Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

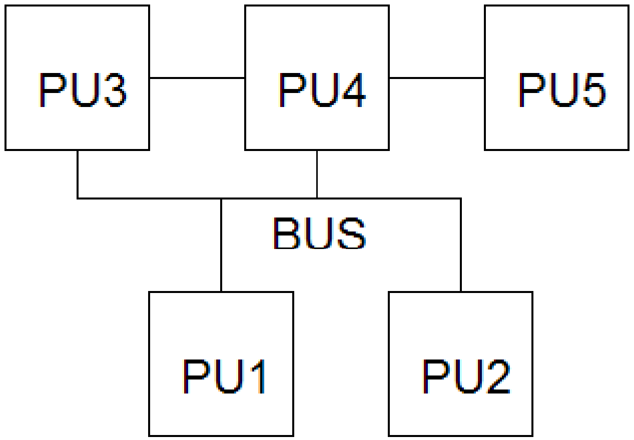
Probabilistic embedded systems design

Contents

- System Definition
- Probabilistic Approach
- Consequences
- Related Work

MPSoC

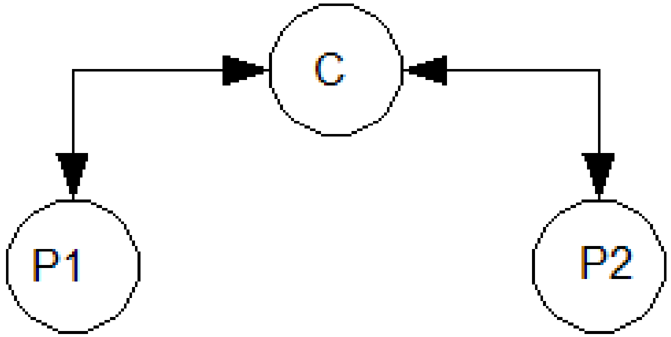
- Multiple processing unit
- Multiple interconnections
- Multiple memory



Hardware Model

H = (P, C, L, φ,λ,τ,σ)

P	processing Units
C	communication units
L	edges
φ	power dependent
λ	available comp. resources
τ	bandwidth
σ	delay



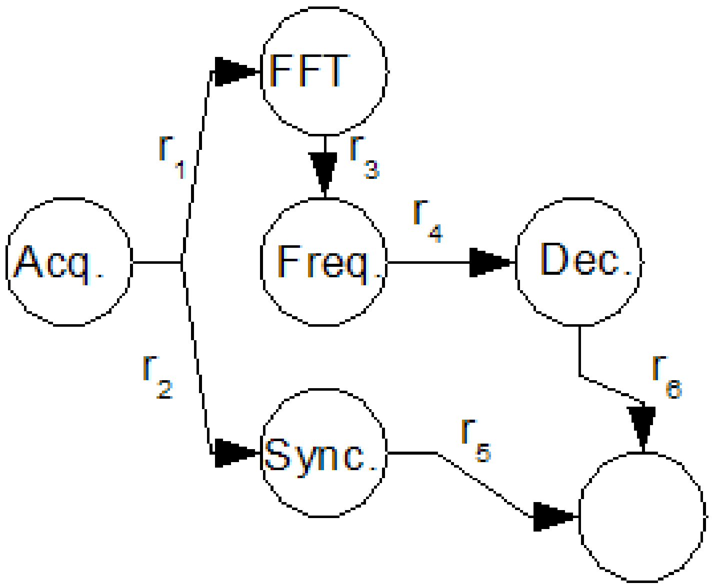
Applications

- Defined by tasks
- Tasks communicate
- Task have an order of execution
- Communication and Computational properties have modes

Application Definition

$A = (T, E, \gamma, \varepsilon, \theta, \delta, \beta, \omega)$

T	Tasks
E	Edges
γ	computational compl.
ε	deadline
θ	app. that share task
ω	rate
δ	delay (edge)
β	bandwidth(edge)



Optimization

- Guarantee throughput
- Guarantee delay
- Minimize energy consumption
- Minimize Hardware resources
- Maximize number of applications

Assumptions / Application relation

- Scenarios
  - Set of Applications (Â)
  - Group of Applications that share tasks Ã
  - Only one application out of a group Ã can be active
  - Applications in Â have specific relations
    - $Â_1 * Â_2 = 0$
  - Â has a probability of execution
    - Future: probability of execution with notion of time

What's the System

- Tasks interact with the Hardware
  - Tasks are mapped to a specific processing unit
  - Proc. Unit have properties that change tasks behavior
- Robustness ?
- Adaptive ?
- Resilient ?
- Exposure: the set of Appl. defined by Scenarios

Probabilistic approach (1)

- Do not design for worst case scenario
- Define probable use-cases
- Probabilities and Exclusion Rules
  - Reachability of conditions

Probabilistic approach (2)

- Optimize according to scenario

$$min\left(\sum_{\forall t_i\in T}\ \sum_{\forall p_j\in P|\mathrm{map}(t_i)=p_j}\left(\frac{\gamma_i\omega_i}{\lambda_j}\mathrm{prob}(\theta_i)\right)\phi_{j0}+\left(1-\frac{\gamma_i\omega_i}{\lambda_j}\mathrm{prob}(\theta_i)\right)\phi_{j1}\right)$$

Advantages

- less hardware, more applications
- smaller proc. units, less energy consumption
- sufficient for “Soft” real-time systems

Problems

- Analytical proof ?
- Design space explosion
- "Hard" Real-Time systems ?
  - Safety critical devices (Airbus A380)

Related Work

[1] N. H. Zamora, X. Hu, R. Marculescu, *System-Level Performance/Power Analysis for Platform-Based Design of Multimedia Applications,* ACM Transactions on Design Automation of Electronic Systems, Vol. 12, No. 1, Article 2, January 2007

[2] S. Hua, G. Qu, S.S. Bhattacharyya, *Energy Reduction Techniques for Multimedia Applications with Tolerance to Deadline Misses,* Proceedings of the Design Automation Conference, pages 131-136, Anaheim CA, June 2003

[3] J. Hu, R. Marculescu, *Energy- and Performance-Aware Mapping for Regular NoC Architectures,* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, No. 4, April 2005

Questions ?

Thanks for your attention