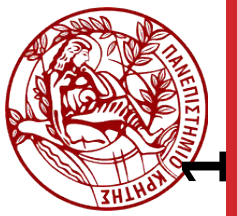
LOGIC SYNTHESIS OF CONCURRENT CONTROLLER SPECIFICATIONS

PAVLOS M. MATTHEAKIS

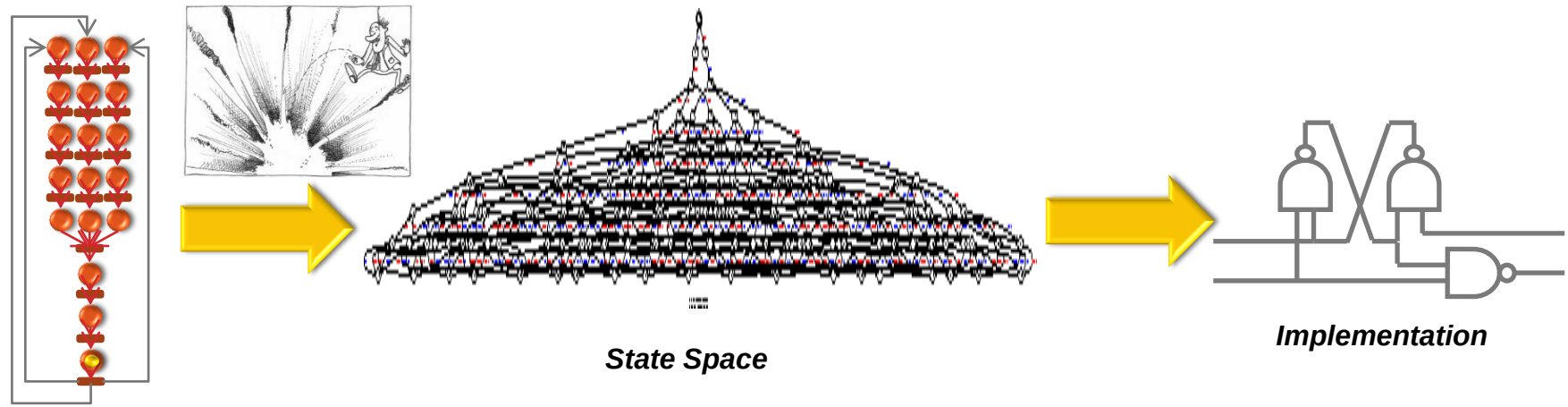
SUPERVISOR: CHRISTOS P. SOTIRIOU



OVERVIEW

- ***Motivation***
- Background
- MSFSMs
- Synthesis
- Verification
- Optimization
- Results
- Conclusions and Future Work

STATE EXPLOSION



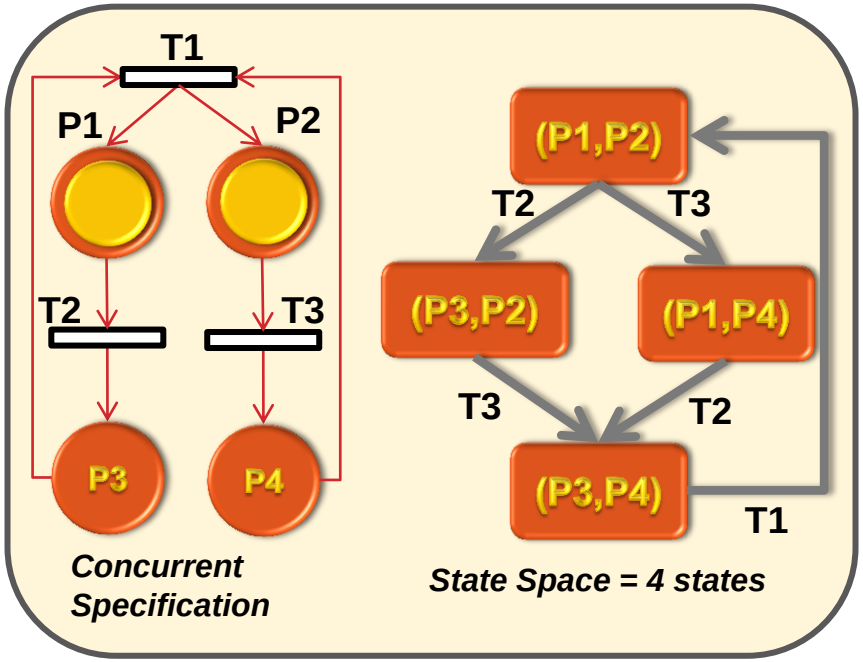
Concurrent Specification

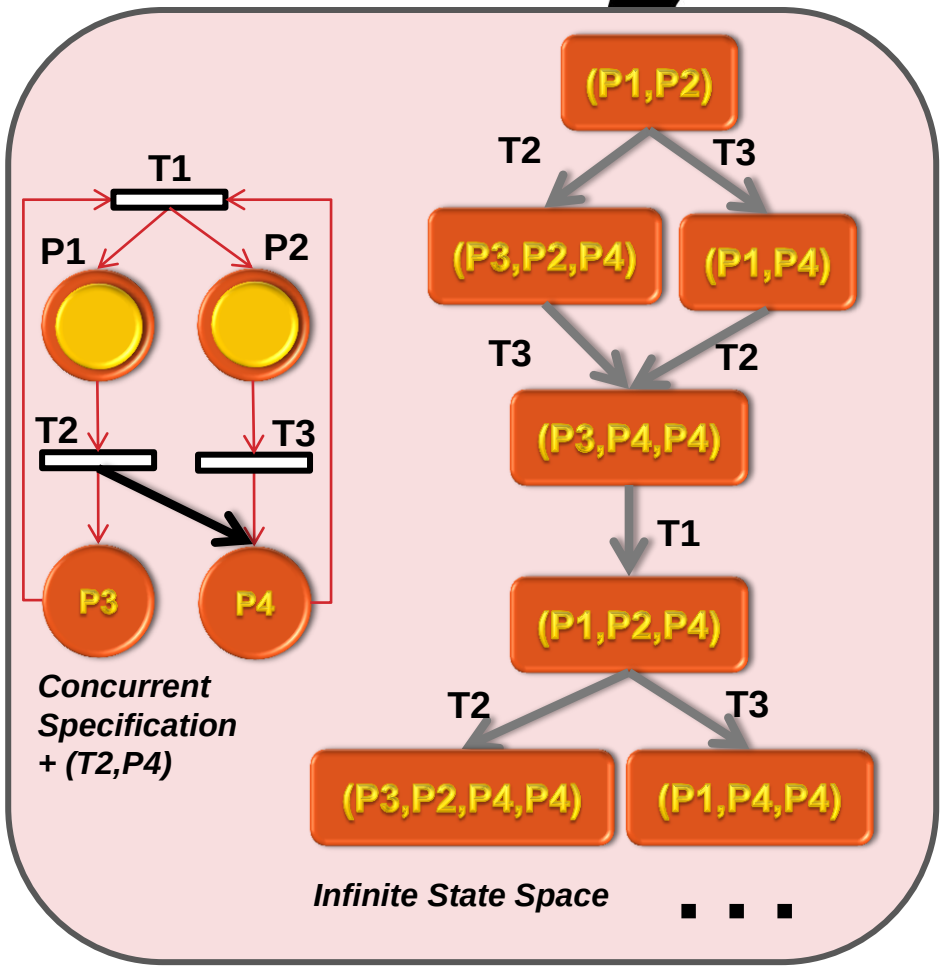
State Space

Implementation

- *Logic Synthesis requires specification's complete state space*
  - *State space's size is exponential compared to the specification*

SPECIFICATION AND STATE CONFLUENCE



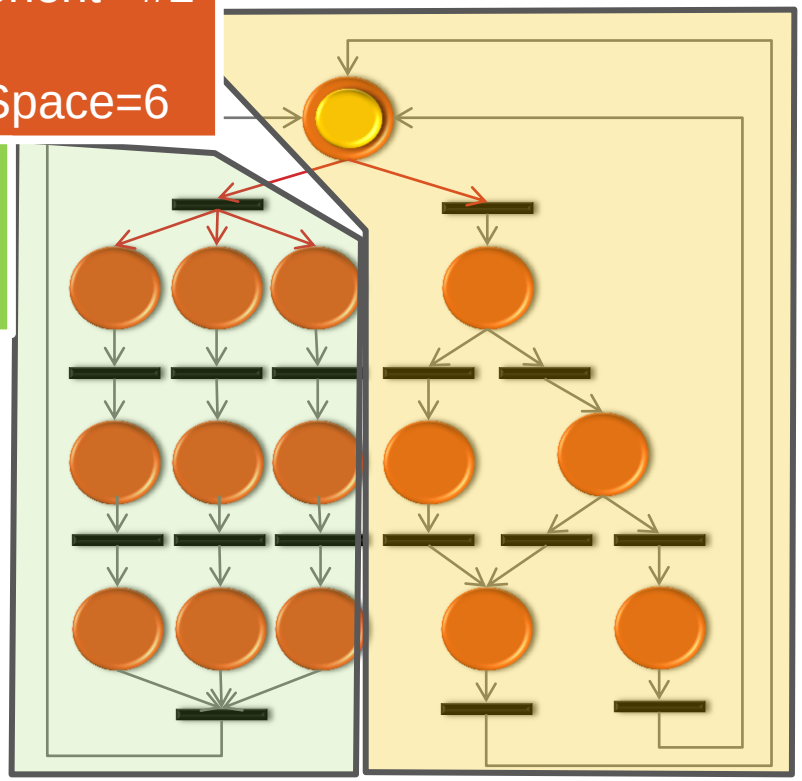


- *Changes at the Concurrent Specification Level Have Unpredictable Results at the State Space*

DECOMPOSABILITY

Component #2
P=6
State Space=6

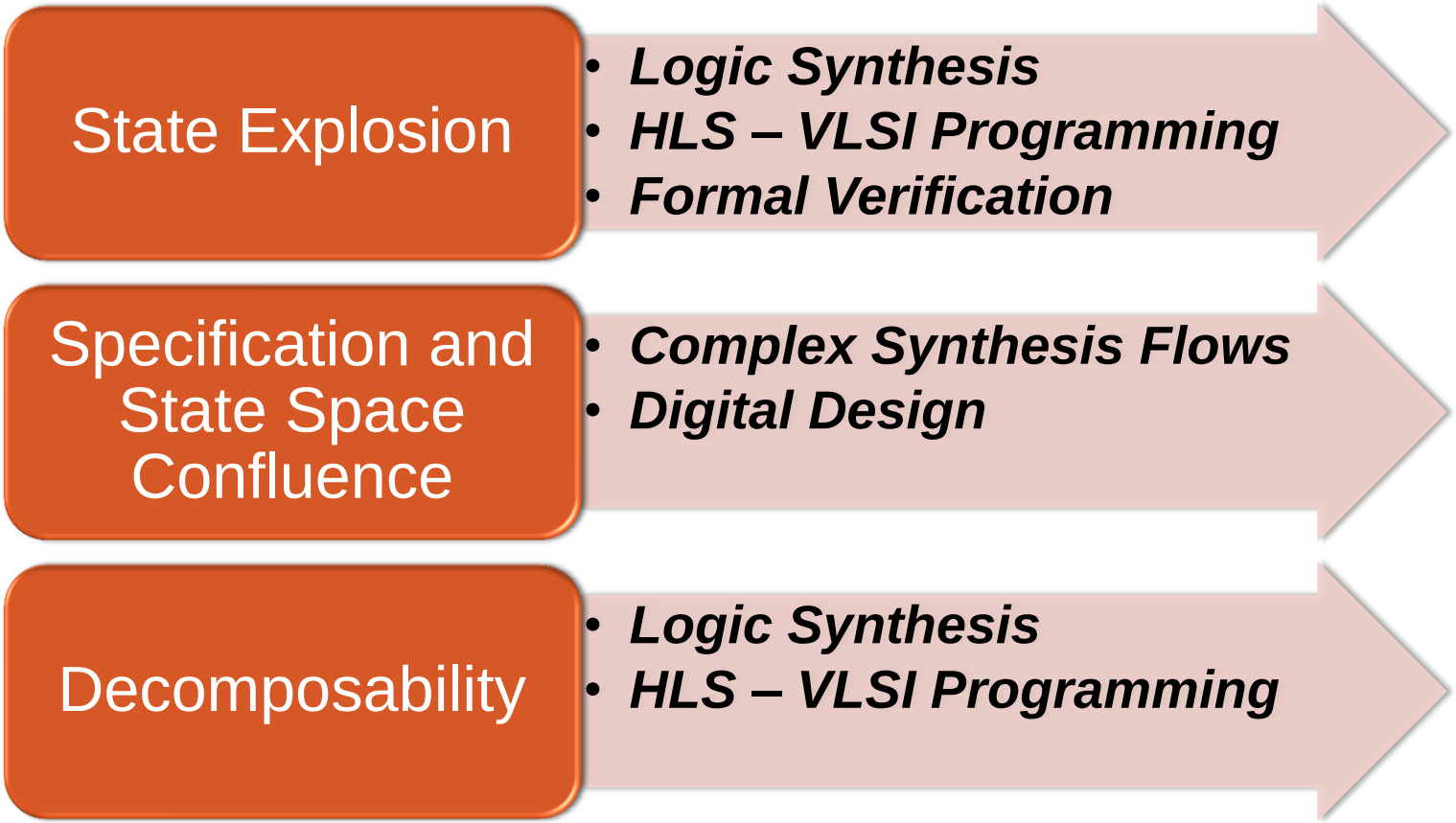
Component #1
P=9
State Space=27



Complete Specification
P=15
State Space=33

- ***Concurrent Specification Decomposition is Unpredictable as it is Evaluated with Specification Level Metrics, e.g. number of places.***

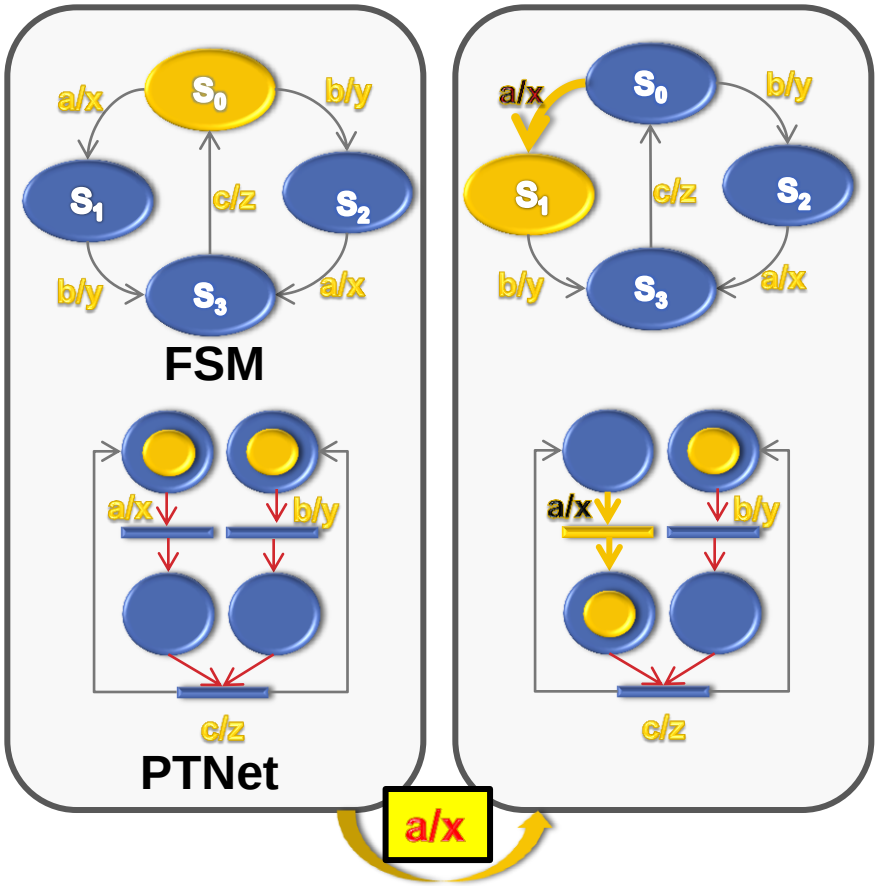
IMPORTANCE


State Explosion
• Logic Synthesis
• HLS – VLSI Programming
• Formal Verification
Specification and State Space Confluence
• Complex Synthesis Flows
• Digital Design
Decomposability
• Logic Synthesis
• HLS – VLSI Programming

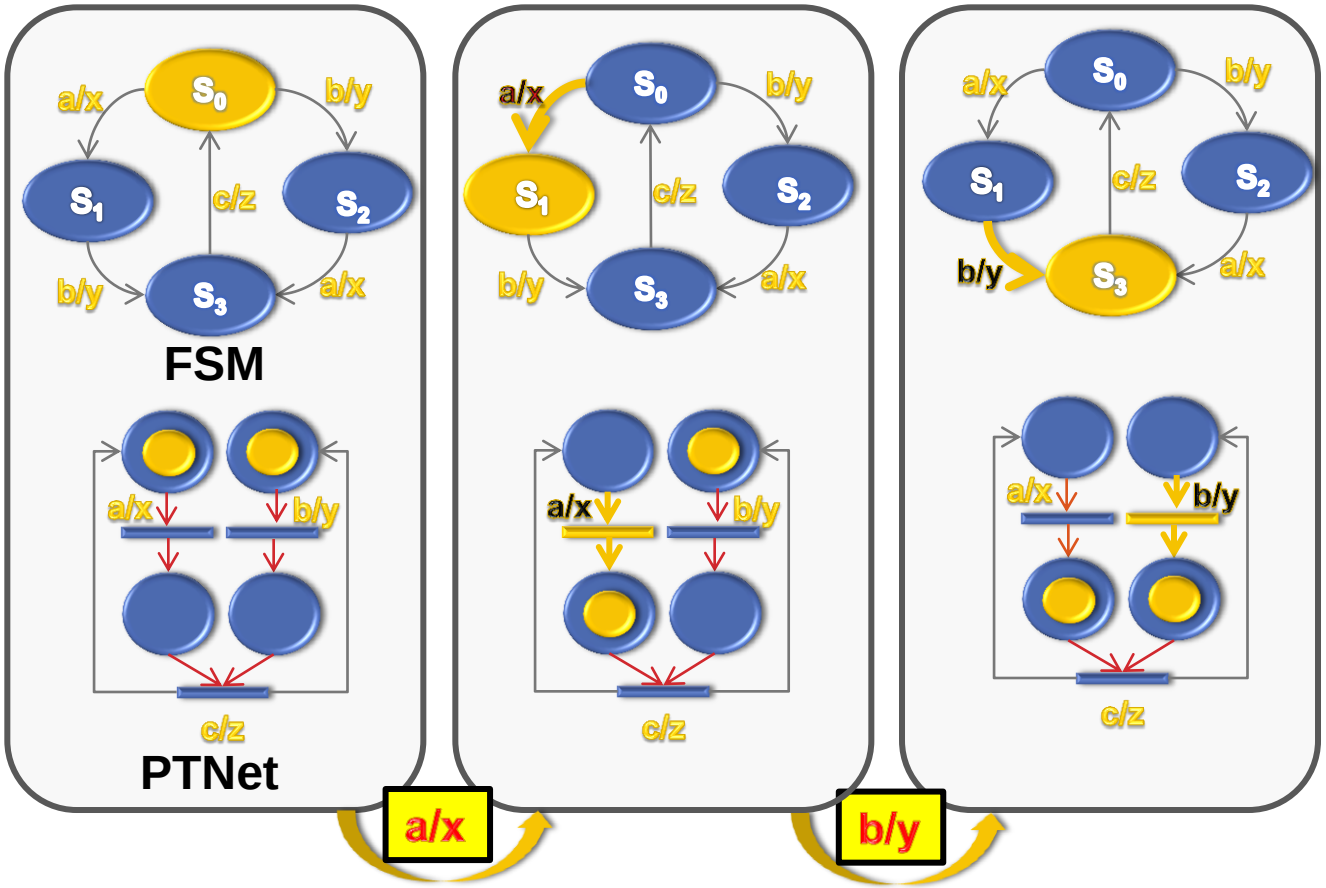
OVERVIEW

- Motivation
- ***Background***
- MSFSMs
- Synthesis
- Verification
- Optimization
- Results
- Conclusions and Future Work

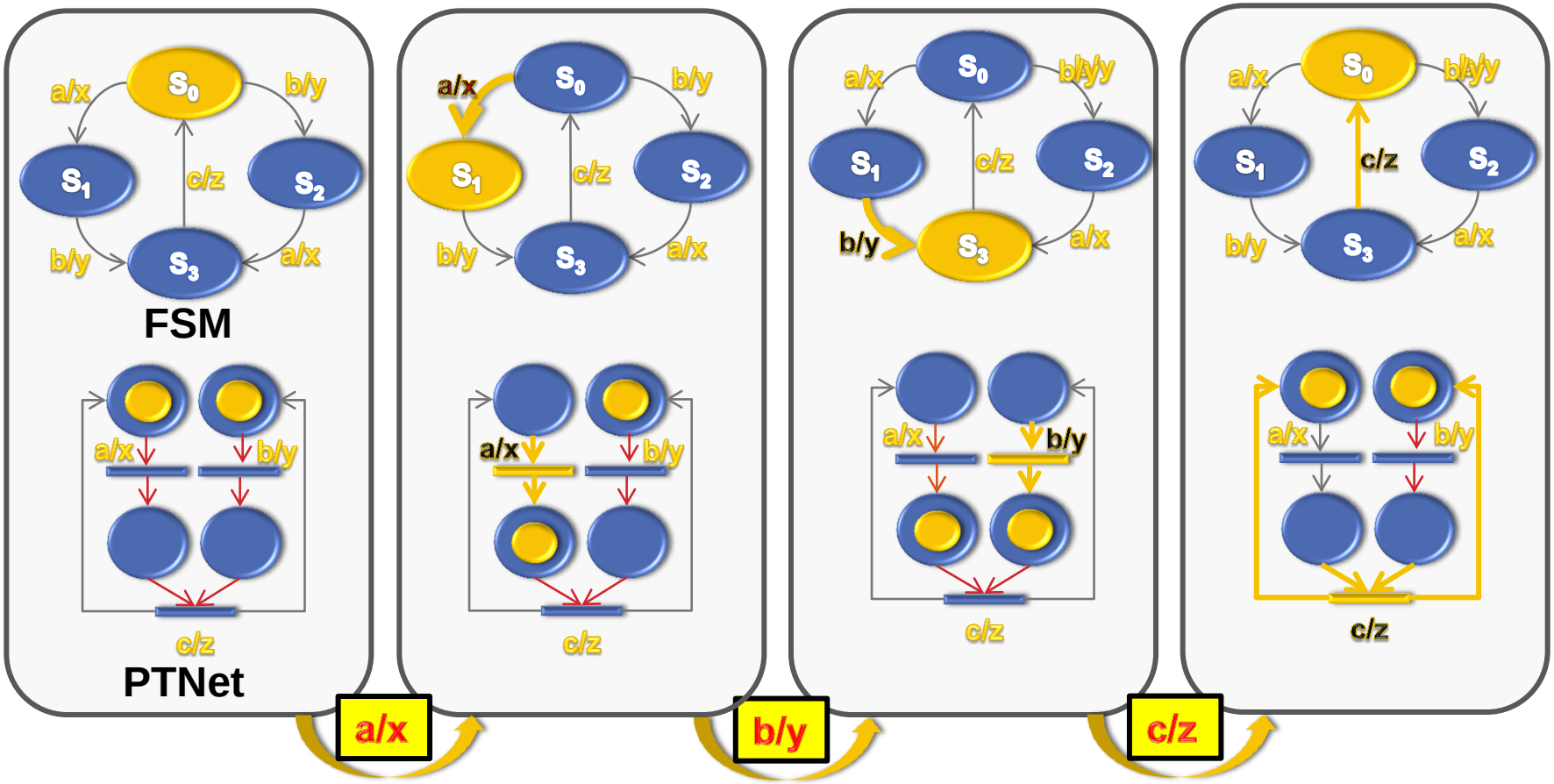
PTNET OPERATION



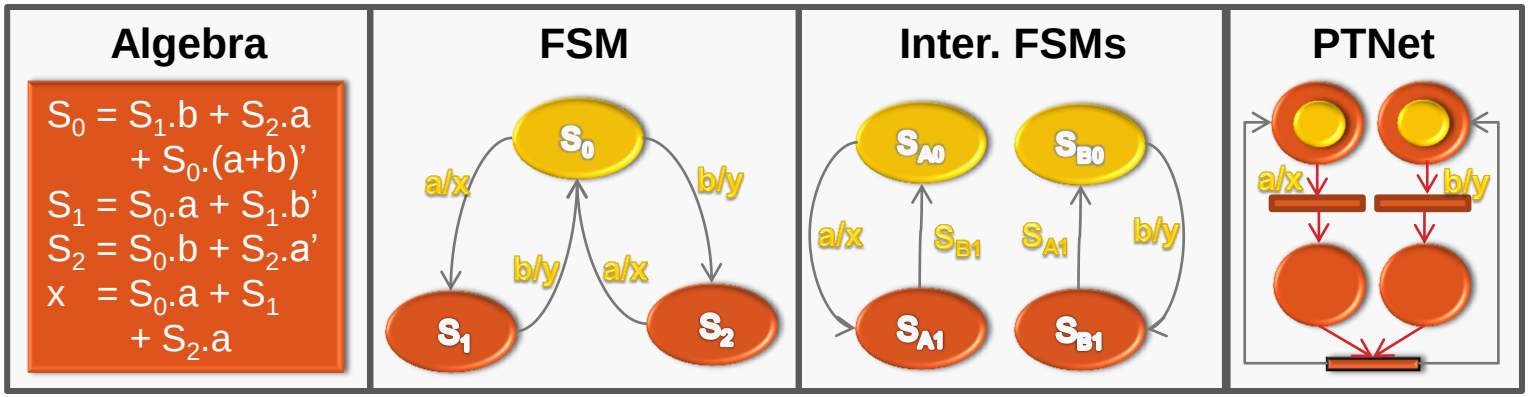
PTNET OPERATION



PTNET OPERATION

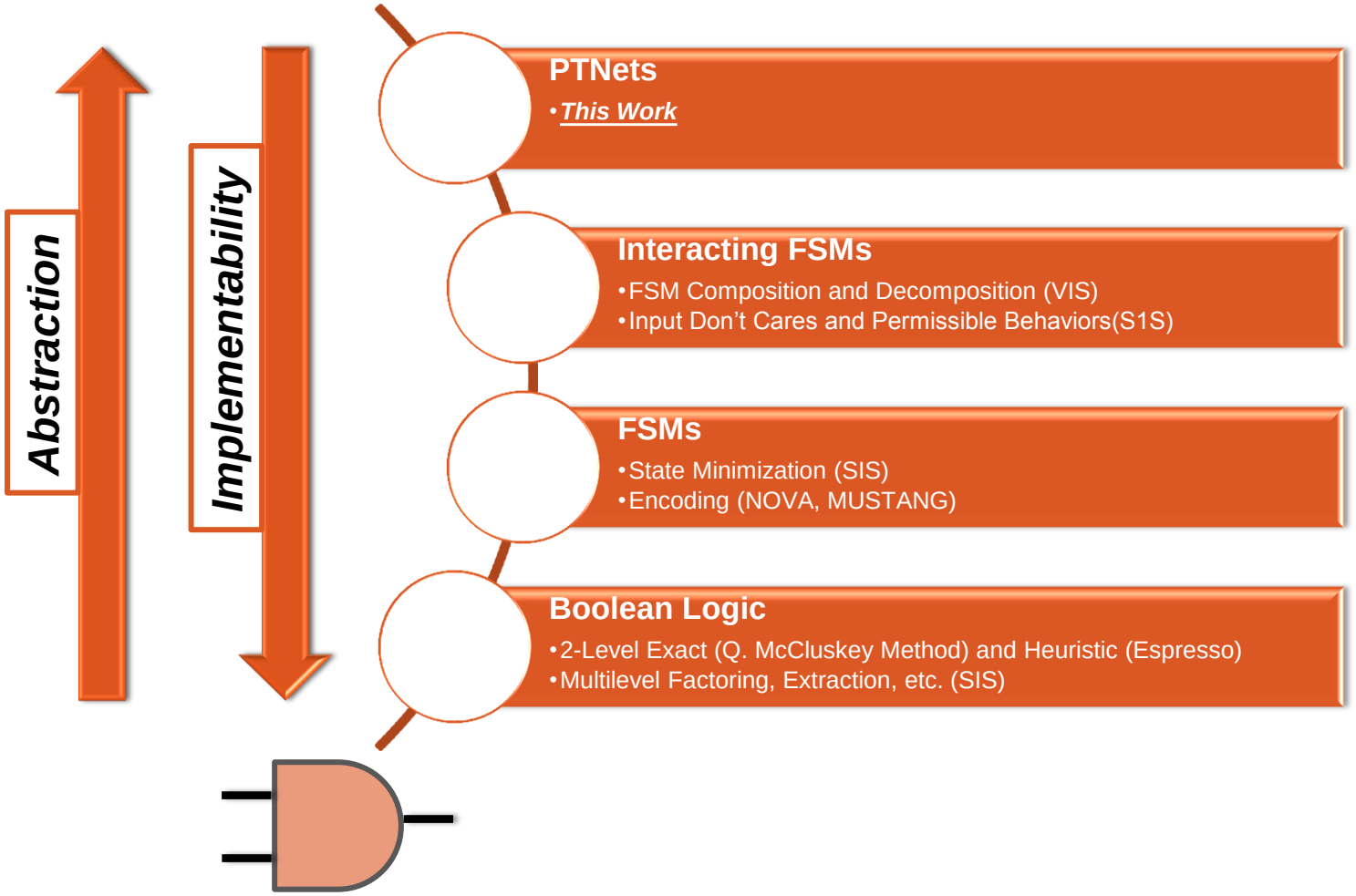


CONTROL MODELS EXPRESSABILITY



Control Model	State Space	Choice	Concurrency	Synchronization
Algebra	Implicit	Implicit	Implicit	Implicit
FSM	Explicit	Explicit	Implicit	Implicit
Inter. FSMs	Implicit	Explicit	Explicit	Implicit
PTNet	Implicit	Explicit	Explicit	Explicit

CONTROL MODELS IMPLEMENTABILITY TO CLOCKED LOGIC



CONTROL MODELS IMPLEMENTABILITY TO SELF-TIMED LOGIC

(A,BM)FSM Synthesis[1]	PTNet Synthesis[2]	Decomposition[4]	Structural Synthesis[5]	Direct Mapping[3]
• Conventional FSM Heuristics • Exponential Concurrent Specifications	• Compact Concurrent Specifications • Exponential Synthesis Time	• Lower Synthesis Time • Infeasible Implementation • Worst Case Exponential	• Lower Synthesis Time • Worst Case Exponential	• Linear Synthesis Time • Suboptimal Results

[1]K. Y. Yun and D. L. Dill, *"Automatic synthesis of extended burst-mode circuits"*, IEEE TCAD, 1999

[2]J. Cortadella et al., *"Logic Synthesis of Asynchronous Controllers and Interfaces"*, Springer-Verlag, 2002.

[3]D. Sokolov et al., *"Direct Mapping of Low-Latency Asynchronous Controllers from STGs"*, IEEE TCAD, 2007

[4]D. Wist et al., *"Signal transition graph decomposition: internal communication for speed independent circuit implementation"*, IET Computers & Digital Techniques, 2011

[5]E. Pastor et al. *"Structural Methods for the Synthesis of Speed-Independent Circuits"*, IEEE TCAD, 1998

OVERVIEW

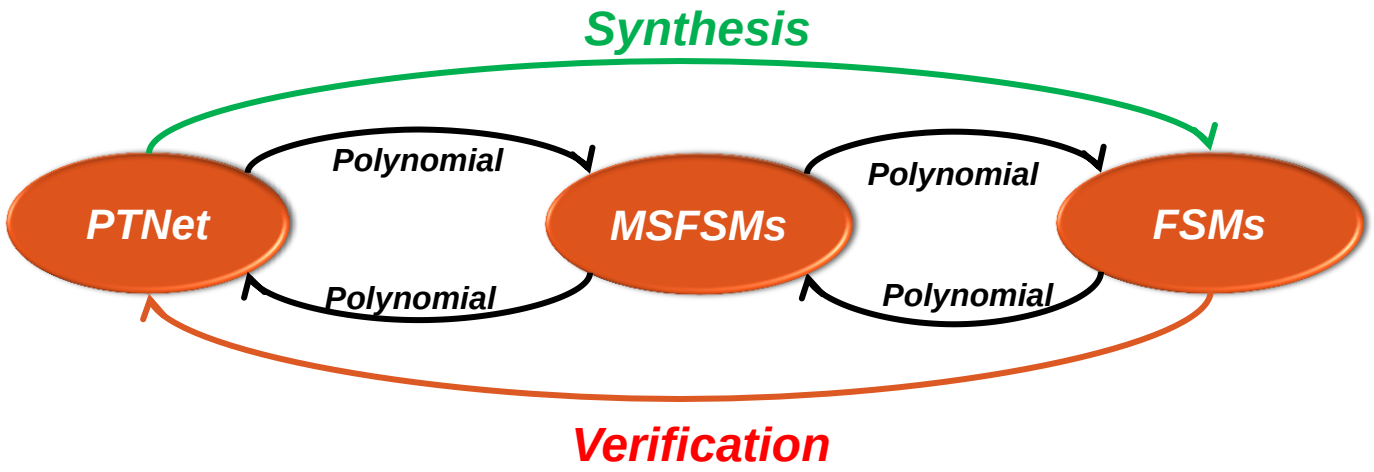
- Motivation
- Background
- ***MSFSMs***
- Synthesis
- Verification
- Optimization
- Results
- Conclusions and Future Work

FORTH
Institute of Computer Science

MULTIPLE SYNCHRONIZED FSMS

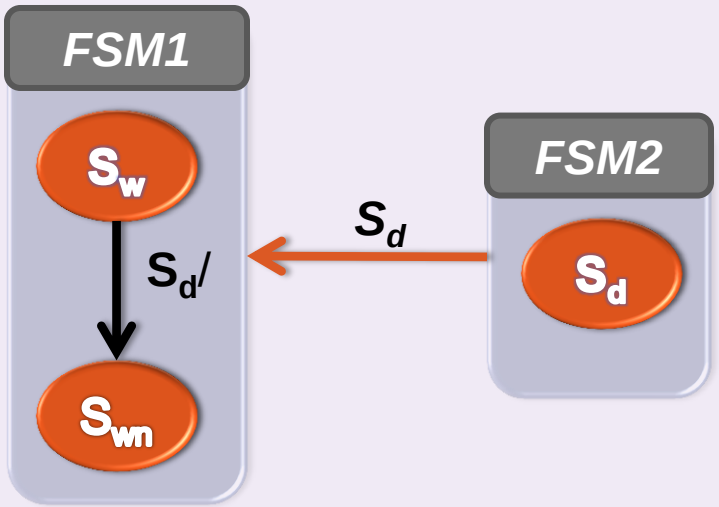
Novel Model for Concurrent Control Specifications

- Parallel Tasks Described with FSMs
- Synchronization explicitly described
    - Transition Barriers, Wait States
- Polynomial Synthesis and Verification Paths



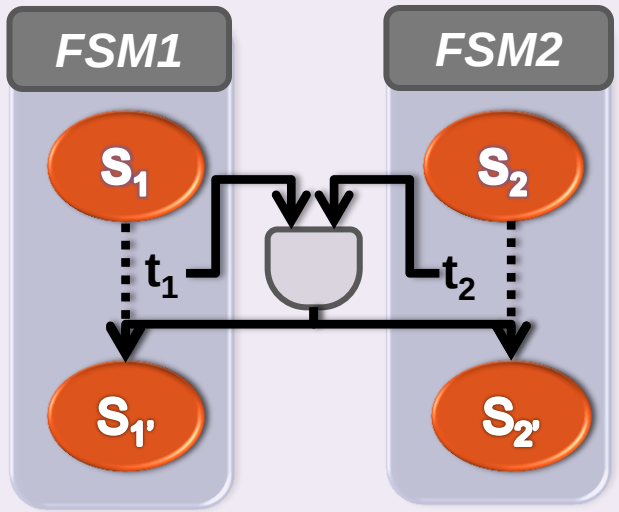
MSFSMS SYNCHRONIZATION PRIMITIVES

Wait State



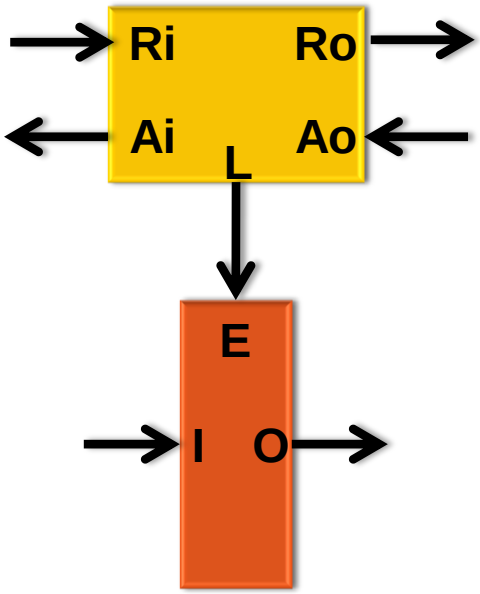
- *FSM1 moves from S_w to S_{wn} if FSM2 is at S_d*

Transition Barrier

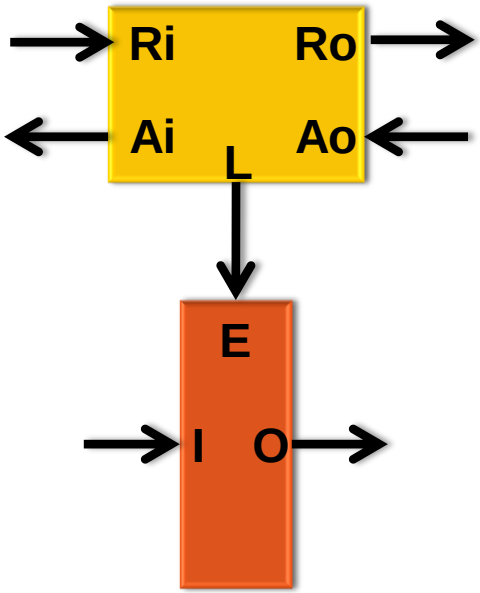


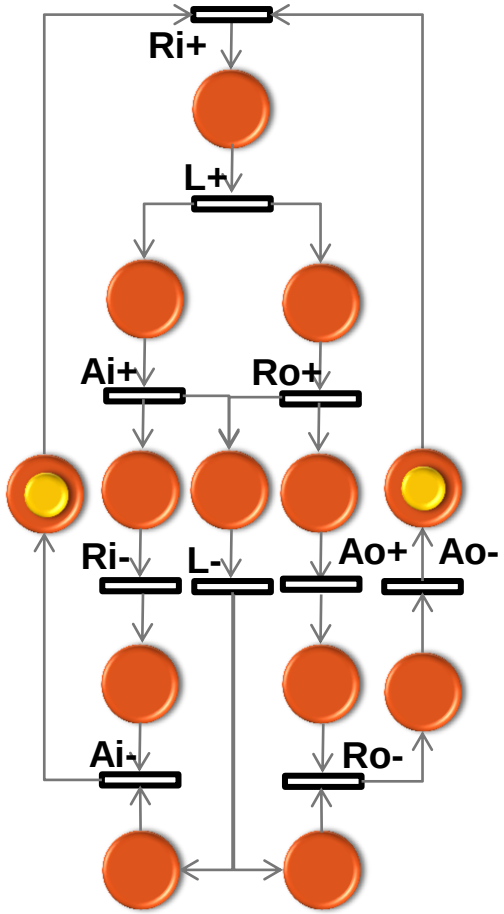
- *FSM1 and FSM2 move to their next states when both t_1 and t_2 are activated*

MSFSMS EXAMPLE: 4-PHASE LATCH CTRL



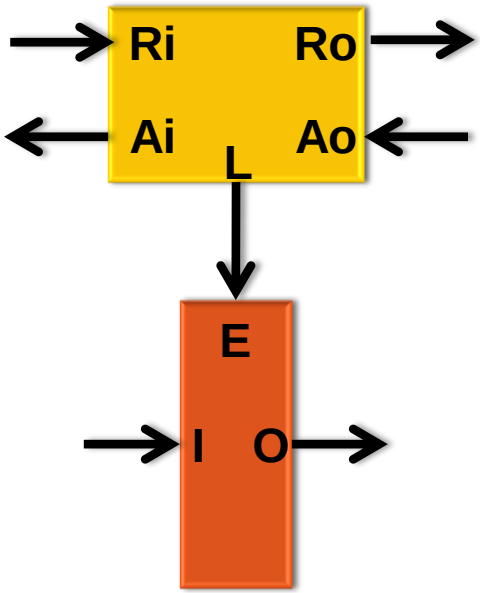
MSFSMS EXAMPLE: 4-PHASE LATCH CTRL

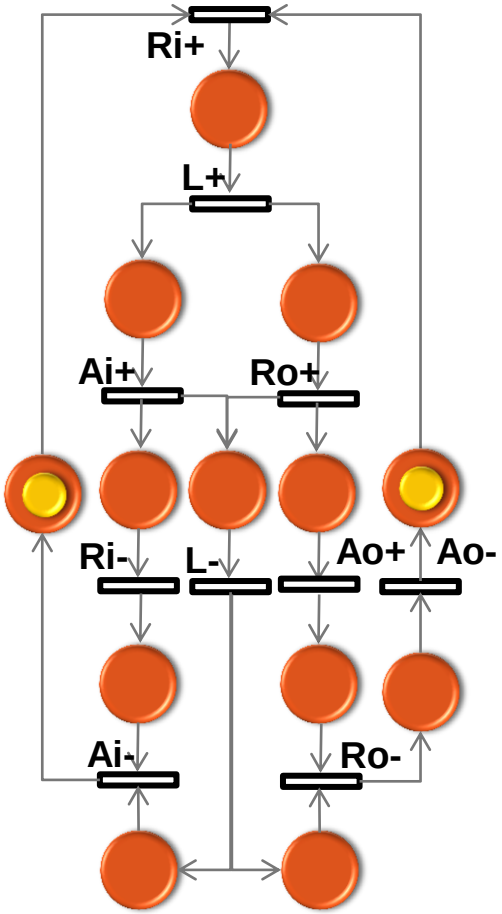




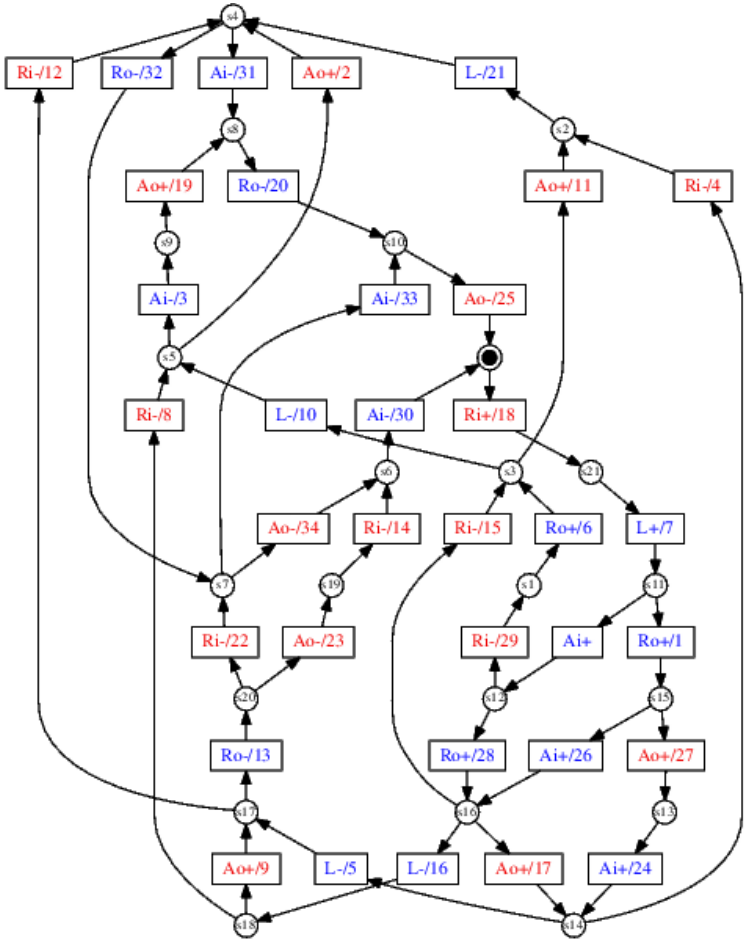
PTNet

MSFSMS EXAMPLE: 4-PHASE LATCH CTRL



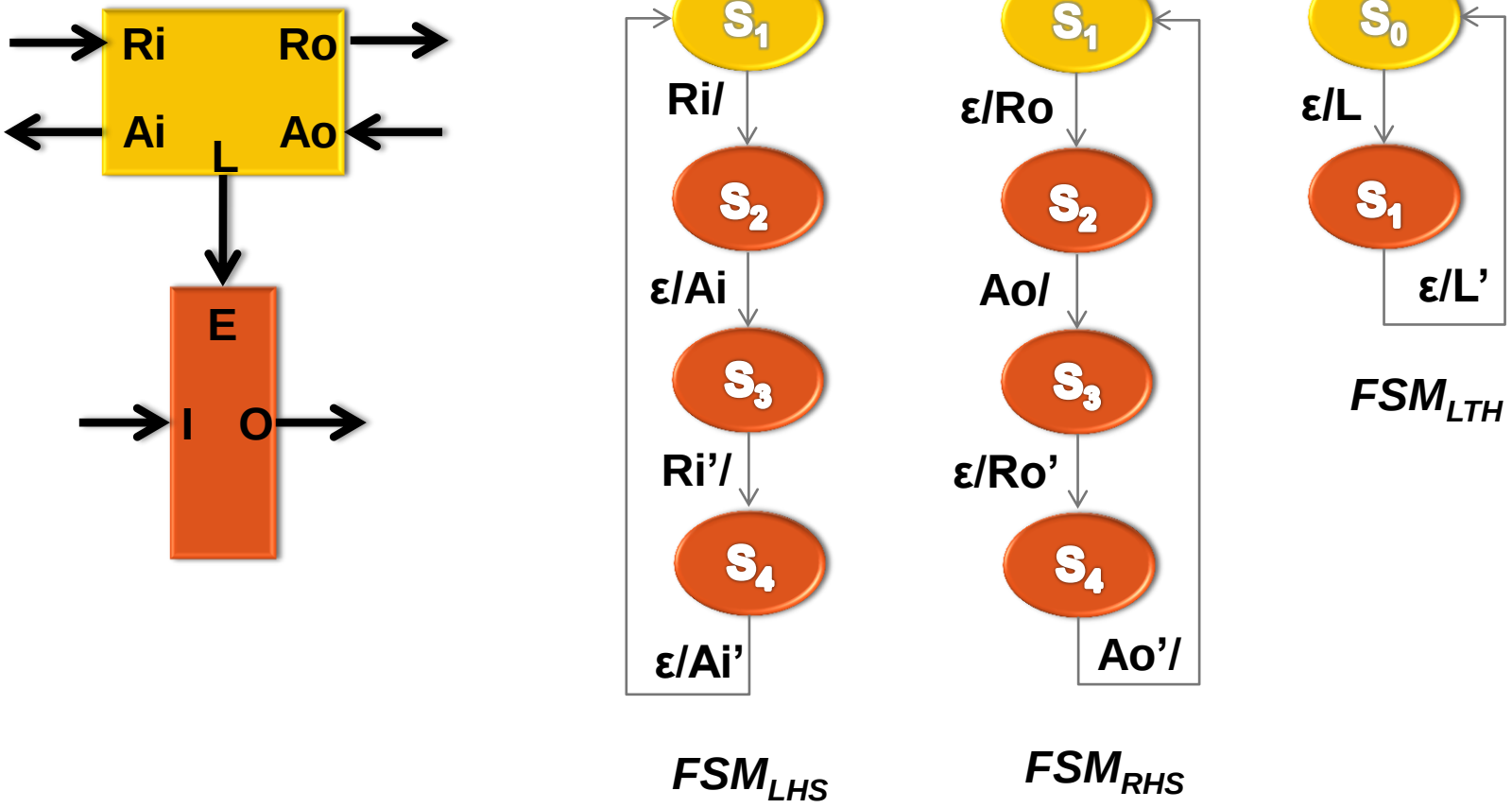


PTNet

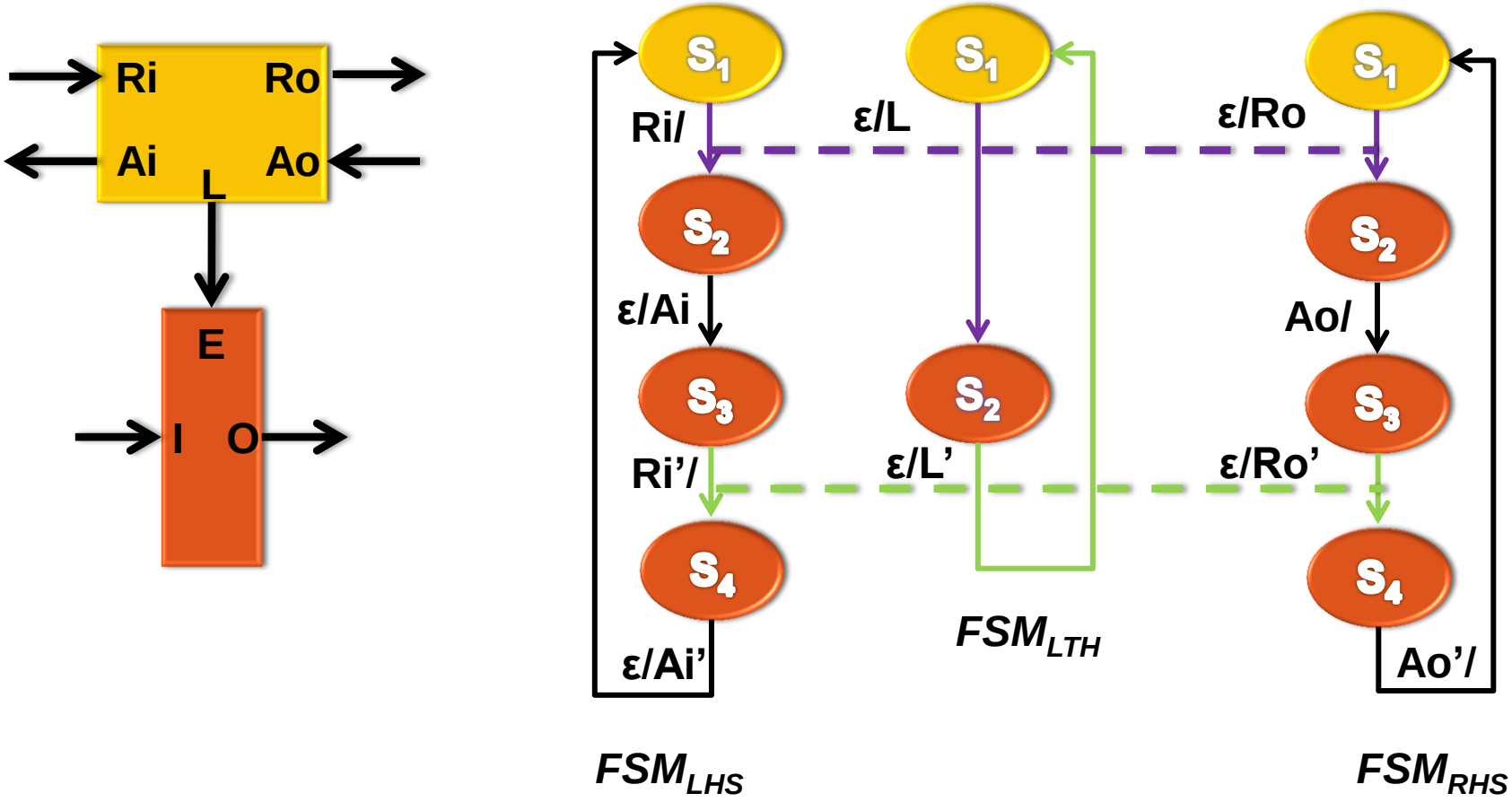


State Graph (FSM)

MSFSMS EXAMPLE: 4-PHASE LATCH CTRL



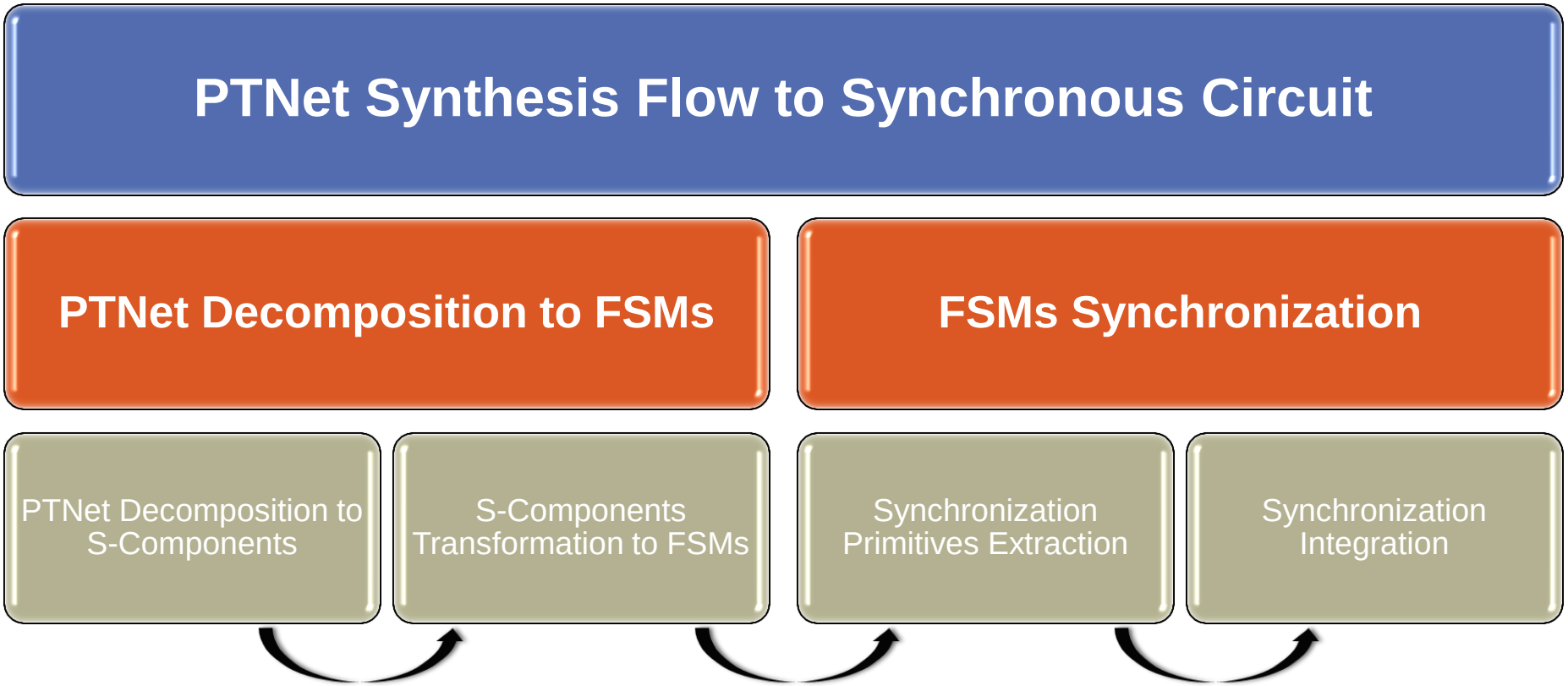
MSFSMS EXAMPLE: 4-PHASE LATCH CTRL



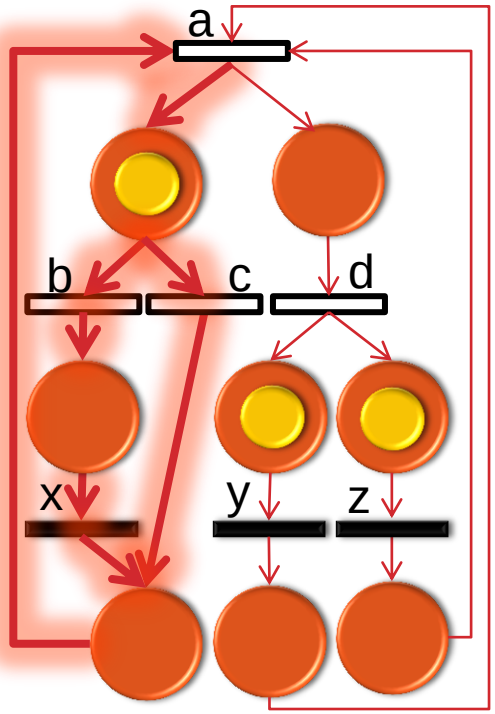
OVERVIEW

- Motivation
- Background
- MSFSMs
- ***Synthesis***
- Verification
- Optimization
- Results
- Conclusions and Future Work

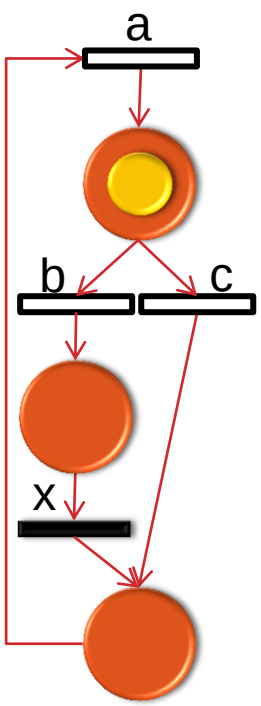
OVERVIEW


PTNet Synthesis Flow to Synchronous Circuit
PTNet Decomposition to FSMs
FSMs Synchronization
PTNet Decomposition to S-Components
S-Components Transformation to FSMs
Synchronization Primitives Extraction
Synchronization Integration

STEPS (1/4)



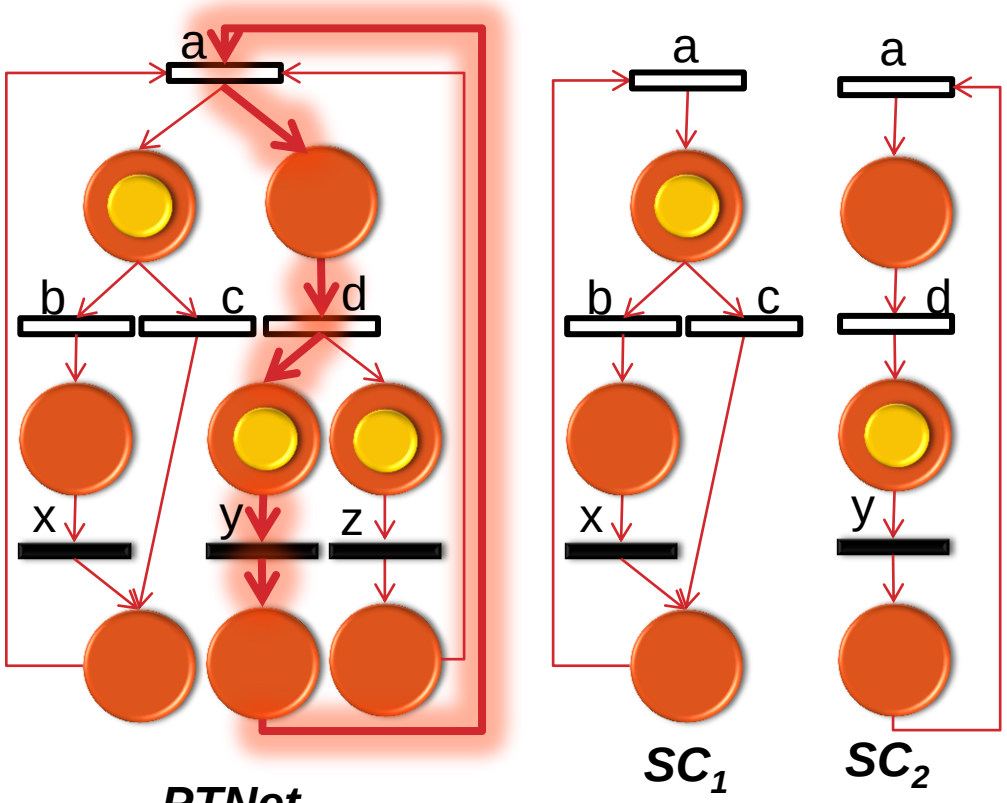
PTNet



SC_1

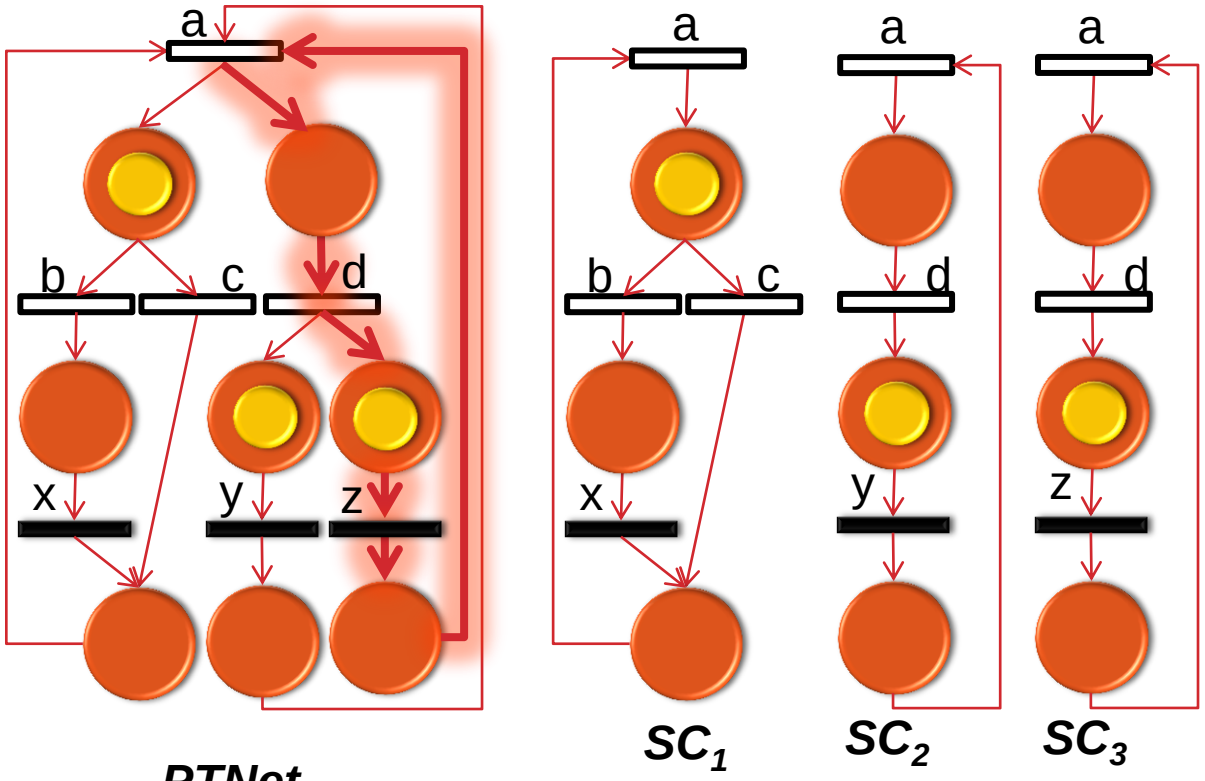
- PTNet to S-Component Decomposition
- S-Component to FSM Mapping
- Synchronization Primitive Extraction
- Synchronization Integration

STEPS (1/4)



- PTNet to S-Component Decomposition
- S-Component to FSM Mapping
- Synchronization Primitive Extraction
- Synchronization Integration

STEPS (1/4)



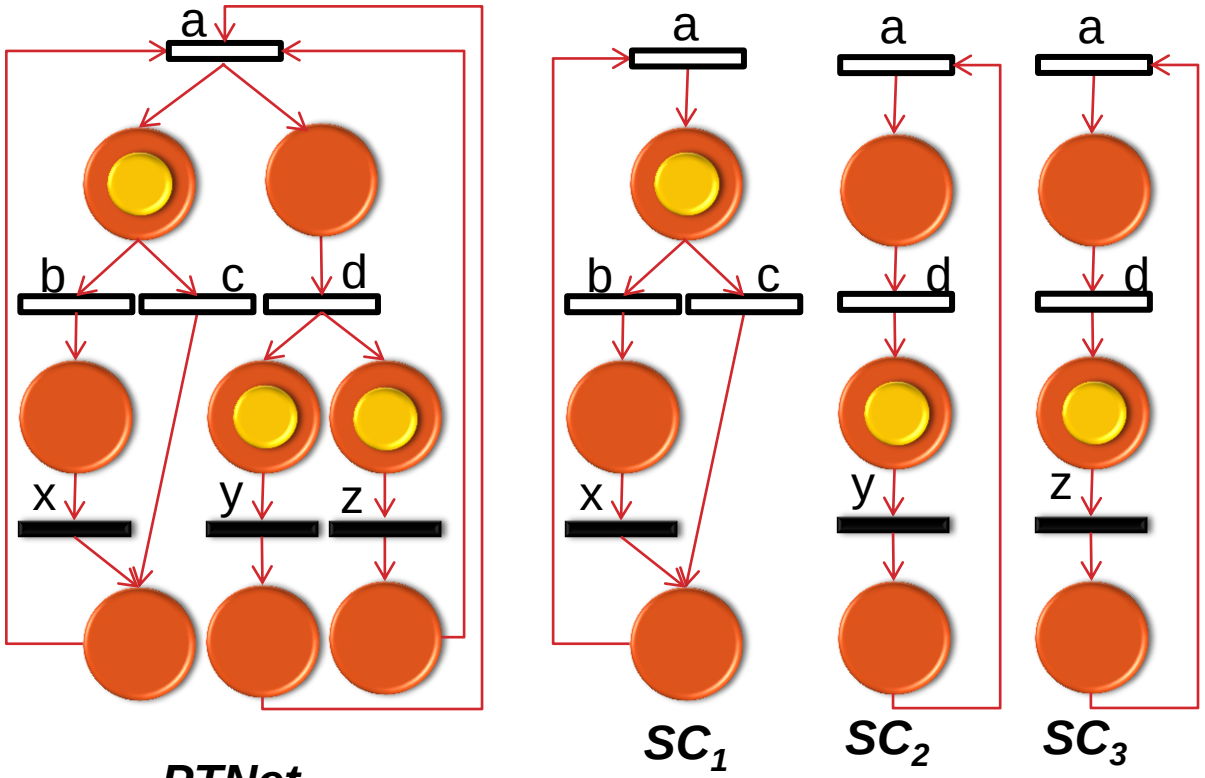
PTNet to S-Component Decomposition

S-Component to FSM Mapping

Synchronization Primitive Extraction

Synchronization Integration

STEPS (1/4)



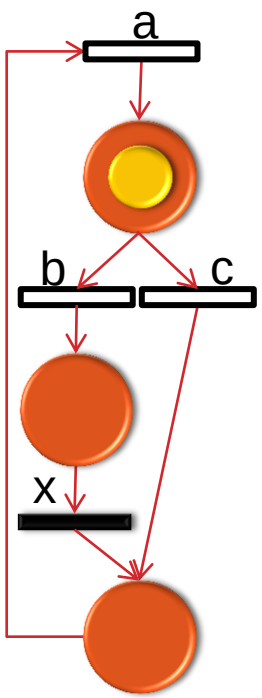
PTNet to S-Component Decomposition

S-Component to FSM Mapping

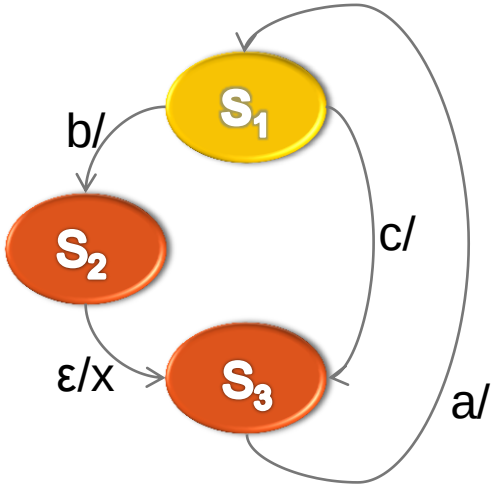
Synchronization Primitive Extraction

Synchronization Integration

STEPS (2/4)



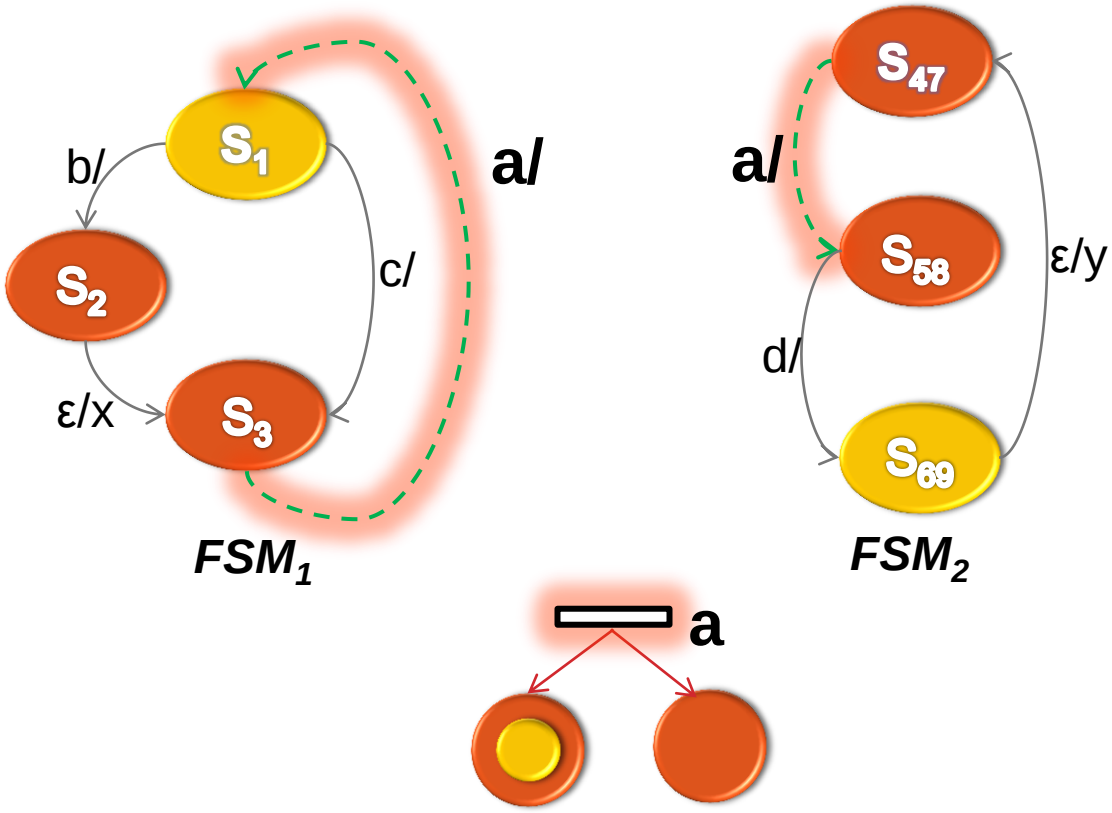
SC_1



FSM_1

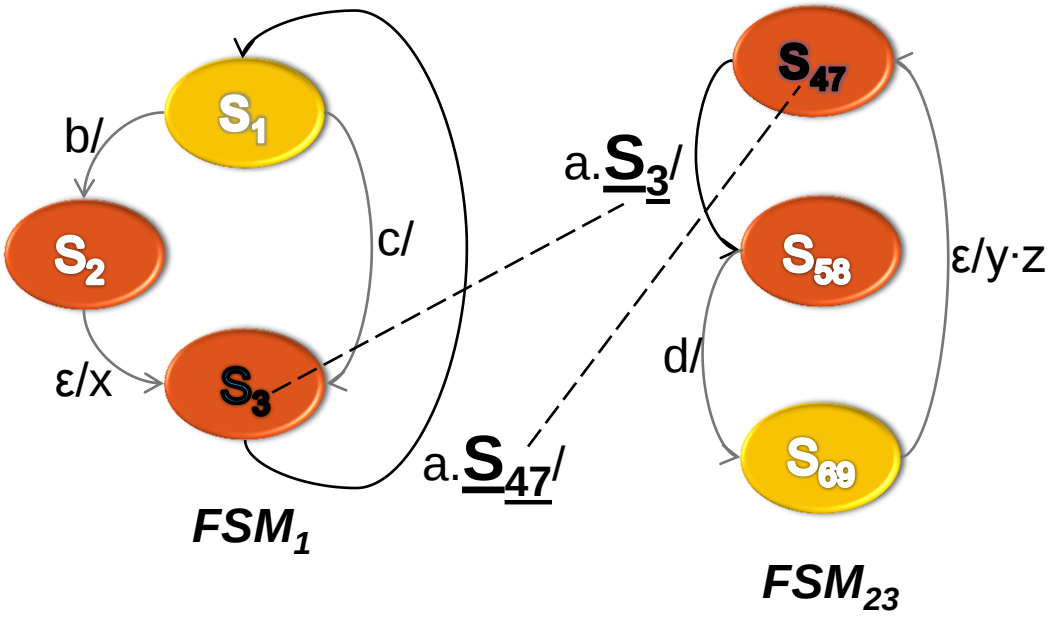
- PTNet to S-Component Decomposition
- S-Component to FSM Mapping
- Synchronization Primitive Extraction
- Synchronization Integration

STEPS (3/4)



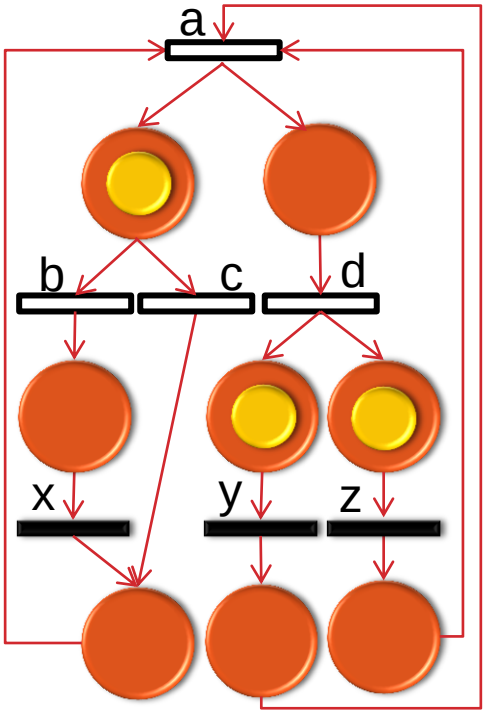
- PTNet to S-Component Decomposition
- S-Component to FSM Mapping
- Synchronization Primitive Extraction
- Synchronization Integration

STEPS (4/4)

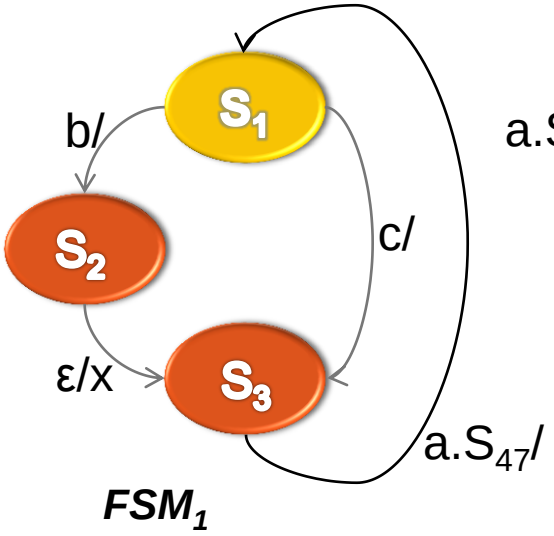


- PTNet to S-Component Decomposition
- S-Component to FSM Mapping
- Synchronization Primitive Extraction
- Synchronization Integration

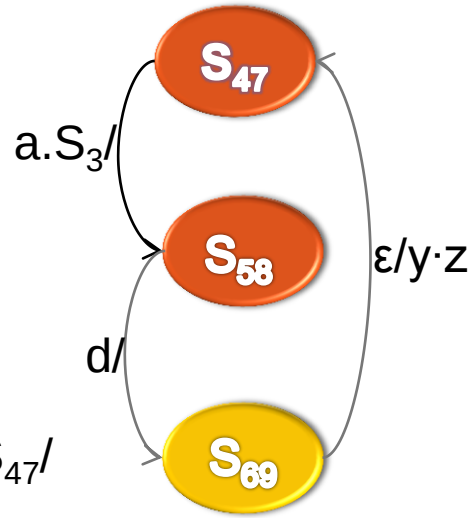
FINAL RESULT



PTNet



FSM1

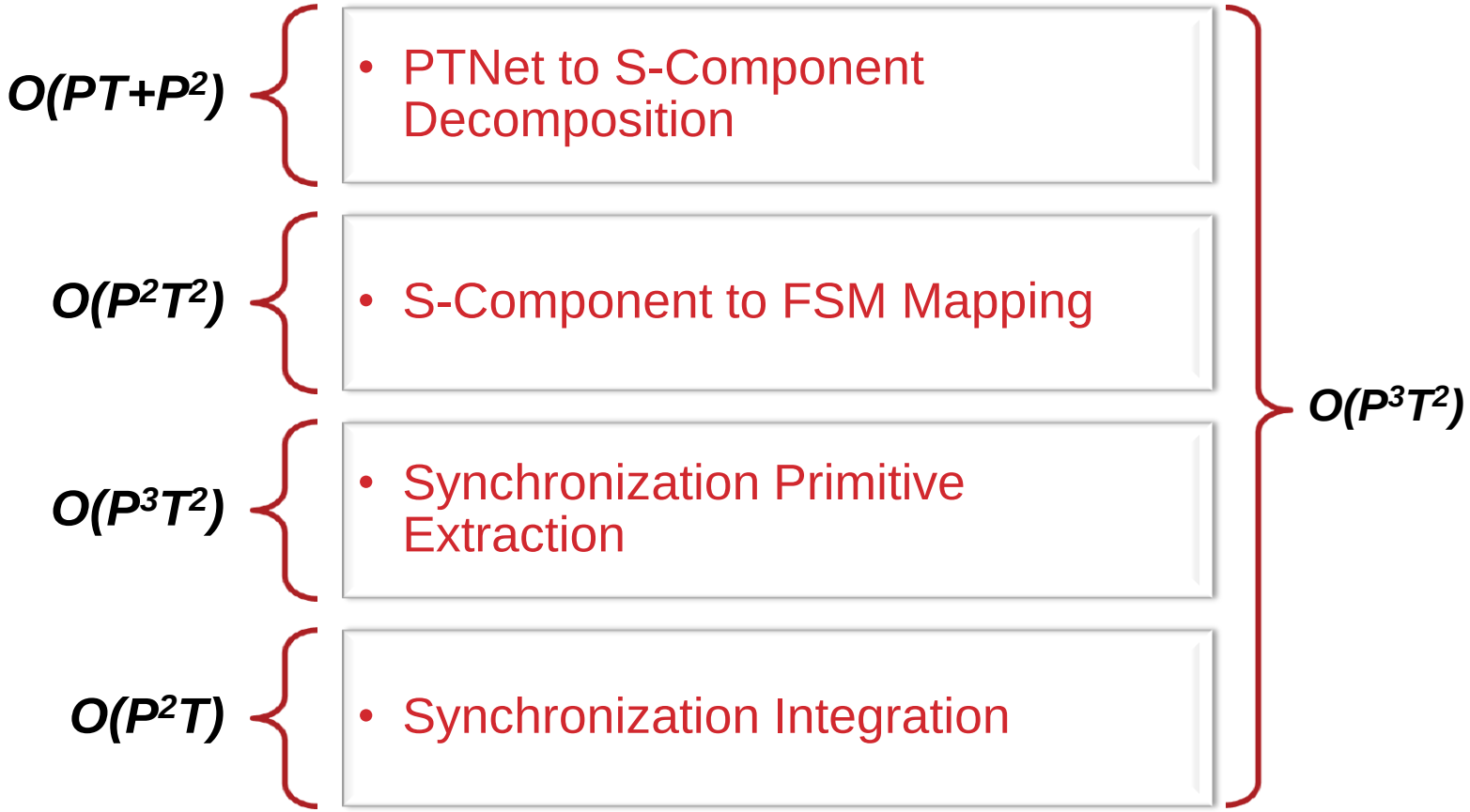


FSM23

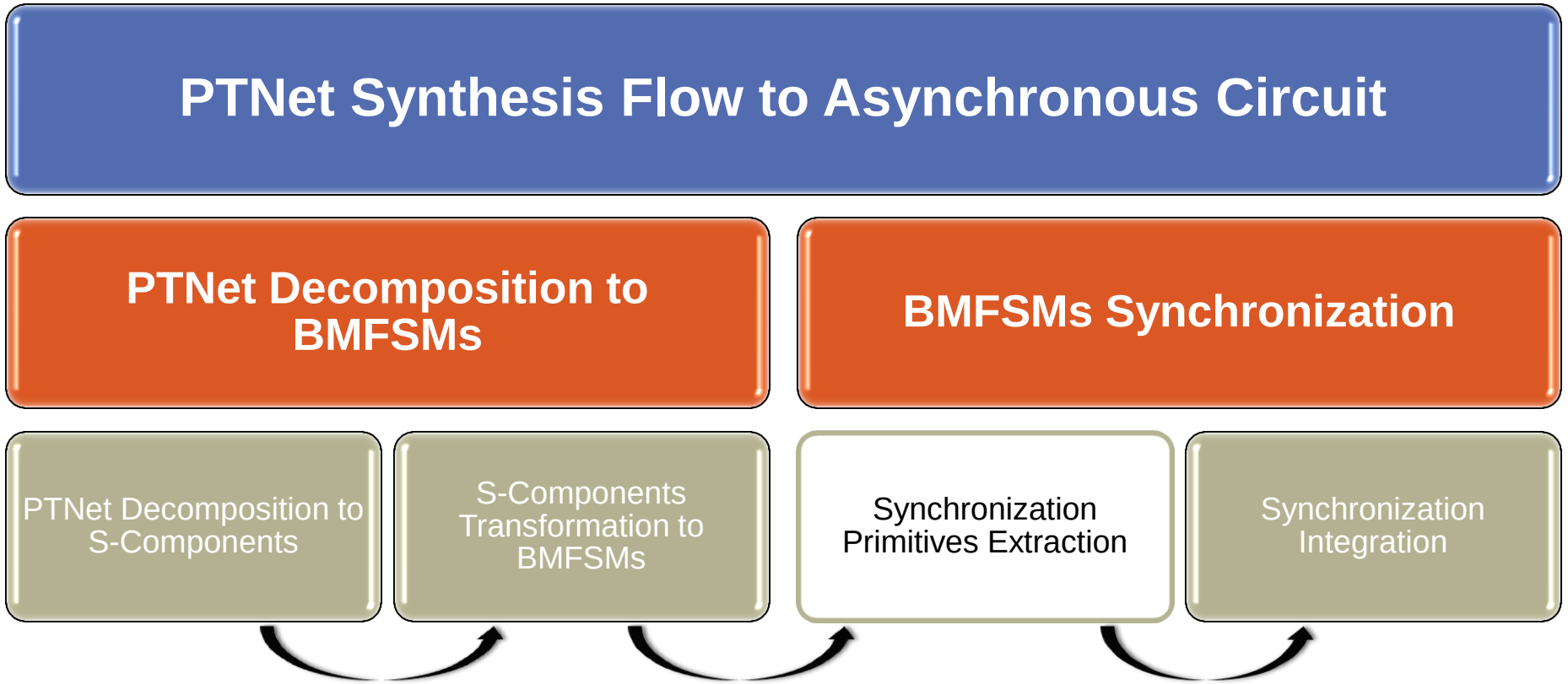
MSFSM

FSMs ***Synchronization***

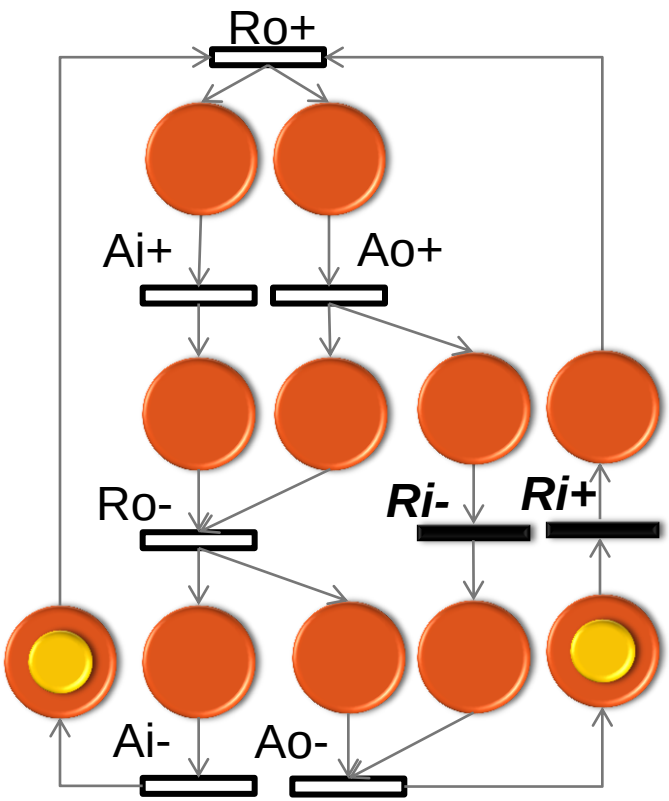
COMPLEXITY


O(PT+P²)
• PTNet to S-Component Decomposition
O(P²T²)
• S-Component to FSM Mapping
O(P³T²)
• Synchronization Primitive Extraction
O(P²T)
• Synchronization Integration
O(P³T²)

OVERVIEW


PTNet Synthesis Flow to Asynchronous Circuit
PTNet Decomposition to BMFSMs
BMFSMs Synchronization
PTNet Decomposition to S-Components
S-Components Transformation to BMFSMs
Synchronization Primitives Extraction
Synchronization Integration

STEPS 1/3



PTNet

1. Choose a signal (Ri) and form all the consecutive transition pairs (Ri+,Ri-) (Ri-,Ri+)

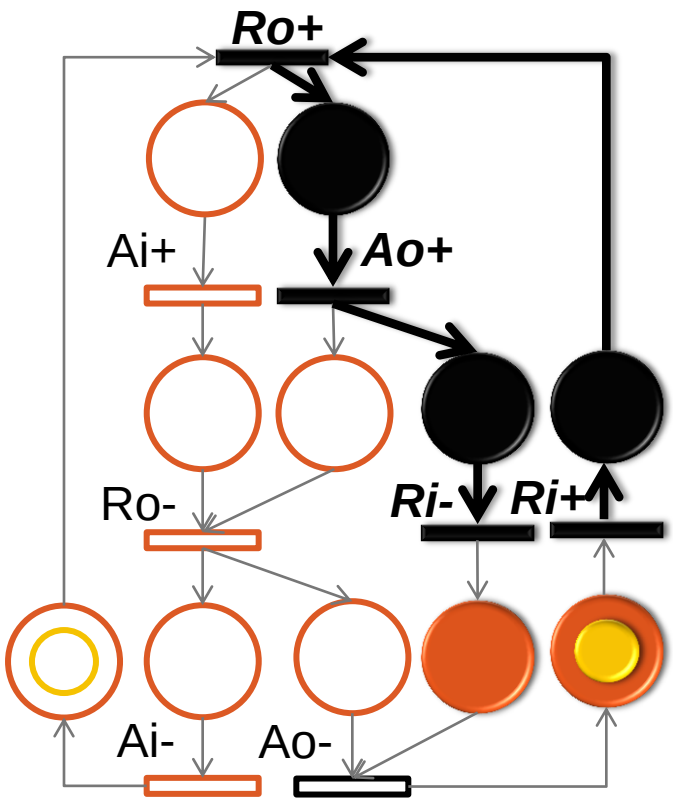
PTNet to S-Component Decomposition

S-Component to BM-FSM Mapping

Synchronization Primitive Extraction

Synchronization Integration

STEPS 1/3



PTNet

1. Choose a signal (Ri) and form all the consecutive transition pairs (Ri+,Ri-) (Ri-,Ri+)

2. For each pair (Ri+,Ri-)
(i)remove all concurrent PTNet components
(ii)DFS to connect pair's transitions

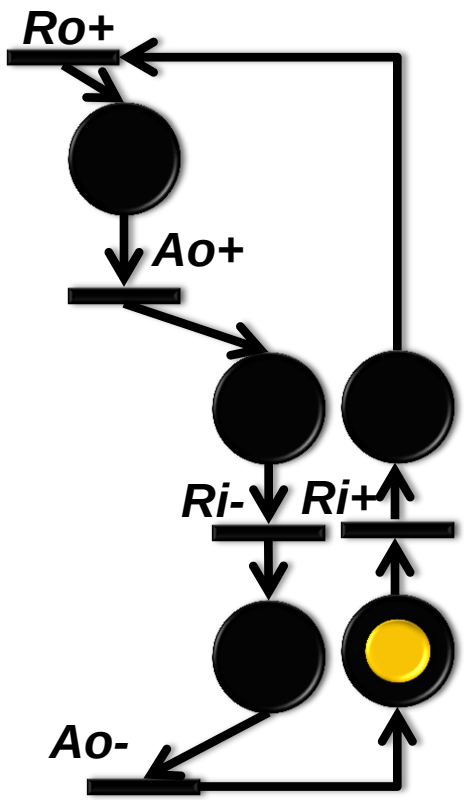
PTNet to S-Component Decomposition

S-Component to BM-FSM Mapping

Synchronization Primitive Extraction

Synchronization Integration

STEPS 1/3



SComponent

1. *Choose a signal (Ri) and form all the consecutive transition pairs (Ri+,Ri-) (Ri-,Ri+)*

2. *For each pair (Ri+,Ri-)*
   (i)remove all concurrent PTNet components
   (ii)DFS to connect pair's transitions

3. *Remove all unmarked PTNet components and their corresponding arcs*

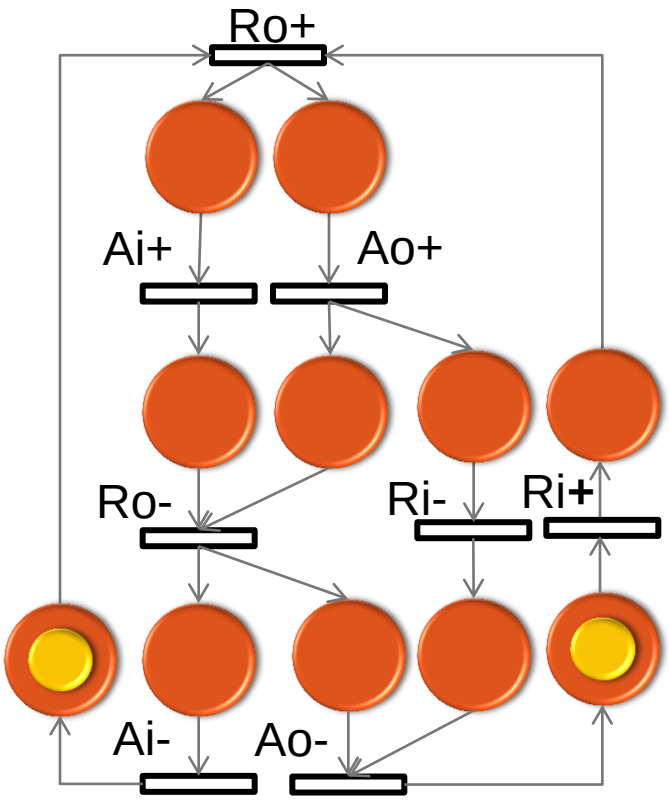
PTNet to S-Component Decomposition

S-Component to BM-FSM Mapping

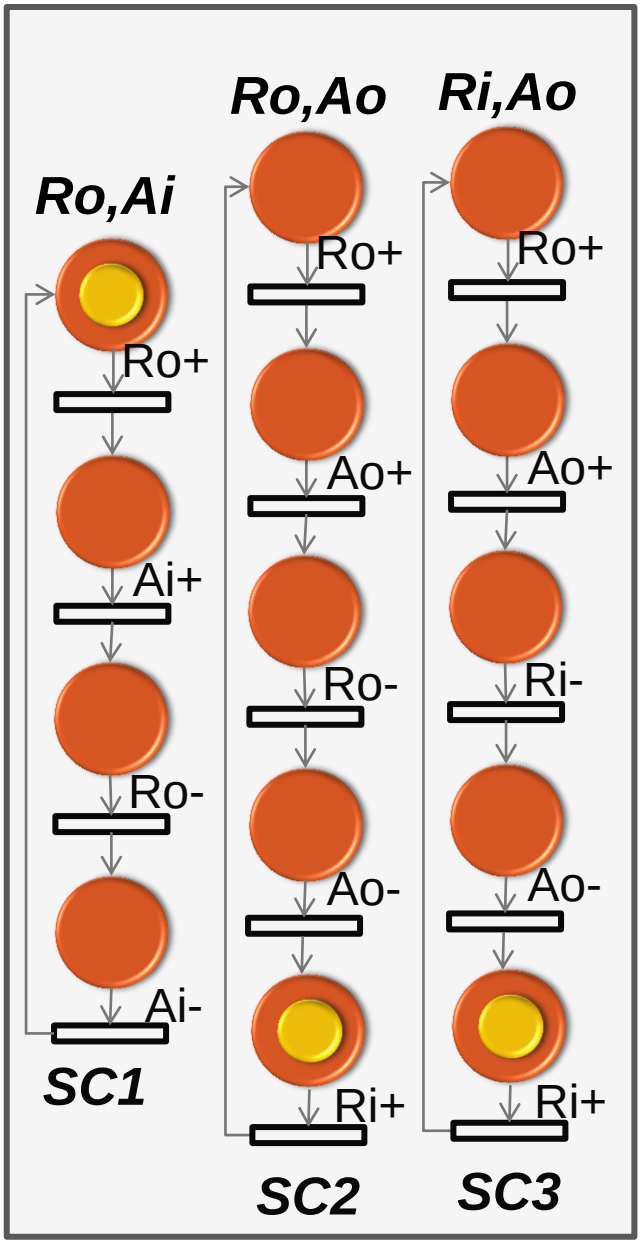
Synchronization Primitive Extraction

Synchronization Integration

STEPS 1/3



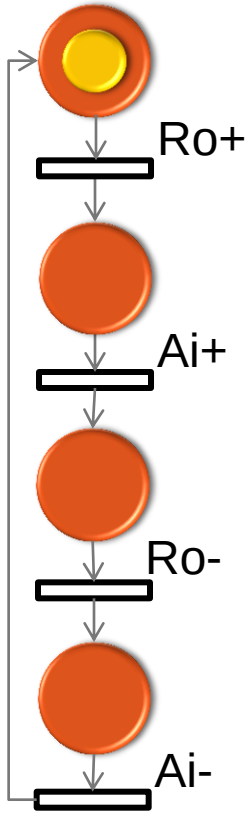
PTNet



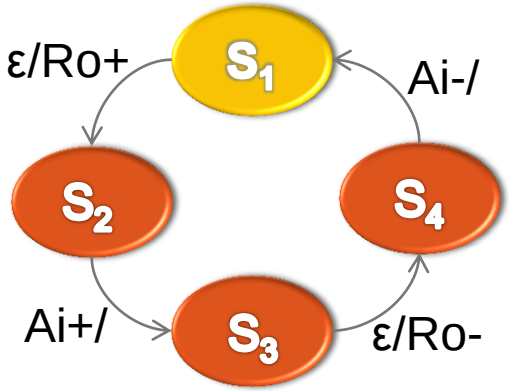
SCover

- PTNet to S-Component Decomposition
- S-Component to BM-FSM Mapping
- Synchronization Primitive Extraction
- Synchronization Integration

STEPS 2/3



SC1



FSM1

PTNet to S-Component Decomposition

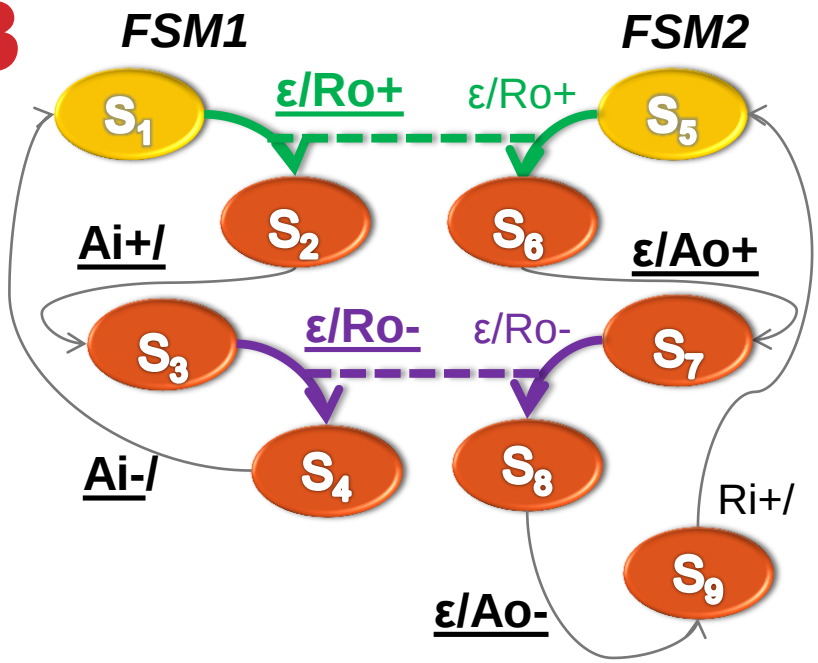
S-Component to BM-FSM Mapping

Synchronization Primitive Extraction

Synchronization Integration

STEPS 3/3

1. Form the Signal Support for each BMFSM



PTNet to S-Component Decomposition

S-Component to BM-FSM Mapping

Synchronization Primitive Extraction

Synchronization Integration

FSM1

	Ai	Ro
S1	0	0
S2	0	1
S3	1	1
S4	1	0

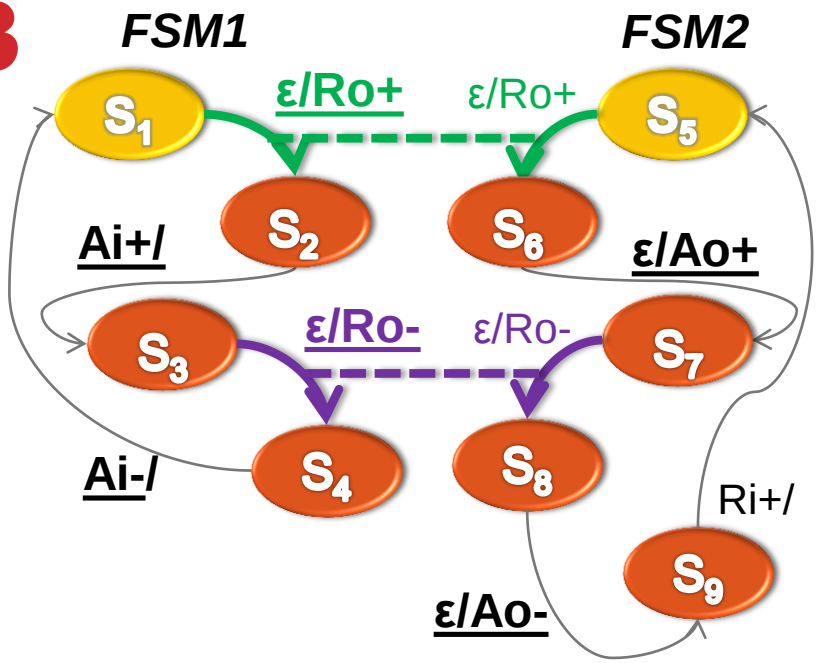
FSM2

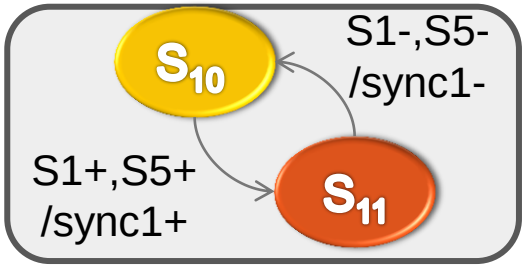
	Ao
S5	0
S6	0
S7	1
S8	1
S9	0

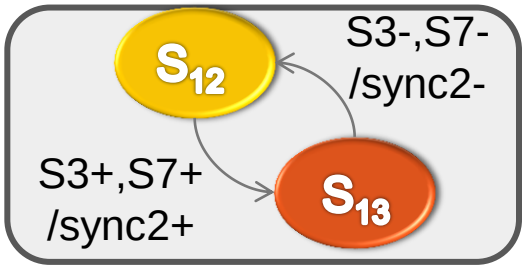
STEPS 3/3

1. *Form the Signal Support for each BMFSM*

2. *Insert Synchronizing BMFSMs*







PTNet to S-Component Decomposition

S-Component to BM-FSM Mapping

Synchronization Primitive Extraction

Synchronization Integration

FSM1

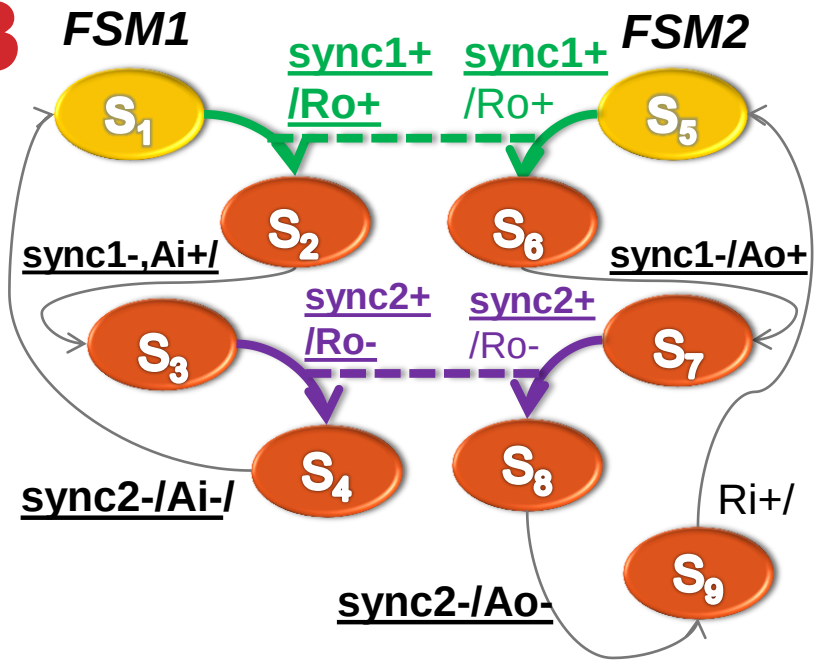
	Ai	Ro
S1	0	0
S2	0	1
S3	1	1
S4	1	0

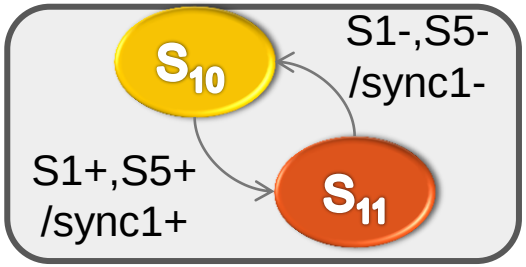
FSM2

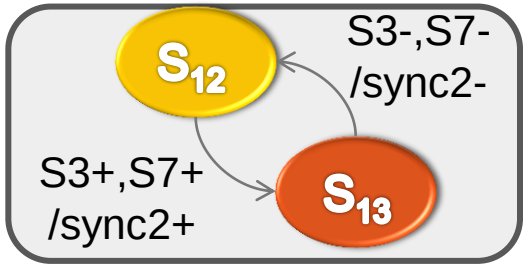
	Ao
S5	0
S6	0
S7	1
S8	1
S9	0

STEPS 3/3

1. *Form the Signal Support for each BMFSM*
2. *Insert Synchronizing BMFSMs*
3. *Insert Synchronization Logic*







PTNet to S-Component Decomposition

S-Component to BM-FSM Mapping

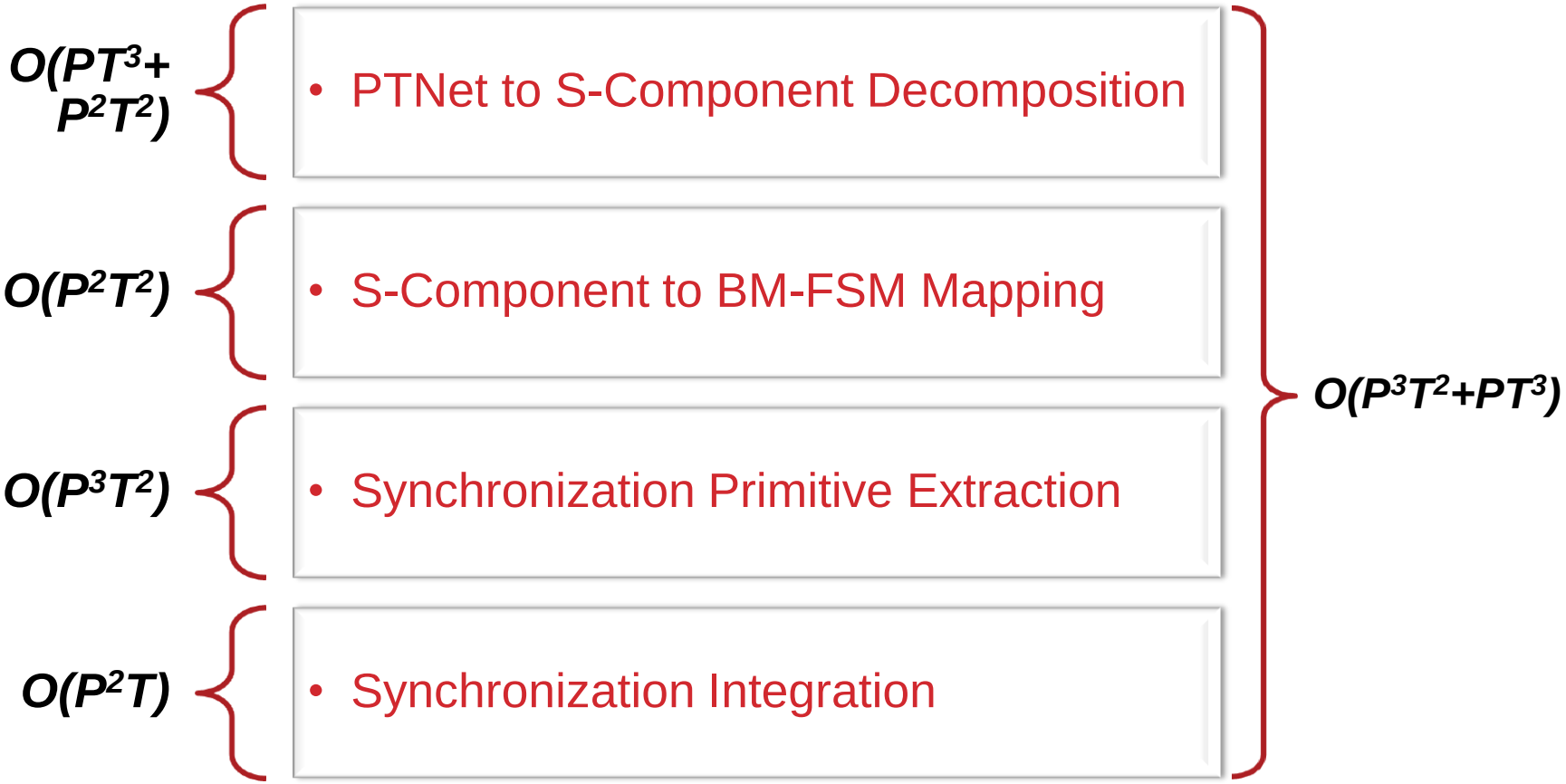
Synchronization Primitive Extraction

Synchronization Integration

FSM1

	Ai	Ro	S1	S3	sync1	sync2
S1	0	0	1	0	0	0
S2	0	1	0	0	1	0
S3	1	1	0	1	0	0
S4	1	0	0	0	0	1

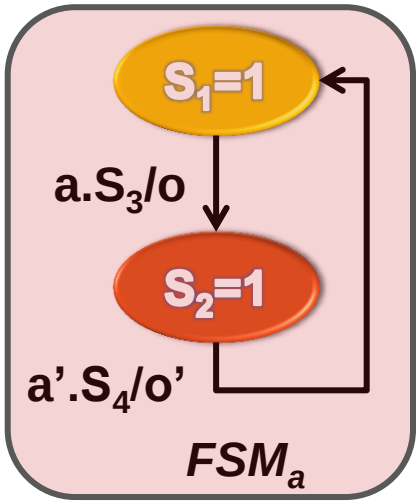
COMPLEXITY

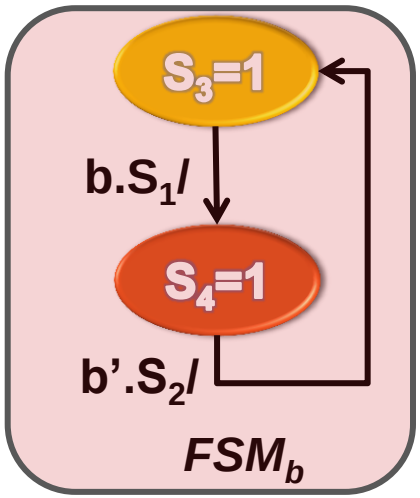

$O(PT^3+P^2T^2)$
• PTNet to S-Component Decomposition
$O(P^2T^2)$
• S-Component to BM-FSM Mapping
$O(P^3T^2)$
• Synchronization Primitive Extraction
$O(P^2T)$
• Synchronization Integration
$O(P^3T^2+PT^3)$

OVERVIEW

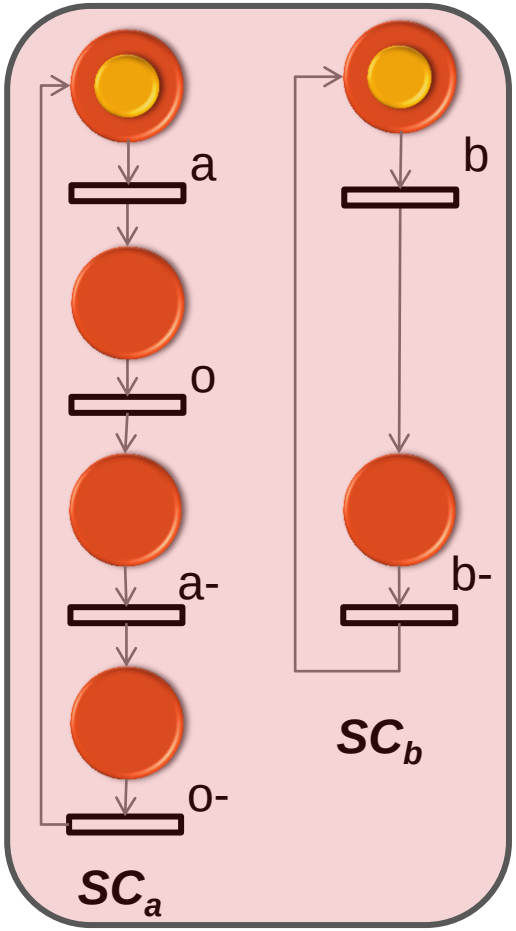
- Motivation
- Background
- MSFSMs
- Synthesis
- ***Verification***
- Optimization
- Results
- Conclusions and Future Work

INTERACTING FSMS TO PTNETS

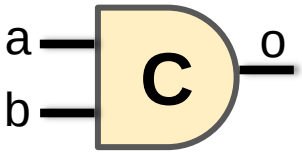


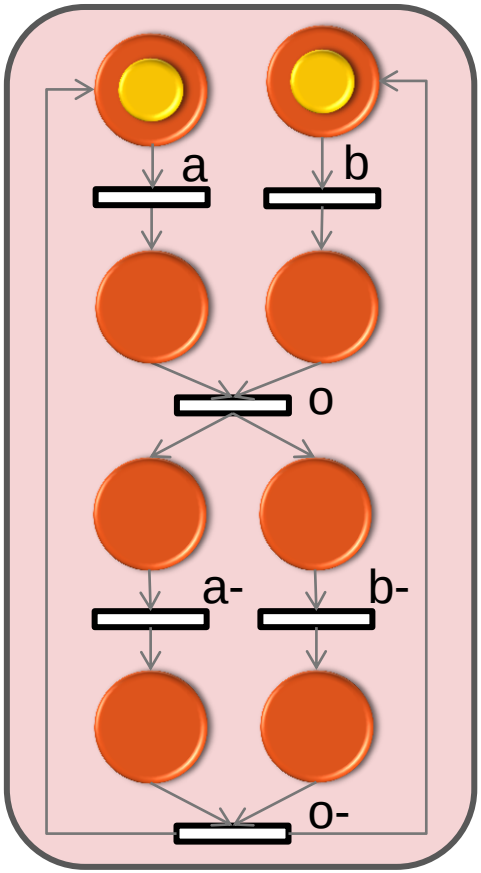


Interacting FSMs



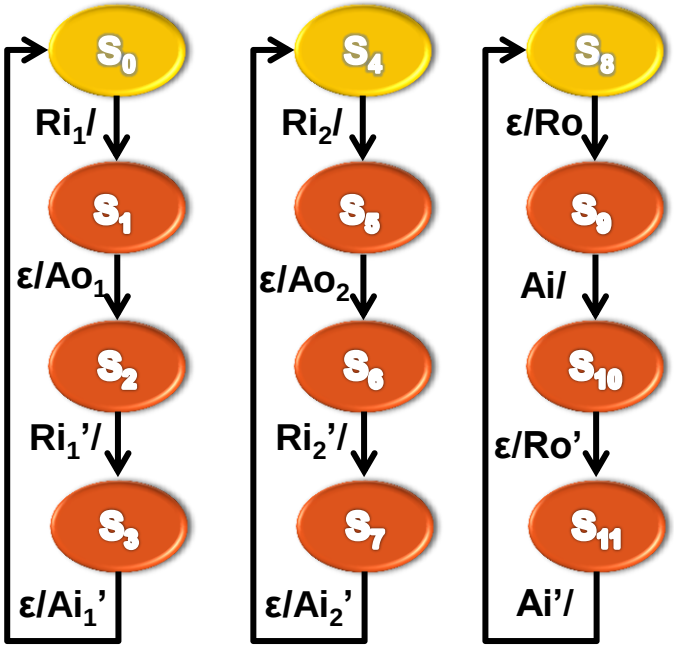
SCover

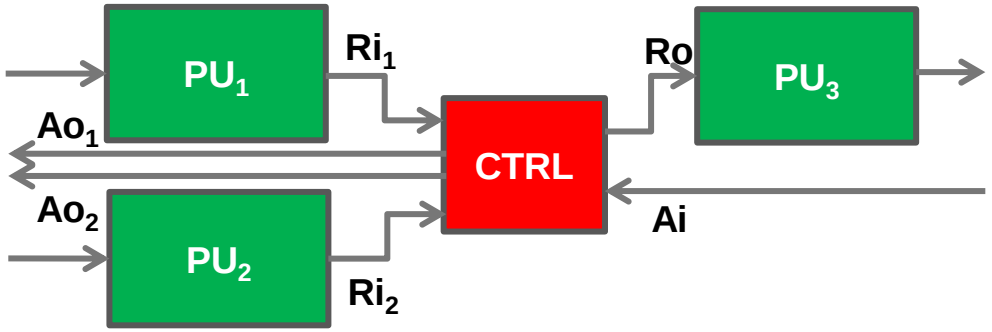




PTNet

MSFSM VERIFICATION

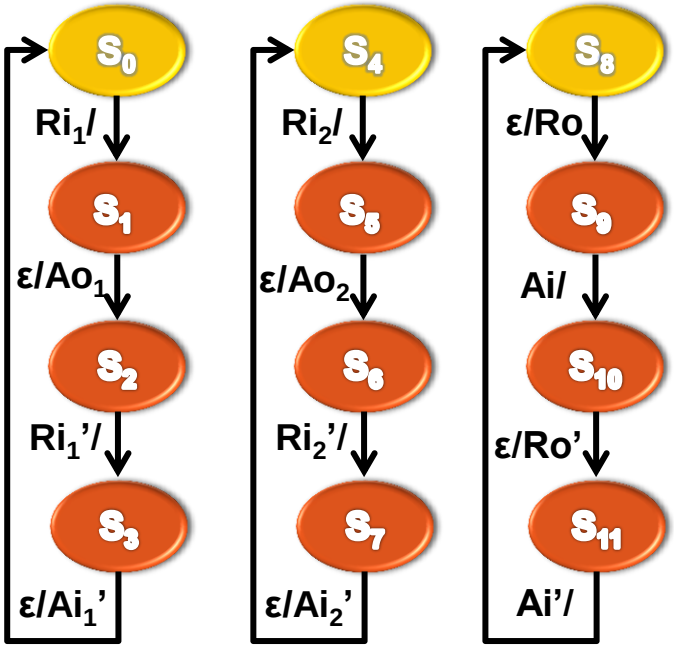




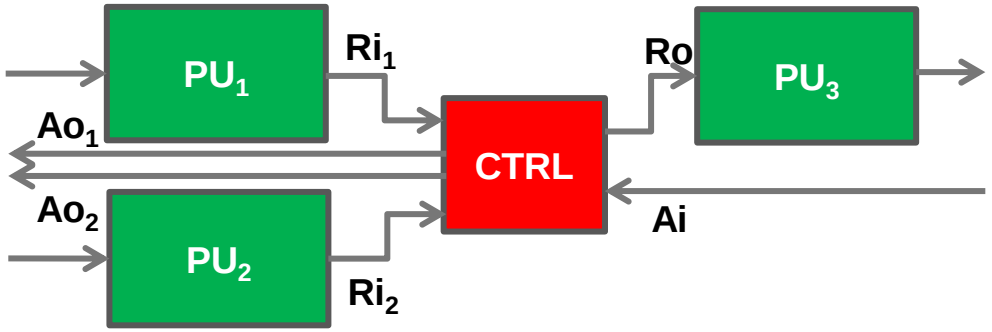
Synchronization Points	
W	*Dependent States*
S8	*S1, S5*
S1	*S10*
S5	*S10*

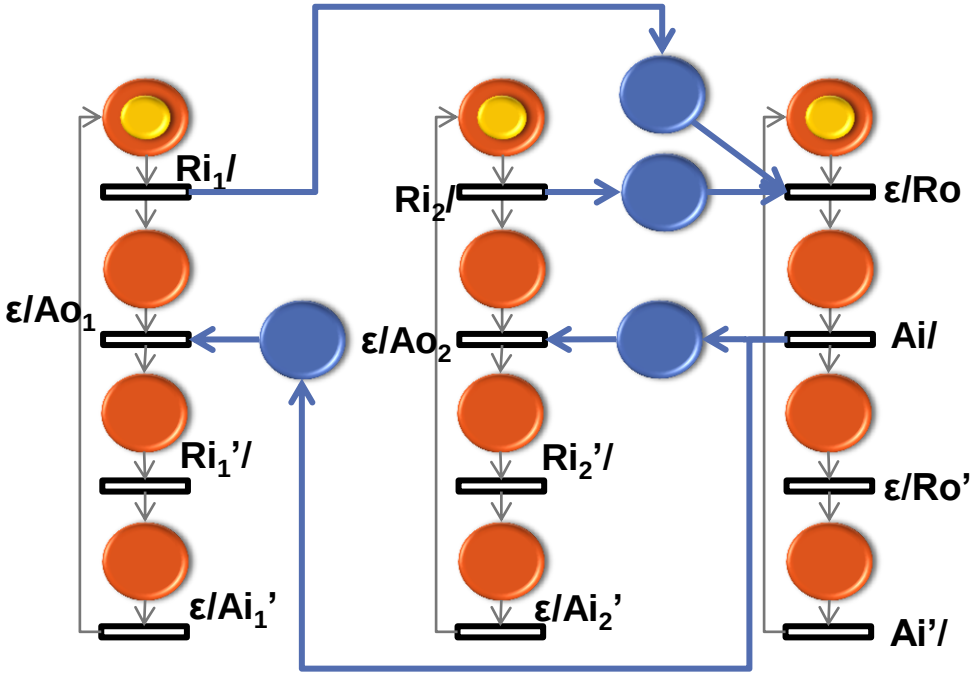
Is the Concurrent System Deadlock-Free?

MSFSM VERIFICATION

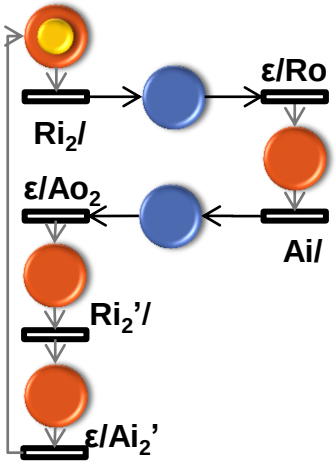


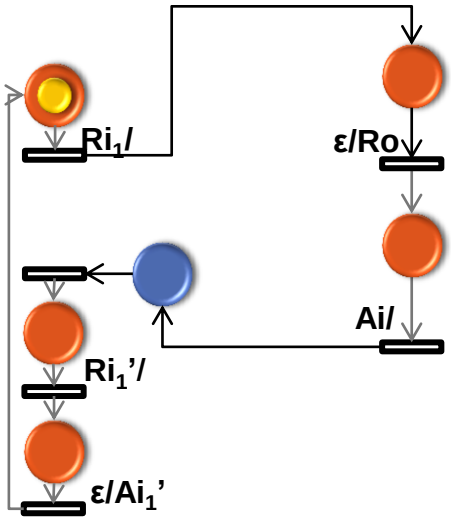
Synchronization Points	
W	*Dependent States*
S8	*S1, S5*
S1	*S10*
S5	*S10*

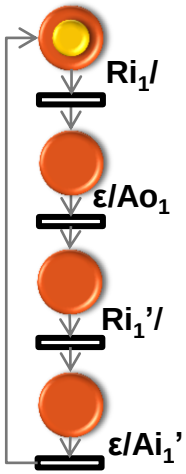


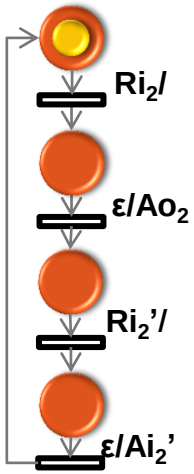


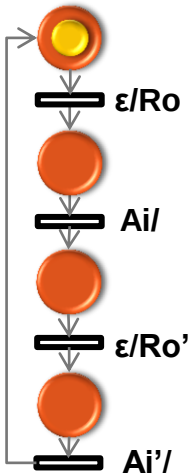
MSFSM VERIFICATION











The system is covered by 5 initially marked minimal siphons.

Deadlock-Free

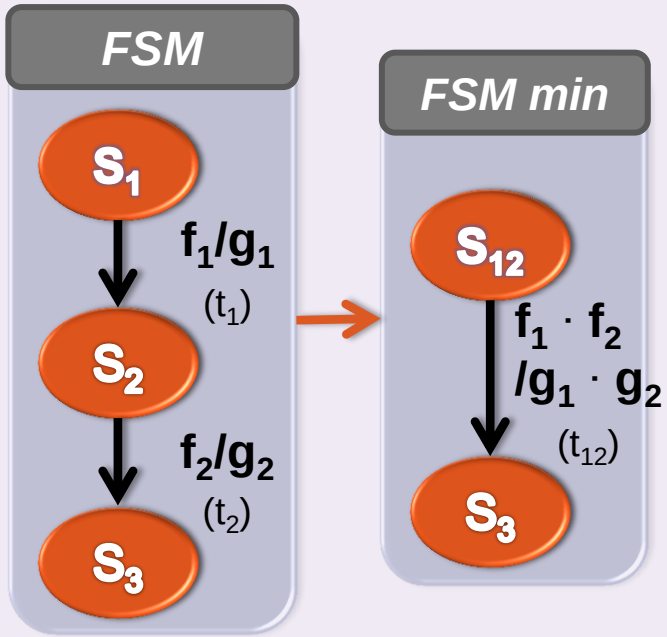
OVERVIEW

- Motivation
- Background
- MSFSMs
- Synthesis
- Verification
- ***Optimization***
- Results
- Conclusions and Future Work

FORTH
Institute of Computer Science

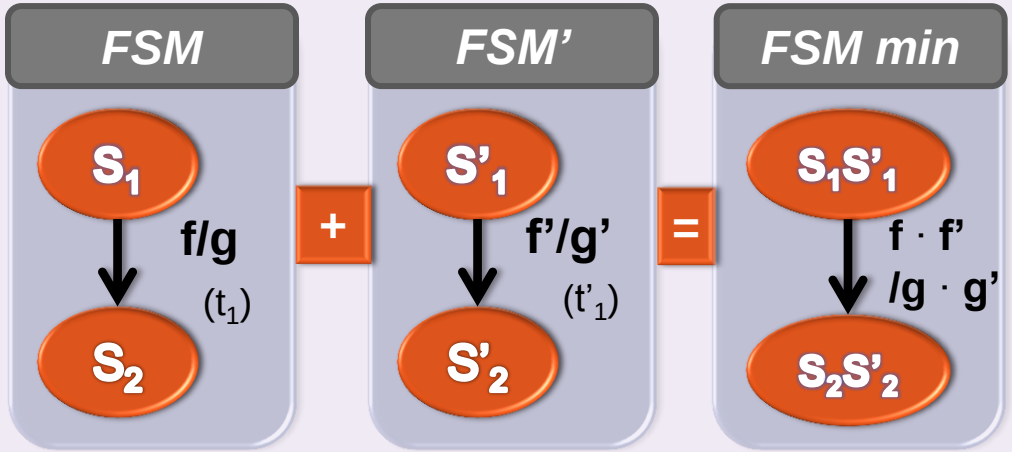
HORIZONTAL AND VERTICAL COLLAPSING

Vertical



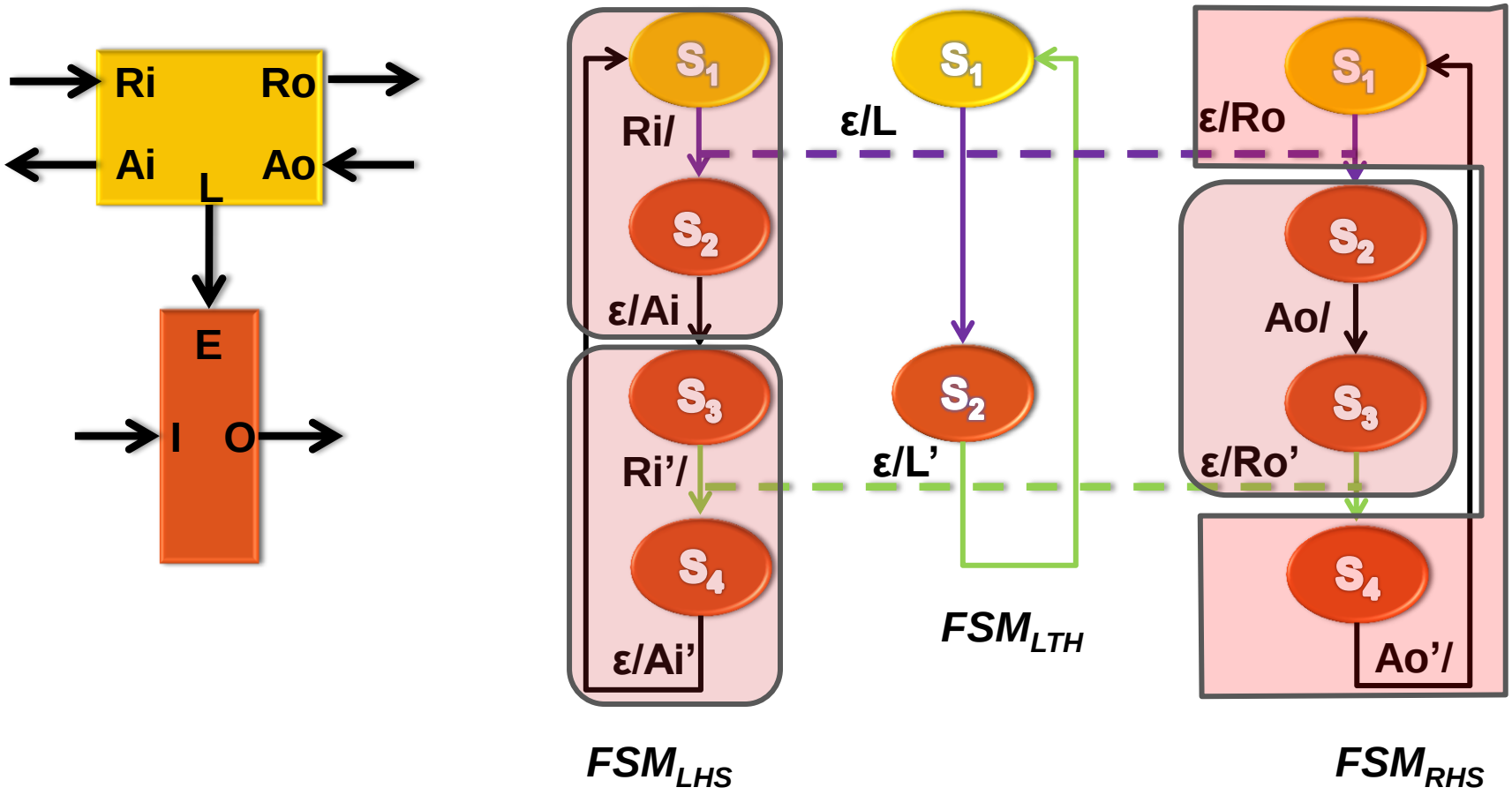
- t_1 and t_2 are not synced and $g_1=\varepsilon$ | $f_2=\varepsilon$
- t_1 is synced and $f_2=\varepsilon$
- t_2 is synced and $g_1=\varepsilon$

Horizontal

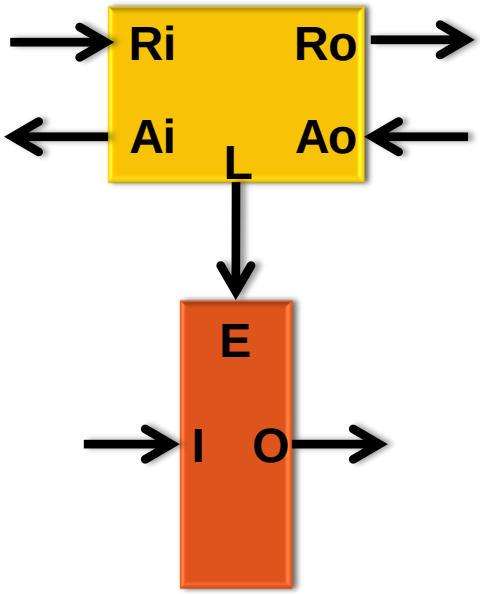


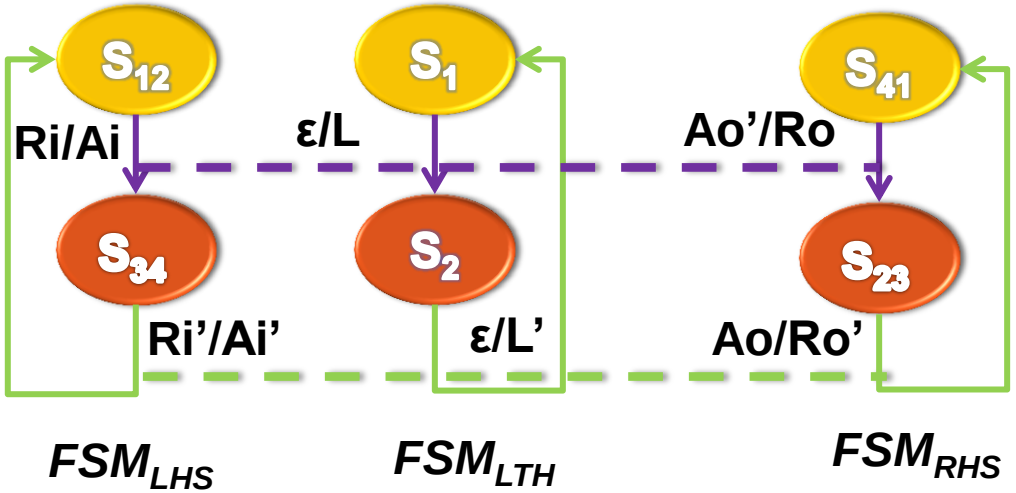
- **FSM and FSM' exhibit concurrency between transitions t and t'**
  - f(t)=ε & f'(t)=ε
  - g(t)=ε & g'(t)=ε

HORIZONTAL AND VERTICAL COLLAPSING

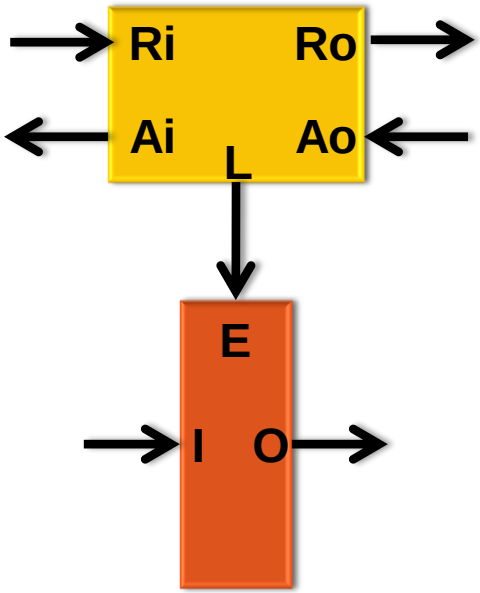


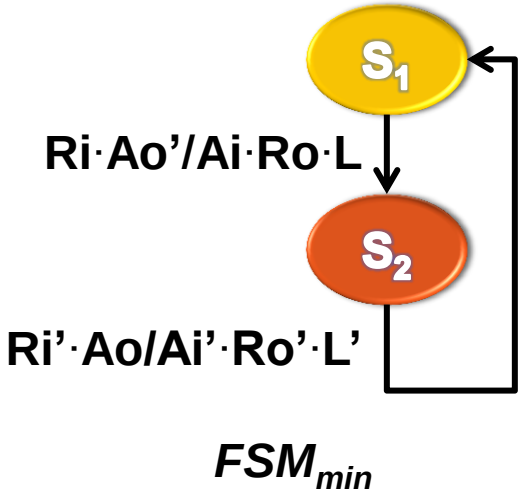
HORIZONTAL AND VERTICAL COLLAPSING





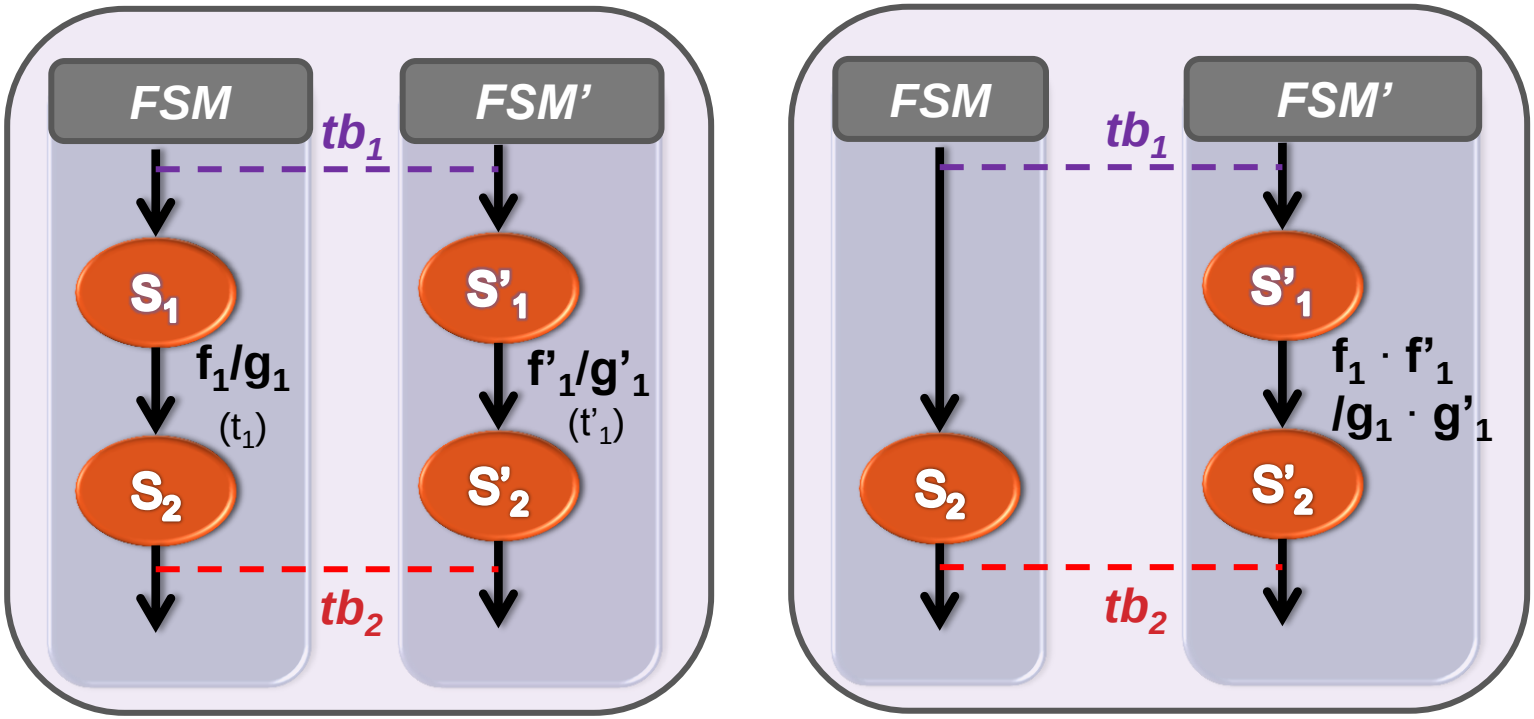
HORIZONTAL AND VERTICAL COLLAPSING





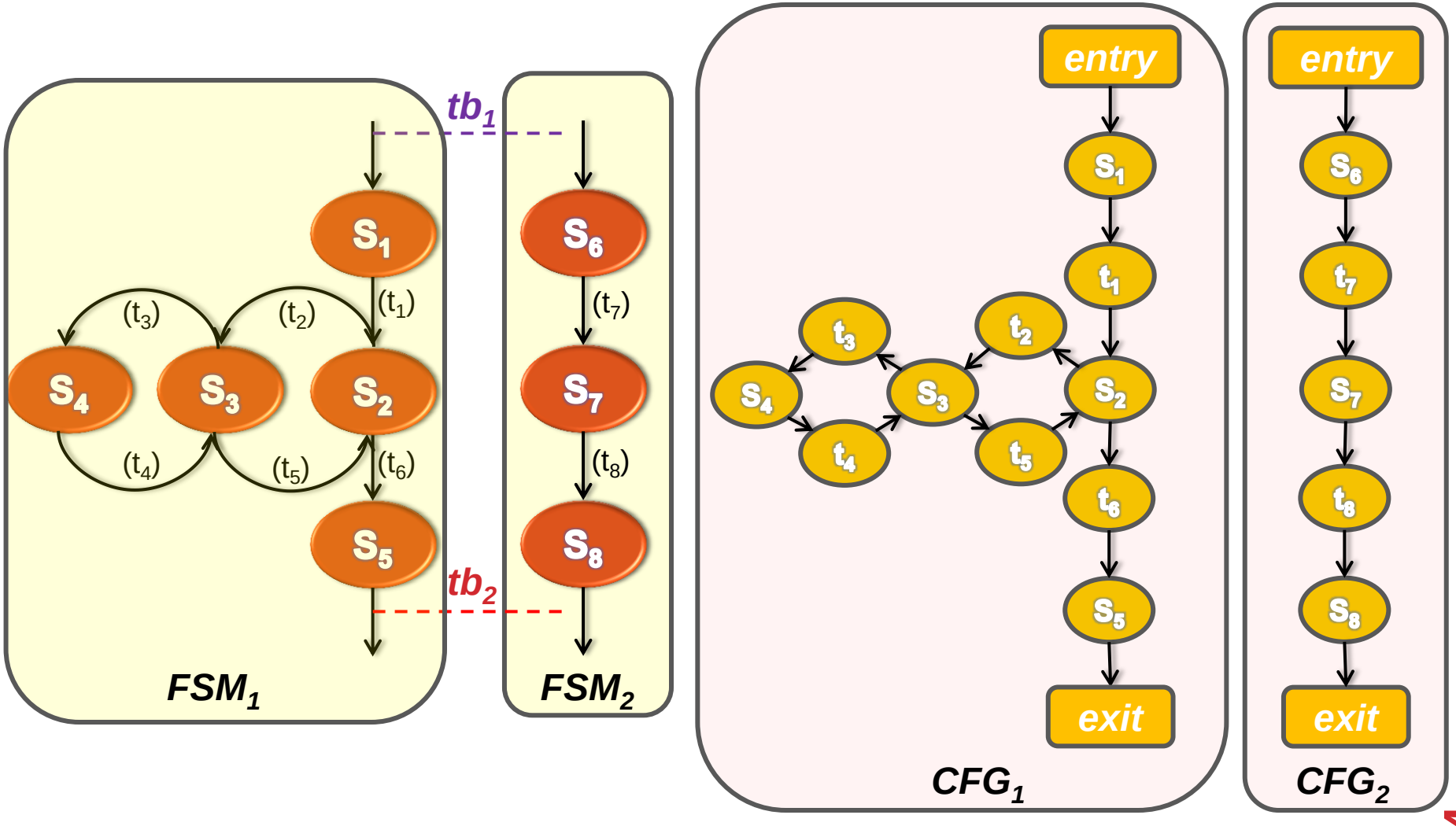
FSM$_{min}$

X-COMPATIBLES

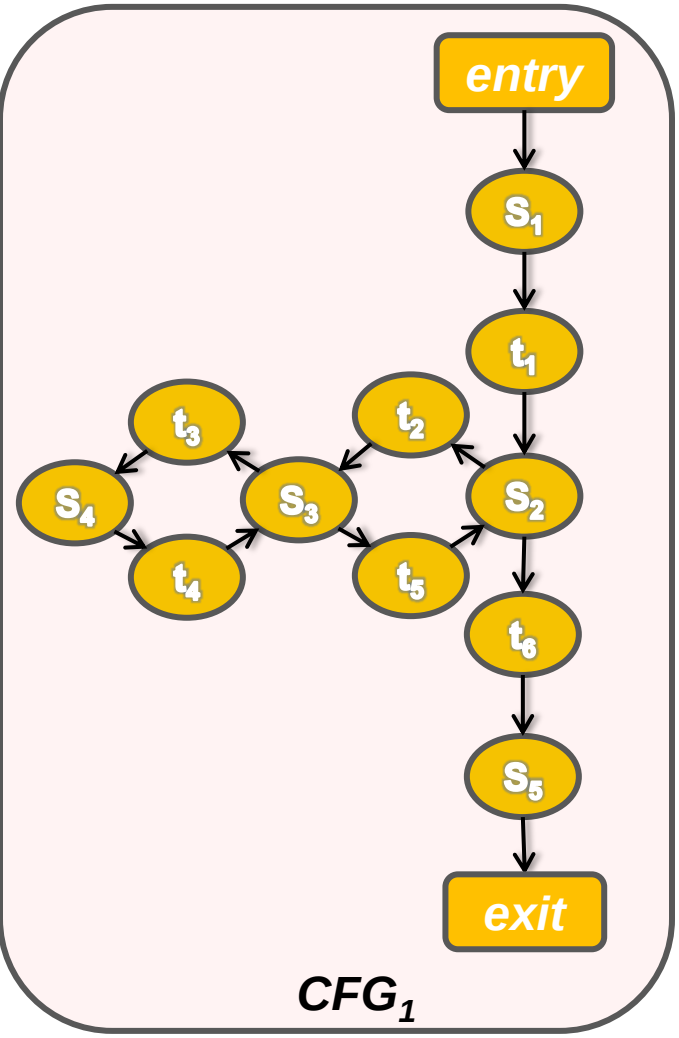


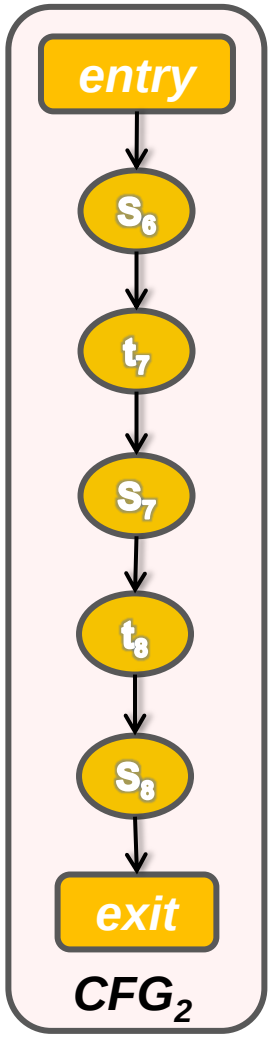
- ***Two transitions are X-Compatible if they are concurrent and mutually inclusive.***
  - *t_1 and t_1' are X-Compatible*

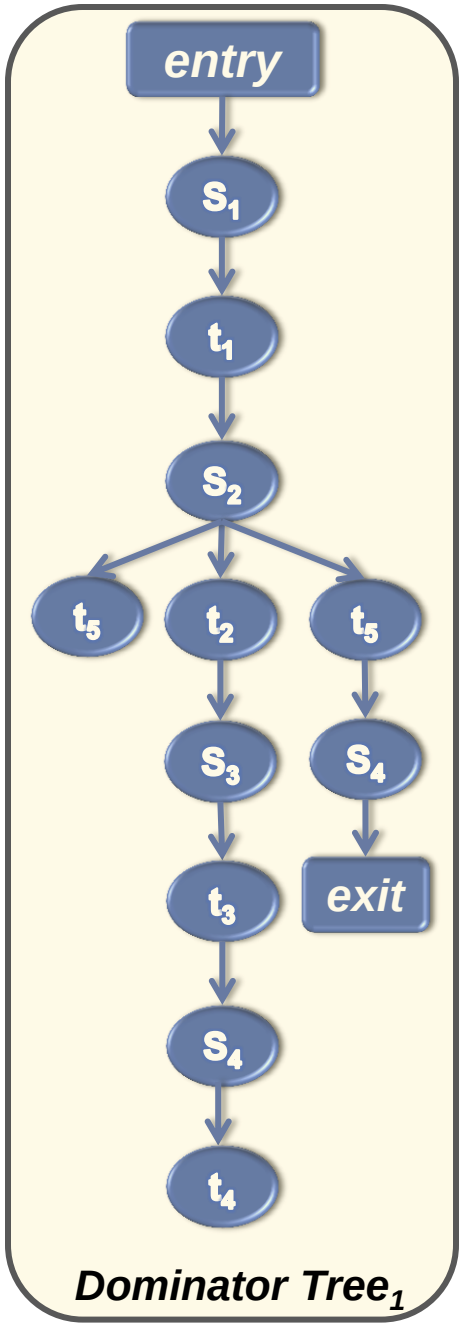
X-COMPATIBLES EXTRACTION

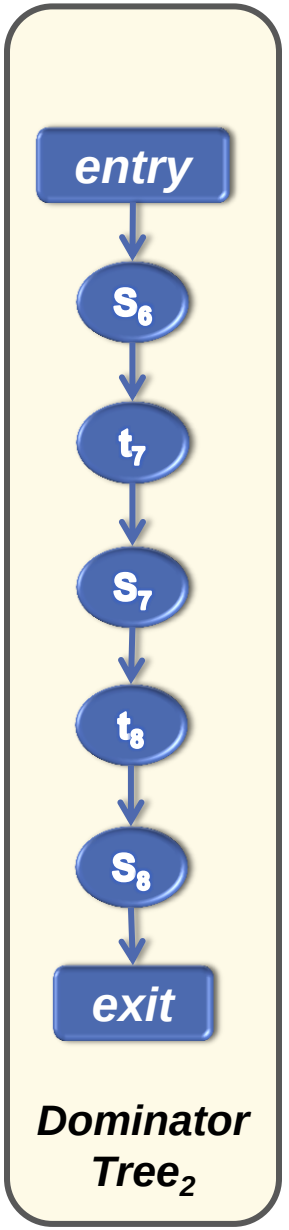


X-COMPATIBLES MINIMIZATION

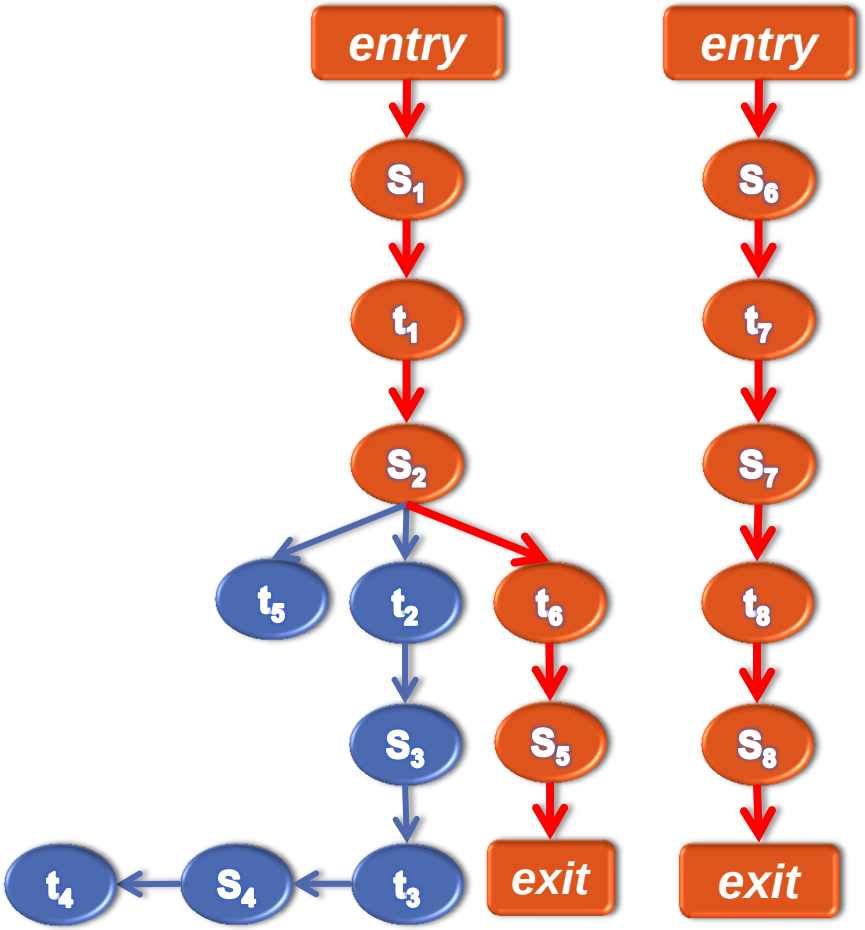






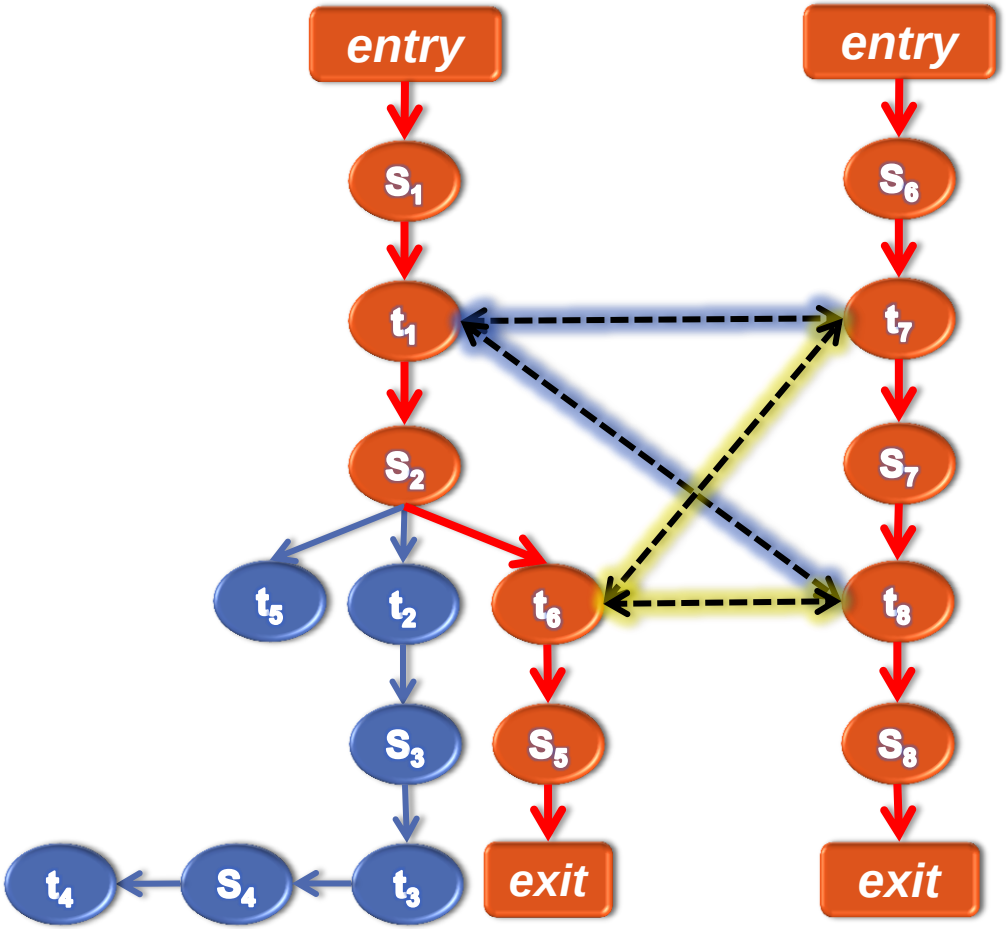


X-COMPATIBLES MINIMIZATION



DFS From entry to exit

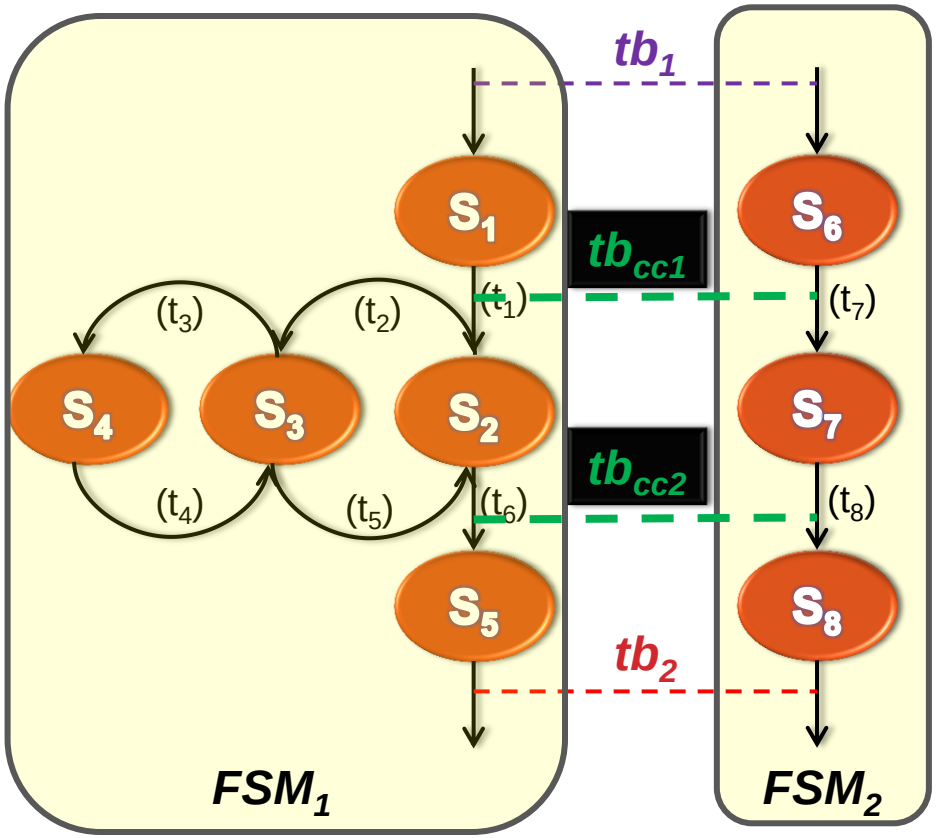
X-COMPATIBLES MINIMIZATION



DFS From entry to exit

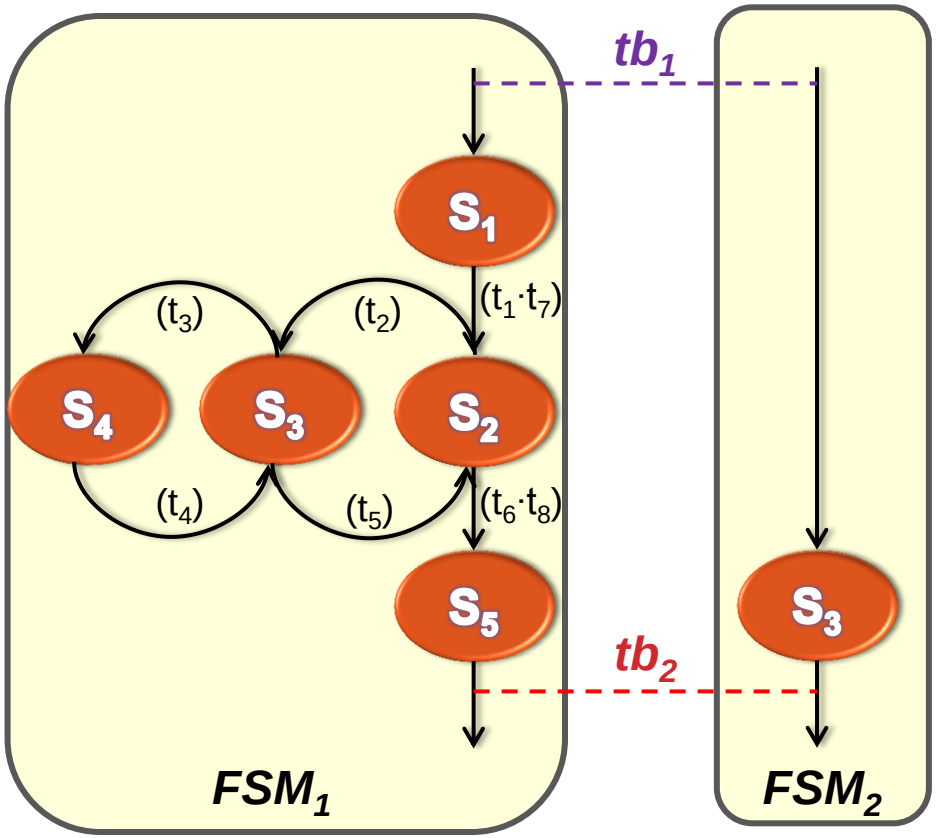
Form X-Compatibles

X-COMPATIBLES MINIMIZATION



- *DFS From entry to exit*
- *Form X-Compatibles*
- *Synchronize*

X-COMPATIBLES MINIMIZATION

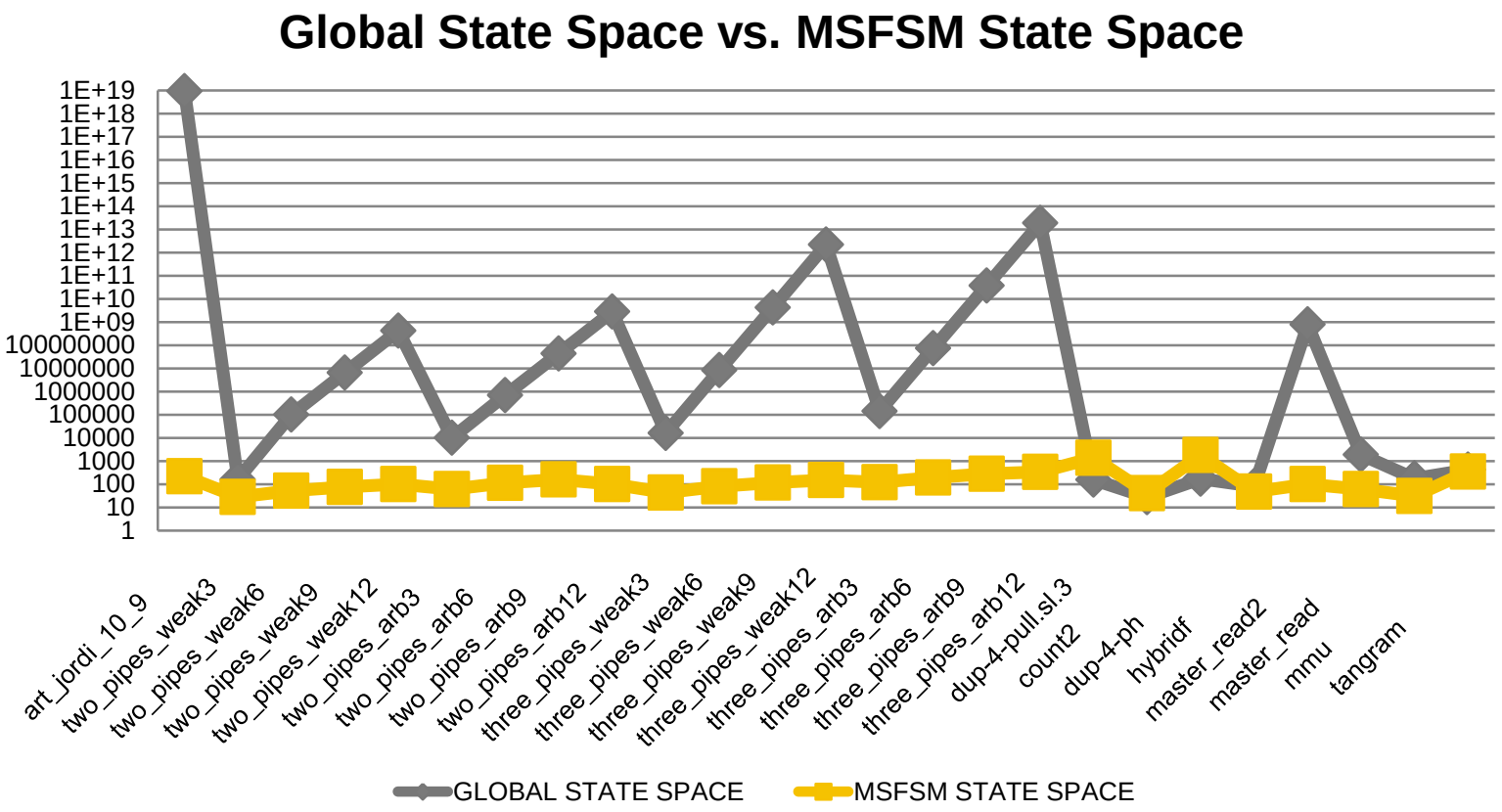


- DFS From entry to exit
- Form X-Compatibles
- Synchronize
- Minimize

OVERVIEW

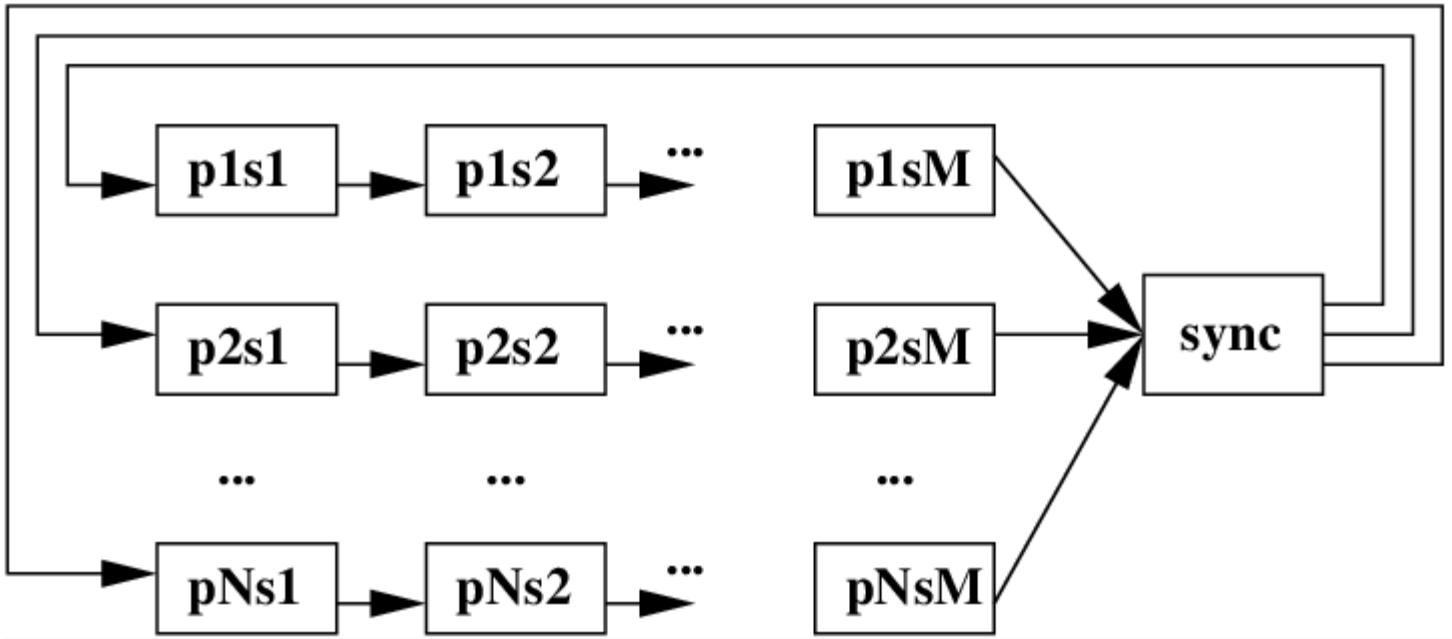
- Motivation
- Background
- MSFSMs
- Synthesis
- Verification
- Optimization
- ***Results***
- Conclusions and Future Work

RESULTS



- *Transforming a PTNet to MSFSMs and then to a synthesizable equivalent does **not** suffer from the state explosion issue.*

ASYNCHRONOUS SYNTHESIS SYNTHETIC BENCHMARK

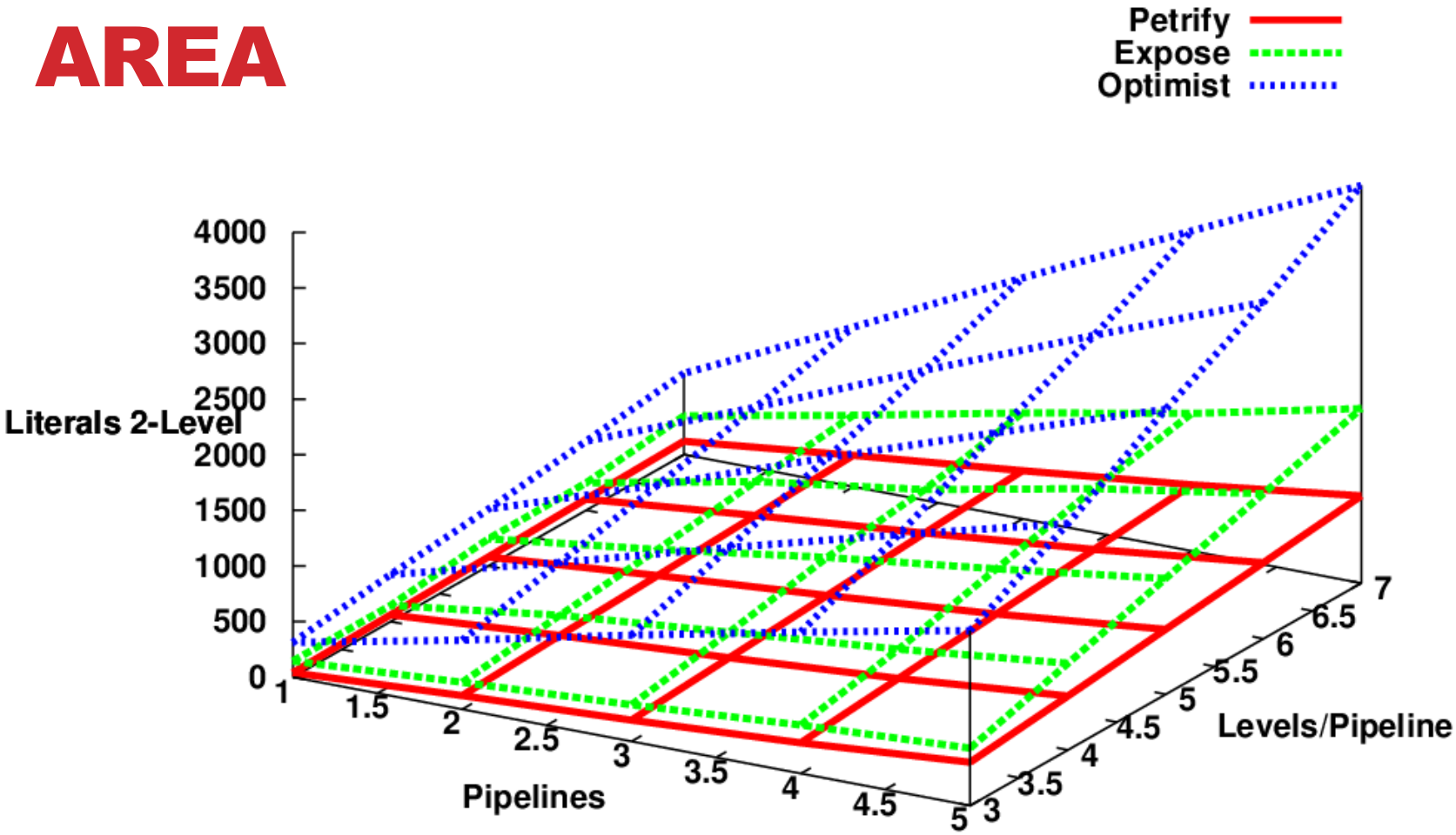


N Parallel Handshake Controllers

M Sequential Controllers Per Pipeline

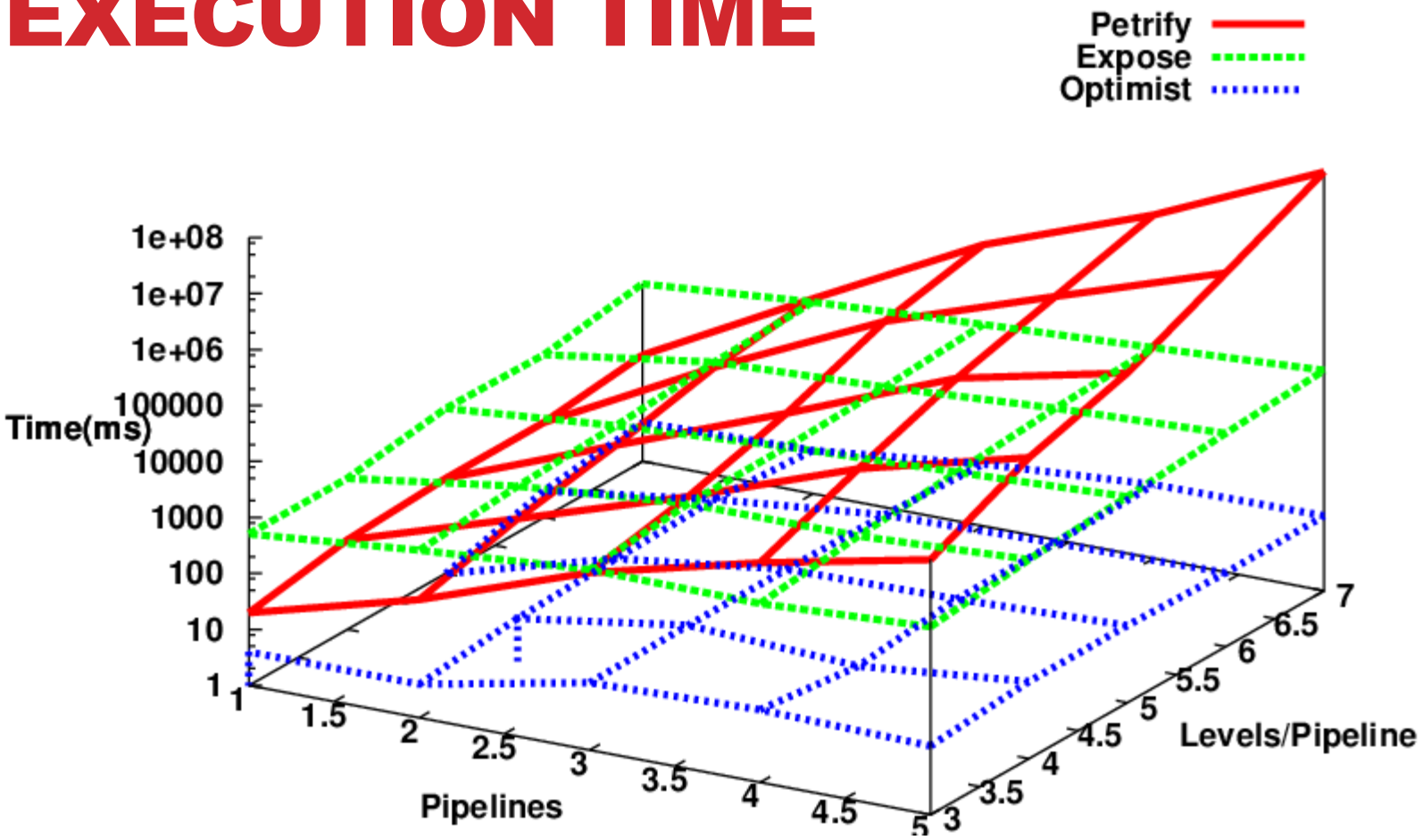
Synchronized at the M stage

AREA



- ***Expose*** *unveils the solution space between the direct mapping approaches and the ones which require the complete state space*

EXECUTION TIME

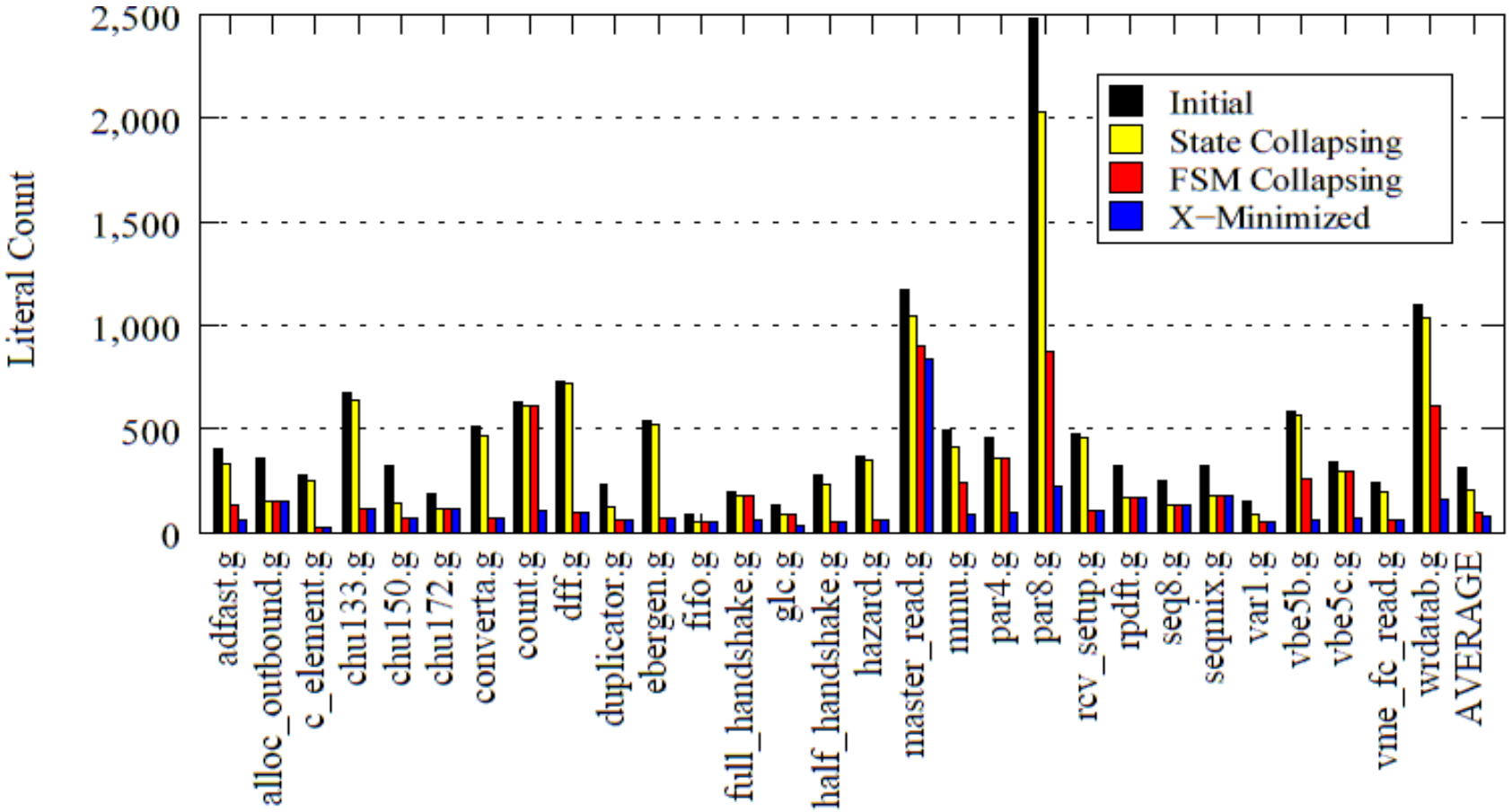


- ***Expose*** *execution time scales with the size of the initial specification*

CONTROL CIRCUITS AREA

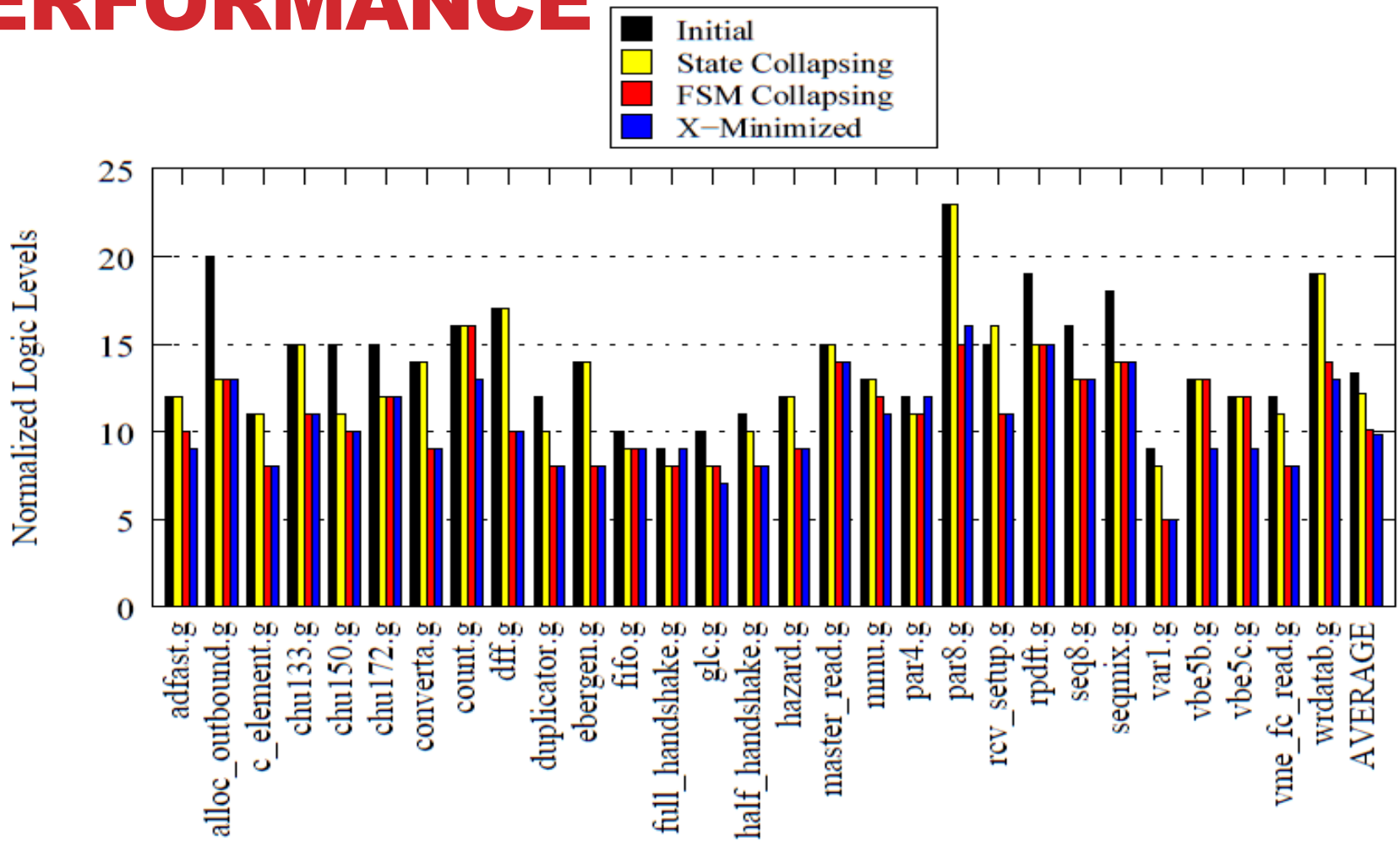
Benchmark	Petrify	Optimist	Expose
alloc-outbound	*66*	*258*	*30*
c3	*12*	*198*	*13*
count2	*Irresolvable CSC*	*302*	*178*
dff	*44*	*304*	*20*
duplicator	*93*	*228*	*111*
full	*44*	*264*	*102*
half	*43*	*198*	*148*
monkey	*N/A*	*1148*	*181*
rpdft	*34*	*214*	*25*
semi-decoupled	*86*	*242*	*208*
vbe6a	*132*	*732*	*237*

OPTIMIZATION AREA



- *Applying the introduced transformations at the MSFSM level predictably reduces area.*

OPTIMIZATION PERFORMANCE



- *Applying the introduced transformations at the MSFSM level typically increases performance.*

OVERVIEW

- Motivation
- Background
- MSFSMs
- Synthesis
- Verification
- Optimization
- Results
- ***Conclusions and Future Work***

CONCLUSIONS

Novel Control Specification Model, MSFSMs

- **Key for Concurrent Specifications Synthesis and Verification**

Logic Synthesis Tool, Expose, Targeting Synchronous and Asynchronous Logic

Optimization Operations with Predictable QoR

FUTURE WORK

Logic Synthesis of Mixed Synchronous and Asynchronous Circuits

Examination of the Concurrency / Area Trade-Off Unveiled by the X-Compatibles Optimization

Expansion of the supported Asynchronous Timing Models

- **Speed Independent**