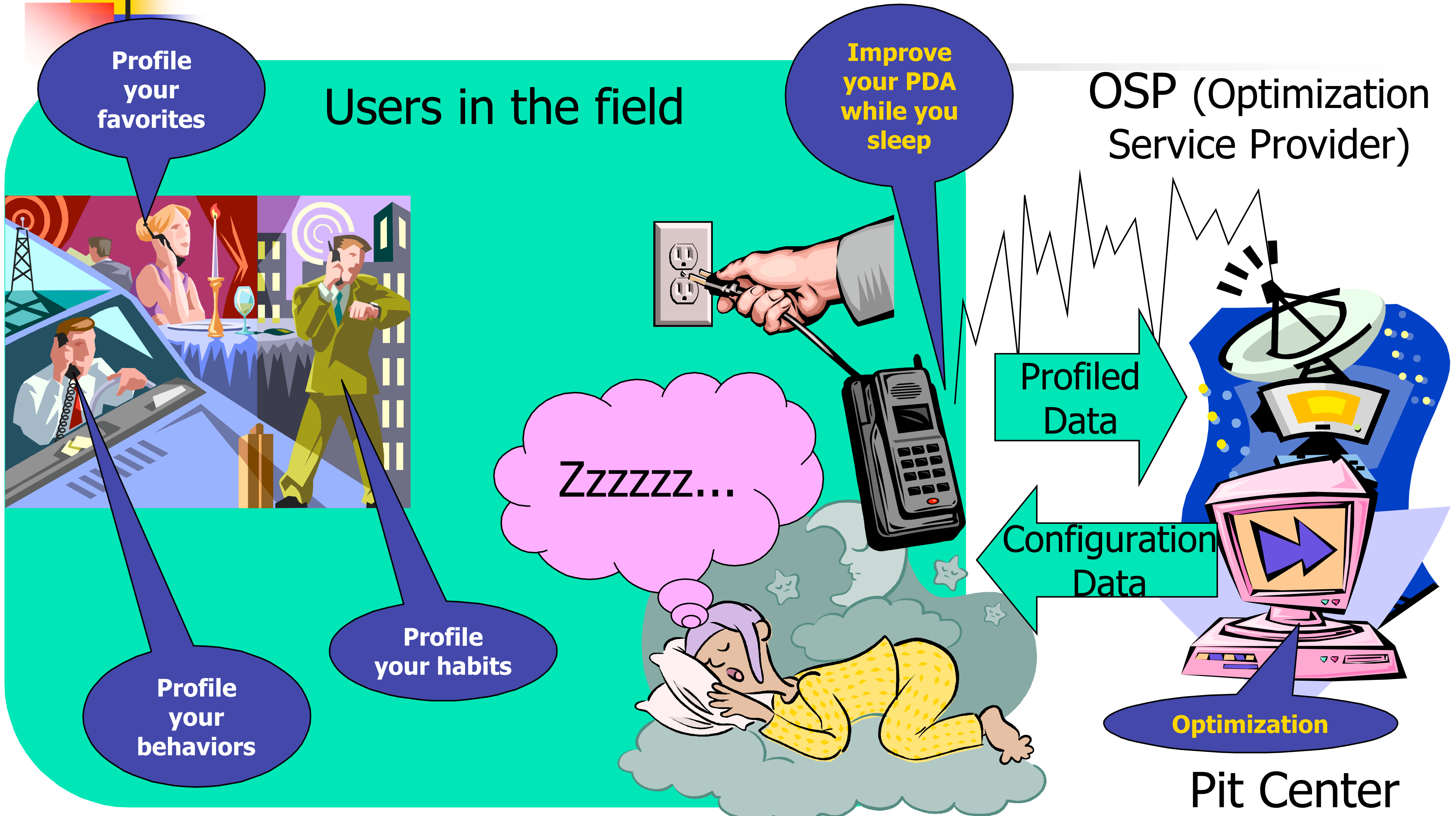
コミット チャージ (KB)	
合計	142460
制限値	551120
最大値	153604

カーネル メモリ (KB)	
合計	46012
ページ	37616
非ページ	8396

プロセス: 44 | CPU 使用率: 1% | メモリ使用量: 142460 KB/551120 KB

A system is not always active. Reconfiguration can be done in idle and sleep time.

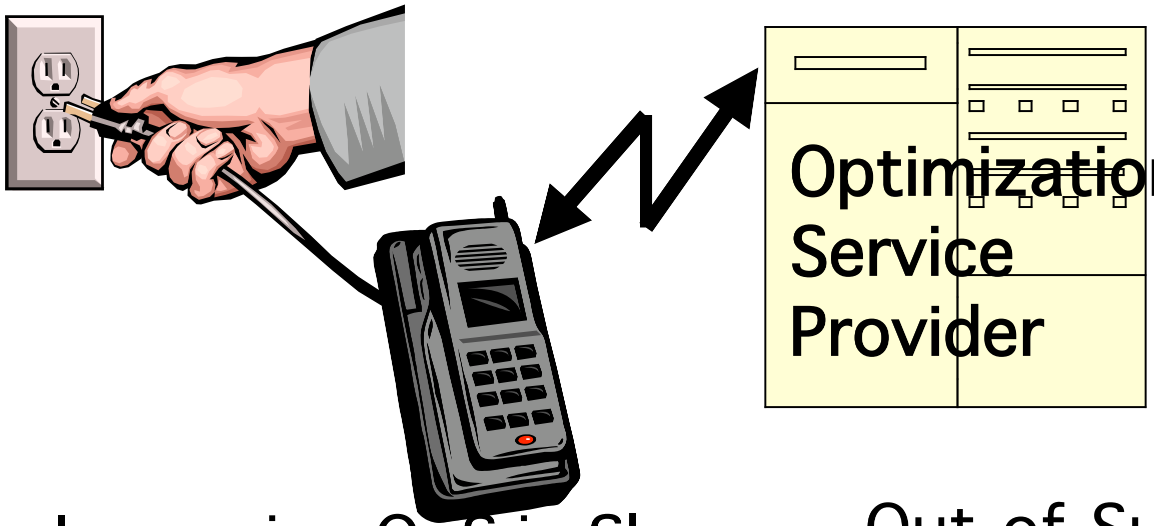
A Possible Business Model: OSP (Optimization Service Provider)



Customizable Mobile Phone

Your phone is evolving every battery charging!

On-line Profiling



Improving QoS in Sleep

- Sound quality
- Battery life
- Key operation
- New services
- Debugging

Out-of-Suit Optimization

Reconfigurable Computing for System on a Chip

- Background and Requirements
- Platforms for Reconfigurable Computing
  - DRP
  - DAP/DNA
- How to use Reconfigurable Computing in SoC
  - SysteMorph
- Conclusion

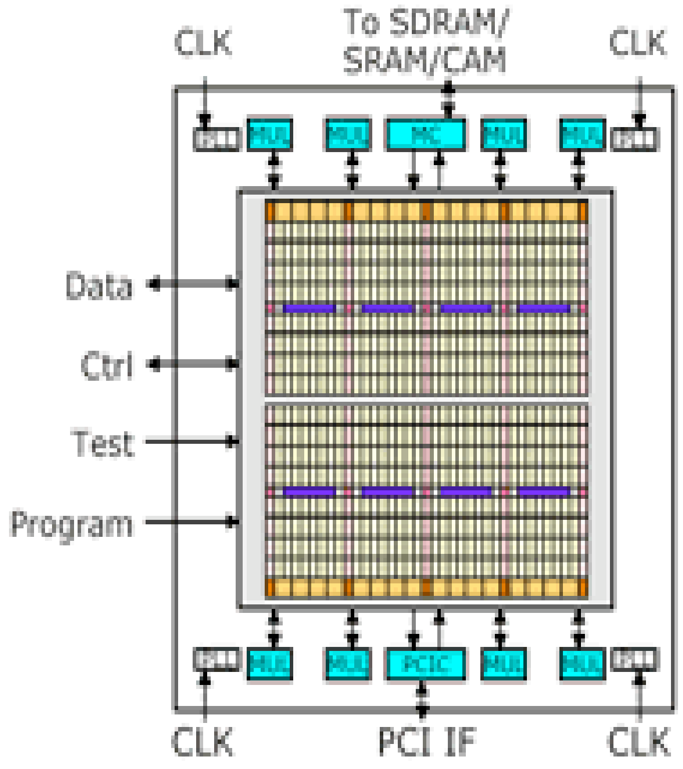
Platforms for Reconfigurable Computing

- Dynamically Reconfigurable Processor: DRP by NEC
- DAP/DNA by IP Flex
- Dynamically Reconfigurable Circuits by Sony

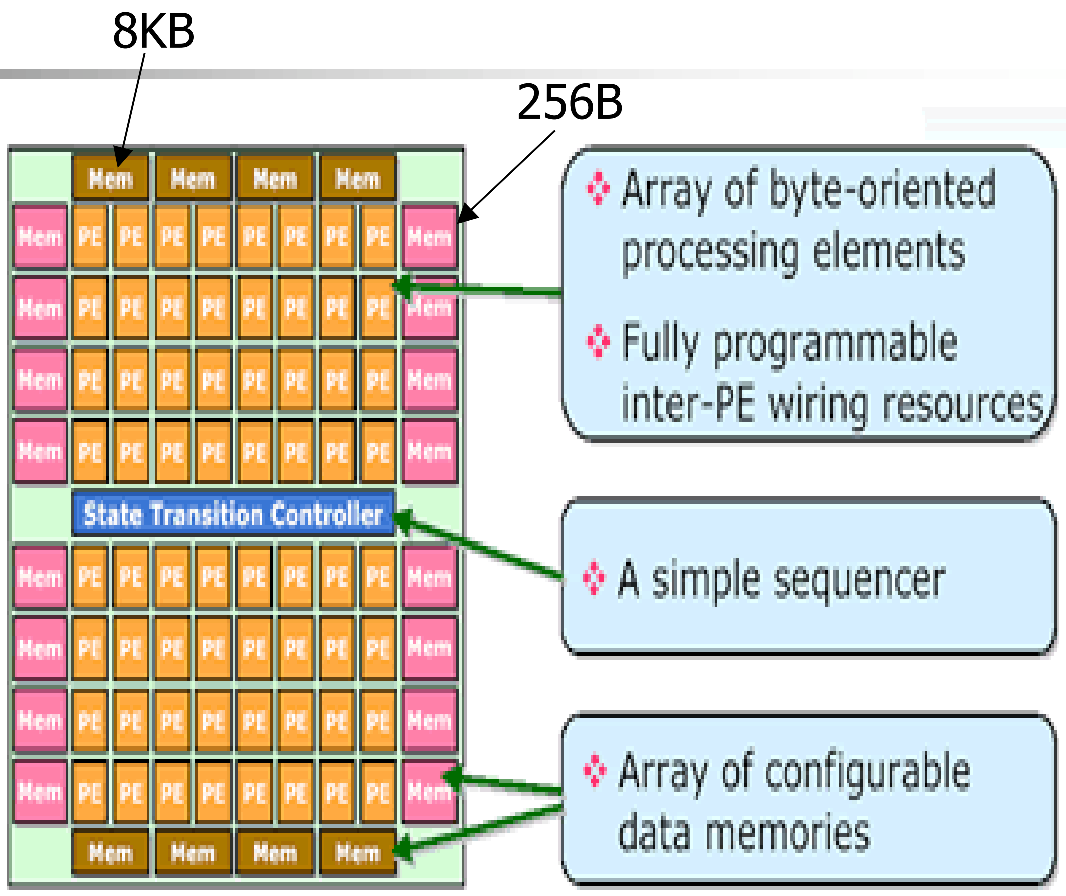
Platforms for Reconfigurable Computing

- Granularity of Reconfiguration
  - A Processor and Software
  - An Processor Array
  - Processing Elements
    - ALU, Multipliers, etc.
  - Logic Gates (FPGA)
- Timing of Reconfiguration
  - Every Clock Cycle
  - Every Task Execution
  - Every Power-on

Dynamically Reconfigurable Processor:DRP



8DRP cores on a Chip

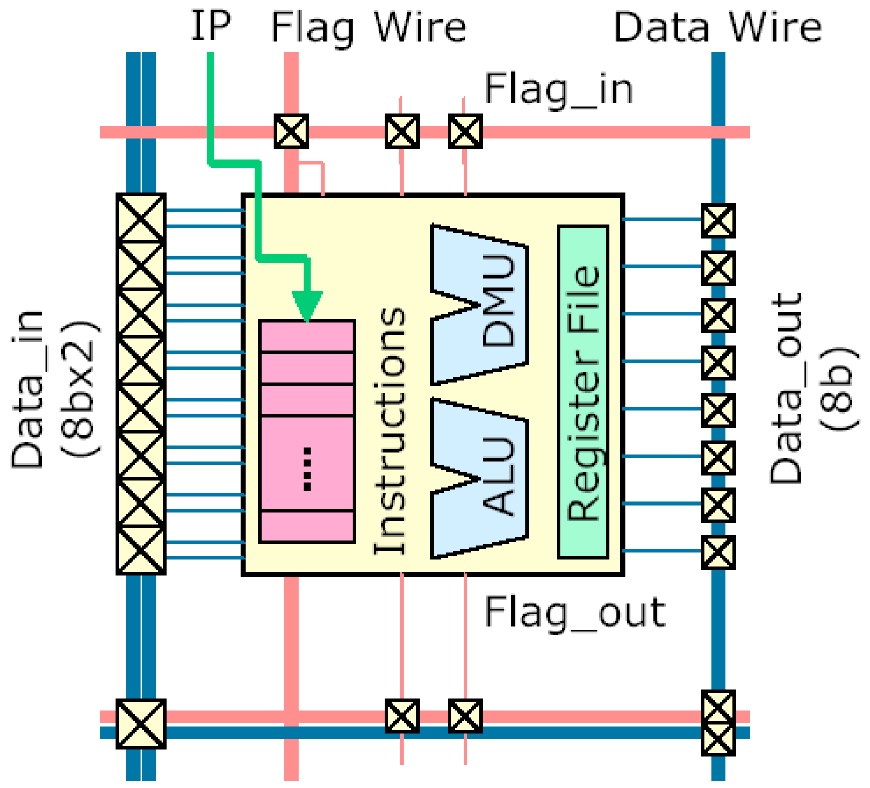


- Array of byte-oriented processing elements
- Fully programmable inter-PE wiring resources
- A simple sequencer
- Array of configurable data memories

A DRP core includes 64 PEs.
STC controls PEs.

By NEC

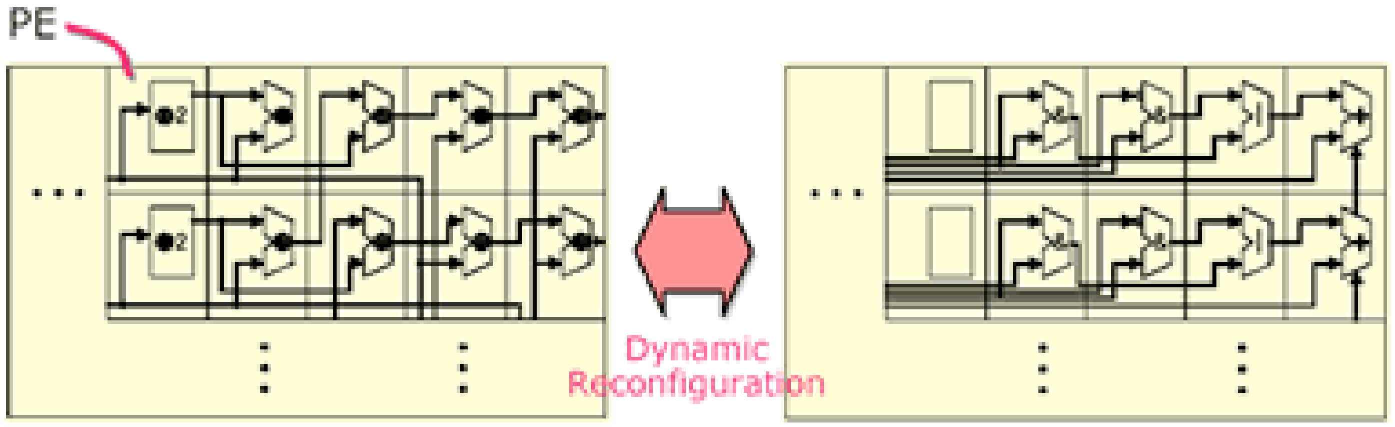
PE of DRP



- 16 instructions can be stored in the instruction memory of each PE.
- An instruction specifies connections and operation of each processor.
- The STC specifies the address of instruction.

By NEC

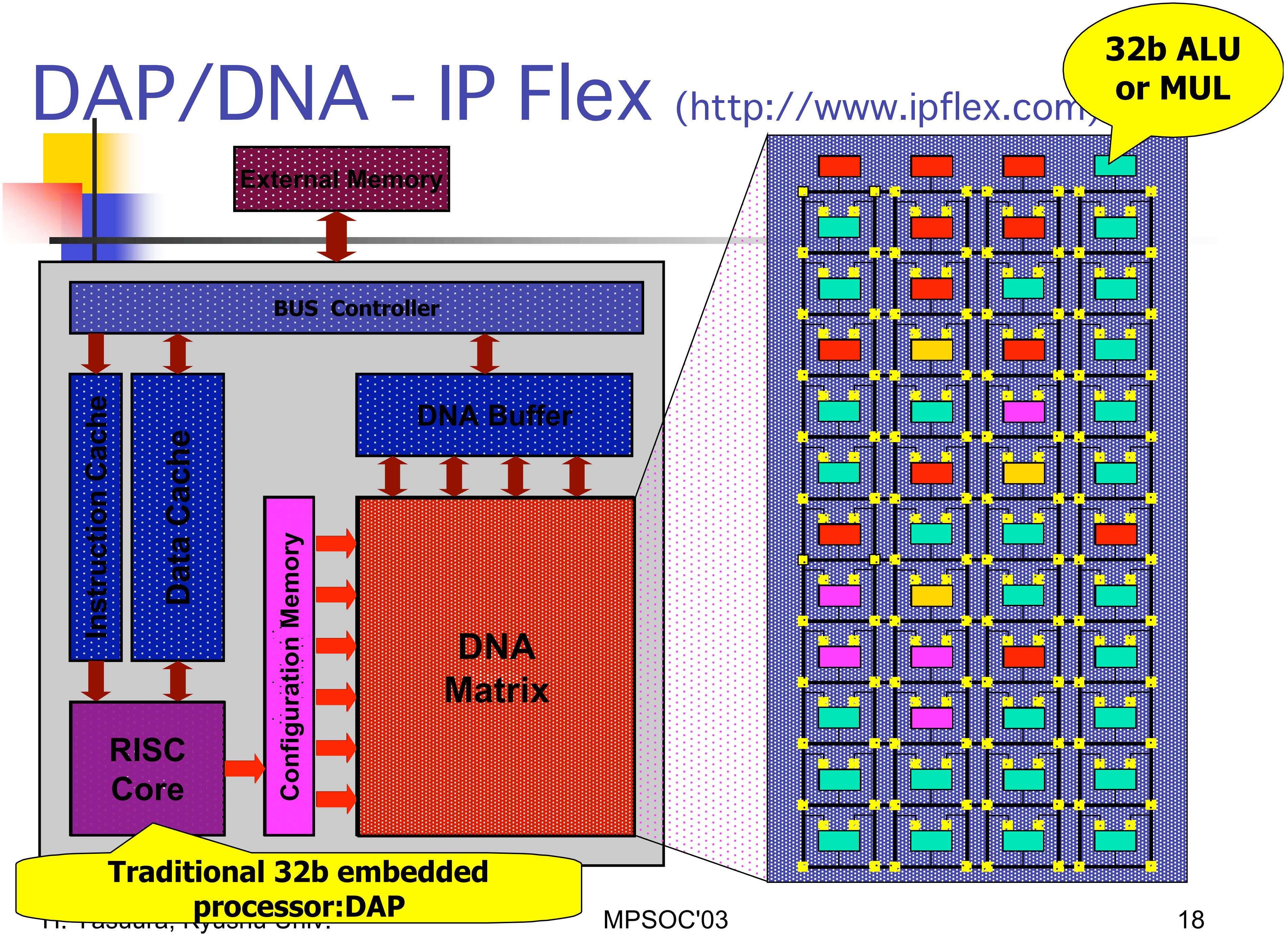
Dynamic Reconfiguration of DRP



The connection among PEs and operations of PEs can be changed in every clock cycle.

By NEC

DAP/DNA - IP Flex (http://www.ipflex.com)



Features of DAP/DNA

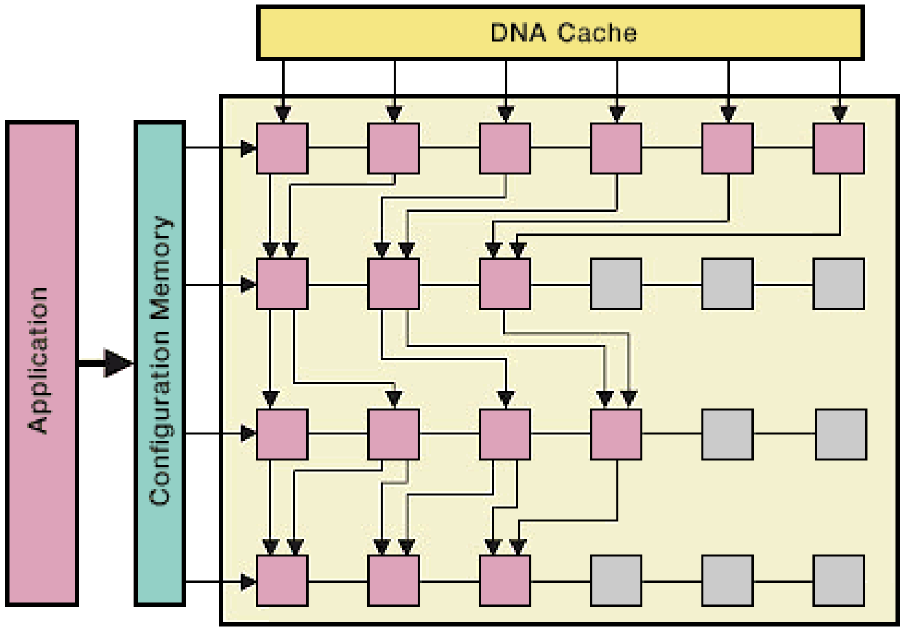
DAP/DNA reconfigurable processor has the advanced features, including:

- The DNA-Matrix architecture with dynamically reconfigurable hardware.
- Reconfiguration of the DNA-Matrix in one clock.
- Parallel data processing, not sequential data processing (Neumann Cycle), and extremely high performance with low power consumption due to the low clock frequency.
- 1-2 digits higher performance compared to existing solutions such as the CPUs and DSPs.
- Dramatic reduction of the development cost and period compared to ASIC and fully custom devices.
- Hardware design with software method (C language) enables flexible the design changes.

600MTr. 225m^2

http://www.ipflex.com/english/product/dapdna_feature.html

DNA(Distributed Network Architecture)



By IP Flex

DNA (Distributed Network Architecture) Matrix Architecture
The DNA-Matrix is a dataflow type accelerator arrayed 148 dynamic reconfigurable operation units. The wiring among elements can be changed dynamically and can quickly constitute parallel/pipeline processing system according to each application operation unit processing. The DNA-Matrix internal constituent information is stored in configuration memory, and its constitution changes in one clock depending on applications.

- 148 of 32bit Operation Units
- Data transfer between elements at single cycle
- Operating Frequency 100MHz

Reconfigurable Computing for System on a Chip

- Background and Requirements
- Platforms for Reconfigurable Computing
  - DRP
  - DAP/DNA
- How to use Reconfigurable Computing in SoC
  - SysteMorph
- Conclusion

SysteMorph K. Murakami (SLRC, Kyushu Univ.)

- Silicon Sea-Belt Project
- Just-in-Time (Dynamic, Online & Adaptive) HW/ISA/SW Co-optimization Technology
- Applications:
  - High Performance Computing
    - Molecular Orbit Computation (Chemistry)
    - Reducing Cost and Energy
  - Mobile Devices
    - Mobile phones
    - Sensor networking
    - Reducing Energy and Increase Service Quality

Design Issues in SysteMorph

- What to profile
- How to profile them
- How to discover hints for optimization
- What to optimize
- How to optimize them
- How to reconfigure HW/ISA/SW

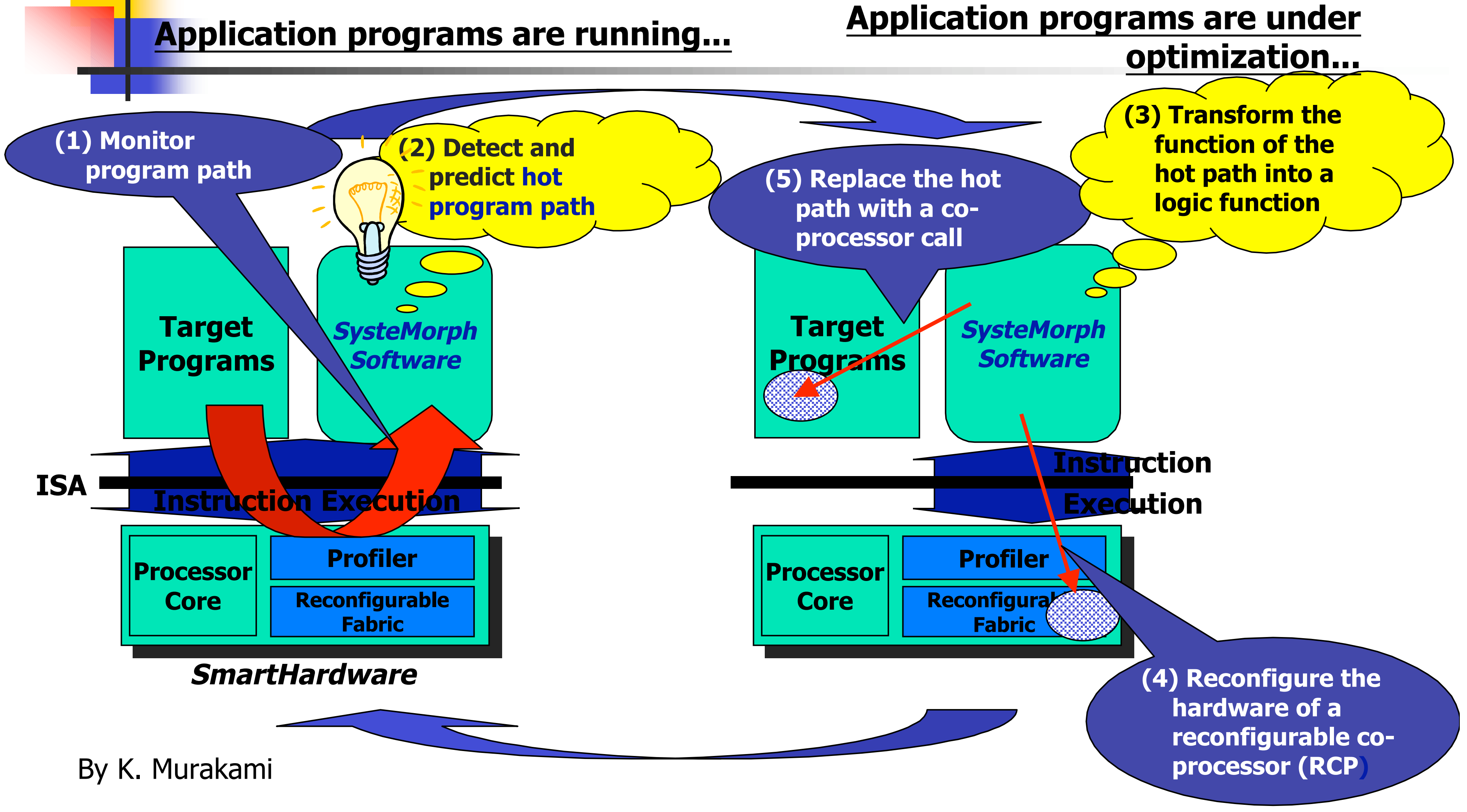
Functionality Morphing: An Example of **SysteMorph**

- Design issues in **SysteMorph**
  - What to profile
  - How to profile them
  - How to discover hints for optimization
  - What to optimize
  - How to optimize them
  - How to reconfigure HW/ISA/SW

- Solutions in functionality morphing
  - Online hot-path profiling
  - Offload the functionality of hot-paths from SW to HW
  - Online HW resynthesis
  - Reconfigurable co-processor
  - Dynamic binary rewriting

By K. Murakami

Functionality Morphing
- Offload Hot Program Path to HW -


Application programs are running...
Application programs are under optimization...
(1) Monitor program path
(2) Detect and predict hot program path
(5) Replace the hot path with a co-processor call
(3) Transform the function of the hot path into a logic function
Target Programs
SysteMorph Software
ISA
Instruction Execution
Processor Core
Profiler
Reconfigurable Fabric
SmartHardware
Target Programs
SysteMorph Software
Instruction Execution
Processor Core
Profiler
Reconfigurable Fabric
(4) Reconfigure the hardware of a reconfigurable co-processor (RCP)

By K. Murakami

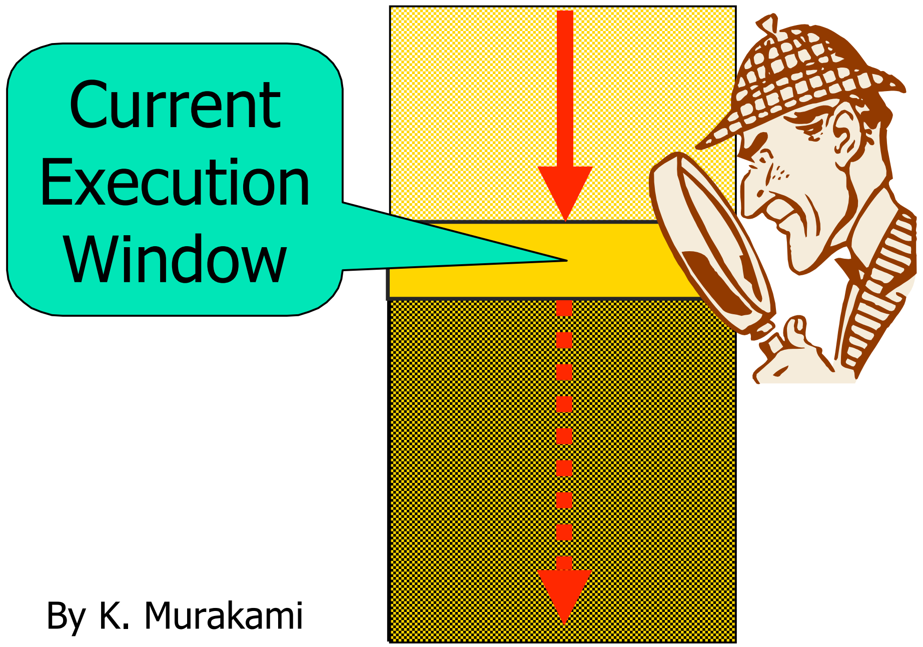
Offline vs. Online Profiling

Offline Profiling

- Summary of program behavior based on whole program trace
- Good for:
  - CO (Compiler Optimization)
  - SRC (Static Reconfigurable Computing)

Online Profiling

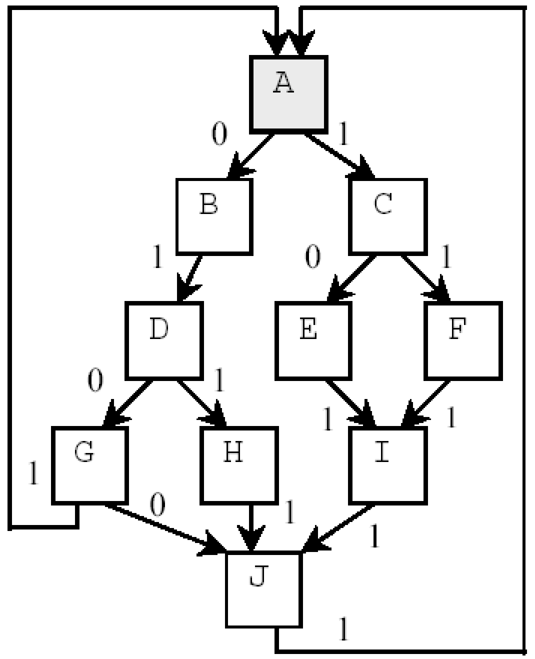
- Prediction based on current execution window of program
- Good for:
  - DCO (Dynamic CO)
  - **SysteMorph**



By K. Murakami

Online Hot-Path Profiling - Algorithm -

- Program Path



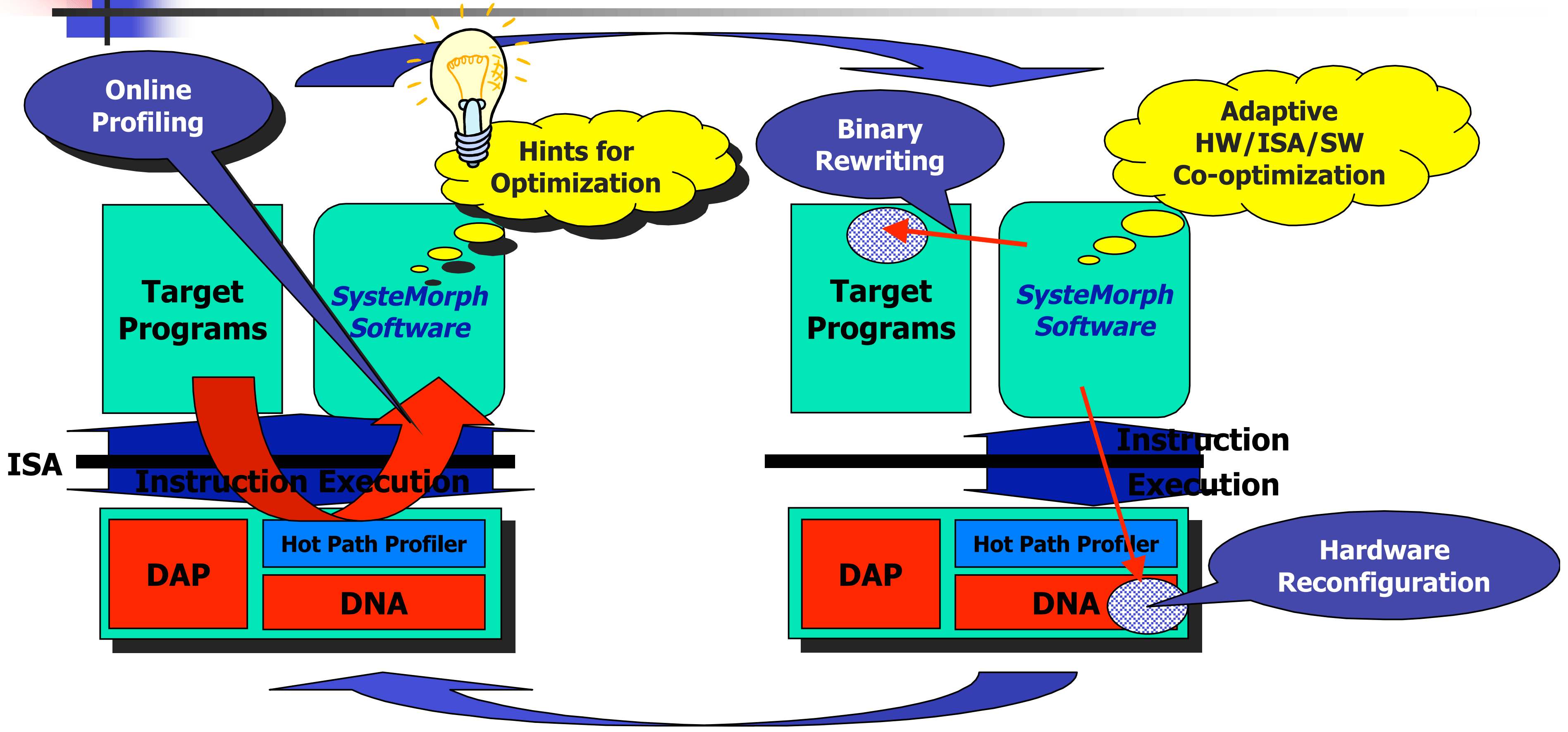
Path:	signature
ABDG:	A.0101
ABDGJ:	A.01001
ABDHJ:	A.01111
ACEIJ:	A.10111
ACFIJ:	A.11111

- Existing Algorithms
  - Efficient path profiling
  - NET prediction

- Our Algorithm
  - Profile the history of branch instruction’s behaviors (taken or not-taken, branch target)
  - If the execution frequency at a path head exceeds the threshold, select the path head (“A” in the figure) as a candidate of the hot path head
  - Traverse the object code, starting with the candidate (“A”), based on the branch history, and predict the hot path

By K. Murakami

Functionality Morphing Prototyping:
IP Flex DAP/DNA Powered by **SysteMorph**



By K. Murakami

Offline vs. Online HW Synthesis

Offline HW Synthesis

HW configuration is synthesized prior to the program execution

Good for:

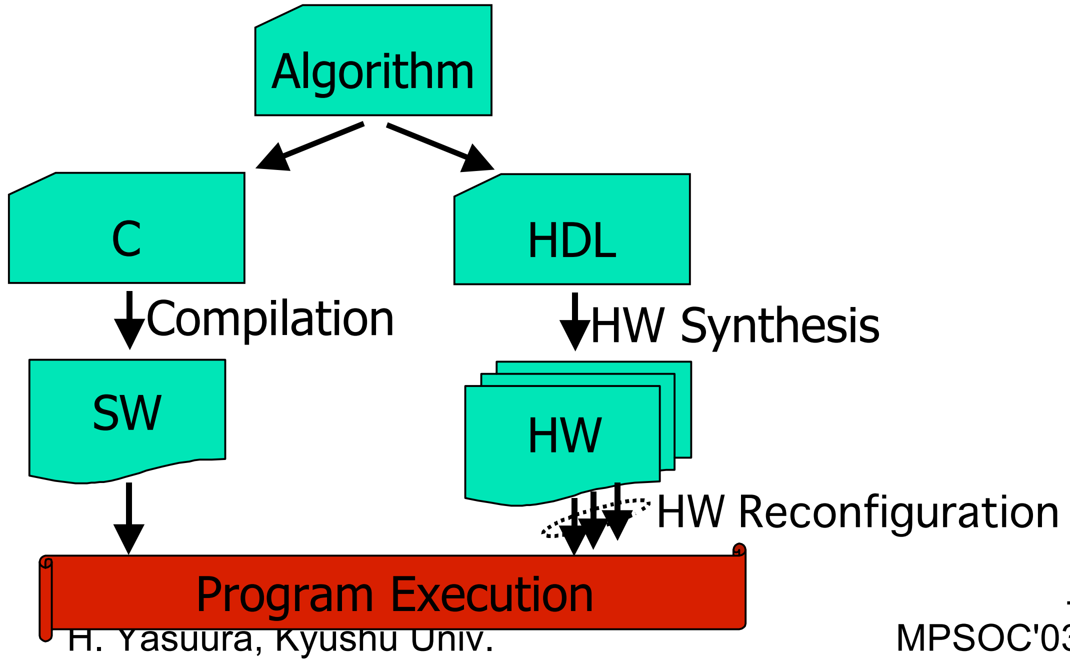
SRC (Static Reconfigurable Computing)

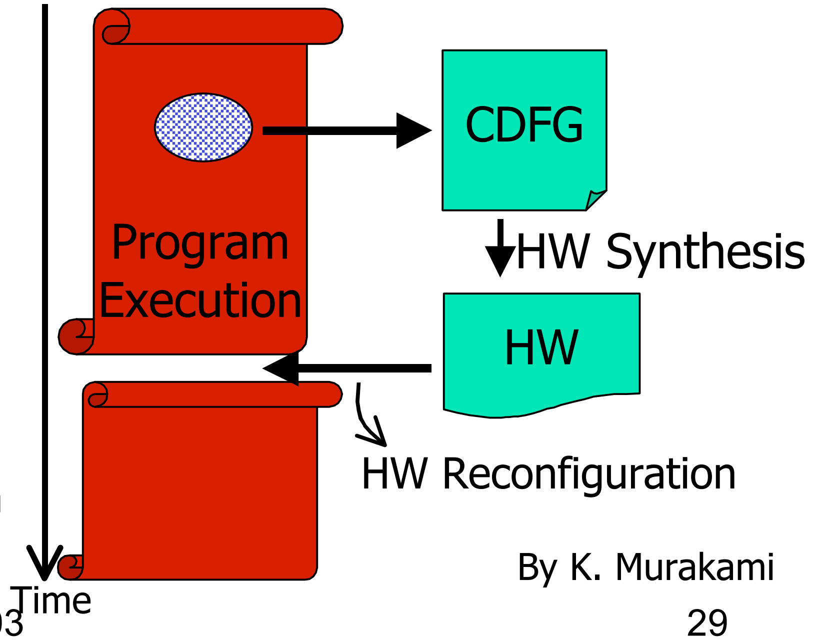
Online HW Synthesis

HW configuration is synthesized in parallel with the program execution

Good for:

SysteMorph





By K. Murakami

Reconfigurable Computing for System on a Chip

- Background and Requirements
- Platforms for Reconfigurable Computing
  - DRP
  - DAP/DNA
- How to use Reconfigurable Computing in SoC
  - SysteMorph
- Conclusion

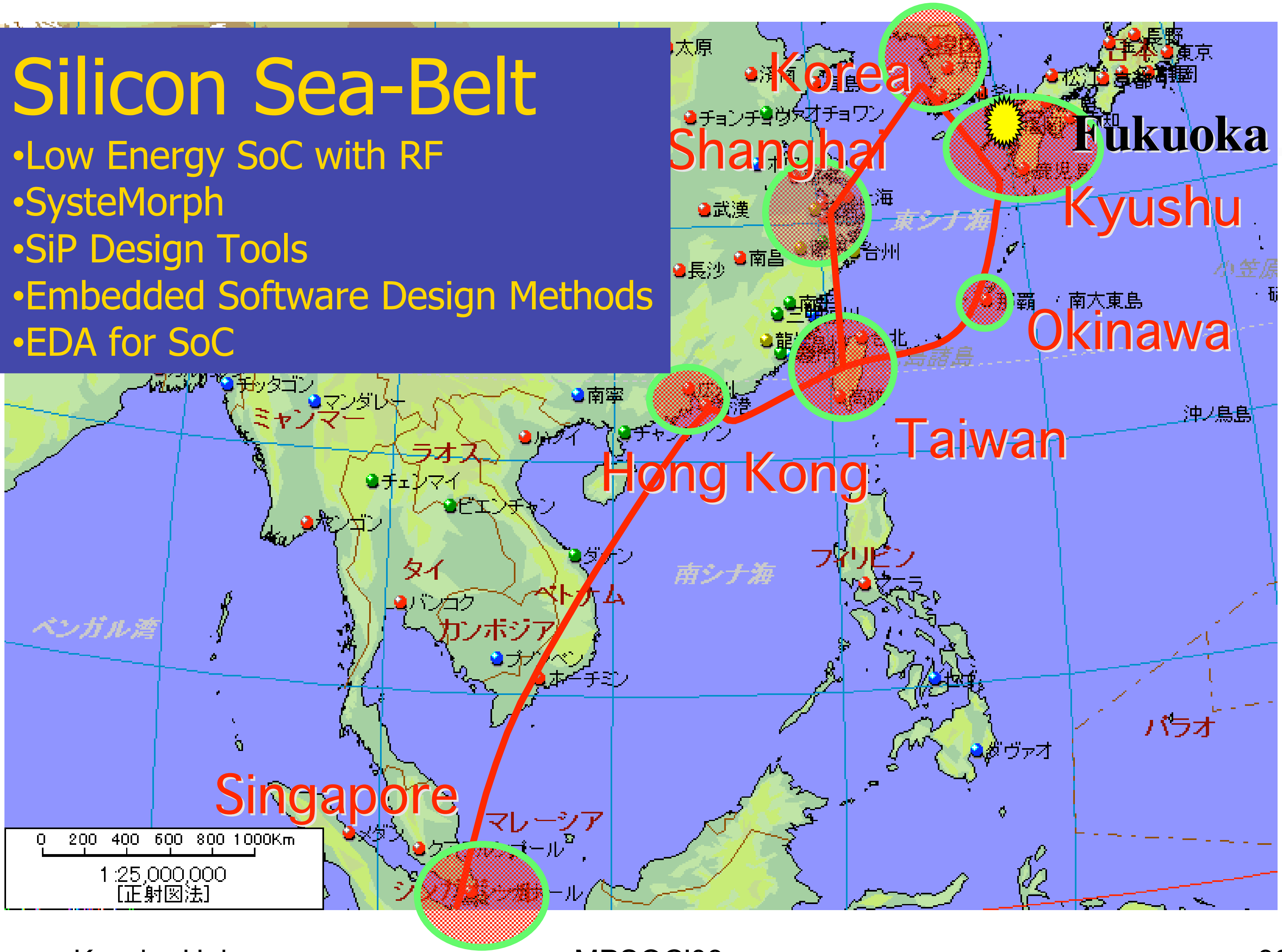
Reconfigurable Computing

- Several platforms are ready to use. System architectures and application techniques for SoC are key issues.
  - Cost Reduction
    - Reduce design, mask, and test costs.
    - Reuse an SoC device into different consumer products.
  - Customer Satisfaction
    - Optimize devices for multiple applications and varieties of usage.
  - Speed-up of Business
    - Introduce new services without change of devices.
  - Reliability and Security
    - Repairs and debugging on customer site.
    - Frequent update of security procedures.
  - Environment and Ecology
    - Reuse devices in revised services.
    - Minimization of energy consumption for each user.

Thank you for your attention.

Silicon Sea-Belt

- Low Energy SoC with RF
- SysteMorph
- SiP Design Tools
- Embedded Software Design Methods
- EDA for SoC



Korea
Fukuoka
Shanghai
Kyushu
Okinawa
Taiwan
Hong Kong
Singapore
0 200 400 600 800 1000Km
1:25,000,000
[正射図法]