Development of Dependable Network-on-Chip Platform

Tomohiro Yoneda

National Institute of Informatics

Masashi Imai
Univ. of Tokyo

Atsushi Matsumoto
Tohoku Univ.

Hiroshi Saito
Univ. of Aizu

NII National Institute of Informatics 国立情報学研究所

Project summary

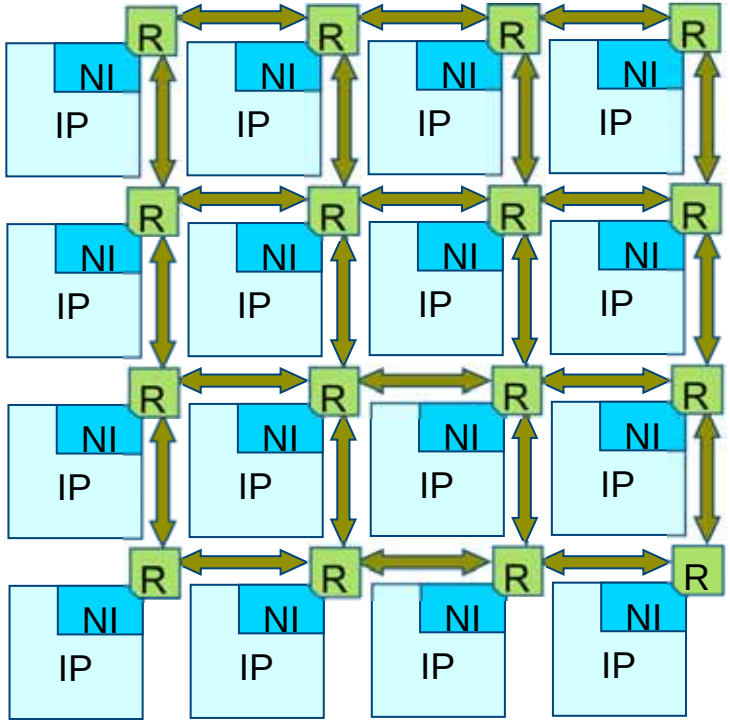
- CREST(Core Research for Evolutional Science and Technology) project
  - sponsored by JST (Japan Science and Technology Agency)
  - 31 research areas and 371 research subjects
  - Research period : 5.5 years or less
  - Research budget : 40 to 120 M. yen per year
- Research area : “Fundamental Technologies for Dependable VLSI system”
  - 7 research subjects

Background and Goal

- Demands for integrating more and more cores into a chip
  - Eg. Automotive electronic systems
    - More than 50 ECUs are used in an automobile
    - Many problems in connecting them
    - New approach
      - Centralized architecture where many ECUs are contained in one chip
- Goal
  - Platform for performing many and various tasks in one chip dependably, efficiently and adaptively

Approach

- Network-on-Chip (NoC)



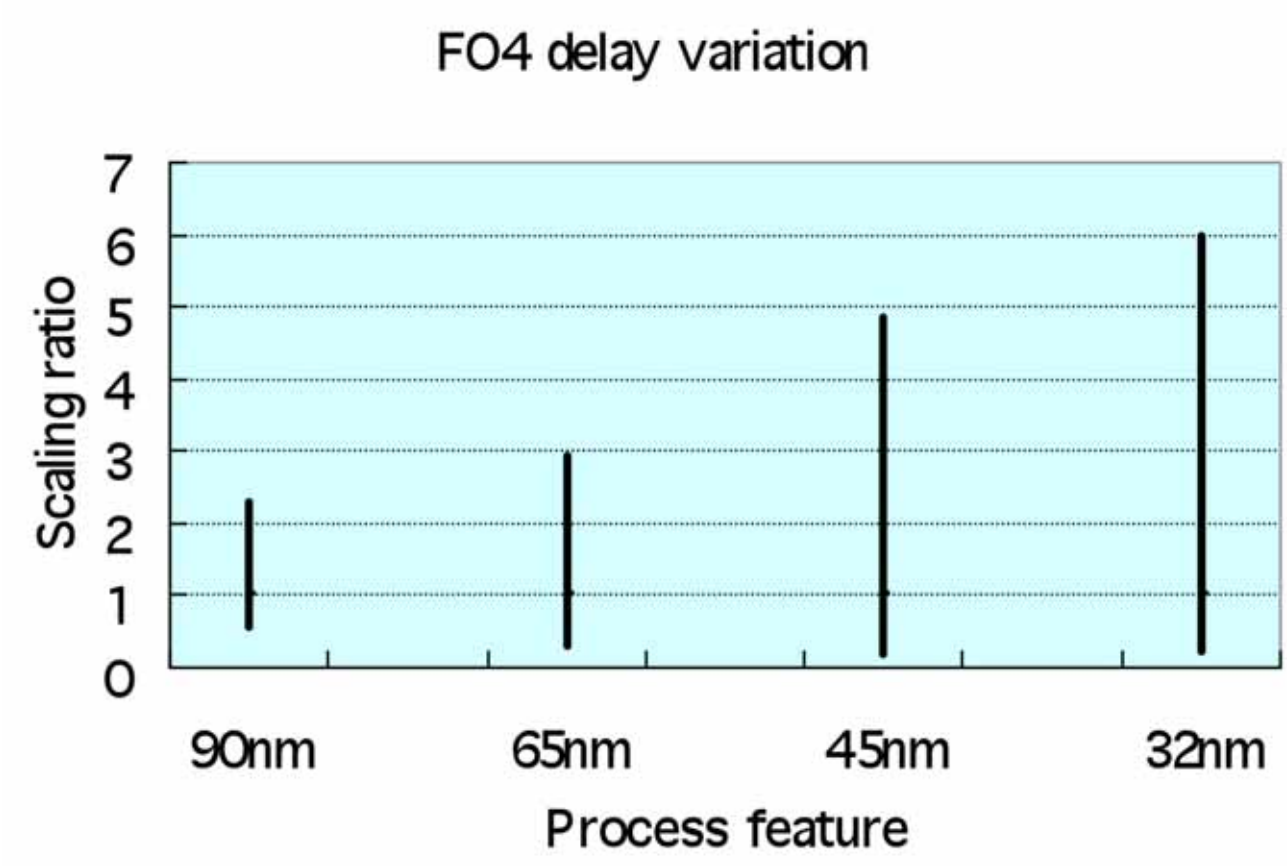
IP : CPU/accelerator core
NI: Network Interface
R: Router

- Globally Asynchronous Locally Synchronous (GALS) systems

Problems to be solved

- GALS-NoC approach still has problems such as
  - Too large variation of performance among local units in a chip, caused by characteristics of advanced semiconductor technologies
  - Too many cores that work only for specific applications, causing inefficient usage of a chip (i.e. many inactive cores)
  - Area and performance overhead caused by GALS-NoC approach

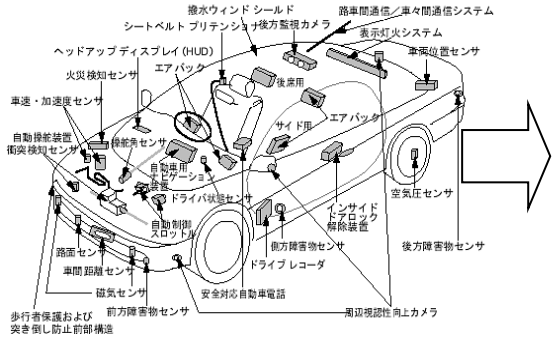
Variation problem

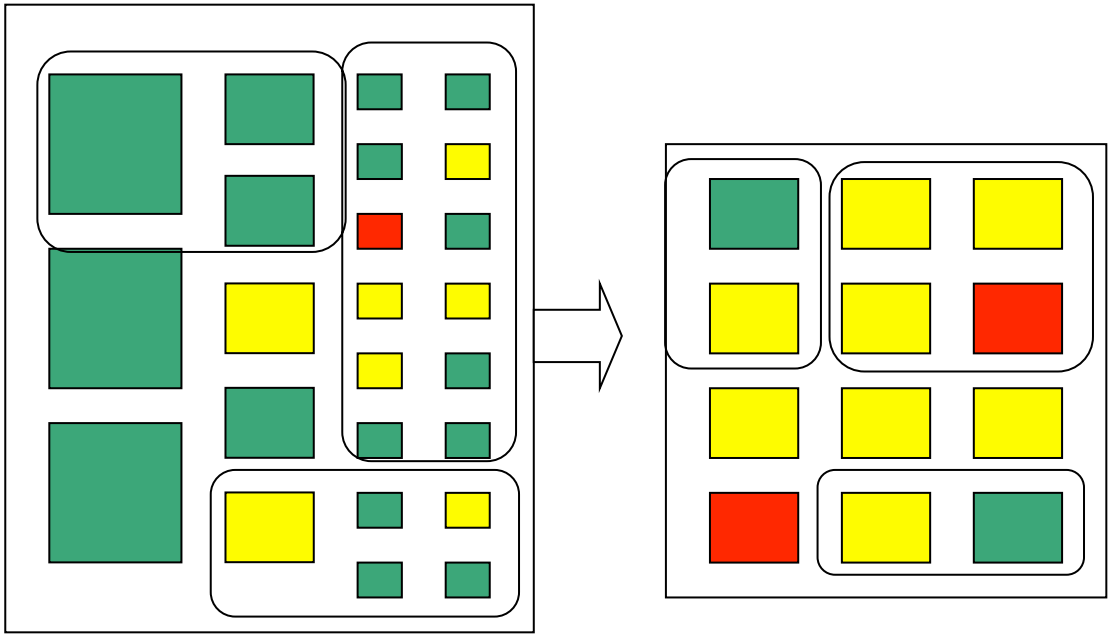


Problems to be solved

- GALS-NoC approach still has problems such as
  - Too large variation of performance among local units in a chip, caused by characteristics of advanced semiconductor technologies
  - Too many cores that work only for specific applications, causing inefficient usage of a chip (i.e. many inactive cores)
  - Area and performance overhead caused by GALS-NoC approach

Inefficient usage of a chip





Each ECU is just put into a chip

Uniform and adaptive cores should be used

high

medium

low

workload

Problems to be solved

- GALS-NoC approach still has problems such as
  - Too large variation of performance among local units in a chip, caused by characteristics of advanced semiconductor technologies
  - Too many cores that work only for specific applications, causing inefficient usage of a chip (i.e. many inactive cores)
  - Area and performance overhead caused by GALS-NoC approach

Research Plan

1. Development of fundamental and essential core technologies for
   - detecting degraded units and assigning less tasks to them in either
     - hardware level (fine-grained)
     - OS level (coarse-grained)
   - implementing adaptive units that can change performance smoothly and efficiently
   - reducing various overhead caused by communication via network

Research Plan (2)

2. Demonstration in automotive control system area through
   - Prototype development
     - Four V850Estar or its asynchronous version
     - A hardware accelerator core
     - Memory and interface cores
     - Modified real-time OS
   - Hardware In the Loop simulation
     - Actual engine, powertrain and environment are modeled by Simulink, and are simulated in real-time

Some preliminary results

- Concurrent processing mechanism with degradation-tolerance
  - For applications that require concurrent processing units of the same operation
  - Use asynchronous processing units
    - avoid sudden illegal operation due to a degraded unit
    - nevertheless, completion of the whole computation may be delayed
  - Use special hardware dispatcher
    - allow each processing unit to get any amount of data
    - select the destination such that degraded processing units are used less often

    → degradation-tolerance

Some preliminary results (2)

- Example of such an application
  - Gauss-Seidel method for solving a system of linear equations

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=0}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}\right)$$

$$\begin{bmatrix} & & & & \\ a_{i1} & \cdots & a_{ii} & \cdots & a_{in} \\ & & & & \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Some preliminary results

- Concurrent processing mechanism with degradation-tolerance
  - For applications that require concurrent processing units of the same operation
  - Use asynchronous processing units
    - avoid sudden illegal operation due to a degraded unit
    - nevertheless, completion of the whole computation may be delayed
  - Use special hardware dispatcher
    - allow each processing unit to get any amount of data
    - select the destination such that degraded processing units are used less often

    → degradation-tolerance

Conclusion

- Introduction of our CREST project
  - Started last October and will end on March of 2014
  - Organized by four groups in
    - NII
    - Univ. of Tokyo
    - Tohoku Univ.
    - Univ. of Aizu
- Research area web site

  http://www.dvlsi.jst.go.jp/english/index.html