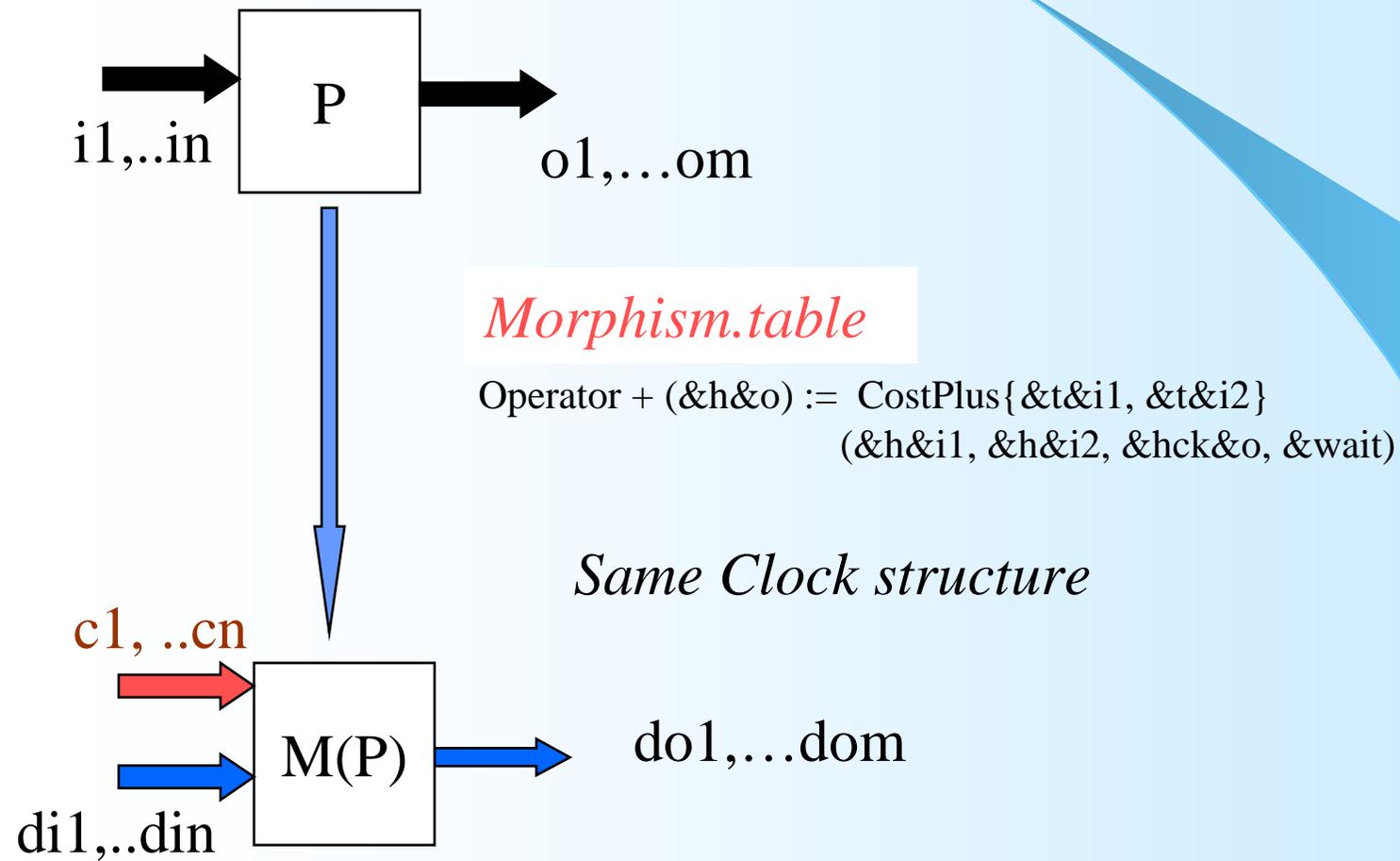


# PERFORMANCE EVALUATION

Loïc Besnard - Thierry Gautier - Paul Le Guernic

- Method
- Experimentations

# METHOD



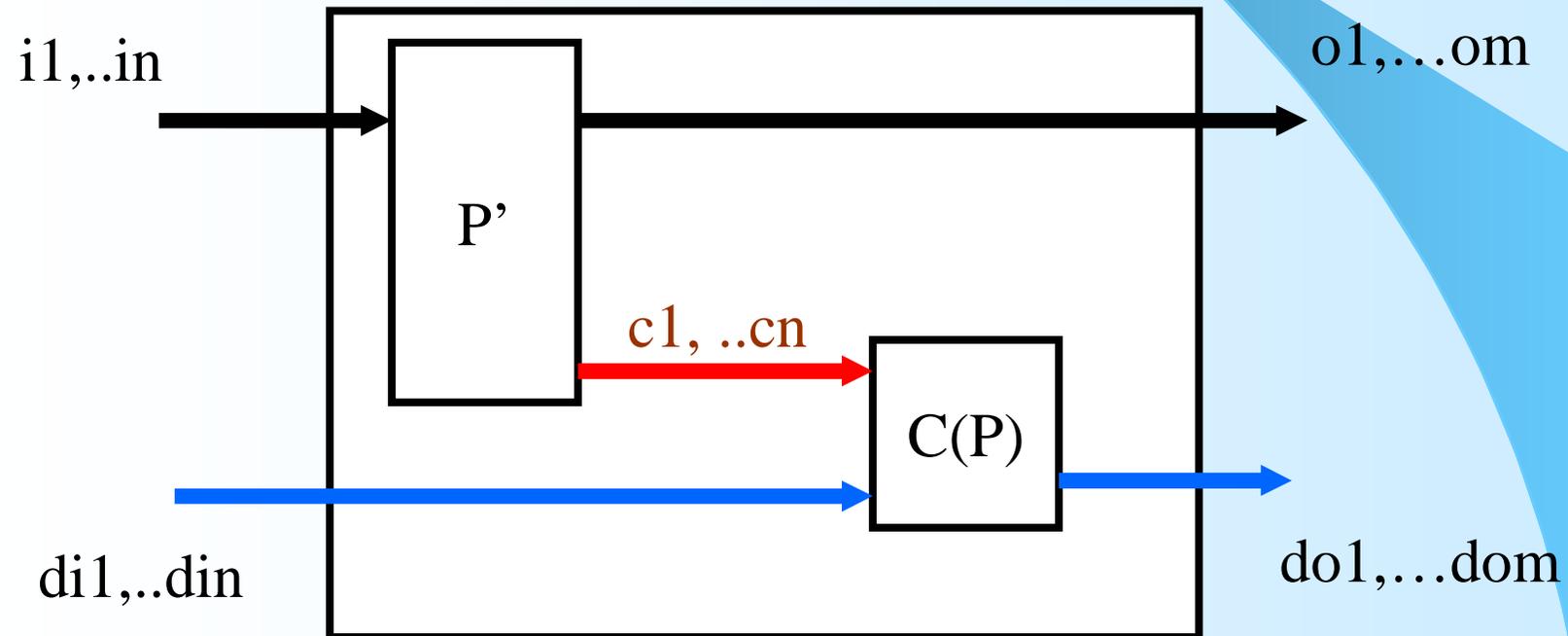
# Method

Morphism.table : can be used for different translations

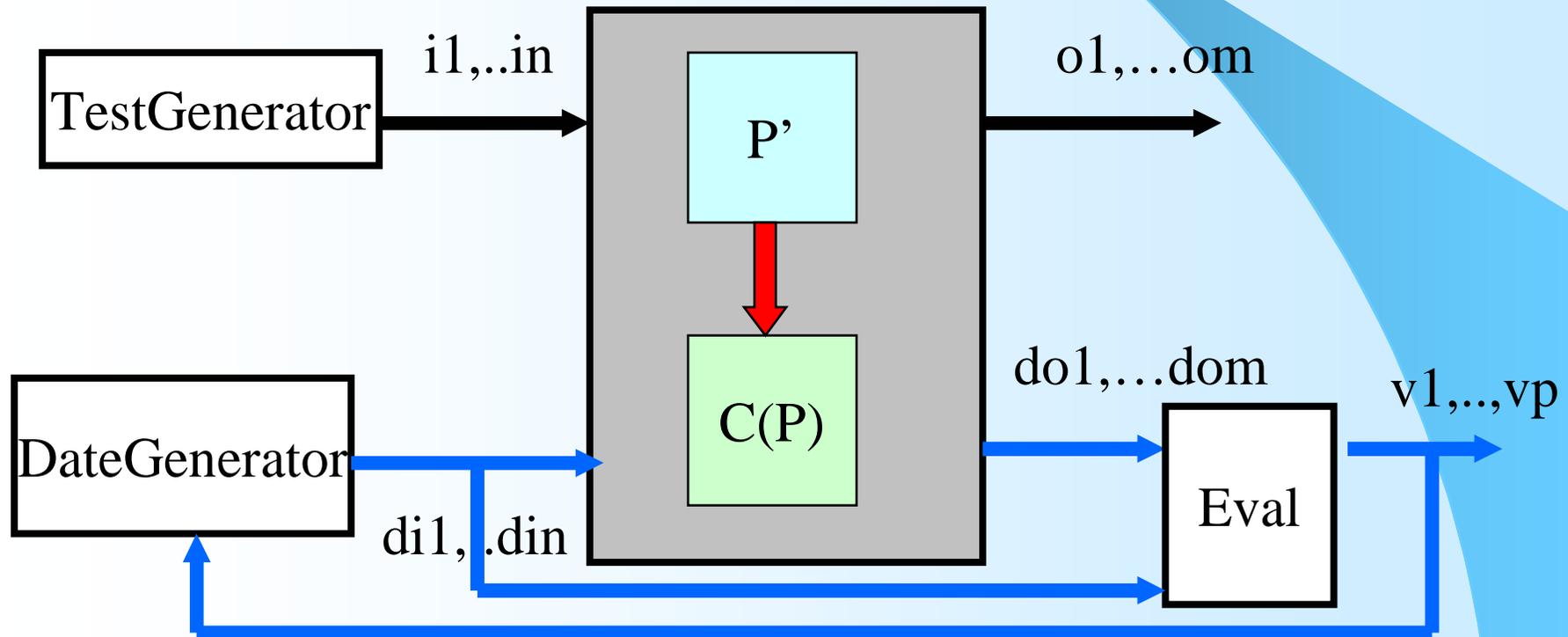
Morphism program needs Library for compiling  
(CostPlus) target specialized

# Experimentations

Co-Simulation



# Experimentations



# Experimentations

Demos

Solving for  $X$   $aX^2 + bX + c = 0$

using iterative Newton method

$$X_0 = \Delta$$

$$n \geq 0, \quad X(n+1) = \frac{X(n) + \Delta}{2X(n)}$$

stop when  $|X(n-1) - X(n)| < \varepsilon$

# Experimentations

1:

$P \dashrightarrow \text{ordering} \rightarrow \text{Morphism}$   
Co-simulation

2:

$P \dashrightarrow \text{Clusters}(P) \rightarrow \text{ordering} \rightarrow \text{Morphism}$   
Co-simulation

# Experimentations

3:

$P = (| Q | R |)$  (mapping)

- \* Global compiling
- \* Partitioning :  $Q'$  and  $R'$  (booleans are added)
- \*  $Q1 = \text{Clusters}(Q'' | C(Q'))$       *separated code gen*
- \*  $R1 = \text{Clusters}(R'' | C(R'))$       *separated code gen*

Then

$(| \text{Abs} ( Q1) | \text{Abs} ( R1) |)$

# Experimentations

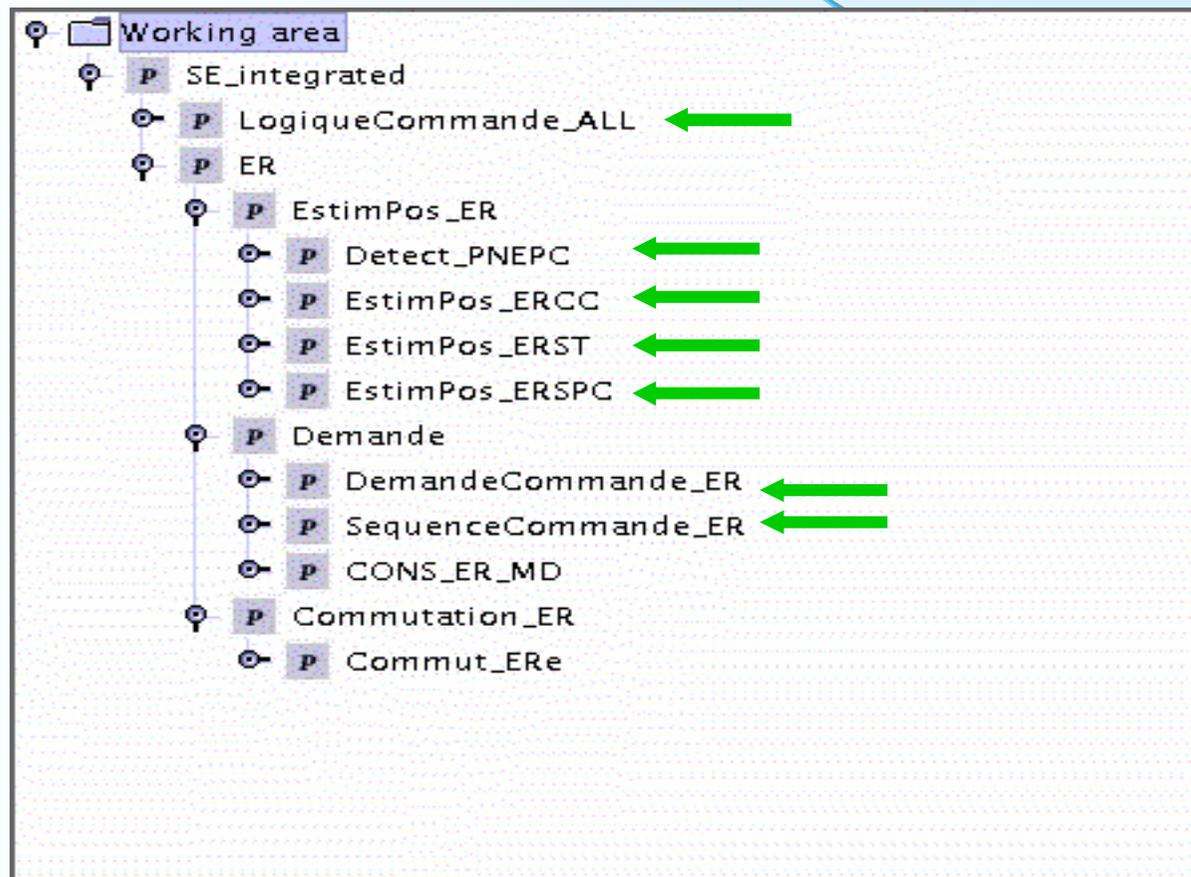
$P = (| Q | R |)$

Global compiling

Partitioning :  $Q'$  and  $R'$  (booleans are added)

$(| \text{step} :: (| \text{Abs}(Q') | \text{Abs}(R') |)$   
 $| \text{latency} :: (| C(Q') | C(R') |)$   
 $| \text{step} \text{ ---> } \text{latency}$   
 $|)$

# Experimentations: Hugues' example



# Conclusion

- Experimentation of a performance evaluation technique
- Efforts:
  - compiler optimizations
  - validating
- Polychrony : set of functionalities