FUNLITE -
A PARALLEL PETRI NET SIMULATOR

Jochen Spranger

jsp@informatik.tu-cottbus.de
http://www.informatik.tu-cottbus.de
INTRODUCTION:

- Provably error-free control software for manufacturing systems
- Live 1-bounded hierarchical Place/Transition nets
- Parallel Petri Net simulator to simulate the tokenflow of a Petri Net
- Goals:
  - fast execution speed
  - low memory consumption
  - low communication overhead
- Transputer system (T9000 & C104)
  INMOS C (CSP model of parallel programming)
PRELIMINARIES:

- Simulation of the tokenflow:
  - the control code is assigned to the transitions
  - the execution of a transition is atomic!

- Problems:
  - Parallelity
  - Conflict resolution
THE SEQUENTIAL PETRI NET SIMULATOR

- **Main problem:**
  speed of the transition enabling test

- **The counter method:**
  - one counter for each transition representing the number of unmarked pre-places
  i.e. \( \text{counter}(t) = 0 \Rightarrow t \text{ enabled} \)
  - after the firing of \( t \), we only have to consider \( t, (\bullet t)^* \) and \( (t^\bullet)^* \)

- For each transition \( t' \) in \( (\bullet t)^* \)
  we increase the counter of \( t' \)
  by the number of common pre-places with \( t \)

- For each transition \( t' \) in \( (t^\bullet)^* \)
  we decrease the counter of \( t' \)
  by the number of common places between the pre-places of \( t' \) and the post-places of \( t \)
COMMUNICATION PLACES:

- Post-communication places

- Pre-many-to-1 communication places

- Pre-many-to-many communication places
Locksets:

- simple conflict resolution
- atomar allocation of more than one token

Goals:

Locksets: (disjointed sets of many-to-many communication places)

Definition: A lockset \( l \) is a minimal set of many-to-many communication places such that holds:

If there are transitions \( t,t' \in T \) with \( \bullet t \cap \bullet t' \)
contains many-to-many communication places of \( l \) then the lockset \( l \) contains all many-to-many communication places of \( \bullet t \cup \bullet t' \).
IMPLEMENTATION:

A lockset \( l \) is implemented by:

- For each transition \( t \) with a many-to-many communication place in \( l \) we introduce a counter representing the number of missing communication place tokens.

- For each place \( p \) in \( l \) we generate an input process which waits for an arriving token and updates the counters of the corresponding transitions.

- An administration process which reacts on token requests from subnets.
CONCLUSION:

- fast and simple conflict resolution
- fast transition enabling test
- minimal network traffic
- low memory consumption

suited for small systems