Using Constraint Programming for Embedded System

Radosław Szymanek

Department of Computer Science

Lund Institute of Technology

Sweden

Email: Radoslaw.Szymanek@cs.lth.se

Embedded System

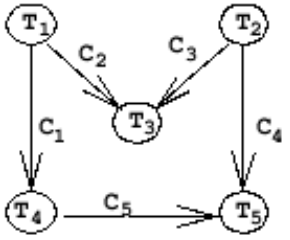
- performs computation but it is not a general purpose computer
- usually performs computation to achieve given goal/functionality
- mobile phones, home appliances, security and safety systems

Issuses within Embedded Systems

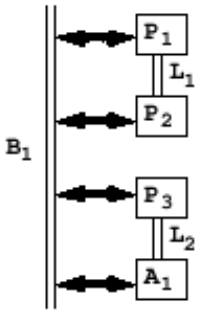
- minimal cost and maximal functionality
- memory issues - increased audio/video requirements
- early good optimization can achieve substanstial gains

Scheduling Problem

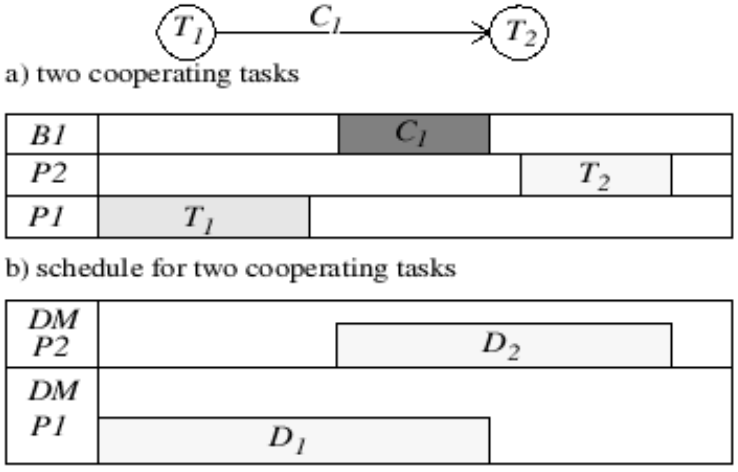
- heterogeneous multiprocessor system
- parallel computation and communication
- heterogeneous constraints - time, memory constraints



a) a task graph example



b) an example of the system architecture



a) two cooperating tasks

b) schedule for two cooperating tasks

c) data memory usage for processors executing these tasks

Constraint description of Scheduling Problem

- diffn, cumulative, element constraints
- arithmetic constraints
- resource constrained scheduling

Optimization

- requires search heuristic
- resource based criteria in optimization function
- reneweable (data memory) and non reneweable resources (time and code memory)

Result

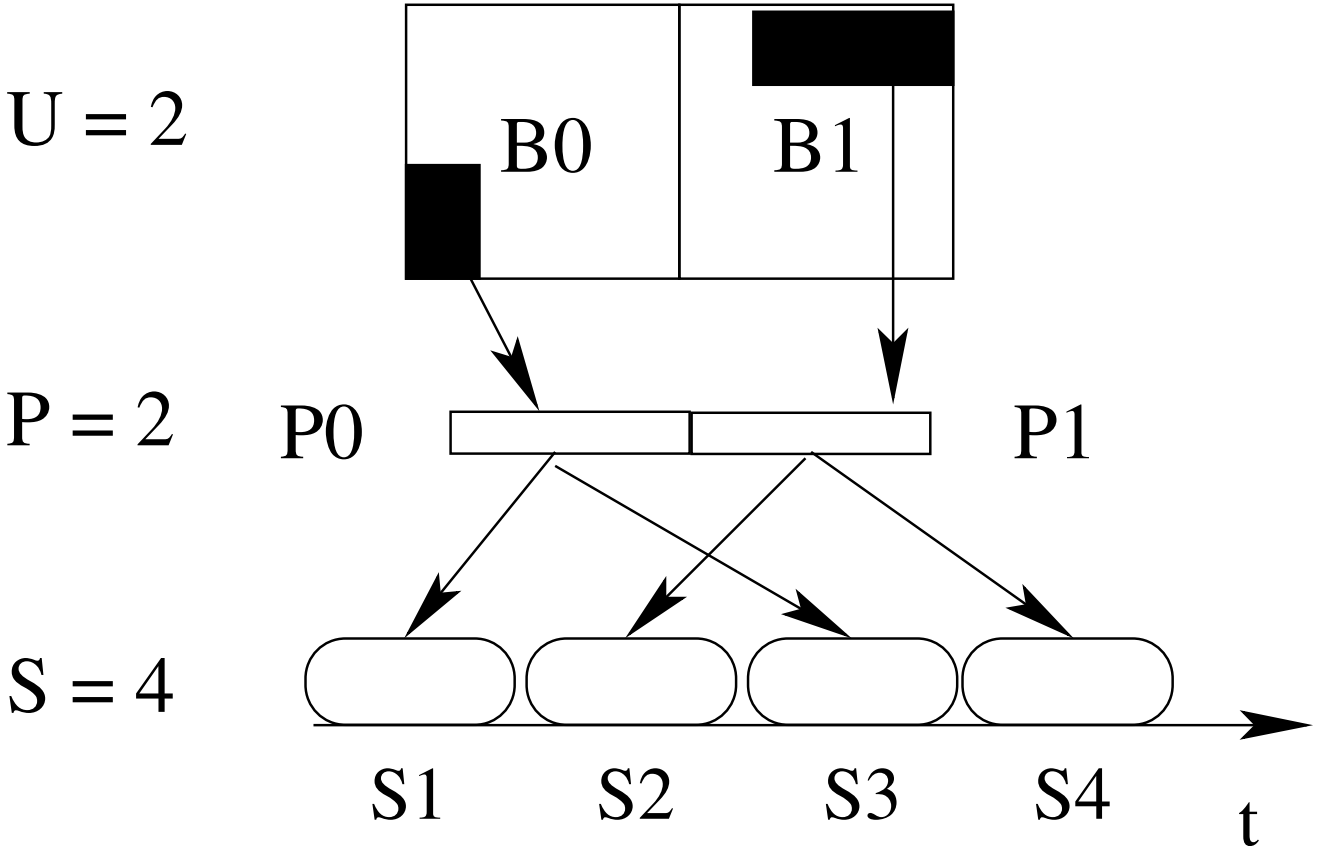
- easy to extend the model
- easy to answer ad-hoc questions, what if?
- guided search methods

Memory Intricacies

- current memory systems allow fast access under given conditions
- high penalties if conditions not respected
- high number of memory modules to increase effectiveness

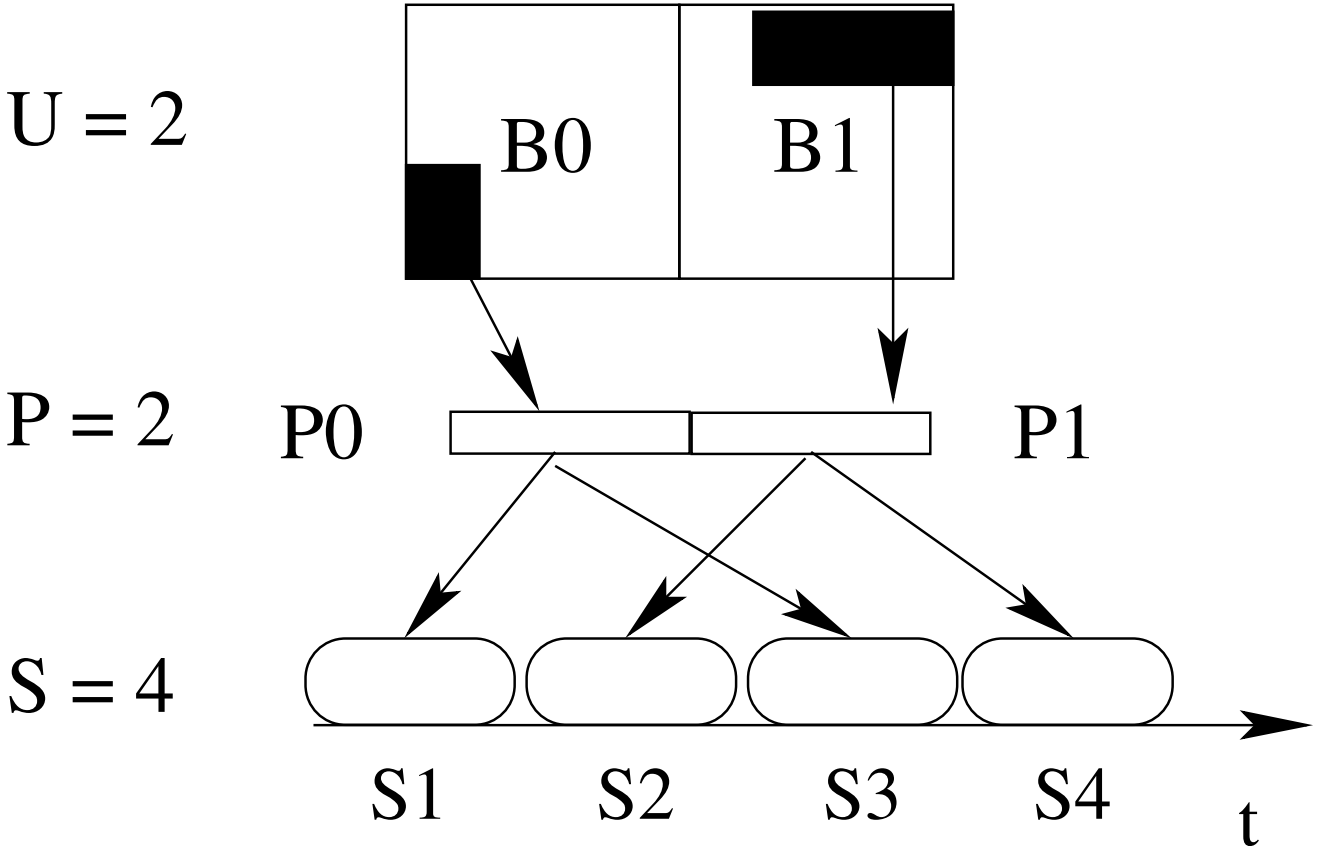
SDRAM memory

- high bandwidth can be obtained if used with care



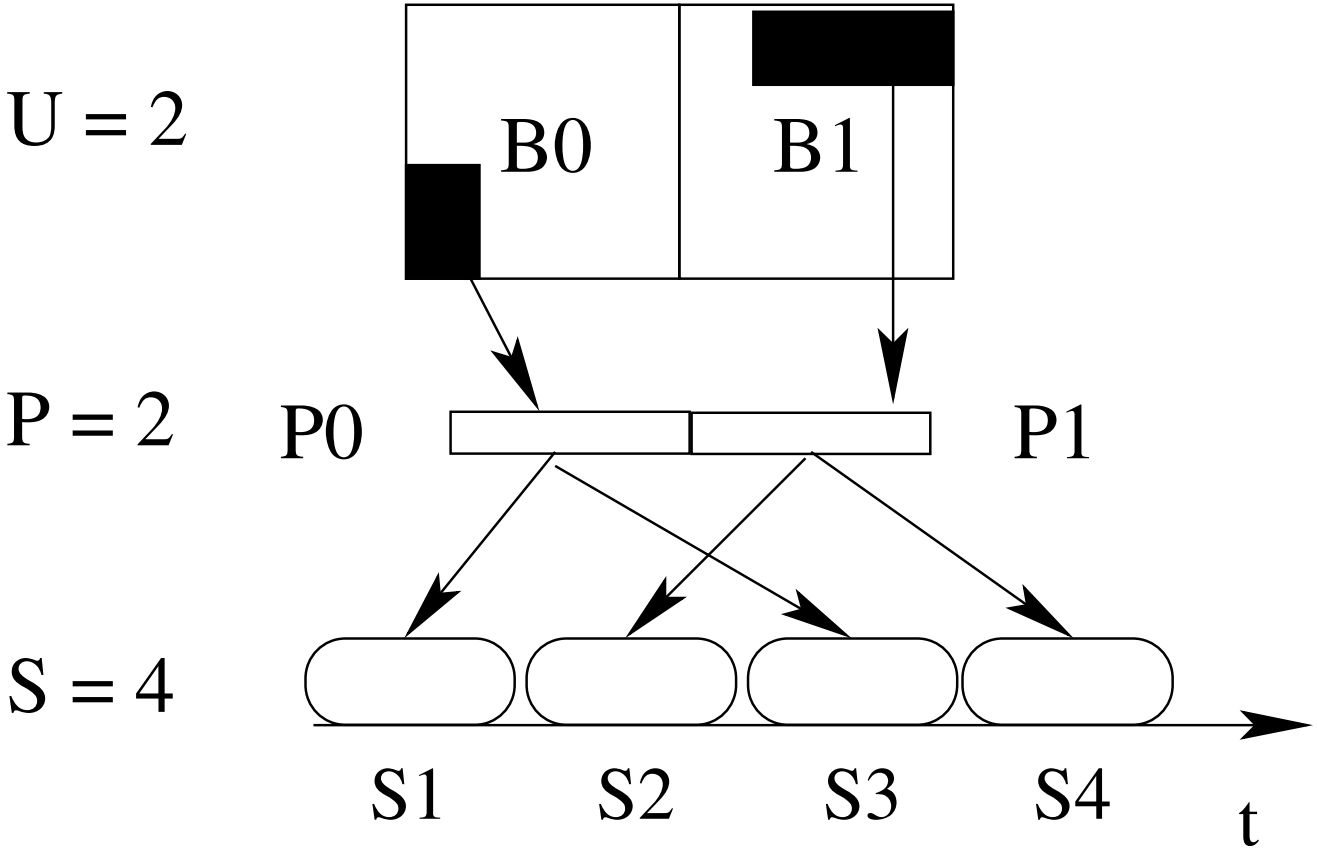
SDRAM memory

- page buffers requires special access patterns



SDRAM memory

- interleaved data access possible from multiple open page buffers



Memory Bandwidth Modeling

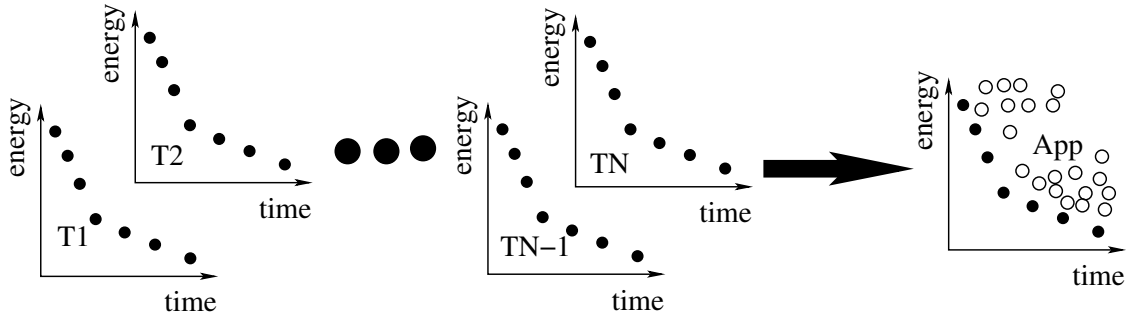
- main bottleneck in current applications
- diffn constraints for page reservation during data access
- cumulative constraints for size constraints
- cumulative for memory bandwidth constraints

Application Execution Flexibility

- each task can execute in fixed number of possible ways
- each execution option indicates optimal operation scheduling within a task
- operation scheduling freedom restricted to few representative choices
- application execution flexibility obtained by task execution flexibility

Time and Energy Pareto Application Diagram

- Hierarchical composition of application task Pareto Diagrams
- Energy consumption is a sum of execution consumption of tasks



- Task energy consumption depends on the option chosen
- Execution time of the application depends on scheduling and task execution option

Conflict Graph Constraints

- Specifies relation between task data which needs to be satisfied
- Memory conflict constraint requires two data to be stored in different memory
- Page conflict constraint requires two data to be stored in such a way that they are accessed from different pages
- Memory compatibility specifies that two data must be stored in the same memory
- Bank compatibility can be used to facilitates page sharing during data access

Application execution optimization

- Find for each task appropriate execution option
- Find best assignment from bandwidth and size point of view
- Find best scheduling for given tasks execution options and data assignment
- Compute constraints for assignment improvement
- Find better assignment and reschedule application

Memory conclusions

- Problem divided into 3 subproblems
- Estimates used for the unknown parameters
- Solution to each step gives possibility to improve results of the previous stage
- Global constraints and tailored search methods

Conclusions

- Heterogeneous problems and specific search methods
- Adaptability
- Global constraints and search methods
- JaCoP - our own constraint solver written in Java