Co-synthesis techniques for embedded systems

Kelvin Yuk

June 5, 2002

EEC282 - Akella

Introduction

- Co-synthesis of high-level systems
- Approaches to co-synthesis
  1. Architectural Co-synthesis Algorithm
  2. Process partitioning and HW/SW Repartitioning Algorithm
- Some drawbacks
- Conclusions

Why do we need good co-synthesis methods

- Exploring the design space of HW/SW design is costly.
- Need to strike a balance between flexibility (SW) and performance (HW) while reducing cost
- Want to automate synthesis of system to meet design requirements. Alternative => heuristics
- Performance parameters:
  - ➢communication channels
  - ➢hardware costs
  - ➢processing time (SW vs. HW)

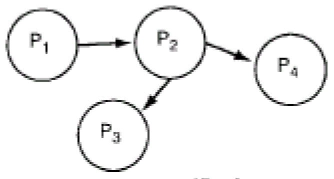
Architectural Co-Synthesis Algorithm

- Models system on a architectural level consisting of processors, memories, I/O controllers etc.
- Targets multiprocessor designs
- Non-hardware specific heuristic algorithm
- Dependant on process scheduling onto available hardware
- Meet specs first, reduce costs in a hierarchical way

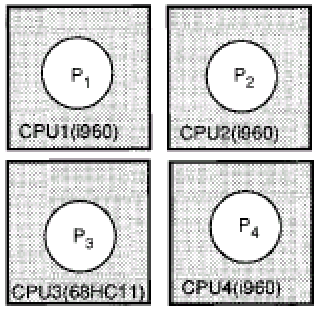
Architectural Co-Synthesis Algorithm cont’d

1. Generate an initial solution by allocating processes to PE’s to meet the specs. Determine communication rates and schedule processes.
2. Reallocate processes to PE’s to minimize PE cost.
3. Reallocate processes again to minimize inter-PE communication
4. Allocate communication channels
5. Allocate devices, either internally to PE’s or externally to communication channels.

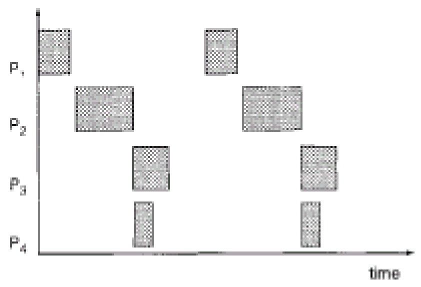
Architectural Co-Synthesis Algorithm and Partial Order Model Partitioning Tree



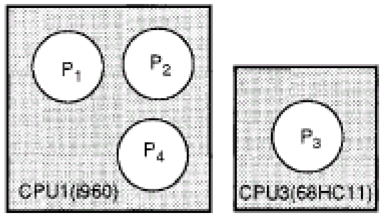
process specification



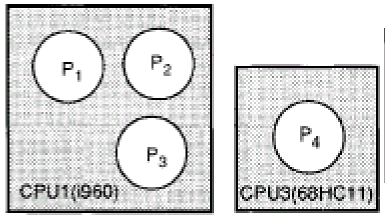
after step 1: initial allocation



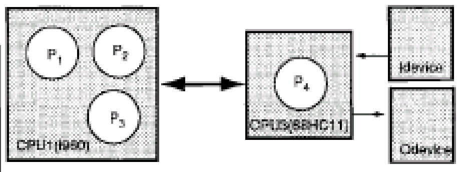
after step 1: initial schedule



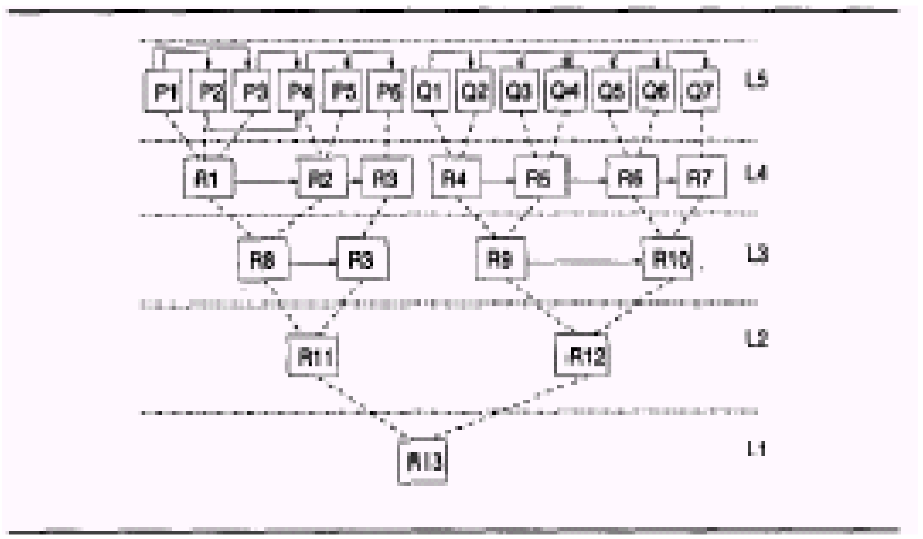
realloation for utilization after step 2



reallocation for communication after step 3



hardware/software architecture with communication and devices after step 5



Process and HW/SW Repartitioning Algorithm

- Based on Hierarchical Partial Order Model (HPOM) which models processes. Performance of HPOM can be estimated. HPOMs can be combined
- Generation of Partitioning Tree (PT) to explore possible configurations of HPOMs
- Initialize system of HPOMs as software (least likely to meet specs) and implement into hardware (most likely to meet specs) where needed

Algorithm

1. System is modeled as a set of basic HPOMs
2. Partitioning tree is generated for sets of HPOMs. Each level of the tree represents a different HPOM realization of the system
3. All HPOMs of all levels are initialized as SW and tested to see if requirements met
4. If specs is not met, the HPOM with the least communication overhead is converted from SW to HW.
5. HW to SW conversion repeats until level meets specs or is fully HW
6. Levels that meet specs are considered as possible implementations of the system

Some Drawbacks

- Partitioning and Repartitioning algorithm is computationally intensive
  - Each HPOM is estimated in terms of speed implementation cost and communication costs
  - HPOMs need to be reestimated when converting from SW to HW
- Multiprocessor algorithm dependant on processes scheduling onto general purpose processors. Lacks flexibility in co-design.
- Architectural: meet specs => reduce costs
- HPOMs: minimal costs => meet specs

Conclusions

- Full exploration of design space is time-consuming
- Co-synthesis algorithms are model dependant
- Heuristic algorithms are not perfect but provide a quick solution in co-design