

# System-Level Design Space Exploration for HMPSoC

*Luigi Pomante*

*luigi.pomante@univaq.it*



**University of L'Aquila – ITALY**  
**Center of Excellence DEWS**



# Overview

- Introduction
- System-Level Design Flow
- Target Architecture
- Application Specification
  - Internal Models
- Design Space Exploration
  - Exploration Strategy
  - System-Level Metrics
- Conclusions

# Overview

- **Introduction**
- System-Level Design Flow
- Target Architecture
- Application Specification
  - Internal Models
- Design Space Exploration
  - Exploration Strategy
  - System-Level Metrics
- Conclusions

# Introduction

- HMPSoC have been recently exploited for a wide range of application domains, for both the embedded and the general purpose products
  - Such systems can include several and different processors, memories, dedicated *ICs* and a set of interconnections between them
- They are so complex that the design methodology plays a major role in determining the success of a product

# Introduction

- For this, in the past few years, a discrete number of research works has focused on system-level co-design of HMPSoC
  - Each of them has proposed a different approach to the design space exploration but all of them always rely on some designer experience to define some aspects of the target architecture
    - In particular, the definition of the communication architecture is often only an input to the design flow (typically imposed by a platform-based approach)

# Introduction

- This talk presents a system-level design space exploration strategy that, starting from the application specification and related constraints, would be able to suggest to the designer
  - an HMPSoC reference architecture
  - an HW/SW partitioning of the given application
  - an allocation of the partitioned entities on the proposed HMPSoC
- Moreover, by means of abstract modeling, the approach try to preserve general applicability and feasibility of the proposed solutions

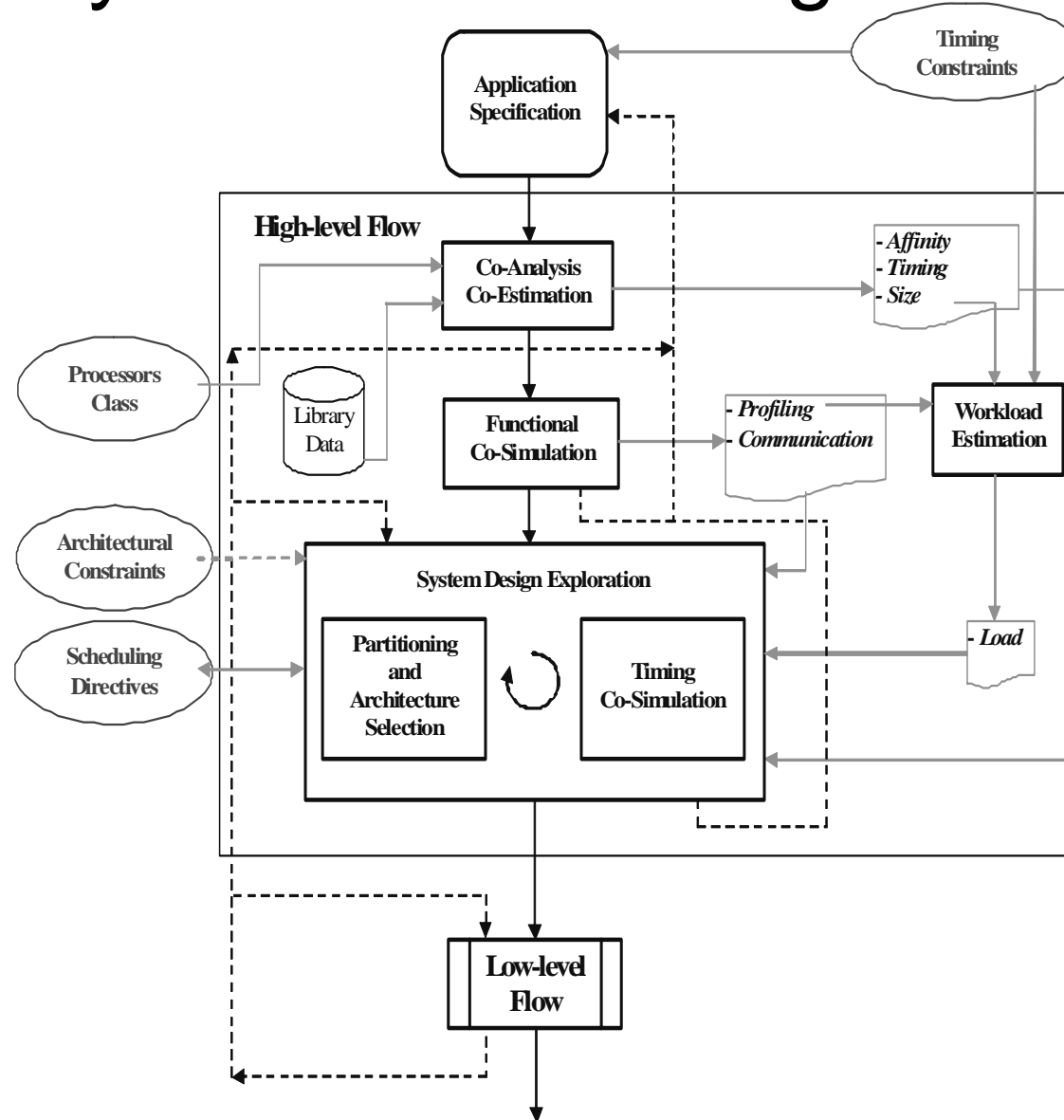
# Introduction

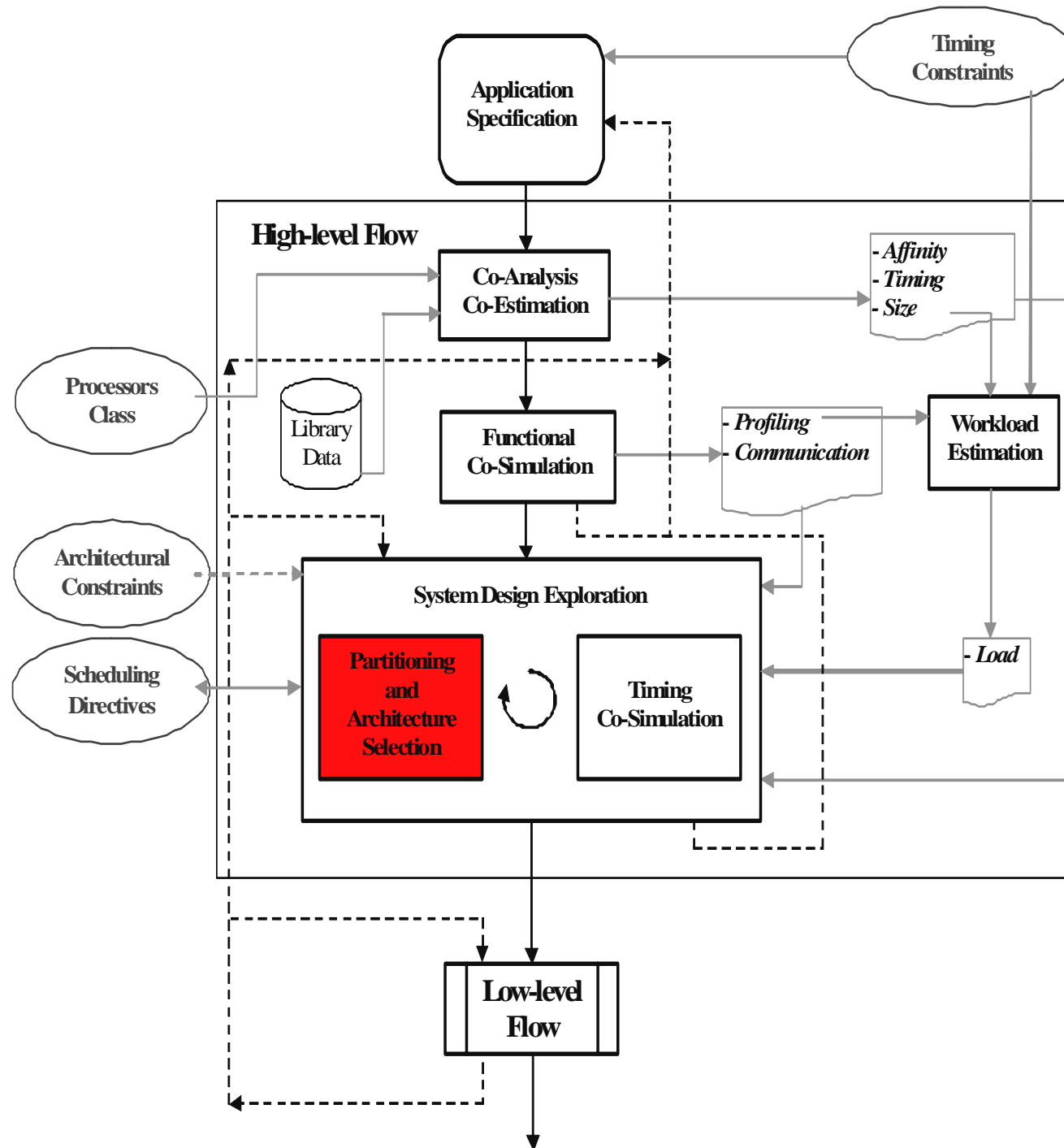
- In particular, this talk focuses on
  - *Modeling Issues*
    - with particular emphasis on model of computation and related “internal models” of representation
  - *System-Level Metrics*
    - used to identify suitable HMPSoC architectures by analyzing the application specification
  - *Design Space Exploration*
    - by means of a genetic algorithm that exploits metrics and some profiling/estimations
      - The approach has been partially validated by checking the consistency with respect to some case studies

# Overview

- Introduction
- **System-Level Design Flow**
- Target Architecture
- Application Specification
  - Internal Models
- Design Space Exploration
  - Exploration Strategy
  - System-Level Metrics
- Conclusions

# System-Level Design Flow



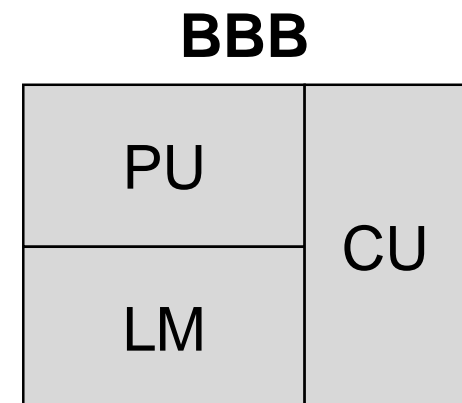


# Overview

- Introduction
- System-Level Design Flow
- **Target Architecture**
- Application Specification
  - Internal Models
- Design Space Exploration
  - Exploration Strategy
  - System-Level Metrics
- Conclusions

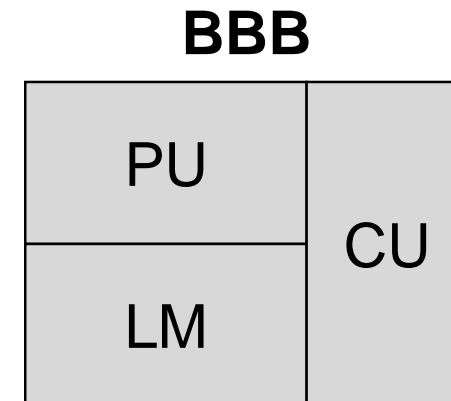
# Target Architecture

- It is an interconnection of some instances of heterogeneous *Basic Building Block*
  - BBB represent the minimal computation, storage and communication unit in the system
    - Distributed memory architecture
  - Each BBB is composed of three main elements
    - The Processing Unit (PU)
    - The Local Memory (LM)
    - The Communication Unit (CU)
      - Number and type of possible instances of interconnection links (IL)



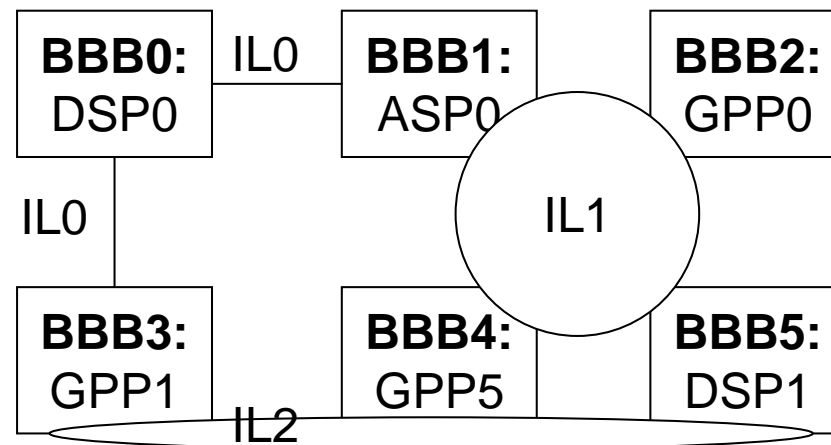
# Target Architecture

- Basic Building Block
  - Processing Unit
    - GPP, DSP, [uC]: { $\epsilon$ ,  $L_{MAX}$ }
    - AS[I]P: { $\epsilon$ ,  $G_{eq\_MAX}$ }
  - Local Memory
    - RAM, ROM: { $\epsilon$ ,  $KB_{MAX}$ }
  - Communication Unit
    - Possible Interconnection links
      - $IL_i$ : { $BW_{MAX}$ ,  $N_{MIN}$ ,  $N_{MAX}$ ,  $\epsilon$ }



# Target Architecture

- Given several instances of BBB and interconnecting them by means of some instances of the available IL it is possible to build a feasible application-specific architecture on which the application can be mapped on

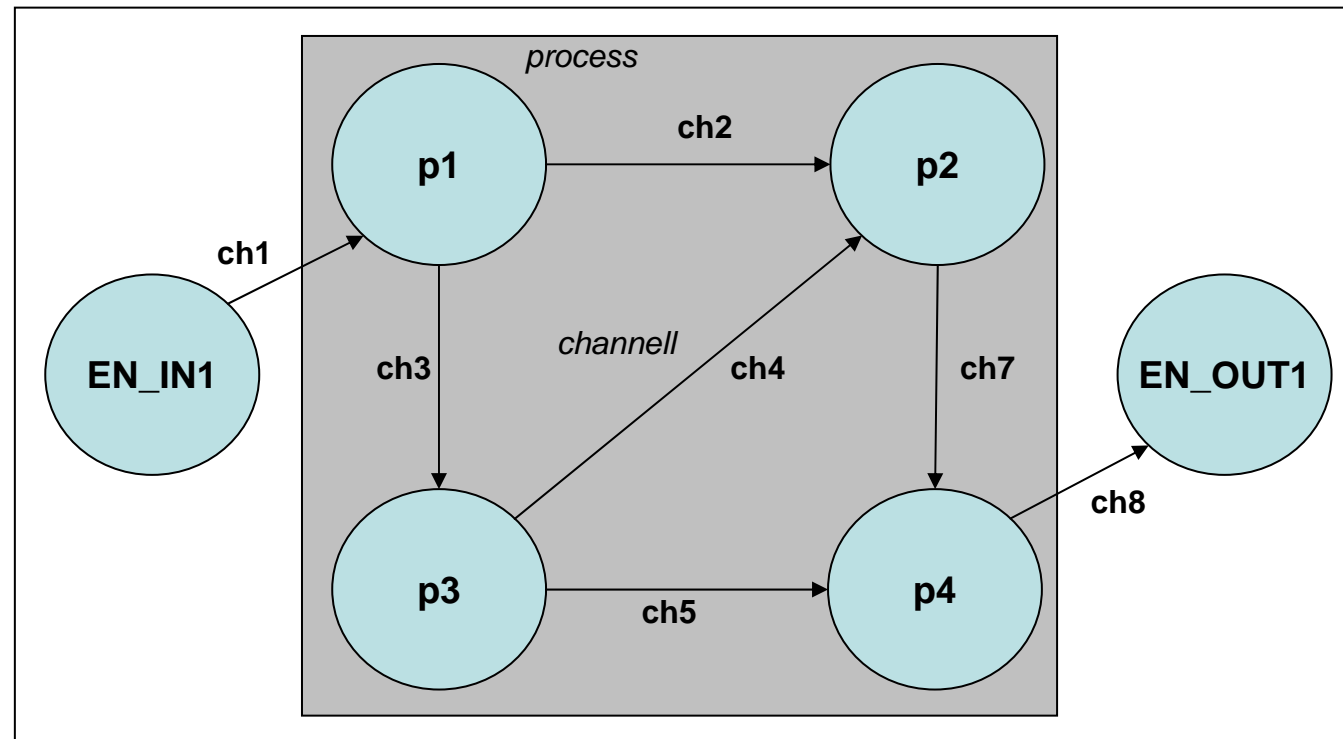


# Overview

- Introduction
- System-Level Design Flow
- Target Architecture
- **Application Specification**
  - **Internal Models**
- Design Space Exploration
  - Exploration Strategy
  - System-Level Metrics
- Conclusions

# Application Specification

- System-Level Model of Computation
  - *Communicating Sequential Processes*
    - e.g.



# Application Specification

- System-Level Specification Languages
  - Some *homogeneous* languages able to support CSP
    - OCCAM
    - HandelC
    - SystemC
    - Simulink
    - ...

# Application Specification

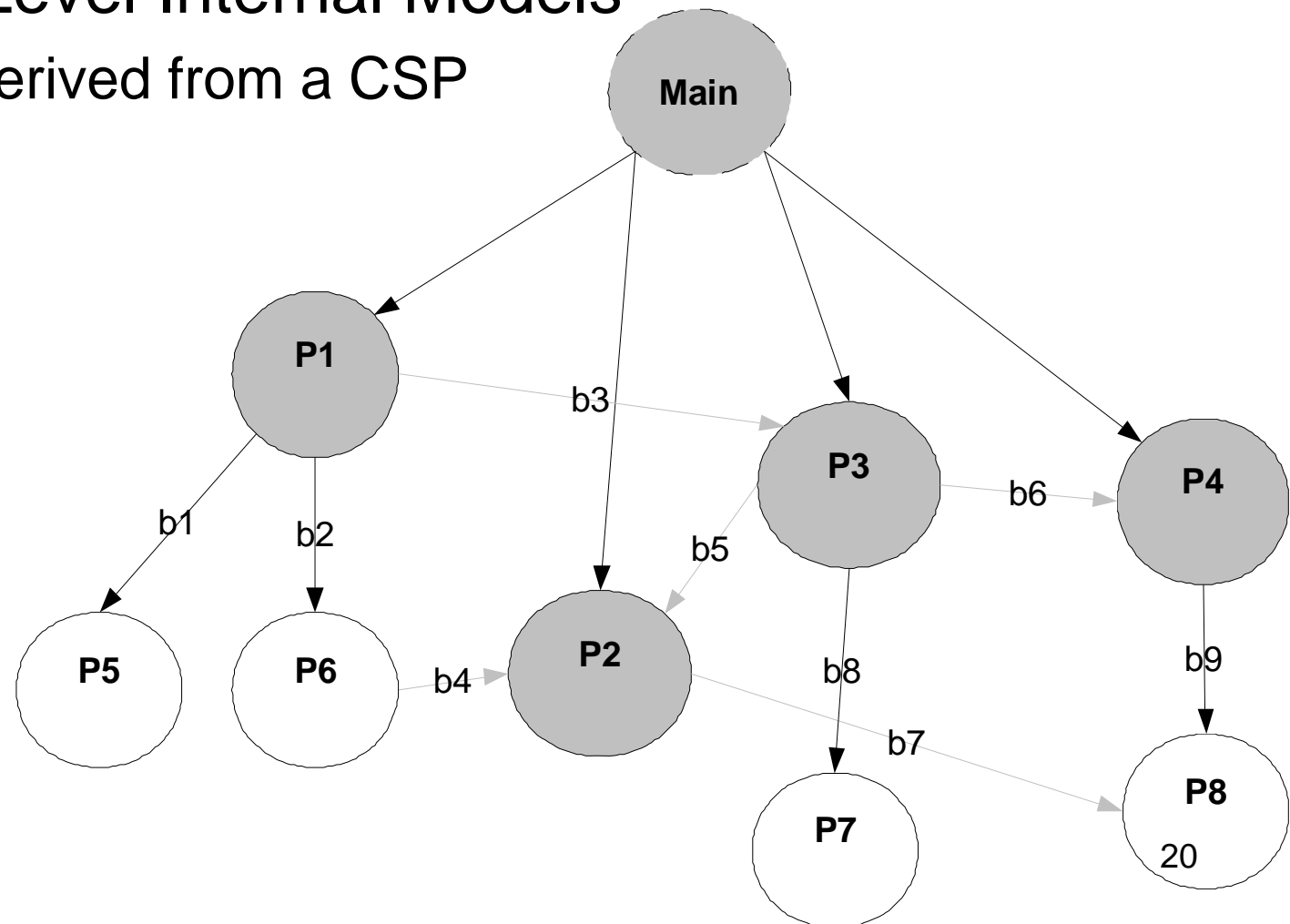
- System-Level Internal Models
  - Different levels of details for IM
    - Statement-level IM
      - Dependent on specification languages and parsers
    - Procedure-level IM
      - Independent from specification languages

# Application Specification

- System-Level Internal Models
  - The procedure-level internal model used in this work during the design space exploration to represent the CSP is called *Procedural Interaction Graph (PING)*
  - The PING is
    - based on the well-known *Procedural Call Graph*
    - able to capture needed information from imperative, possibly *object-oriented*, specification

# Application Specification

- System-Level Internal Models
  - PING derived from a CSP
    - e.g.

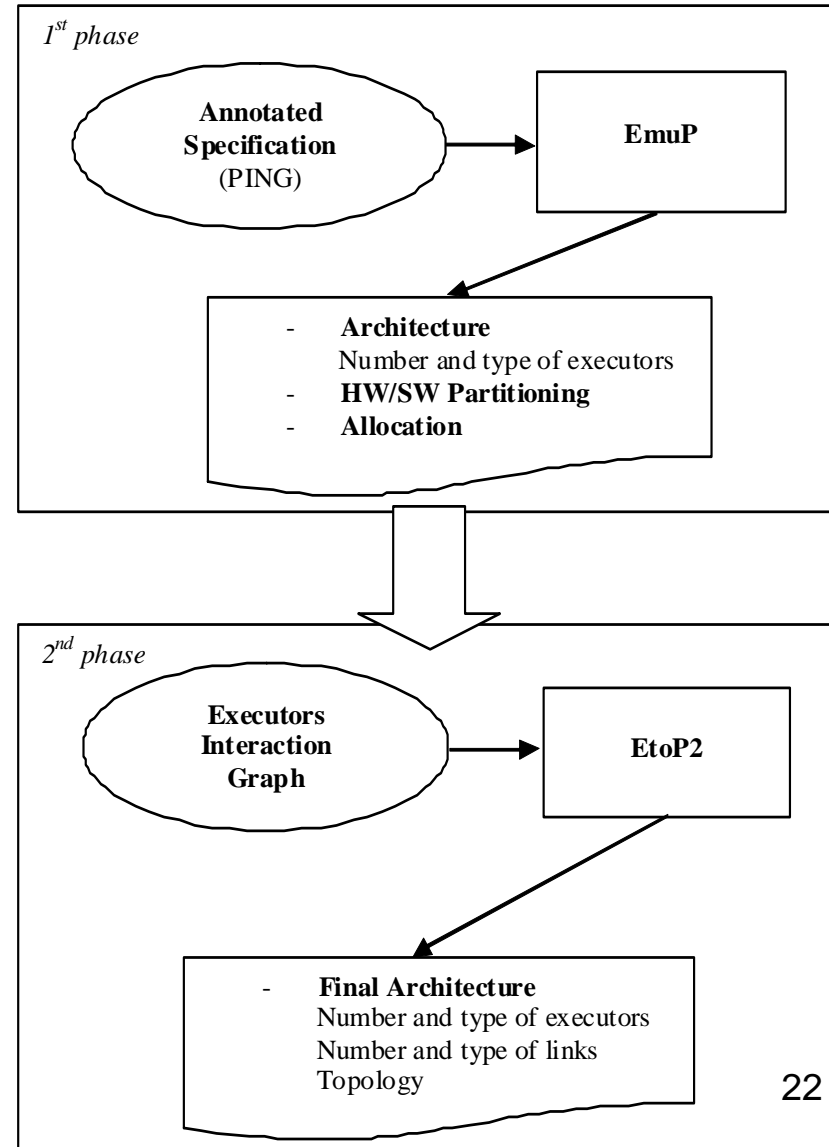


# Overview

- Introduction
- System-Level Design Flow
- Target Architecture
- Application Specification
  - Internal Models
- **Design Space Exploration**
  - **Exploration Strategy**
  - **System-Level Metrics**
- Conclusions

# Design Space Exploration

- The goal of DSE is the automatic identification of both the architecture and the mapping that optimizes a given CF

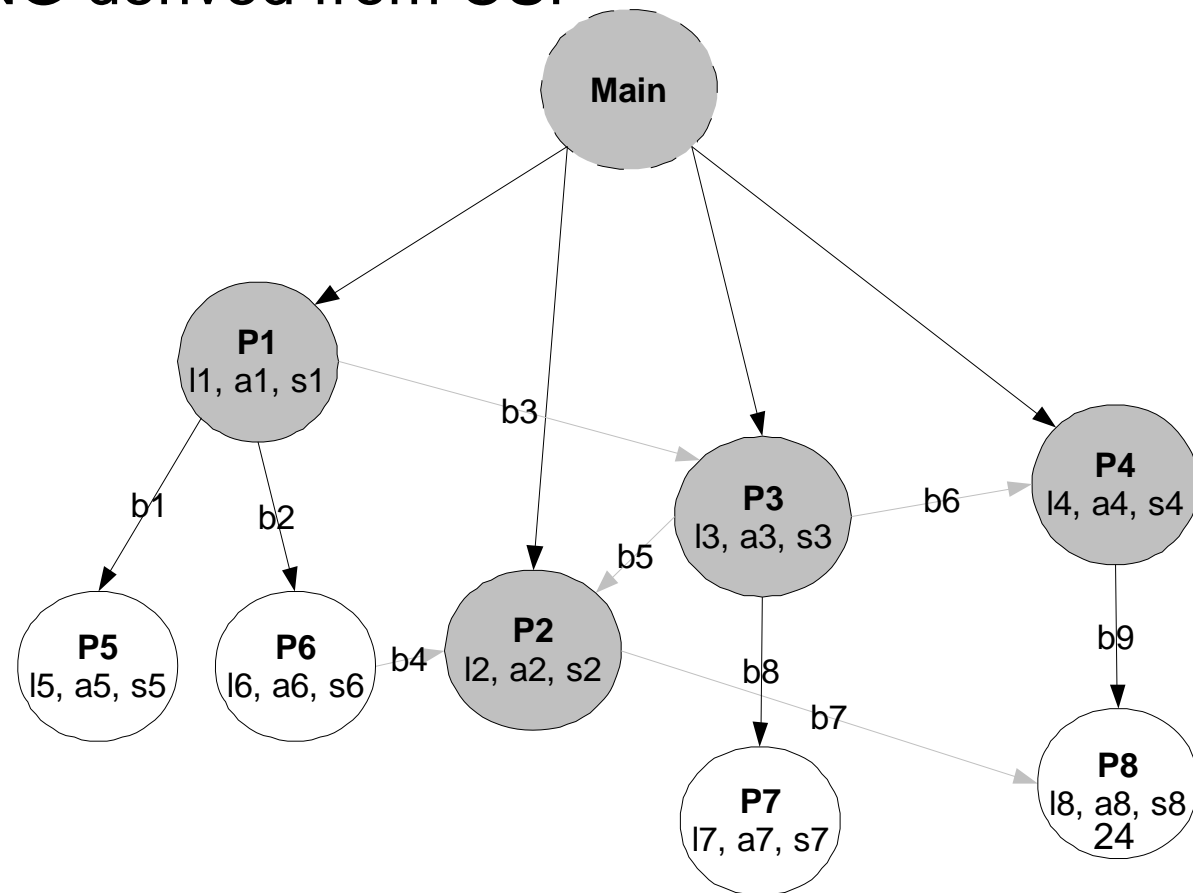


# Design Space Exploration

- Each procedure in the PING is supposed to be annotated by different metrics
  - Load imposed to a single GPP
    - $l_i$
  - Bandwidth needed to communicate with other procedures
    - $b_i$
  - Size
    - $s_i$ : KB or  $G_{eq}$
  - Affinity towards GPP, DSP and ASP
    - $a_i$

# Design Space Exploration

- System-Level Internal Models
  - Annotated PING derived from CSP
    - e.g.



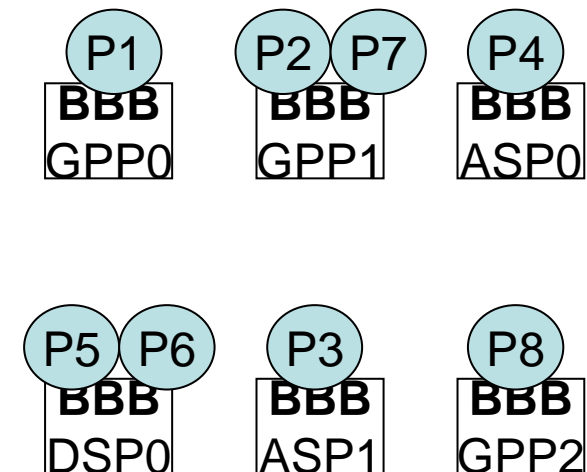
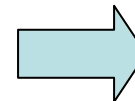
# Design Space Exploration

- 1<sup>st</sup> phase
  - Starting from PING to determine number and type of BBB/PU...
    - Goals
      - Minimize the cost of the set of BBB/PU
        - » Max number for each kind of PU could be provided by the designer
      - Exploit the potential parallelism expressed in the PING
      - Keep the load of each PU near but under its  $L_{MAX}$
      - Minimize communications between different BBB/PU
      - Keep the used size near but under  $KB_{MAX}$  or  $G_{eq\_MAX}$
      - Exploit affinity between BBB/PU and the procedures

# Design Space Exploration

- 1<sup>st</sup> phase
  - ...minimizing a cost function by means of a genetic approach

P1	P2	P3	P4	P5	P6	P7	P8
GPP	GPP	ASP	ASP	DSP	DSP	GPP	GPP
0	1	1	0	0	0	1	2



# Design Space Exploration

- System-Level Metrics (1<sup>st</sup> phase)
  - *Affinity Index* ( $I_A$ ) [0, 1]
    - Matching between the functionalities and the processing elements on which they have been allocated
  - *Load Indexes* ( $I_{Lsw}$ ,  $I_{Lhw}$ ) [0, 1]
    - Balancing of the workload over the available HW and SW processing elements
  - *Communication Index* ( $I_C$ ) [0, 1]
    - Exchanged data size between functionalities allocated on different processing elements
  - *Physical Cost Index* ( $I_{\$}$ ) [0, 1]
    - Cost of the solution with respect to the most expensive one

# Design Space Exploration

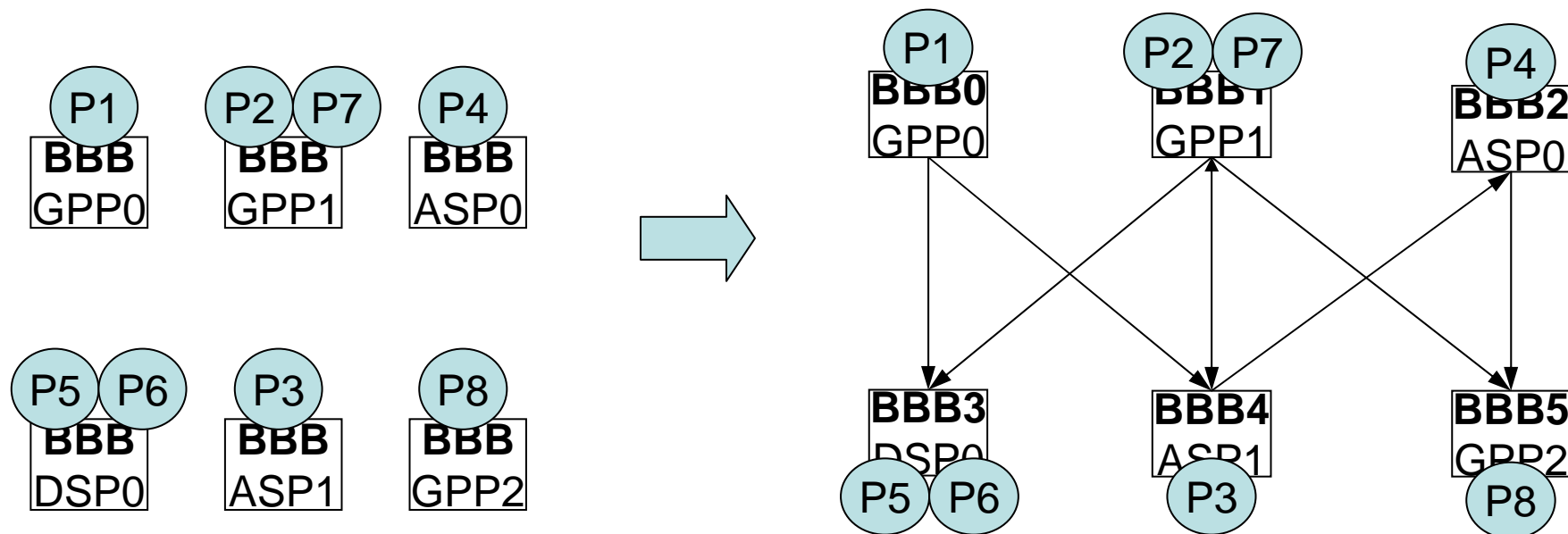
- System-Level Metrics (1<sup>st</sup> phase)
  - By combining the metrics a cost function has been built to compare different solutions

$$CF = w_A \cdot I_A + w_{Lsw} \cdot I_{Lsw} + w_{Lhw} \cdot I_{Lhw} + w_C \cdot I_C + w_{\$} \cdot I_{\$}$$

- Affinity and load parameters
  - Tend mainly to separate the functionalities to balance the load and exploit the processing elements features
- Communications and physical cost
  - Tend to keep together the functionalities to minimize the number of processing elements

# Design Space Exploration

- 2<sup>nd</sup> phase
  - From the results of the 1<sup>st</sup> phase and the PING it is possible to build a BBB Interaction Graph (BING)
  - The BING is a model used to represent the partial system at the end of the first phase
    - Each edge will present a proper  $B=f(b_i)$



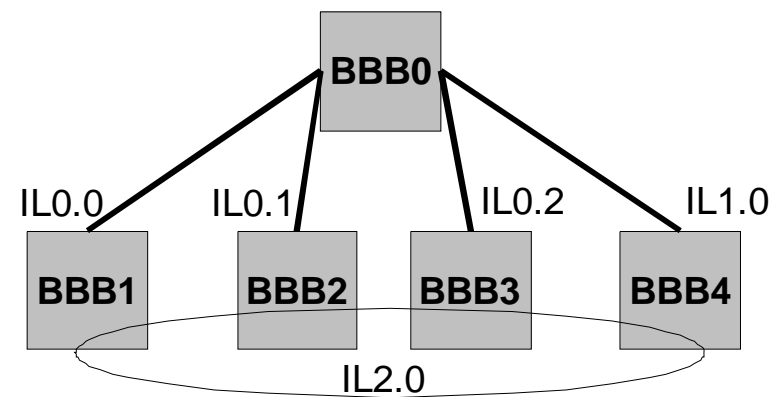
# Design Space Exploration

- 2<sup>nd</sup> phase
  - Starting from BING to determine number and type of IL between PU...
    - Goals
      - Minimize cost of  $IL_i$ 
        - » Max number of instances for each IL can be specified by the designer
      - Keep the bandwidth of each  $IL_i$  under but near  $BW_{MAX}$
      - Keep the number of executors of each  $IL_i$  under but near  $N_{MAX}$  (and  $> N_{MIN}$ )
      - Satisfy eventual constraints on the latency
      - Keeping feasibility while respecting CU characterization

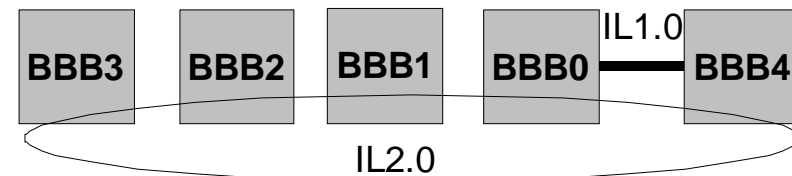
# Design Space Exploration

- 2<sup>nd</sup> phase
  - ...minimizing a cost function by means of a genetic approach

<i>BBB0</i>					
<i>BBB1</i>	<b>IL0.0</b>				
<i>BBB2</i>	<b>IL0.1</b>	<b>IL2.0</b>			
<i>BBB3</i>	<b>IL0.2</b>	<b>0</b>	<b>0</b>		
<i>BBB4</i>	<b>IL1.0</b>	<b>IL2.0</b>	<b>IL2.0</b>	<b>0</b>	
	<i>BBB0</i>	<i>BBB1</i>	<i>BBB2</i>	<i>BBB3</i>	<i>BBB4</i>



<i>BBB0</i>					
<i>BBB1</i>	<b>IL2.0</b>				
<i>BBB2</i>	<b>IL2.0</b>	<b>IL2.0</b>			
<i>BBB3</i>	<b>IL2.0</b>	<b>0</b>	<b>0</b>		
<i>BBB4</i>	<b>IL1.0</b>	<b>IL2.0</b>	<b>IL2.0</b>	<b>0</b>	
	<i>BBB0</i>	<i>BBB1</i>	<i>BBB2</i>	<i>BBB3</i>	<i>BBB4</i>



# Design Space Exploration

- System-Level Metrics (2<sup>nd</sup> phase)
  - *Saturation Index* ( $I_B$ ) [0, 1]
    - Respect of max bandwidth offered by the IL
  - *Exploitation Index* ( $I_E$ ) [0, 1]
    - Respect min/max number of BBB that can use a single IL instance
  - *Physical Cost Index* ( $I_{\epsilon}$ ) [0, 1]
    - Cost with respect to the expensive solution
  - *Feasibility Index* ( $I_F$ ) [0, 1]
    - A pair of CU should be able to manage at least a common IL in order to allow the related BBB to directly communicate

# Design Space Exploration

- System-Level Metrics (2<sup>nd</sup> phase)
  - CU characterization for feasibility
    - e.g.

	IL0	IL1	IL2	IL3	IL4
<b>BBB0-CU</b>	X	X	X		X
<b>BBB1-CU</b>	X	X	X	X	X
<b>BBB2-CU</b>	X		X		X
<b>BBB3-CU</b>	X		X	X	
<b>BBB4-CU</b>		X	X	X	X

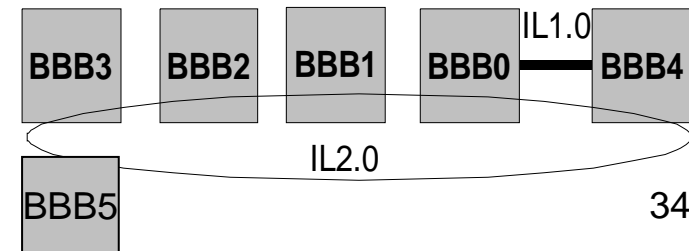
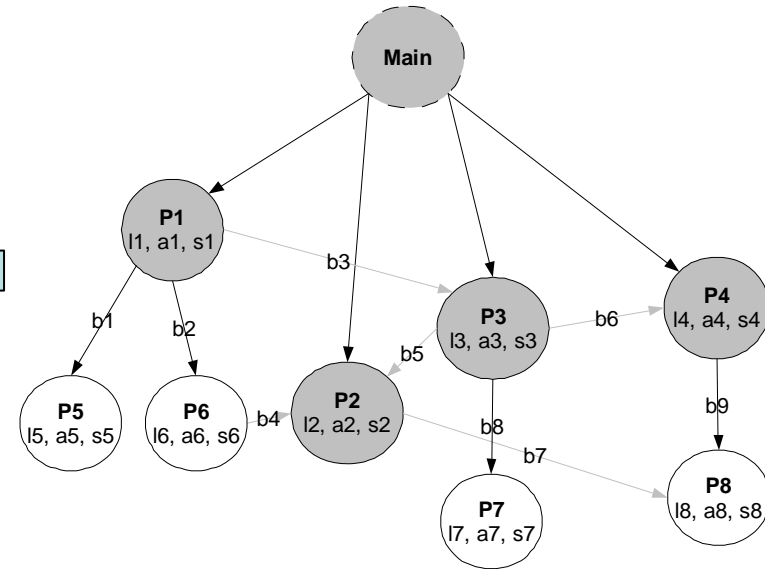
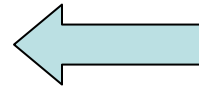
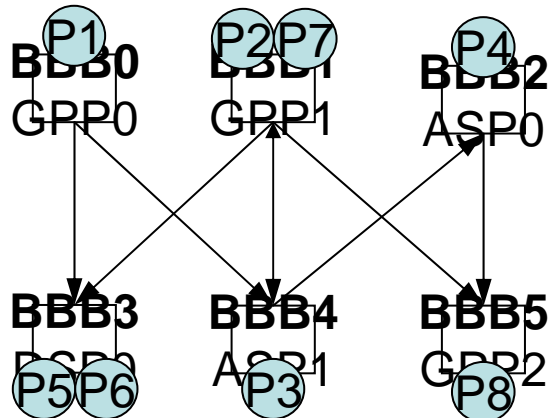
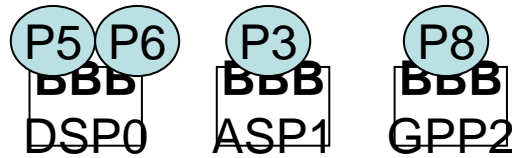
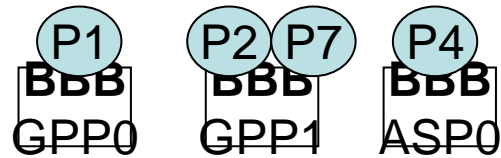
## – Cost Function

- Used to compares different proposed solutions identifying the one that better tradeoffs different parameters

$$CF = w_B \cdot I_B + w_E \cdot I_E + w_{\epsilon} \cdot I_{\epsilon} + I_F$$

# Design Space Exploration

- Summary



# Overview

- Introduction
- System-Level Design Flow
- Target Architecture
- Application Specification
  - Internal Models
- Design Space Exploration
  - Exploration Strategy
  - System-Level Metrics
- **Conclusions**

# Conclusions

- This talk has presented a System-Level Design Space Exploration strategy for HMPSoC
  - It has presented a methodology able to propose an HW/SW partitioning of the specification, mapping this one onto an automatically selected architecture
    - The methodology has been only partially validated
- The experimental results are encouraging and justify further research efforts in this direction may be also thanks to your comments...