Combining Memory Optimization with Mapping of Multimedia Applications for MPSoC

Gabriela Nicolescu

Ecole Polytechnique de Montréal
Tel : (514) 340 4711 ext 5434
Fax: (514) 340 3240
Email : gabriela.nicolescu@polymtl.ca

ÉCOLE POLYTECHNIQUE MONTRÉAL

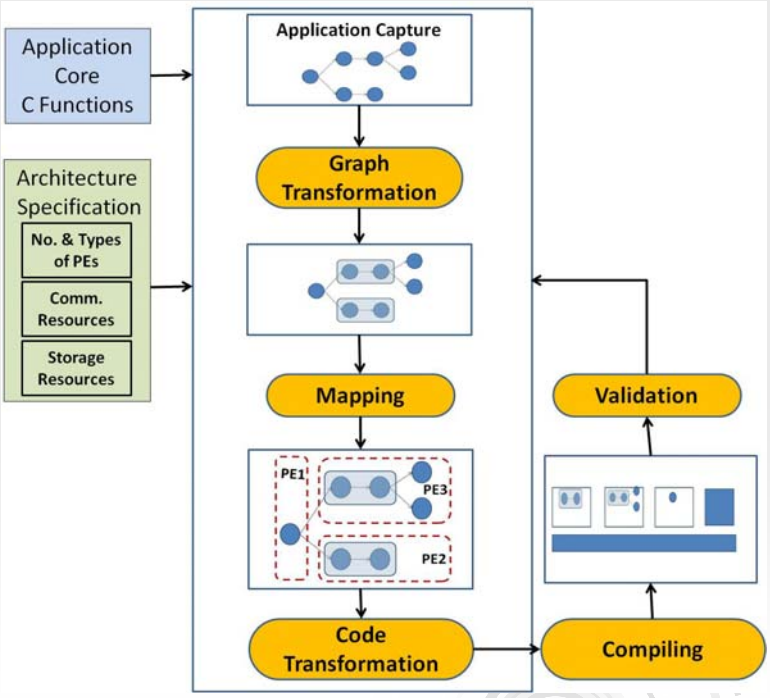
Memory Optimization & Mapping: Key Issues

- Data-oriented applications are one of the main drivers for MPSoC systems
  - Video codecs, image quality improvement
  - Digital still camera
  - 3D graphics, etc.
- Their characteristics challenge the system-level design
  - High number of memory accesses
  - Large memory space
  - High energy consumption
  - Tight real-time constraints
- These characteristics have to be considered while applications are mapped on an MPSoC platform
- Main issue: Combining memory optimization & mapping design stages

Application-Level Transformations in MPSoC

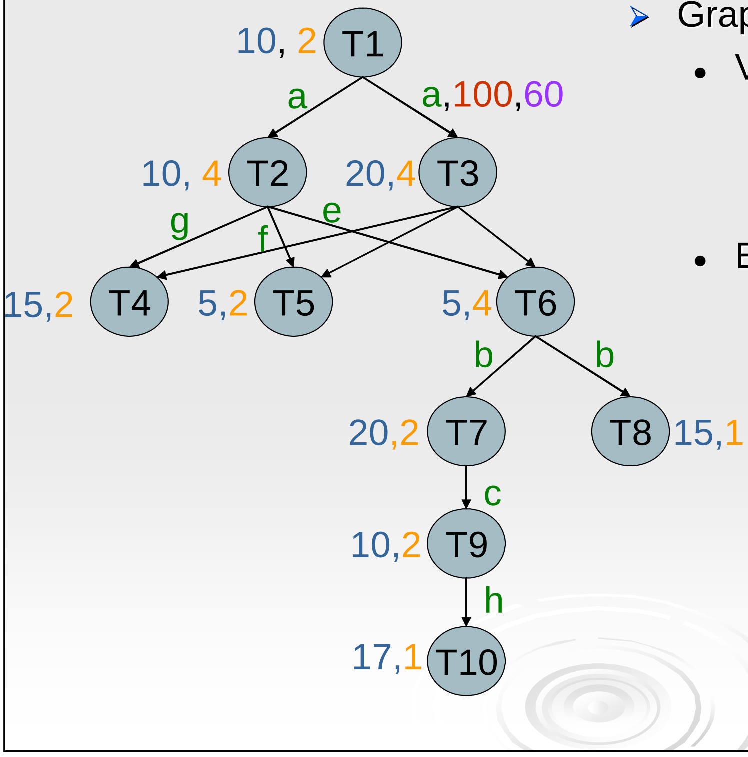
- Streaming Programming model
- Memory optimization transformations
  - Loop Fusion & Buffer allocation
- Linked to MultiFlex STMicroelectronics platform mapping tools
  - Mapped to ST MPSoC streaming platform
- Applications
  - Digital Still Camera: Demosaic

Design Flow



- Streaming Programming Model for Application Capture
- Graph transformation
  - The application model is analyzed and transformed by grouping the tasks for further exploitation of memory optimizations
- Mapping (Task Assignment)
  - Tasks are assigned to processors
- Code transformations
  - Memory optimizations techniques applied locally for each processor

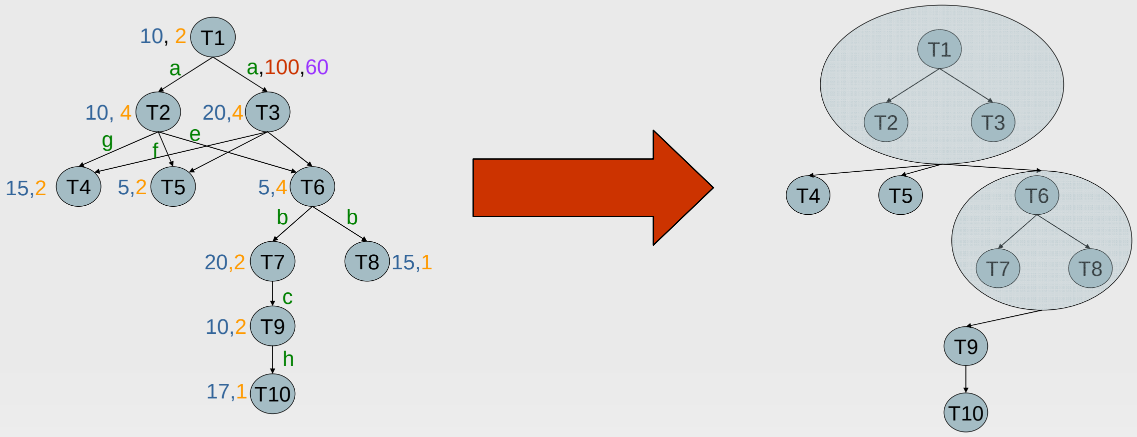
Application Capture



- Graph of communicating tasks
  - Vertex Vi annotations
    - proc. load
    - no. of edges (channels) incident to the task
  - Edge Ei annotations
    - The name of streams (Si) transferred between the task communicating by ei
    - The size of Si
    - The gain(Si) the memory gain that represents the improvement in memory if the tasks connected by ei are merged and uses buffer allocation optimizations.

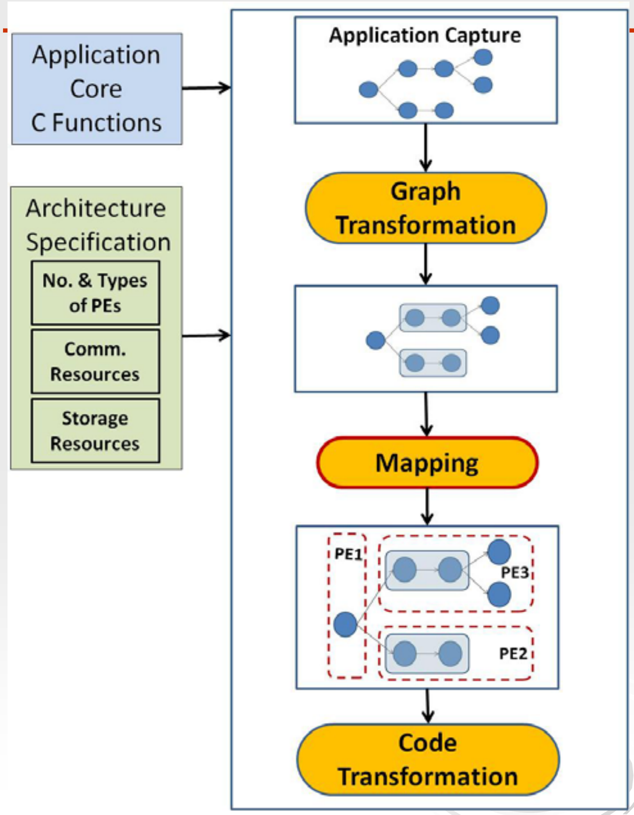
Application Capture Transformation

- Application Capture
- Transformed Application



- The set of tasks with the required properties for the memory optimization transformations (i.e sharing the same array) are grouped
- Constraints
  - Load balancing
  - Communication
  - Memory size

Design Flow



Application Core C Functions

Architecture Specification

No. & Types of PEs

Comm. Resources

Storage Resources

Application Capture

Graph Transformation

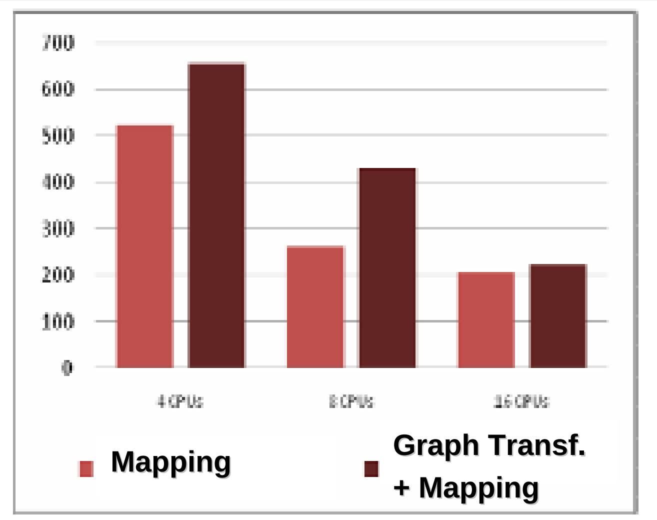
Mapping

Code Transformation

- Mapping based on the following algorithms
  - Ready-list
  - Critical path
  - *Genetic*

Results: Memory Gain

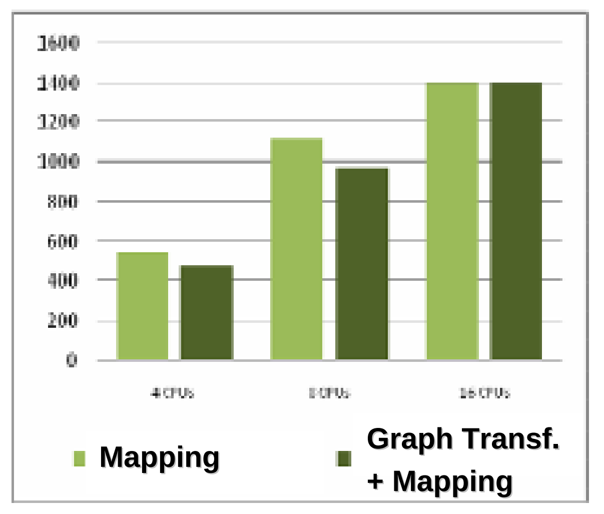
- Demosaic Application



- Increasing the Memory Gain by
  - 4CPUs: 25%
  - 8CPUs: 64%
  - 16CPUs: 9%

Results: Communication

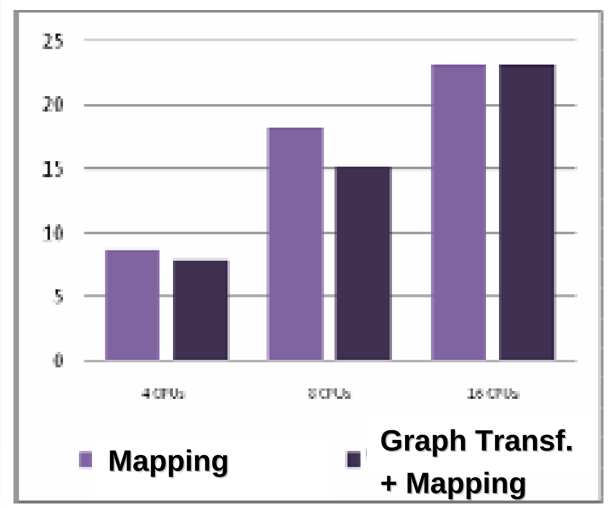
- Demosaic Application



- Decreasing the Communication Load by
  - 4CPUs: 13%
  - 8CPUs: 14%
  - 16CPUs: 0%

Results: Physical Links

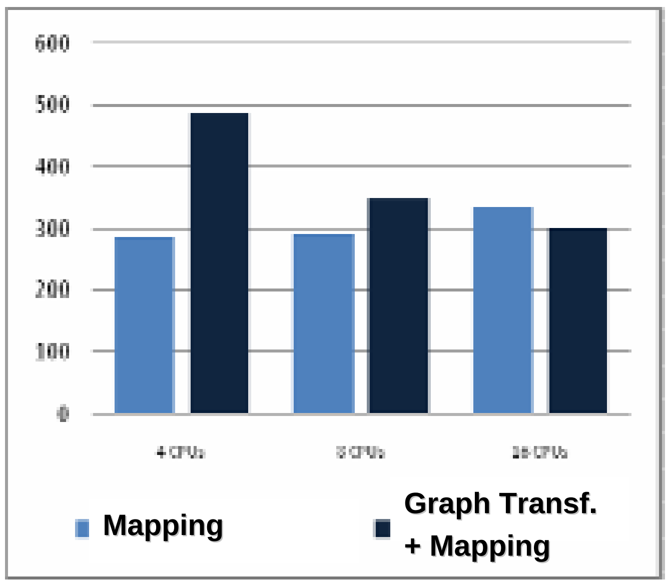
- Demosaic Application



- Decreasing the no. of Physical Links by
  - 4CPUs: 11%
  - 8CPUs: 17%
  - 16CPUs: 0%

Results: Load Balancing

- Demosaic Application



- Increasing the Load Fluctuation by
  - 4CPUs: 69%
  - 8CPUs: 20%
- The impact of memory transformations on power consumption was not considered

Conclusion

- Data-oriented applications presents specific characteristics challenging the system-level design
- Combining memory optimization and mapping algorithms can improve performances for MPSoC data-oriented applications
- First results for a digital still camera application