



# An Efficient Performance Estimation Method of MPSoC with Configurable Multi-layer Bus System

MASAHARU IMAI, SALITA SOMBATSIRI AND YOSHINORI TAKEUCHI  
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY  
OSAKA UNIVERSITY



## Agenda



### Introduction

- Definition of Models:  
System-Level Model and Architectural Model
- Architecture Exploration Method for MPSoC
- Performance Estimation Method
- Configurable Multi-layer Bus-based SoC
- Case Study
- Conclusion and Future Work

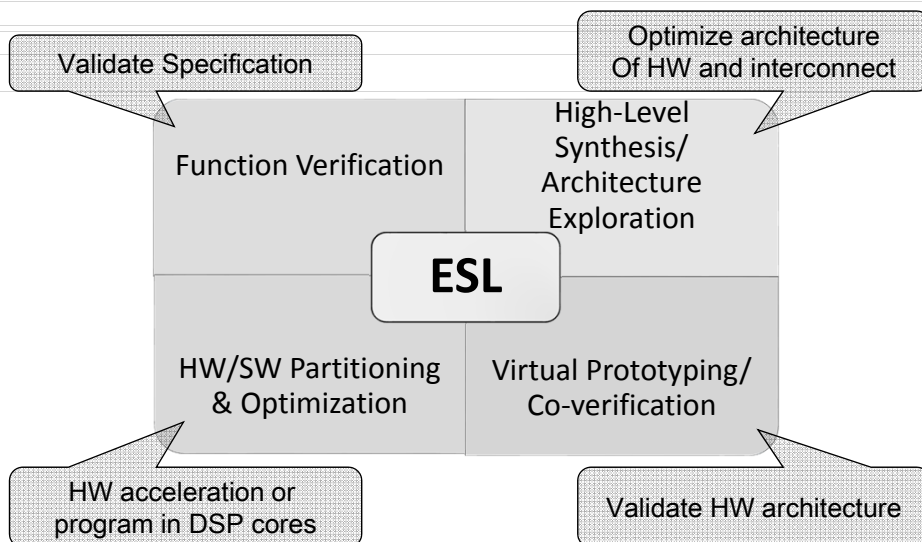
# High-performance System-on-a-Chip (SoC)

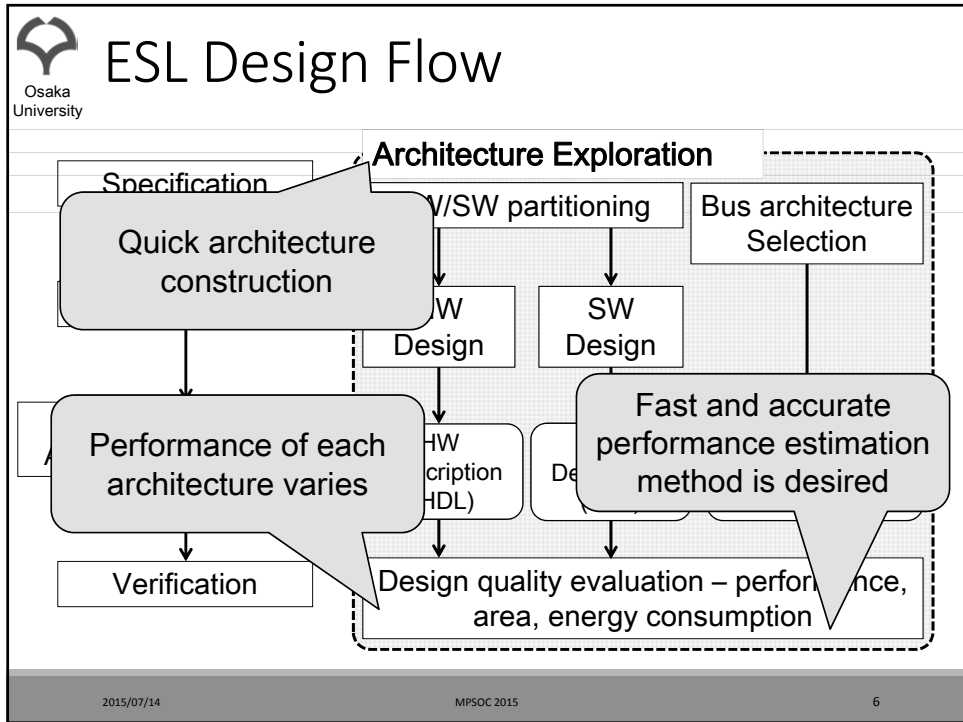
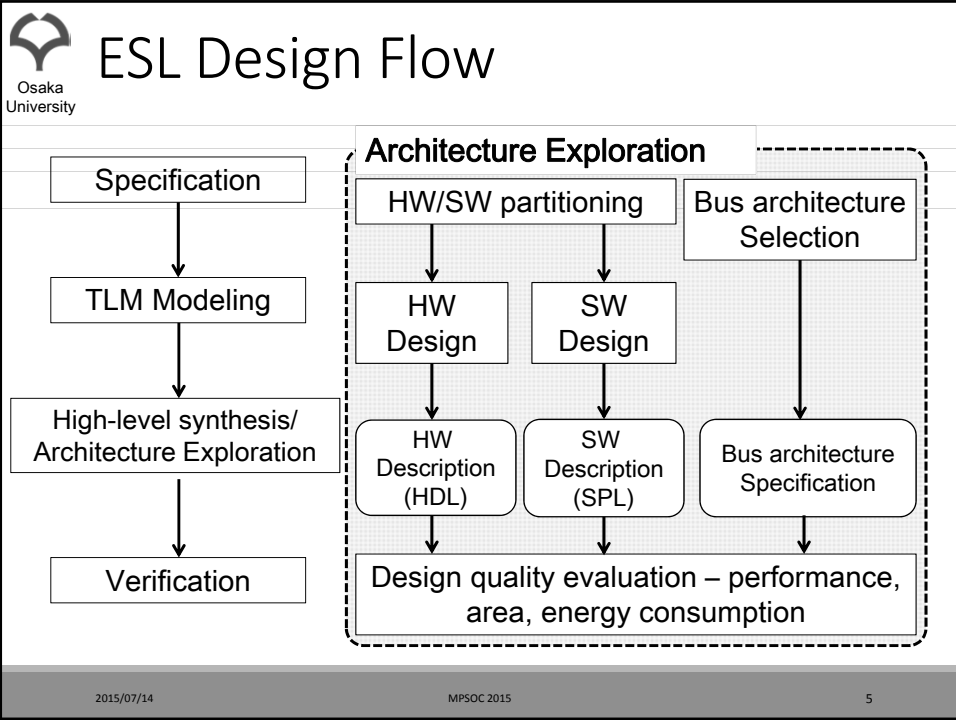
- The advancement in process technology
  - High-performance and multi-function SoC
    - Multiple Processing Elements (PE)
    - Multicore or many core implementation
    - High speed bus – shared bus, multi-layer bus
  - Strict design constraints
    - High performance, small area, low energy consumption
- Standard bus specifications
  - AMBA, CoreConnect etc.



Electronic System-Level to quickly explore design space to find architectures that satisfy all constraints

# Electronics System-Level(ESL) Design






## Contribution of this research

---

- Architecture exploration method for multi-processor SoC
- Efficient performance estimation method for multi-layer bus-based SoC
  - Model standard bus protocols' features
    - Dynamic behavior such as pipeline transfer, burst transfer, split response operation, error response operation, bus preemption
  - Efficient performance estimation method by analyzing Architecture-Level Execution Dependency Graph (AL-EDG)
    - Recognize bus contention
    - Predict behavior of shared buses and multi-layer bus during performance estimation
    - Estimate performance by analyzing AL-EDG according to speculated bus behavior

## Agenda

---

- Introduction
-  Definition of Models:  
System-Level Model and Architectural Model
- Architecture Exploration Method for MPSoC
- Performance Estimation Method
- Configurable Multi-layer Bus-based SoC
- Case Study
- Conclusion and Future Work

# Model of Computation (MoC)

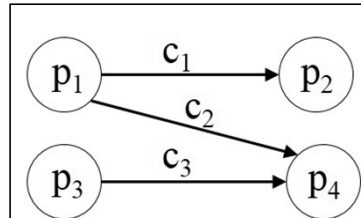
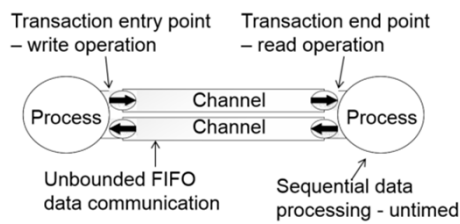
## ● System-Level Model (SLM)

SLM  $M_{sl} = (P, C)$

Process set  $P = \{p_i | i = 1, 2, 3, \dots\}$

Channel set  $C = \{c_i | i = 1, 2, 3, \dots\}$

- Based on Kahn-Process Network
- Represent data flow of the system
- Abstraction level : Loosely-timed model of TLM 2.0



Example :

$M_{sl} = (P, C)$

$P = \{p_1, p_2, p_3, p_4\}$

$C = \{c_1, c_2, c_3\}$

# Architectural Model

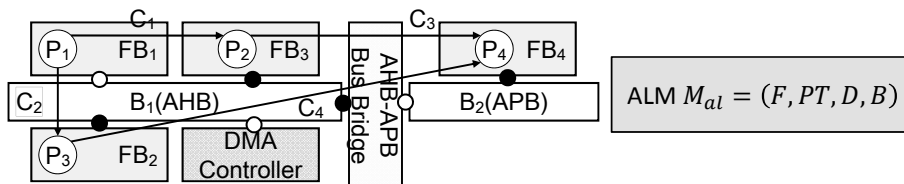
## ● Architecture-Level Model (ALM)

- Represent architecture of SoC

- Components of ALM,

- Functional block set  $F$  – instance of Intellectual Property (IP) Module
- Communication port set  $PT$  – master and slave ports of functional blocks
- Direct Memory Access Controller (DMAC) set  $D$
- Shared bus set  $B$


- Information about mapping SLM to ALM and communication path
- Information of components, e.g. execution frequency, bus width etc.



ALM  $M_{al} = (F, PT, D, B)$

# Agenda

---

- Introduction
- Definition of Models:  
System-Level Model and Architectural Model
-  Architecture Exploration Method for MPSoC
  - Performance Estimation Method
  - Configurable Multi-layer Bus-based SoC
  - Case Study
  - Conclusion and Future Work

# Related Work

---

- Communication architecture exploration
  - Bus architecture optimization based on bus template[1]
    - Mapping data transfer to bus template
  - Bus architecture optimization[2]
    - Explore a bus architecture for a fixed set of functional blocks
- Bus matrix optimization
  - Bus matrix optimization by slave clustering[3]
    - Clustering slaves to find the minimum number of buses on bus matrix under Throughput Constraint Path
  - Bus matrix optimization by traffic overlap analysis[4]
    - Clustering slaves and masters that do not violate traffic overlap threshold
    - Cannot find multiple masters and AHB subsystem architecture
    - Aim : To find the minimum area under performance constraint

[1] Parischa et. al., Proc. 42<sup>nd</sup> DAC, 2005.

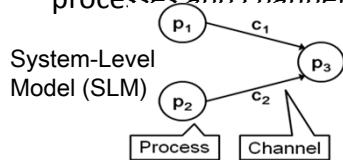
[2] Lahiri et. al., IEEE Trans. Comput.-Aided Des. Intgr Circuits System, Vol. 23, No. 6, 2004.

[3] Parischa et. al., IEEE Trans. Comput.-Aided Des. Intgr Circuits System, Vol. 26, No. 8, 2007.

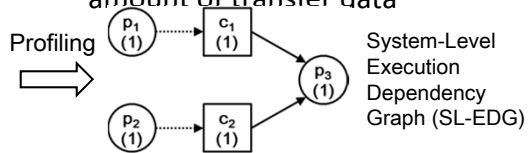
[4] Murali et. al., IEEE Trans. Comput.-Aided Des. Intgr Circuits System, Vol. 26, No. 7, 2007.

# Architecture Exploration Method

- Model application with processes and channels

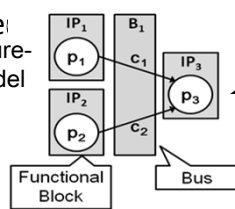


- Extract execution order and amount of transfer data



## Architecture Exploration

- Architecture-Level Model (ALM)



- Design quality estimation
- Performance estimation
- Area estimation
- Energy consumption estimation

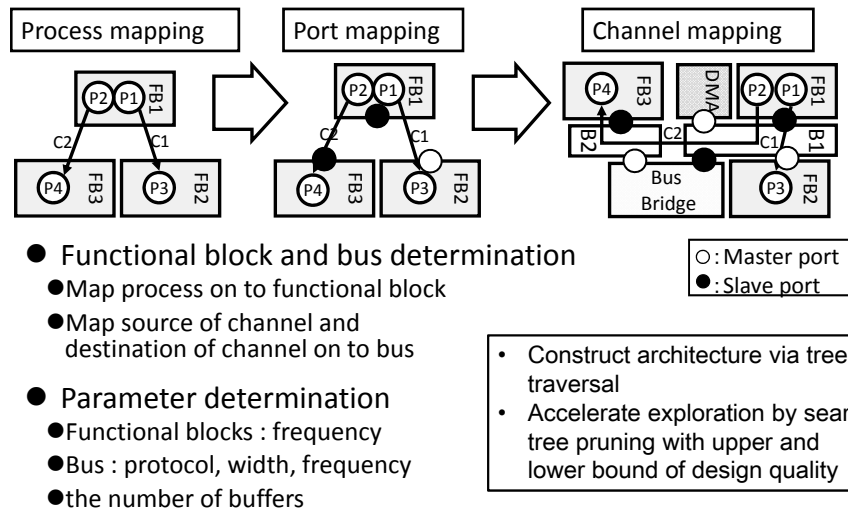


\* Intellectual Property (IP) : Hardware module

# Definition of the Performance Estimation Method

- Input
  - System-Level Model (SLM)
  - IP database
    - Gate count (gates), Mappable process and clock cycle, Number of ports
  - Bus protocol database
    - Protocol name, frequency candidate, bit width candidate
  - Design constraints
    - Architecture constraint, design quality constraint
  - Profiling Information obtained from system level profiling using loosely-timed simulation
    - Execution order, the amount of transfer data
- Output
  - All Pareto solutions of the target architecture and corresponding design quality, represented by Architecture-Level Model (ALM)
  - Design quality functions - Performance estimation function, hardware area function, energy consumption function

# AMBA Hierarchical Shared Bus Architecture Exploration



# Design Quality Estimation Functions

- Performance estimation function
  - Execution time estimation by analyzing static model
  - Detail in efficient performance estimation method section
- Hardware area function
  - Summation of hardware area
- Energy consumption function
  - Estimation of energy consumed by every hardware in the system when executing application specified by SLM



## Experiment Setup (1/2)

- Objective
  - To demonstrate that the proposed method allows AMBA shared bus architecture to be explored
- Machine : 2.80 GHz intel core i7, 8GB memory, CentOS6.2
- Input
  - SLM and profiling information



Data size(bits)	24 bits	8 bits	12 bits	12 bits	12 bits	8 bits
Maximum data	64 data	64 data	64 data	64 data	64 data	256 data

- Bus database

Protocol name	Bus width candidate	Frequency candidate	# of master interfaces	# of slave interfaces
AHB	32, 64 bits	50, 100 MHz	16	Not specified
APB	16, 32 bits	30 MHz	1	Not specified

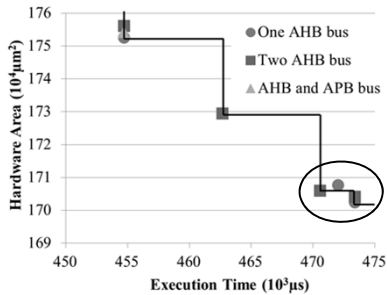
## Experiment Setup (2/2)

- IP Database

IP name	Area [gate]	Frequency [MHz]	# of master port	# of slave port	Functional block (Mapped process[cycle])
IP <sub>0</sub>	3,295	50,100	0	1	BS(BS[70])
IP <sub>1</sub>	19,249	100	0	1	CT(CT[1,345])
IP <sub>2</sub>	18,739	100	1	0	DCT(DCT[3,617])
IP <sub>3</sub>	7,713	100	0	1	ZZ(ZZ[64])
IP <sub>4</sub>	10,754	50	1	0	Q(Q[1,280])
IP <sub>5</sub>	47,148	100	0	1	VLC(VLC[251])
IP <sub>6</sub>	24,036	100	1	0	WRT(writer[769])

- Design constraints
  - Maximum number of bus in an architecture : 2
  - Maximum bus bridge in an architecture : 1
  - Number of buffers : 1,2
- Area estimation parameters
  - 0.56  $\mu\text{m}$  wire pitch, 0.18  $\mu\text{m}$  CMOS library, 0.95 over-the-cell ratio

# Architecture Exploration Results

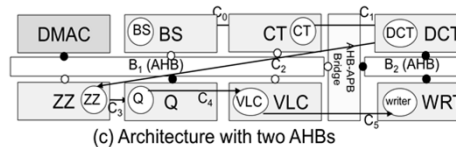
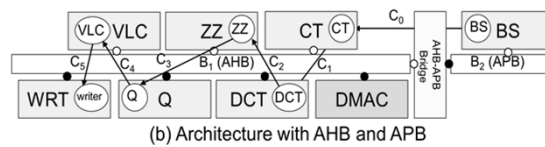
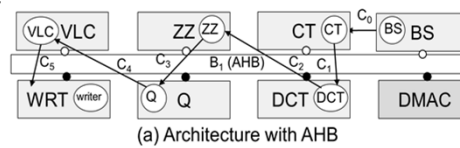


- Point : Relationship between area and execution time of each solution
- Line : Trade-off boundary of each functional block set and bus architecture
- 7 Pareto solutions found
  - Design space size : ~4 billion architectures
  - Exploration time : 19 hours


Proposed method actualizes the bus architectures to AMBA AHB and APB to be optimized in the design space exploration

# Explored Architecture

- Various bus architecture
  - (a) One AHB bus
  - (b) One AHB bus and one APB bus
  - (c) Two AHB buses
- (c) is the architecture closest to the origin point of the trade-off graph

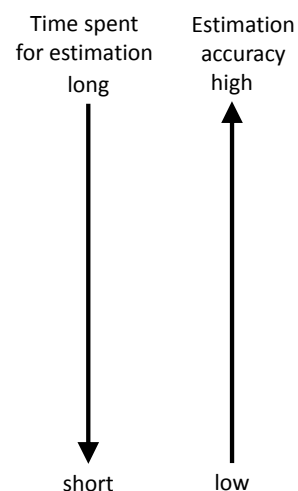


# Agenda

- Introduction
- Definition of Models:  
System-Level Model and Architectural Model
- Architecture Exploration Method for MPSoC
-  Performance Estimation Method
  - Configurable Multi-layer Bus-based SoC
  - Case Study
  - Conclusion and Future Work

# Performance Estimation Method

- Conventional RTL simulation
  - Involve a large number of signals
- Hardware-software co-simulation
  - Simulate a complete behavior of a system architecture
- Simulation of models using high-level languages
  - SpecC, SystemC etc.
  - Models in various abstraction levels
- Analysis of static Model of Computation
  - Synchronous Data Flow, Stochastic timed marked graph



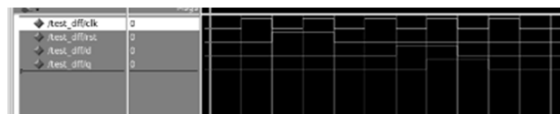
## Conventional RTL Simulation

- Register-Transfer-Level (RTL) abstraction of hardware
  - Implemented in using Hardware Description Language
  - Simulation using HDL simulator
  - All signals are simulated

```

1 use IEEE STD_LOGIC_1164.ALL;
2 entity DFF is
3   port(
4     Q : out std_logic;
5     CLK : in std_logic;
6     RST : in std_logic;
7     D : in std_logic
8   );
9 end DFF;
10
11
12 architecture rtl of DFF is
13 begin
14   process(CLK)
15   begin
16     if ( rising_edge(CLK) ) then
17       if ( RST = '1' ) then
18         Q <= '0';
19       else
20         Q <= D;
21       end if;
22     end if;
23   end process;
24 end rtl;
25

```



## Related Work: Hardware-Software Co-simulation

- Models for hardware-software co-simulation
  - Metropolis Meta-model [5], Ptolemy [6]
  - ➔ Accelerate simulation in orders of magnitude, but must be rebuilt for each of the architectures
- Flexible bus model-based approach for communication refinement [7,8]
  - Repeat only performance analysis for various bus architecture
  - ➔ Remodeling and simulation still needed to collect communication trace for architectures that contains different processing element set

[5] Balarin et. al., Proc. CODES'02, 2002.

[6] Buck et. al., Int. Journal of Computer Simulation, Vol. 4, 1994.

[7] M. Takahashi et. al., Proc. 11th SASIMI, apr 2003, pp. 345–350.

[8] K. Lahiri et. al., IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 20, no. 6, pp. 768–783, jun 2001.



## Related Work: Simulation of Models using High-level Languages

- High-level language : SpecC, SystemC
- Several abstraction level can be implemented
  - Cycle Accurate (CA) model [9]
    - 10-100 times faster than RTL simulation
  - Bus Cycle Accurate (BCA)
    - 19-90 times faster than RTL simulation [10]
  - Bus Cycle Accurate at Transaction Boundaries [11]
  - Timed-model
    - Approx. 20 times faster than BCA simulation [10]
- ➔ Simulation speed still slow and individual high-level abstraction model required for each architecture

[9] Loghi et. al., Proc. DATE'04, 2004.

[10] Baganne et. al., Int. Journal of Computer Simulation, Vol. 4, 1994.

[11] Pasricha et. al., ACM TECS, Vol. 7, Issue 2, feb 2008.

2015/07/14

MPSOC 2015

25



## Related Work: Analysis of static Model of Computation

- Worst case performance estimation
  - Formal model used for approximating performance of AMBA shared bus and detecting deadlock [12]
  - Synchronous Data Flow (SDF) for estimating hierarchical shared bus [13]
- Statistical performance estimation
  - Stochastic timed marked graph [14]
  - Timed marked graph [15]
- System bus latency estimation for shared bus and multi-layer bus[16]
- ➔ Fail to capture dynamic bus contention during system execution

[12] Madl et. al., Proc. 6th ACM & IEEE EMSOFT'06, oct 2006, pp. 311-320.

[13] Lee et. al., J. Signal Process. Syst., Vol. 58, No.2, pp.193-213, 2010.

[14] Li et. al., J. Signal Process. Syst., Vol. 58, No.2, pp.105-116, 2010.

[15] Liu et. al., Proc. DATE'12, pp. 641-646, 2012.

[16] Cho et.al., Proc. SLIP'06, pp.67-74, 2006.

2015/07/14

MPSOC 2015

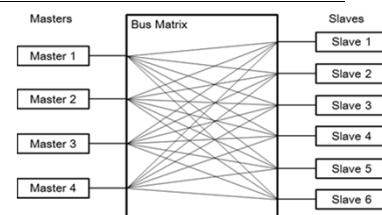
26

# Agenda

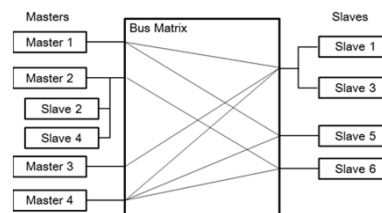
- Introduction
- Definition of Models:  
System-Level Model and Architectural Model
- Architecture Exploration Method for MPSoC
- Performance Estimation Method
- ☞ **Configurable Multi-layer Bus-based SoC**
  - Case Study
  - Conclusion and Future Work

# Configurable Multi-layer Bus

- Bus matrix as main interconnect fabric
- Bus matrix architecture
  - Full bus matrix architecture
    - Routing difficulty
  - Maximally connected reduced bus matrix architecture
    - Unnecessary buses are eliminated
  - Partially bus matrix architecture
    - Optimize bus matrix with heterogeneous configurations
    - Performance-area trade-off
    - Ease of routing problems



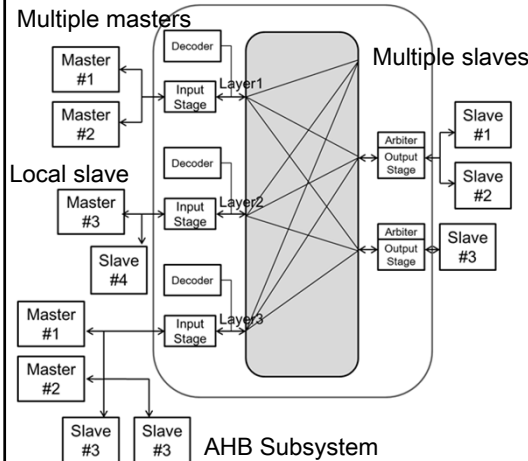
Full Bus Matrix Architecture



Partial Bus Matrix Architecture

# Configurable Multi-layer Bus and Behavior

## BUS MATRIX CONFIGURATIONS



## BEHAVIOR

- Pipeline Transfer
  - Address phase and the last data phase of the previous transaction overlap
- Burst Transaction
  - Consecutive data transfer to save arbitration clock cycle
- Split/Retry Operation
  - Slave responses with split/retry response when it is not ready to serve the request
- Lock Transfer
  - Master initiate a lock transaction so that it is not preempted

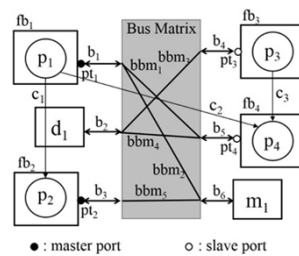
# Architectural Model

## ● Architecture-Level Model (ALM)

### ● Represent architecture of SoC

#### ● Components of ALM,

- Functional block set  $F$  – instance of Intellectual Property (IP) Module
- Communication port set  $PT$  – master and slave ports of functional blocks
- Direct Memory Access Controller (DMAC) set  $D$
- Memory set  $M$
- Shared bus set  $B$
- Bus Matrix set  $BM$
- Information about mapping SLM to ALM and communication path
- Information of each component such as execution frequency, bus width etc.

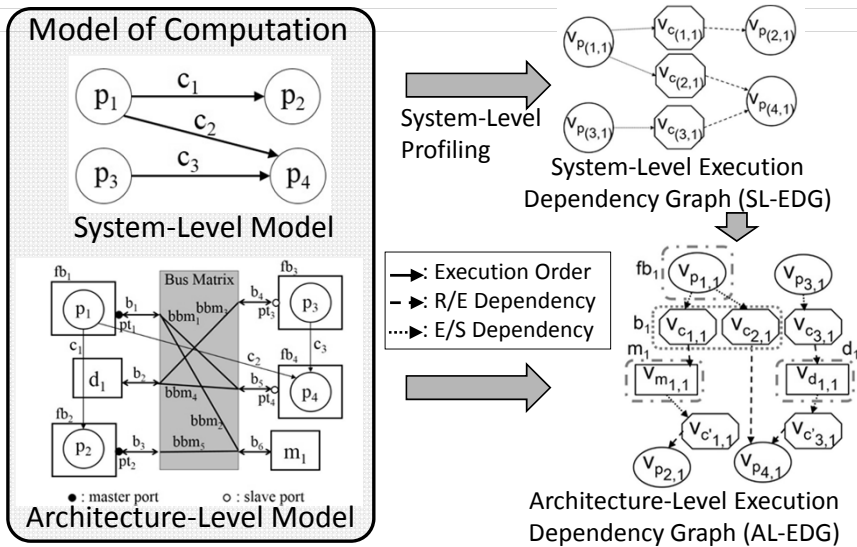


$$ALM M_{al} = (F, PT, D, M, B, BM)$$

# Definition of the Performance Estimation Method

- Input
  - $M_{sl}$  : An SLM describing behavior of a system
  - $M_{al}$  : ALMs specifying components and mappings of an architecture
  
- Output
  - $T_{sys}$  : Total execution time of a system described by  $M_{sl}$  when executed on  $M_{al}$ , considering concurrent data processings and transfers

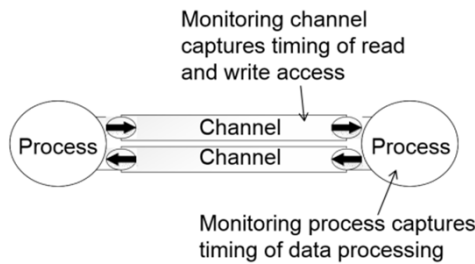
# Efficient Performance Estimation Method





# System-Level Profiling using SystemC

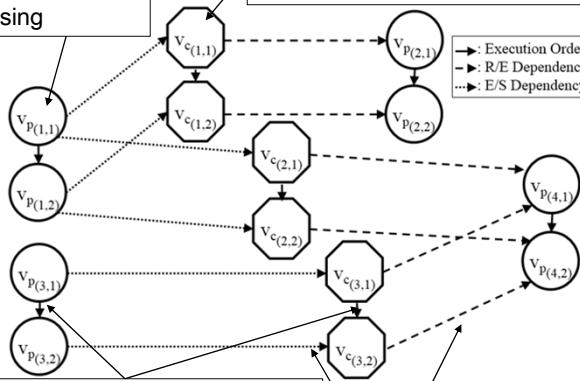
- Profile executions of data processing and data transfer
- Tool :
  - Language : SystemC
  - Simulator : SystemC simulator



# SL-EDG Construction

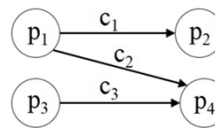
Vertex representing execution of data processing

Vertex representing execution of data communication



Vertices and edges constructed based on timing information from system-level profiling

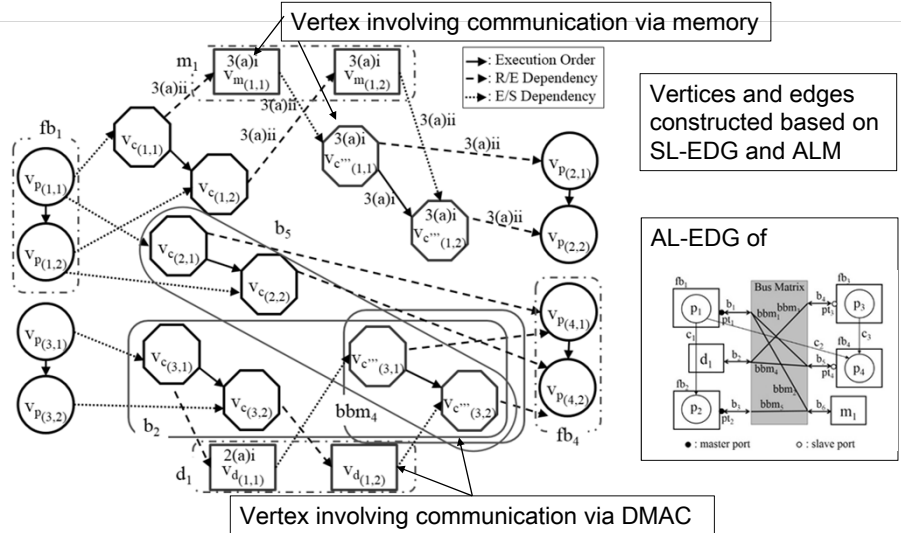
SL-EDG of



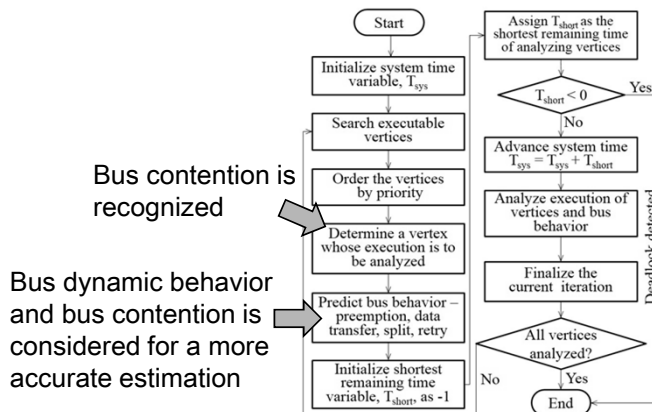
Edge representing execution order

Edge representing execution dependency

# AL-EDG Construction




# AL-EDG Analysis – Execution Time Estimation



# Agenda

---

- Introduction
- Definition of Models:  
System-Level Model and Architectural Model
- Architecture Exploration Method for MPSoC
- Performance Estimation Method
- Configurable Multi-layer Bus-based SoC
-  Case Study
  - Conclusion and Future Work

# Modeling of Multi-layer AHB Protocol

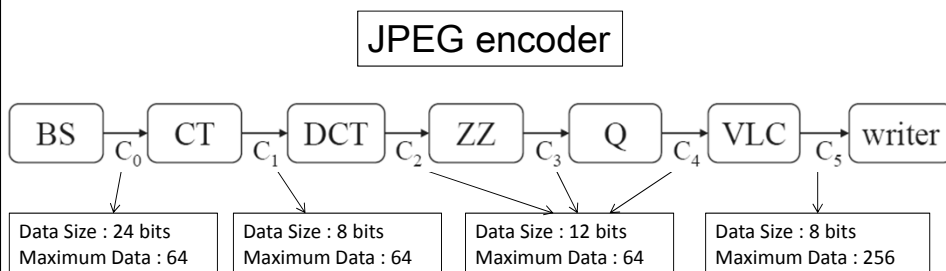
---

- Protocol related parameters
  - Address cycle : 1 cycle
  - Data cycle : 1 cycle(AHB)
  - Burst beats : 1,2,4,8,16 beats except for preemption
  - Split response, Retry response : 2 cycles
- Bus dynamic behavior
  - Pipeline transfer
  - Burst transfer
  - Lock transfer : for communication via bus matrix
  - Split and retry response
  - Bus preemption

# Experimental Setup

- Machine Spec
  - 3.60 GHz Intel Xeon
  - 32 GB memory
  - 64-bit CentOS
- Implementation of SLM and system-level profiling
  - SystemC 2.3.0
- Implementation of proposed performance estimation method
  - Language : C
  - Compiler : GNU gcc 4.1.2
- Implementation of conventional RTL
  - Language : VHDL
  - Simulator : ModelSim SE-64 10.3

# Target Application



# ALMs used in Case Study

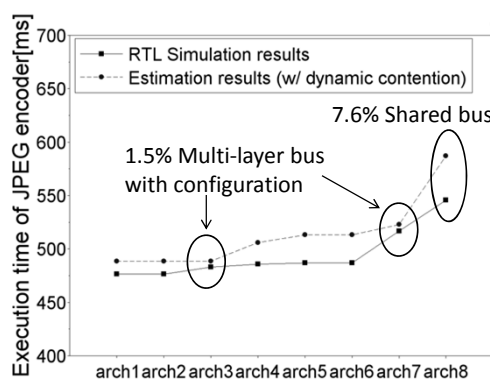
Information of buses  
 Bus width : 32 bits  
 Frequency : 50 MHz

Information of functional blocks

FB name <sup>-1</sup>	Exe.cycle [cycle]	Port
BS	67	1 Slave
CT	68	1 Master
DCT	368	1 Slave
ZZ	67	1 Slave
Q	68	1 Master
VLC	200-265	1 Master
WRT	258	1 Slave

2015/07/14      MPSOC 2015      41

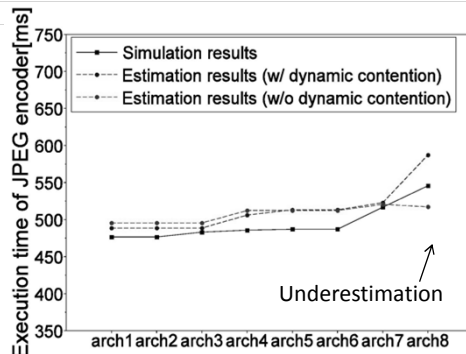
# Accuracy of the Proposed Method



- Estimation results when recognizing bus dynamic contention
- Smallest error : 1.5% (arch3, arch7)
- Biggest error : 7.6% (arch8)
  - Accumulated error on shared bus
- Average error : 3.8%
  - Worst case execution time is used

JPEG encoder application  
 Image size : 1,024 × 1,024 pixels

# Effects of Recognizing Bus Contention



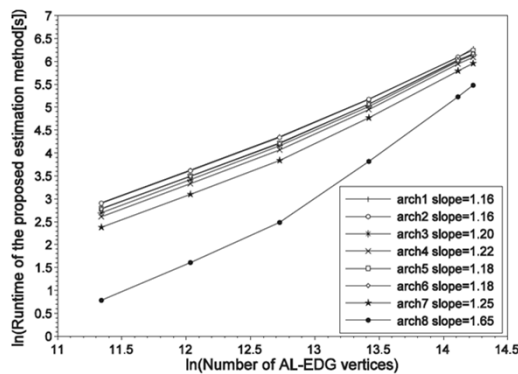
JPEG encoder application  
Image size : 1,024 × 1,024 pixels

Since data processing of JPEG encoder dominates data communication between IPs, the impact of considering dynamic bus contention and behavior is not so large

- ▶ Estimation results when not recognizing bus contention and bus dynamic behavior
- ▶ Less accurate than the proposed method considering bus contention and dynamic behavior
- ▶ Underestimation due to too optimistic about bus contention

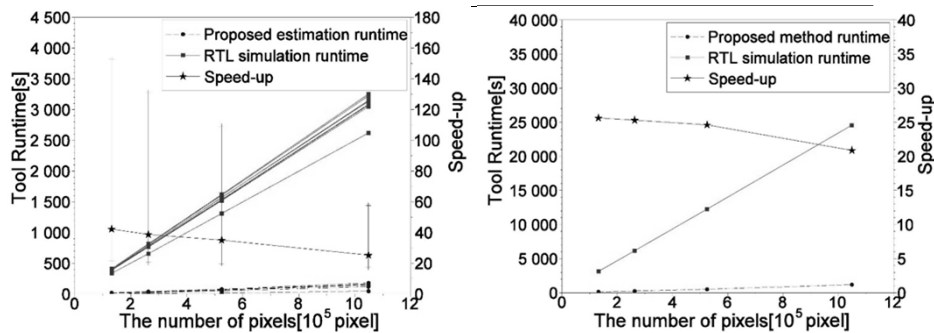
# Computational Complexity

- Time complexity of performance analysis algorithm
  - Worst case :  $O(n^3)$
- From the experiment



- ▶ Time complexity in common cases is between  $O(n)$  and  $O(n^2)$ 
  - ▶ Executable vertices of each component exists as few as no more than 5-6 vertices at a time
  - ▶ Reduce complexity for ordering vertex according to priority

## Tool Runtime and Speed-up



- Speed-up for estimating performance of each individual architecture compared to RTL simulation
  - Speed-up range : 17.4~152.6 times

- Speed-up for estimating performance of 8 architectures compared to RTL simulation
  - Maximum speed-up : 25.6 times

## Discussion

- Abstraction level is between untimed- and timed-model
  - Loosely-timed simulation takes place in system-level profiling procedure
  - Static analysis repeatedly executed to estimate performance
- Advantage over dynamic simulation (RTL, CA, BCA)
  - Require less modeling effort
    - One SLM is implemented and its profiling information used for performance estimation of many ALMs
  - Require less time for performance estimation
    - Approximately 30-35 times faster than CA simulation
  - Require less memory resource

# Agenda

---

- Introduction
- Definition of Models:  
System-Level Model and Architectural Model
- Architecture Exploration Method for MPSoC
- Performance Estimation Method
- Configurable Multi-layer Bus-based SoC
- Case Study



Conclusion and Future Work

# Conclusion and Future work

---

- Design space exploration method
  - Find architecture candidate via tree traversal for components and parameters
  - Accelerate exploration by search tree pruning with upper bound and lower bound of design quality
- Efficiency of performance estimation method for multi-layer bus-based SoC is fast and accurate
  - Compared to RTL simulation : estimation error is within 8% and 25.6 times speed-up is achieved
- Future work
  - Variable timing behavior and statistical analysis must be considered in future study





**Thank you  
for your attention!**

