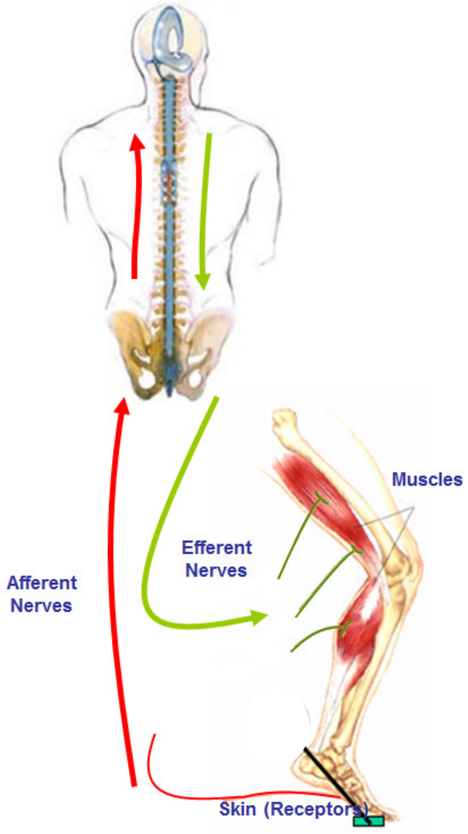
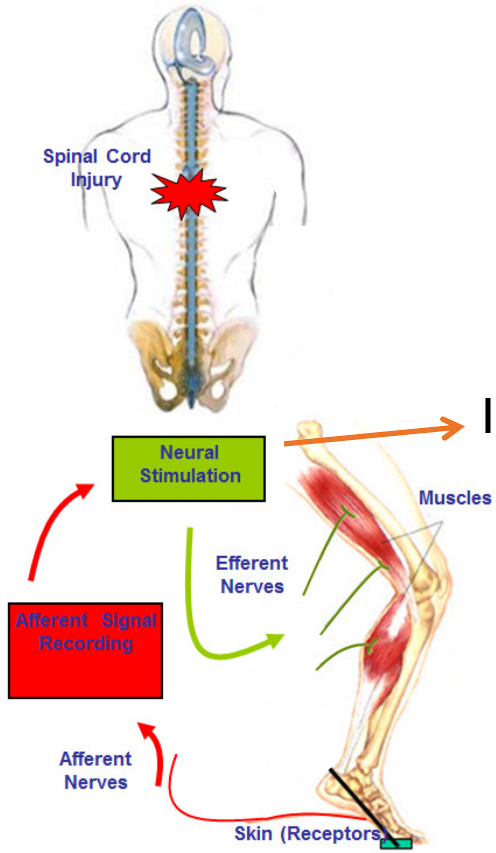
Complex Digital System Design: a methodology and its application to medical implants

Hélène LEROUX, INRIA DEMAR LIRMM
Karen GODARY-DEJEAN, LIRMM
David ANDREU, INRIA DEMAR LIRMM

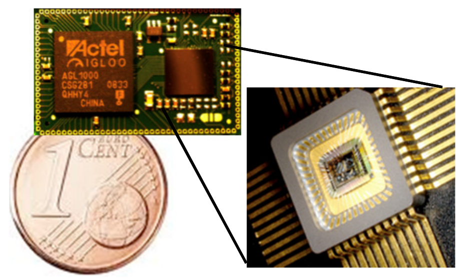
September 2012

Context : The functional electrical stimulation

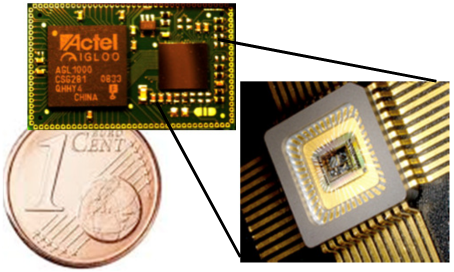




Implantable active medical devices



Context of implantable active medical devices



<u>FPGA :</u>

- flexibility
- reconfigurable
- power and area meet requirements of embedded system

<u>Constraints of the IAMD:</u>

- reliable system
- efficient implementation : limited size, limited power consumption

- convenient for designers

Table of contents

I. The HILECOP methodology

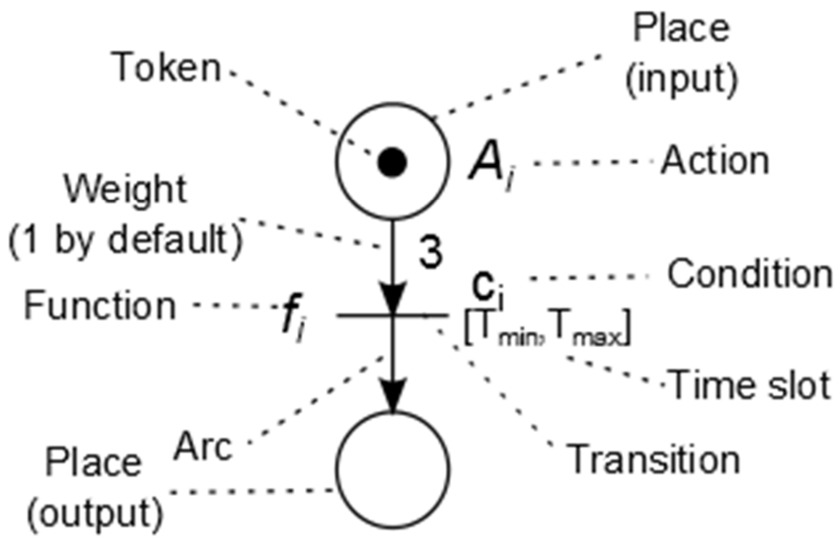
II. Use of analysis in HILECOP methodology

III. Application on an industrial system : a distributed stimulation unit

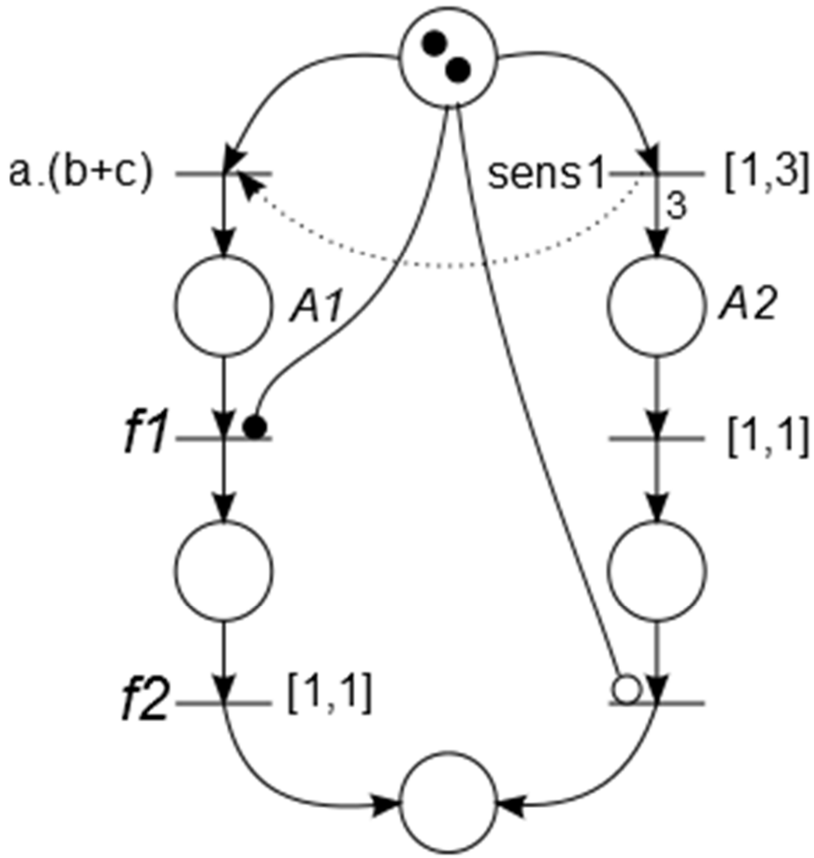
I. The Hilecop methodology

1. Presentation of IPrTPN
2. Hilecop-components
3. Principle of the implementation

1. Interpreted Prioritized Time Petri Nets

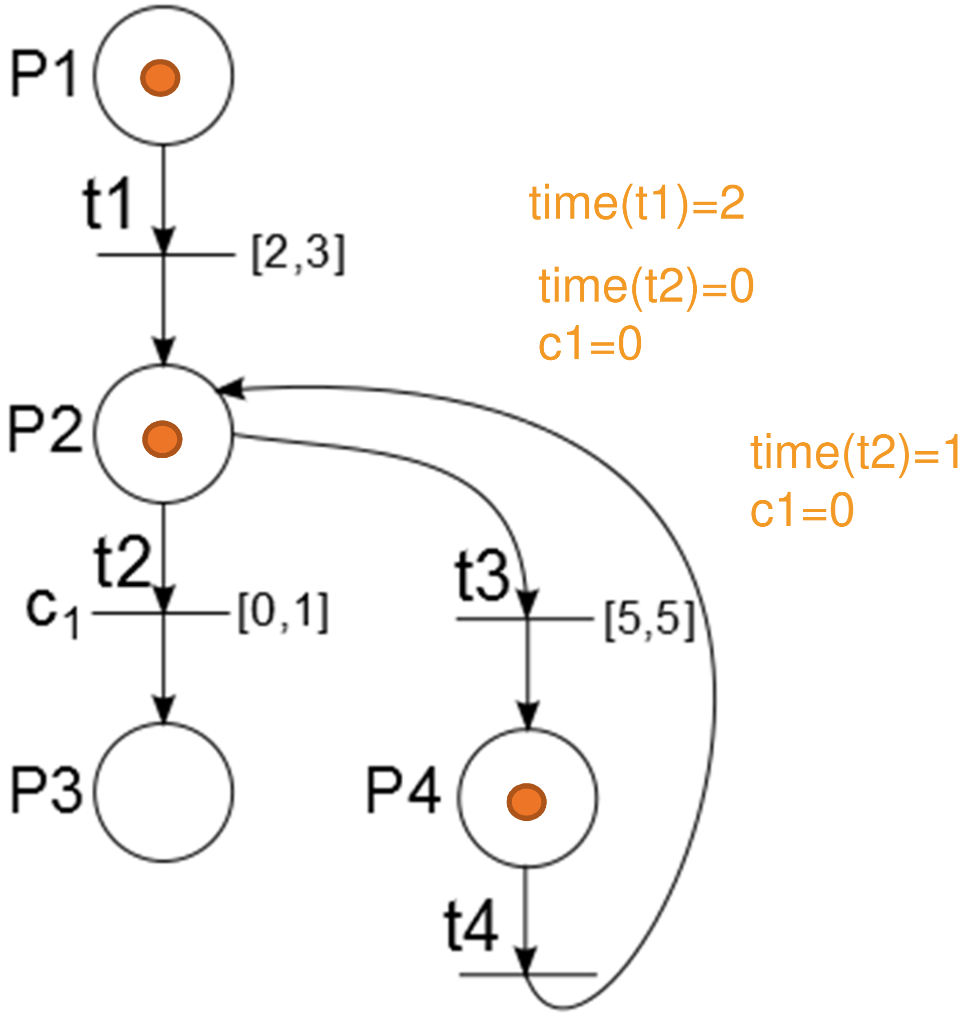


elements of IPrTPN



example of an IPrTPN

Specific semantics of IPrTPN



- State of a IPrTPN= (M,I)
  - M : marking of the IPrTPN
  - I : time slot for every enabled transition
- 3 types of evolution :
  - Discrete transition
  - Continuous transition
  - Blocking transition

M = (1,0,0,0)
I=([0,1])

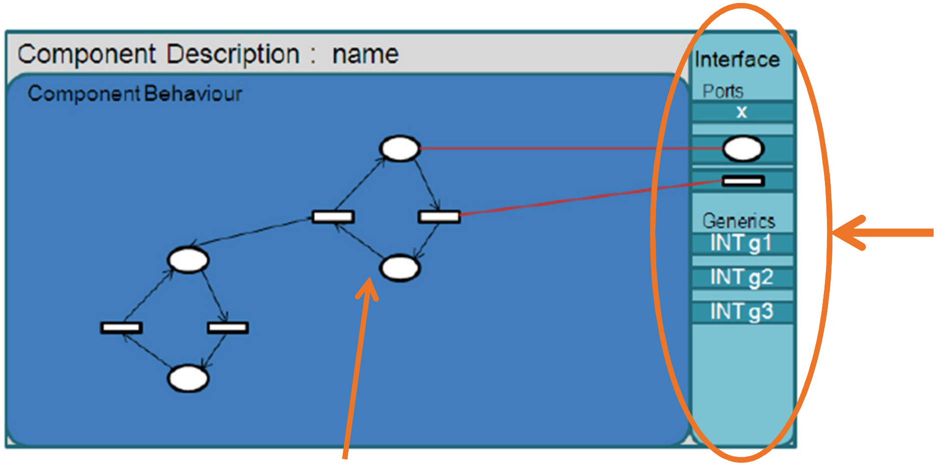
M = (0,1,0,0)
I=([0,1],[5,5])

M = (0,1,0,0)
I=([0,0],[4,4])

M = (0,1,0,0)
I=(Ø,[4,4])

2. Hilecop-component

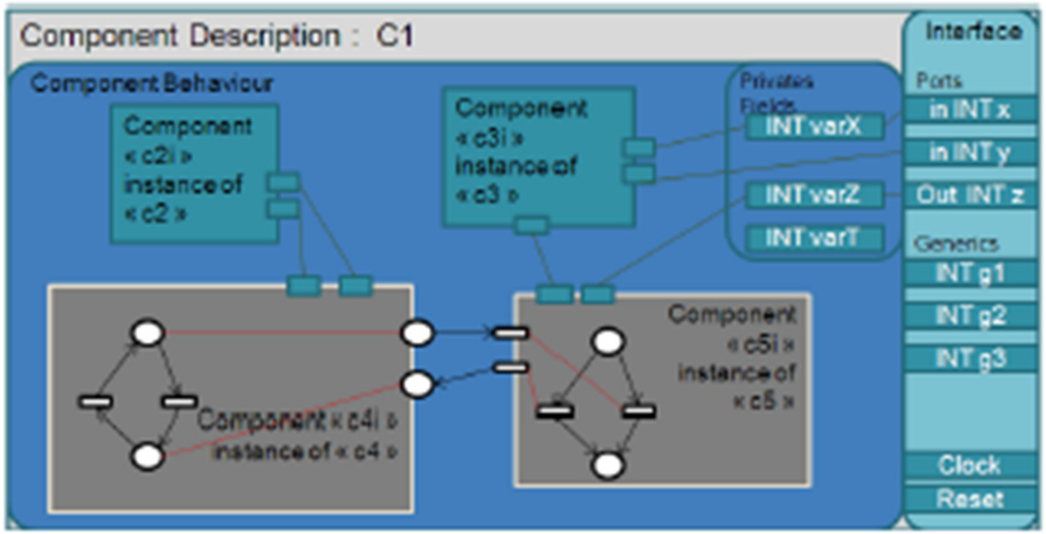
Hilecop-component = component behavior + interface



Ports :
- digital signals
- nodes of the behavior

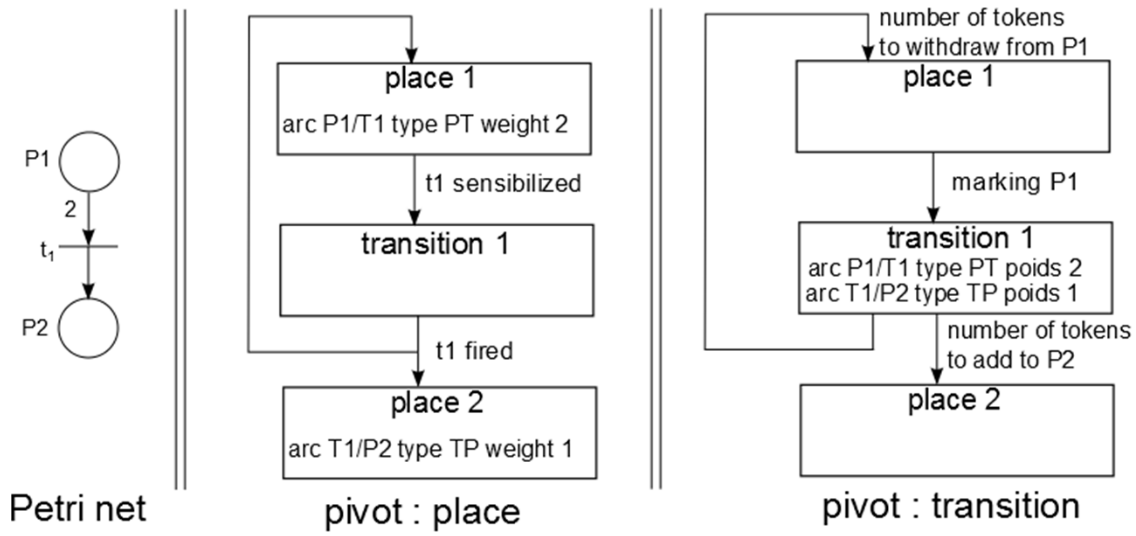
Component behavior : IPrTPN

Composition + aggregation
of components



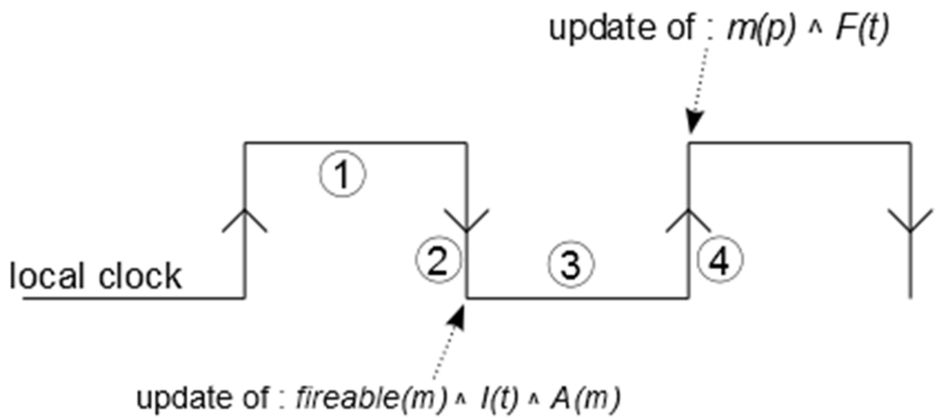
3. Principle of the implementation

Components of the VHDL code and their interconnections:

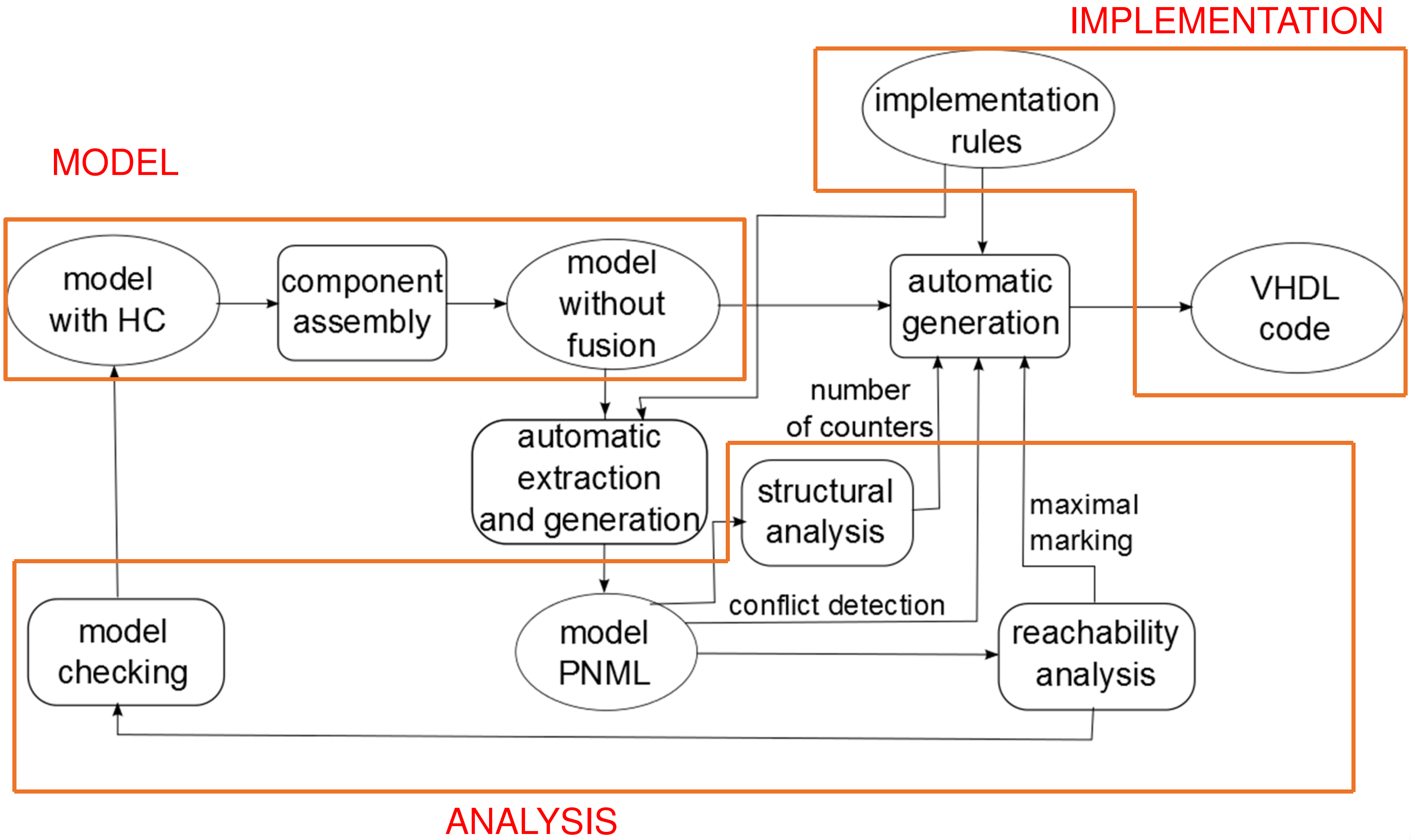


Synchronous execution on FPGA:

1 : Are transitions sensibilized?
2 : firing of transitions
3 : Which transitions have been fired?
4 : Actualization of marking

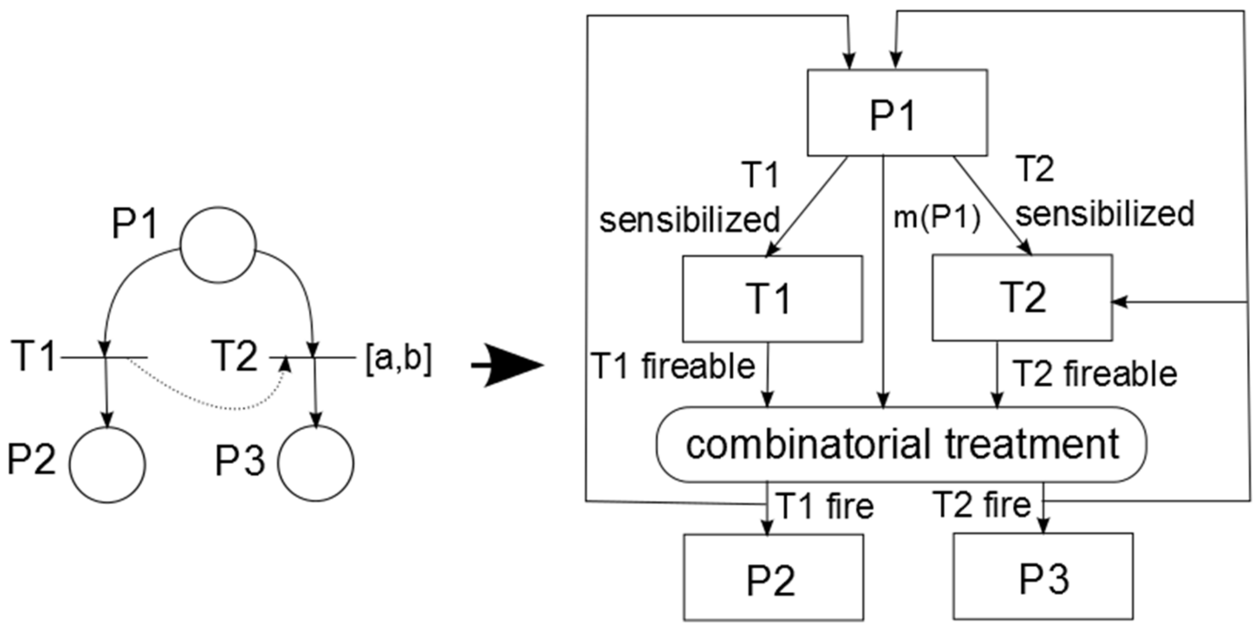


II. Use of analysis in HILECOP methodology



Handling effective conflicts

- Why? : Synchronous implementation of the PN
- How? :
- Automatic detection of all structural conflicts offline
- Automatic treatment of all structural conflicts online to determine if the conflict is effective and which transitions must be fired



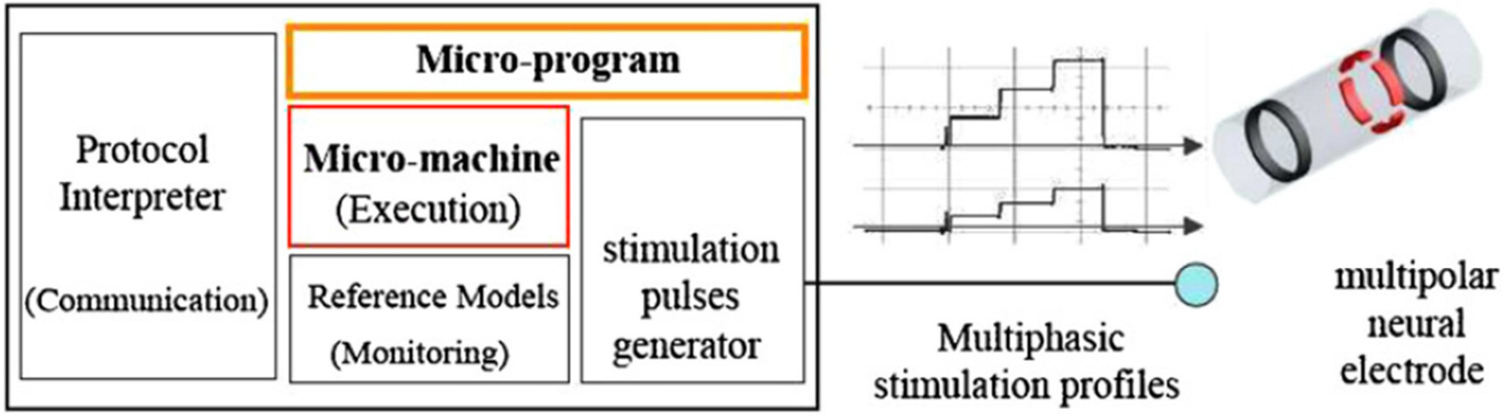
Examples of optimization of the implementation

- Finding the maximal marking of each place :
  - Why? :
  - safety : risk of loss of activity (state or actions) or creation of deadlocks (because of an overrun)
  - efficiency : precise number of bits for each marking

  - How?
  - Reachability analysis : finding the maximal marking for each place in all possible states
  - Structural analysis : use of P-invariant

- Determining the number of necessary counters for temporal transitions:
  - Why? :
    - efficiency : limited number of counters without reliability loss → gain in circuit's size
  - How? :
    - Structural analysis : use of T-invariant to determine all transitions that can share the same counter

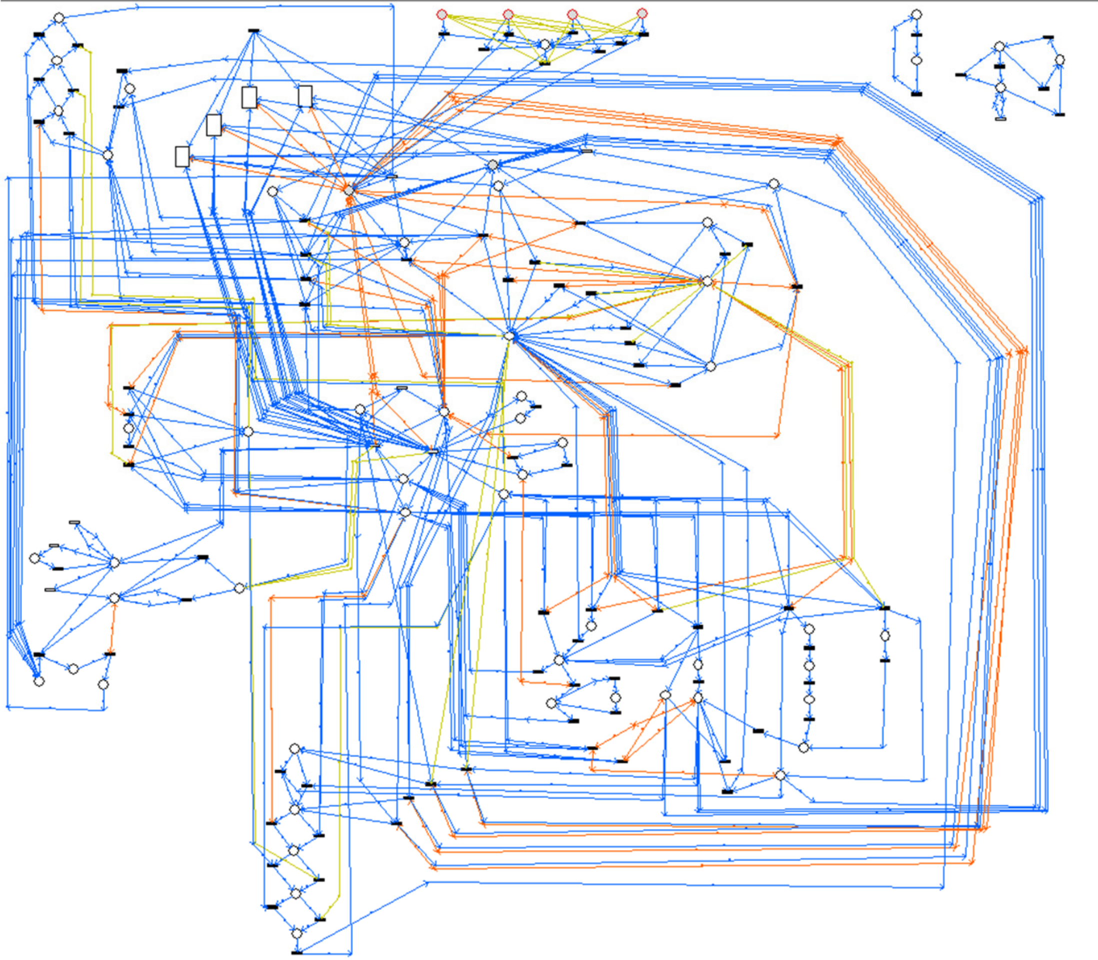
III. Application on an industrial system : A distributed stimulation unit



Micro-machine: small instruction interpreter engine
Micro-program: based on a reduced FES specific instruction set

Complete model of the DSU (1st generation) :
650 places and 770 transitions

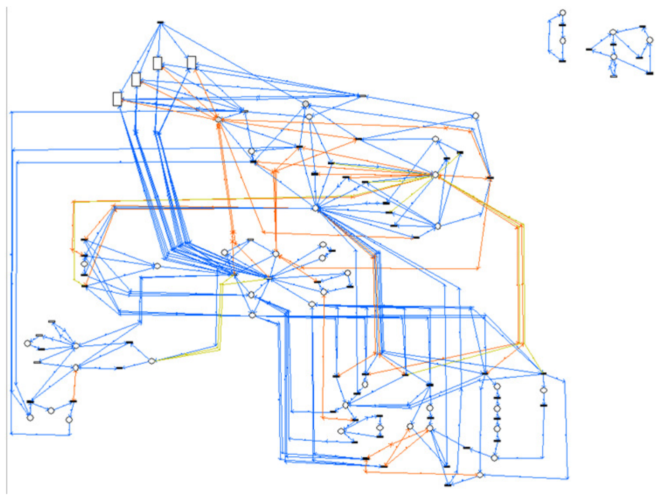
Model of the micro-machine



Behavior model of the micro-machine

61 places
98 transitions
4 Hilecop-components

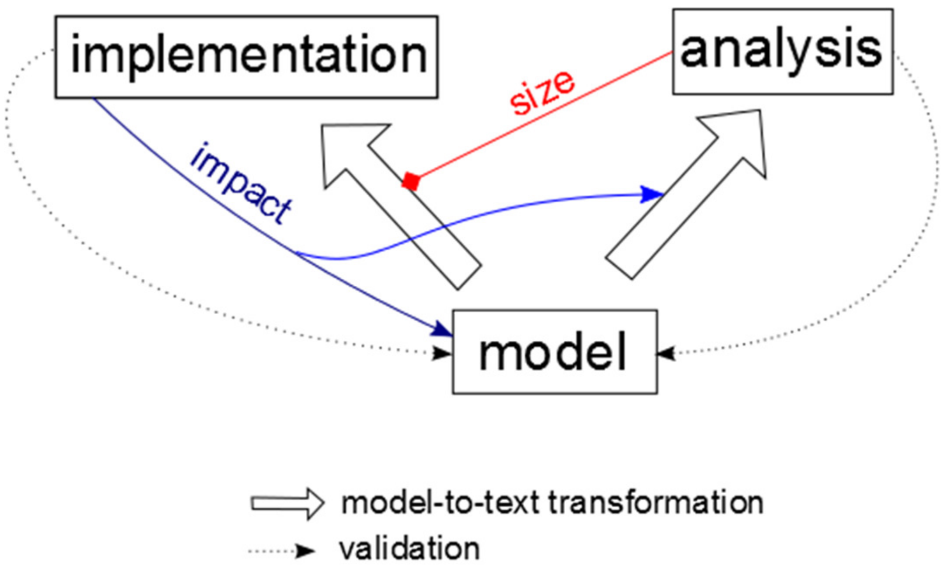
- Priority: 6973 logic blocs without priority / 7035 with priority (+0,9%)
- Maximal marking : PN dealing with the normal behavior is binary



Normal behavior

Conclusion

- Goals of the HILECOP methodology :
  - Correspondence between the model and the implementation
  - Increasing reliability
  - Optimization of the implementation



- Ongoing :
  - Integration of these contributions into the HILECOP tool
  - Validation of a new formalism for exception handling

Thank you for your attention!