Bridging the Design Methodologies of Burst-Mode Specifications and Signal Transition Graphs

Alex Chan[†], Danil Sokolov[‡], Victor Khomenko[‡], Alex Yakovlev[†]

[†]Newcastle University, United Kingdom

[‡]Dialog Semiconductor, United Kingdom

Newcastle University

dialog SEMICONDUCTOR

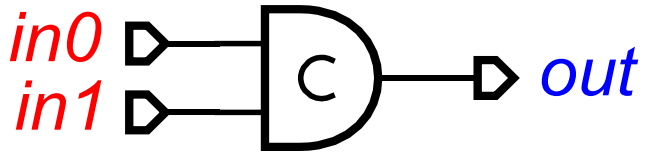
Workcraft

Outline

- Briefly cover Burst-Mode Specifications (BMs) and Signal Transition Graphs (STGs).
- Review current design methodologies with BMs and STGs.
- Motivating example involving a handshake decoupler.
- Propose a ‘co-design’ methodology using Burst Automata (BAs).
- Highlight the benefits of our ‘co-design’ methodology with a snippet of experimental results.

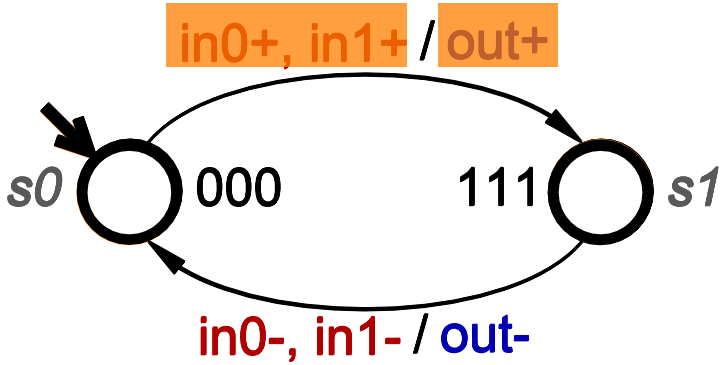
Introductory to BMs and STGs

C-element Gate



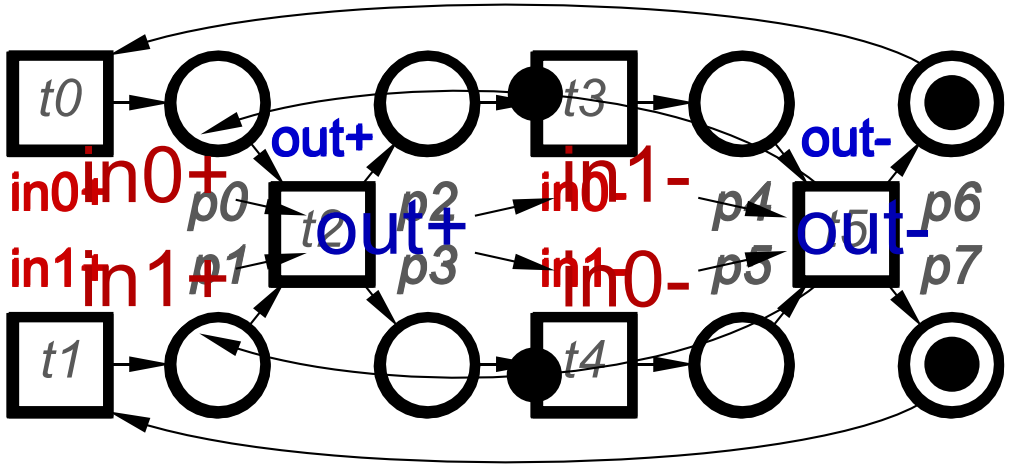
in0	in1	out
0	0	0
0	1	–
1	0	–
1	1	1

BM Model



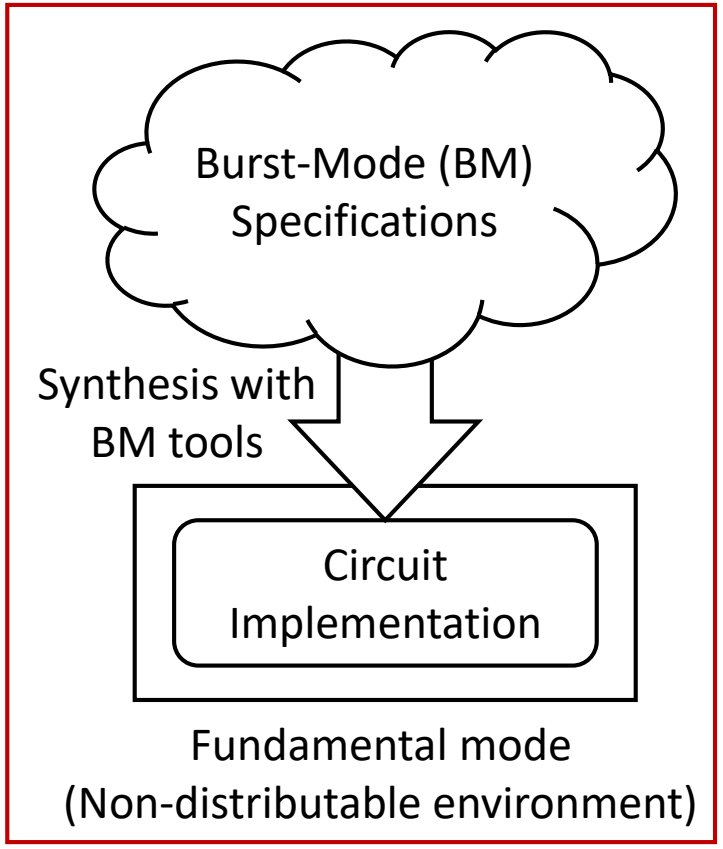
Encoding: in0 in1 out

STG Model

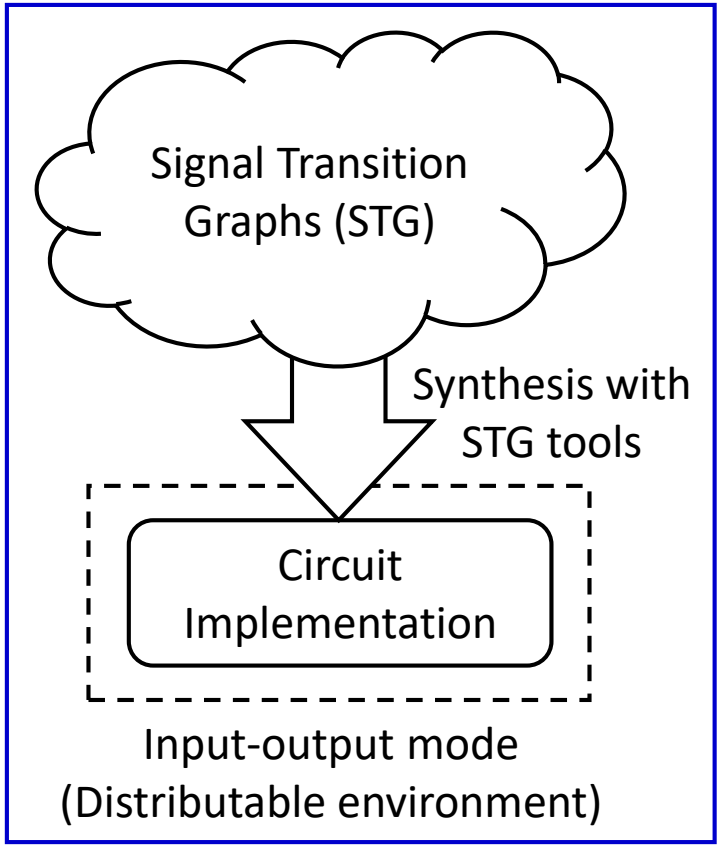


Current Design Methodologies

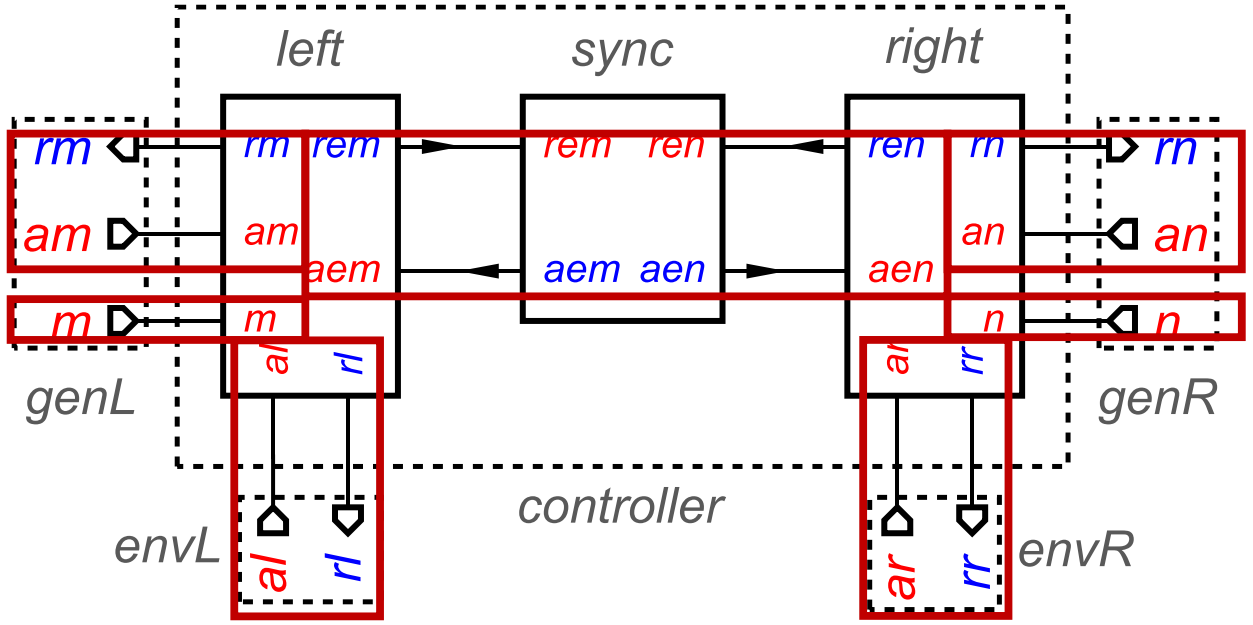
State-based methods, i.e.
the “legacy” route



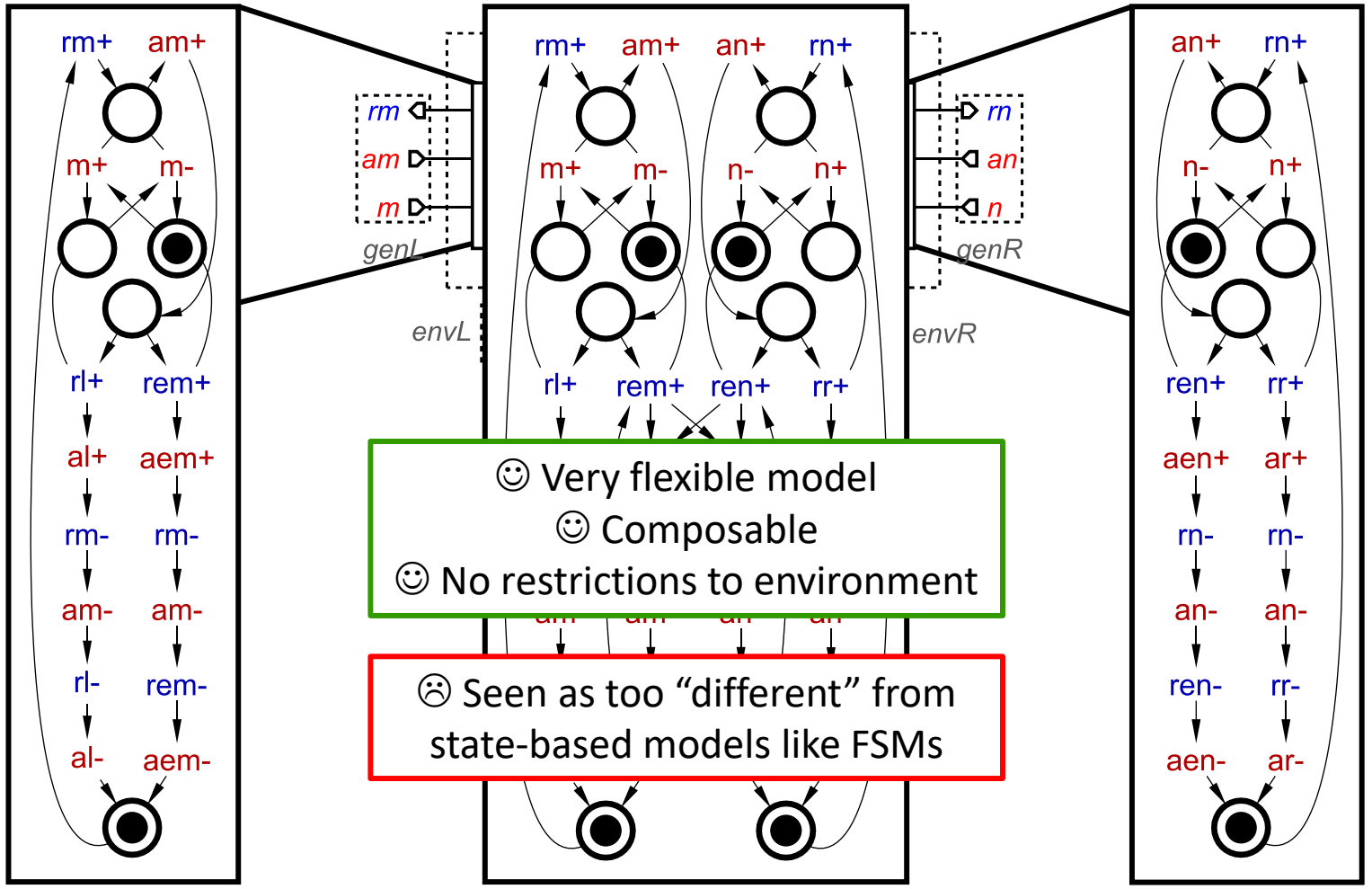
Event-based methods, i.e.
the “disruptive” Route



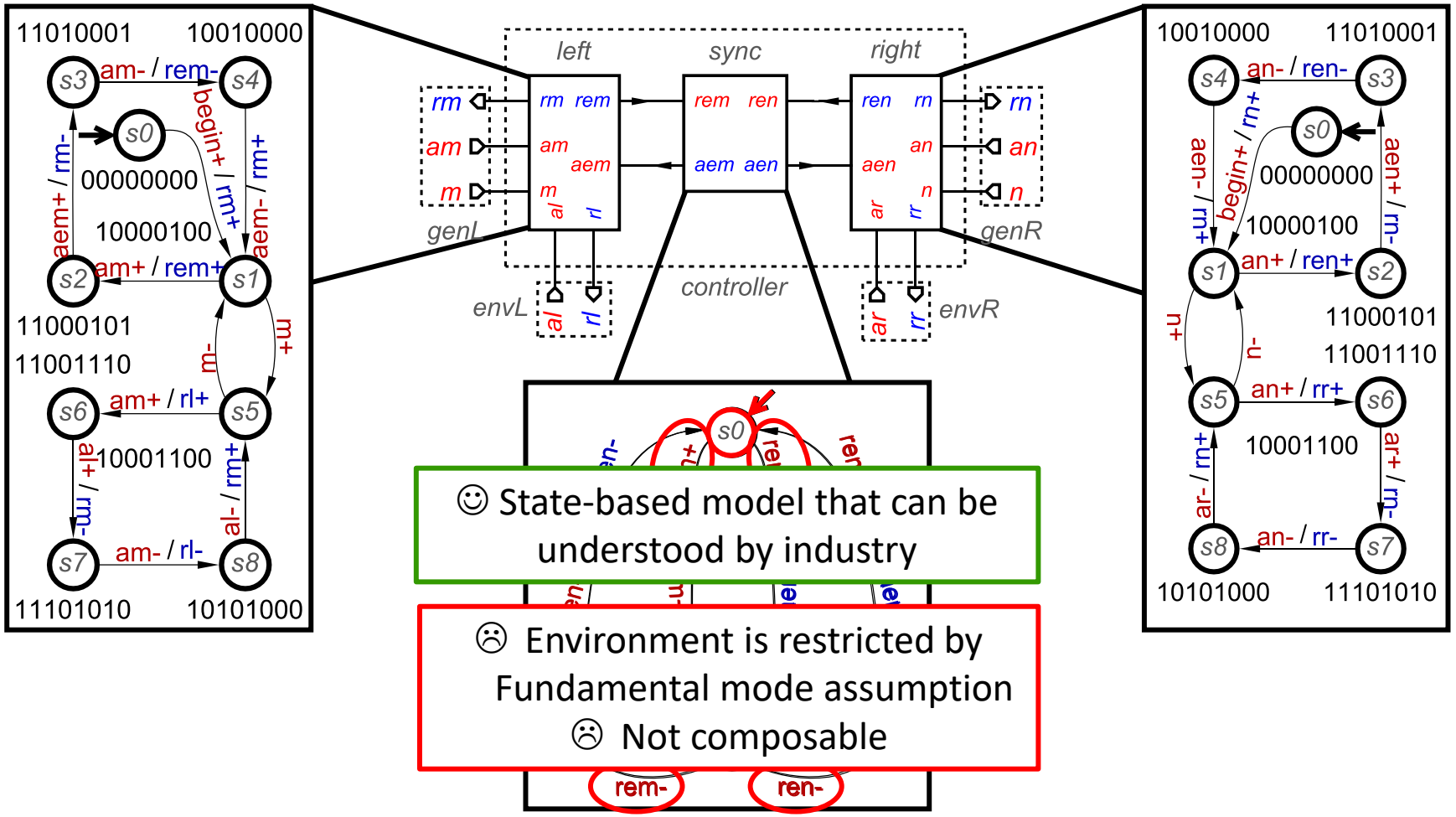
Handshake Decoupler Example



Example using STGs



Example using BMs

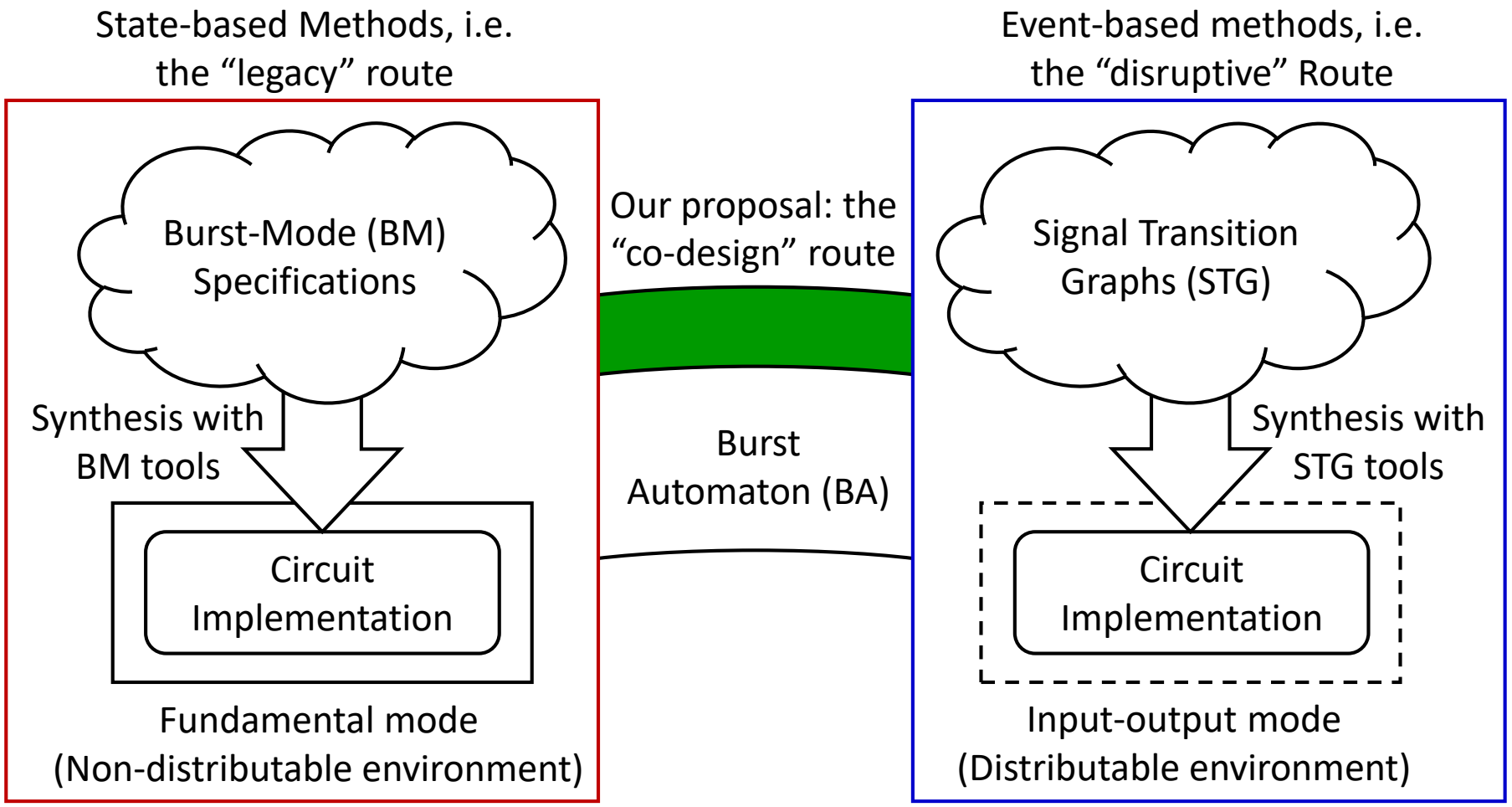


☺ State-based model that can be understood by industry

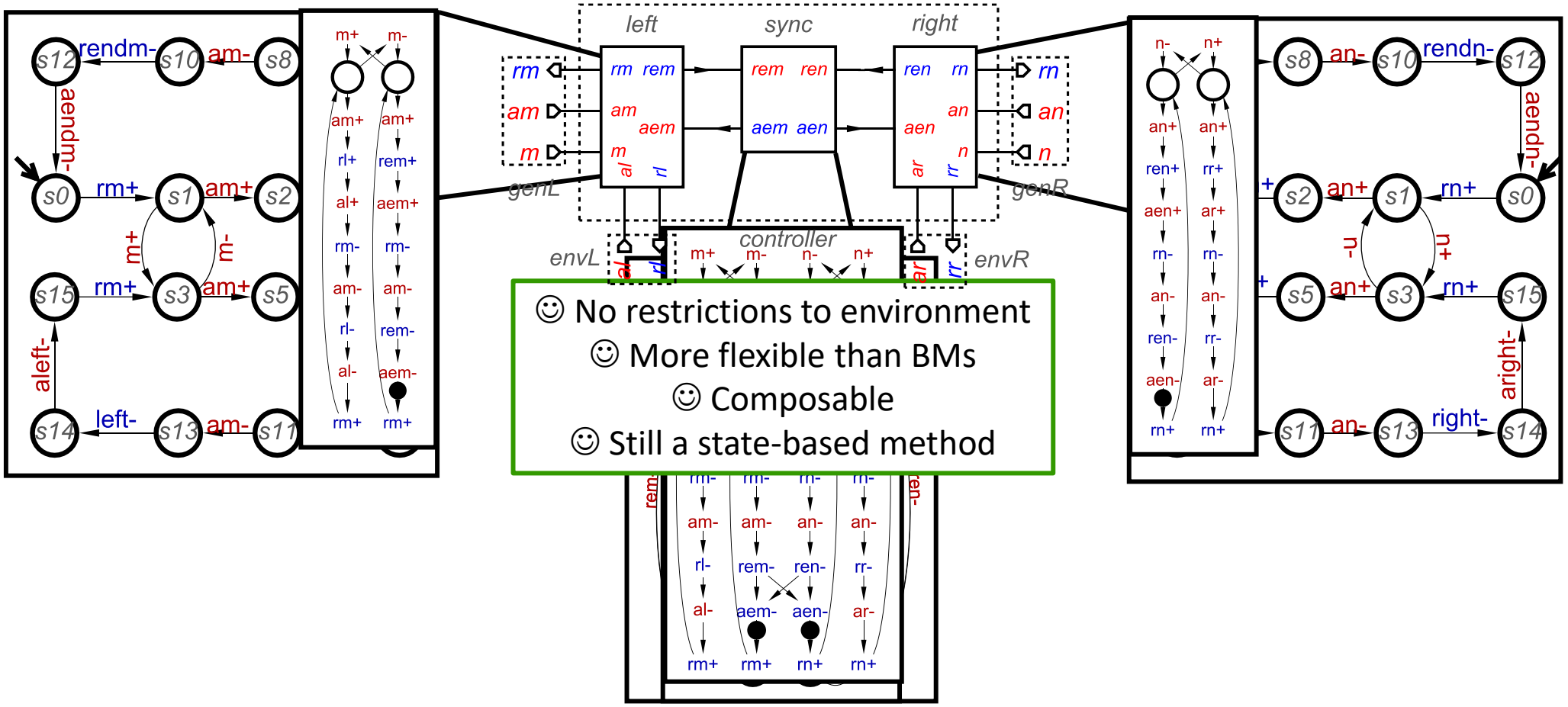
☹ Environment is restricted by Fundamental mode assumption

☹ Not composable

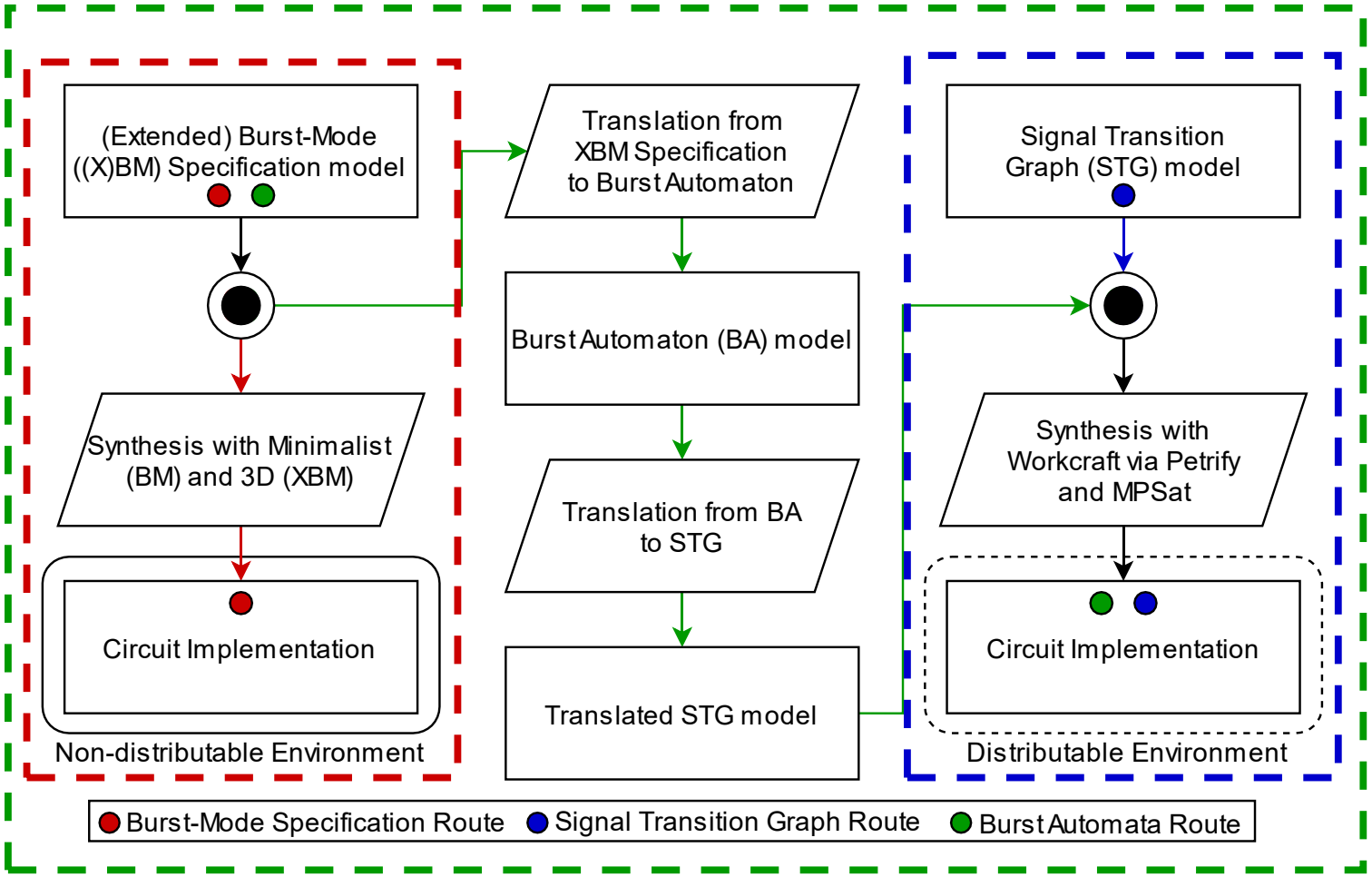
Towards a Co-design Methodology



Example using BAs



Co-design Methodology Flow



Snippet of Experimental Results

Specification	BA 'Co-Design' Options			
	BM Tools		STG Tools	
	Minimalist	3D	Petrify	MPSat
buck-no-zc	**4**	**4**	**4**	**4**
buck-late-zc	Failed (Inexpressable)	Failed (Inexpressable)	**4**	**4**
buck-early-zc	Failed (Inexpressable)	Failed (Inexpressable)	**7**	**7**
buck-composed	Failed (Inexpressable)	Failed (Inexpressable)	**7**	**7**
gcd-controller	Failed (Inexpressable)	177	Failed (CSC)	**92**
stetson-p1	**230**	Failed (Error)	Failed (CSC)	Failed (CSC)
stetson-p2	**141**	Failed (Error)	Failed (CSC)	Failed (CSC)
stetson-p3	13	11	**6**	**6**
token-distributor	42	39	28	**26**

Summary

- Identify some issues with the “legacy” and “disruptive” design methodologies and address them with our “co-design” methodology.
- BMs are a state-based method, but their environment is restrictive, and they cannot be composed.
- STGs are flexible, their environment is non-restrictive, and are easily composable, but they are an event-based method.
- BAs takes the best of both by bridging the gap between BMs and STGs, and enables a design flow where both models can leverage the other’s benefits.
- Experimental results show how our “co-design” methodology enables collaboration between BMs and STGs, with ability to fallback to either model’s set of tools if the other fails.

Thank you!

Any questions?