A Constraint Logic Programming Framework for the Synthesis of Fault-Tolerant Schedules for Distributed Embedded Systems

Kåre Harbo Poulsen,

Paul Pop, Viacheslav Izosimov

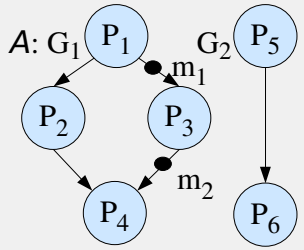
August 23, 2007

Domain

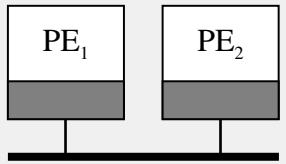
- Safety critical embedded systems
  - Hard real-time
    - Static schedules are necessary
  - Reliability is critical
    - Fault-tolerant
  - Battery powered
    - Energy consumption must be minimised

Example

- Application with
  - Hard deadline
  - Reliability goal: 0.999,999,9
  - Minimise energy



	N_1	N_2
P_1	10	X
P_2	70	X
P_3	X	40
P_4	40	X
P_5	X	40
P_6	X	50

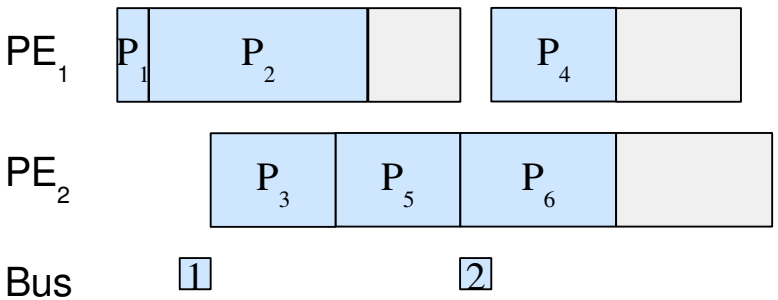


Voltage levels

N_1	100%	66%	33%
N_2	100%	66%	33%

k = 1

Fastest Schedule:



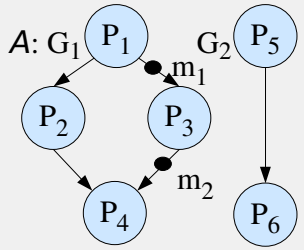
Deadline

R=0.999 999 987

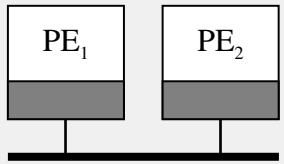
100% E_0

Example

- Application with
  - Hard deadline
  - Reliability goal: 0.999,999,9
  - Minimise energy



	N_1	N_2
P_1	10	X
P_2	70	X
P_3	X	40
P_4	40	X
P_5	X	40
P_6	X	50

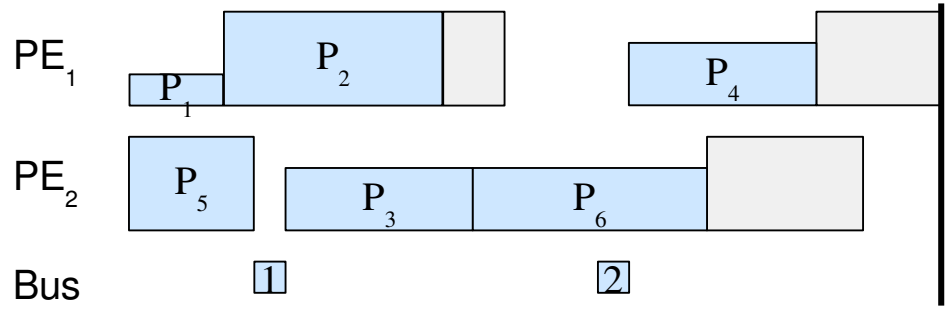


Voltage levels

N_1	100%	66%	33%
N_2	100%	66%	33%

k = 1

Energy Minimised:

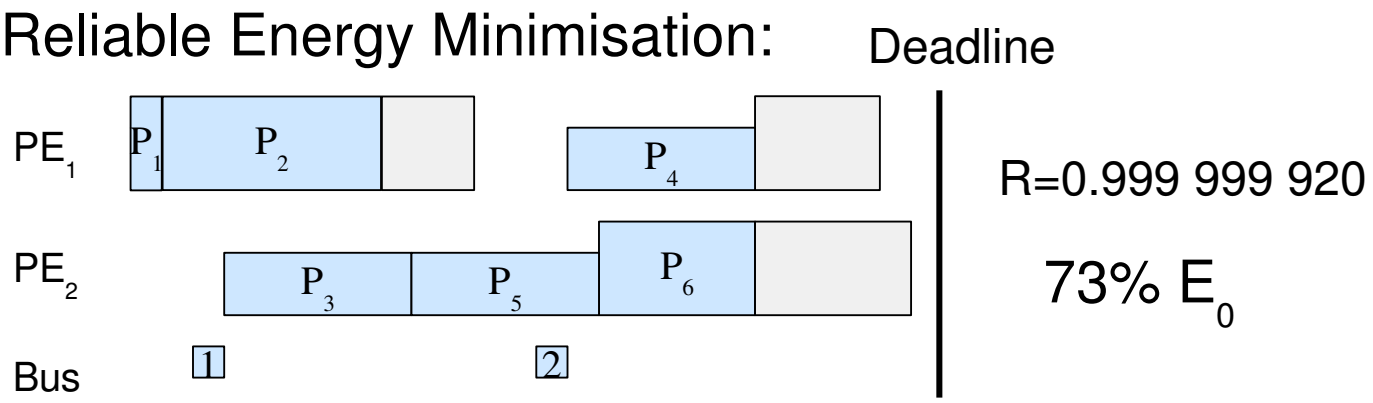


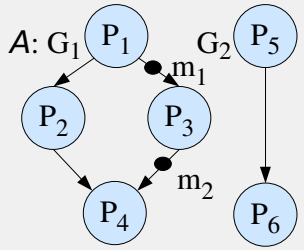
R=0.999 999 878

68% E_0

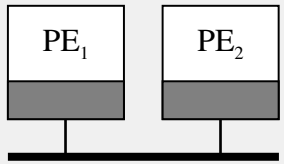
Example

- Application with
  - Hard deadline
  - Reliability goal: 0.999,999,9
  - Minimise energy
    - Sacrifice 5% energy to meet reliability





	N_1	N_2
P_1	10	X
P_2	70	X
P_3	X	40
P_4	40	X
P_5	X	40
P_6	X	50



Voltage levels

N_1	100%	66%	33%
N_2	100%	66%	33%

k = 1

Design Optimisation Tool

- Constraint Logic Programming
  - Good performance for NP-complete problems
  - Easily extendable model
- Heuristics
  - Complete: Variable- and value selection to guide search
  - Incomplete: Credit search to limit search to feasible space/time