Automatic Generation of Cycle Accurate and Cycle Count Accurate Transaction Level Bus Models from a Formal Model

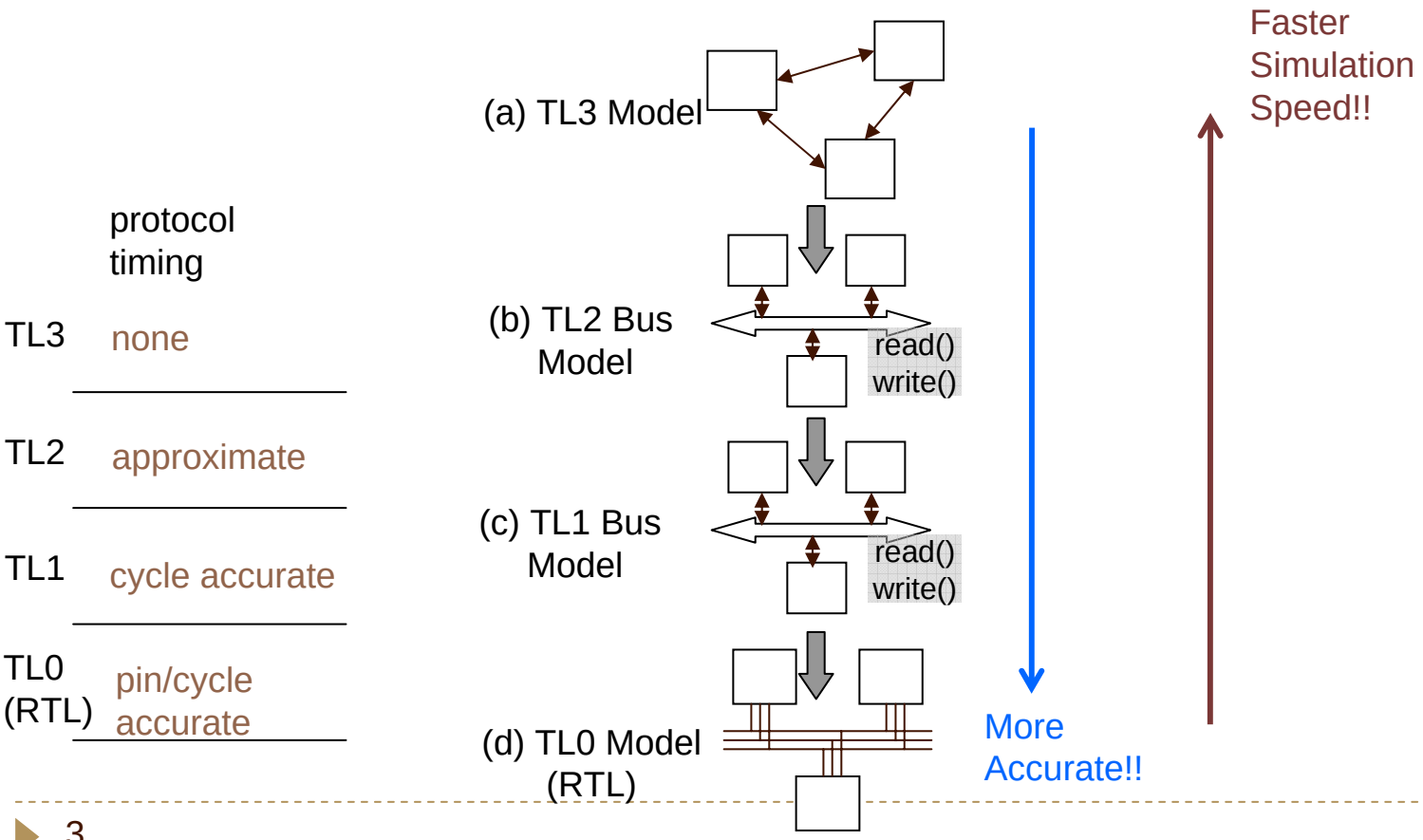
Chen Kang Lo, Ren Song Tsay,
Logos Advanced System Lab,
National Tsing-Hua University, Hsinchu, Taiwan

Outline

- Introduction
- Related Work
- Problem Formulation
- Transaction Level Bus Model Generation
- Experimental Results
- Conclusion and Future Work

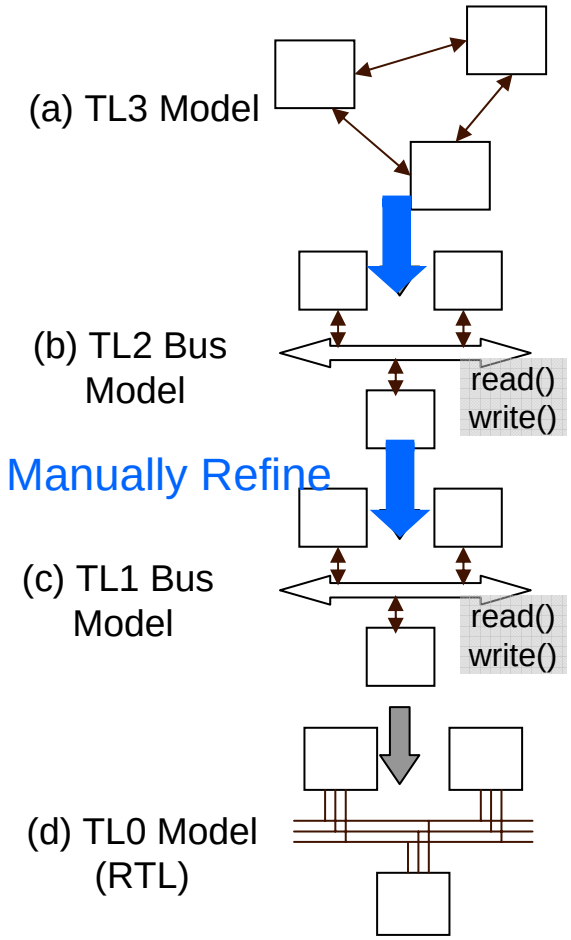
Introduction to TLM

- Transaction Level Modeling (TLM) is proposed to perform architecture exploration and verification.



Problems of Manual Refinement

	protocol timing
TL3	none
TL2	approximate
TL1	cycle accurate
TL0 (RTL)	pin/cycle accurate

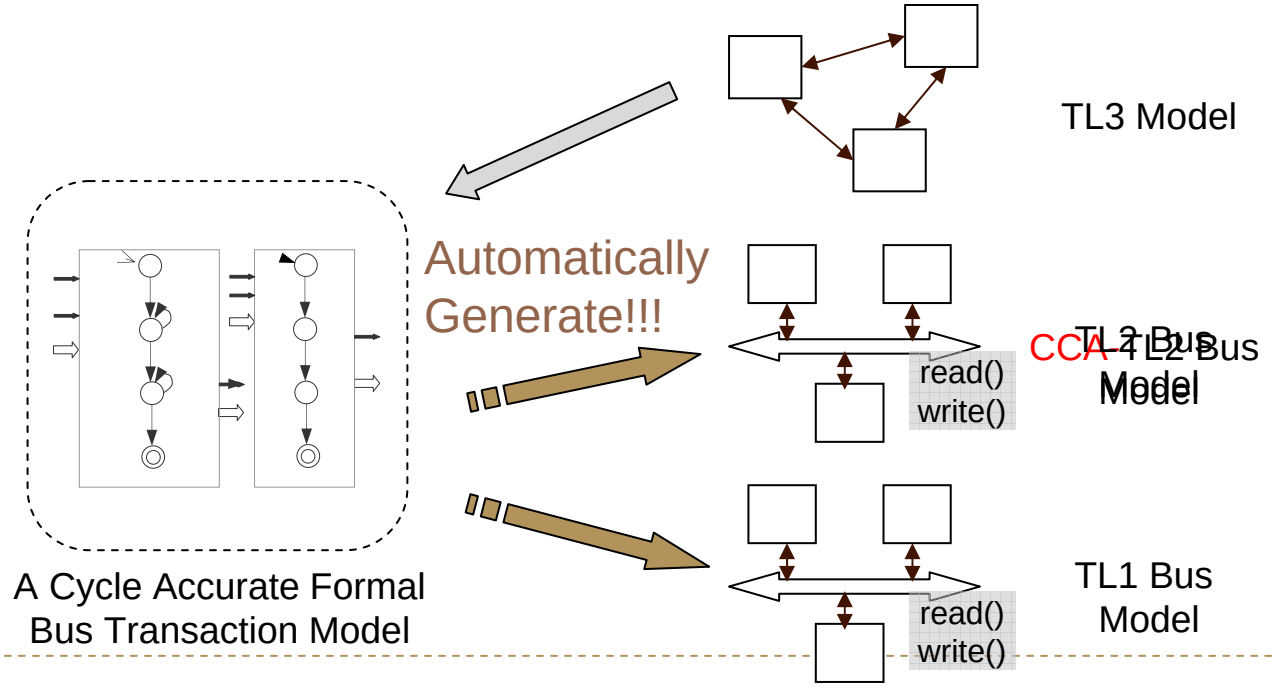


Difficult Consistency Checking!!

Tedious and Error-prone Re-modeling!!

Proposed Automatic Approach

- An automatic approach to generate:
  - TL1 (Cycle Accurate) model
  - Cycle Count Accurate TL2 model (CCA-TL2)



Outline

- Introduction
- Related Work
- Problem Formulation
- Transaction Level Bus Model Generation
- Experimental Results
- Conclusion and Future Work

Related Work

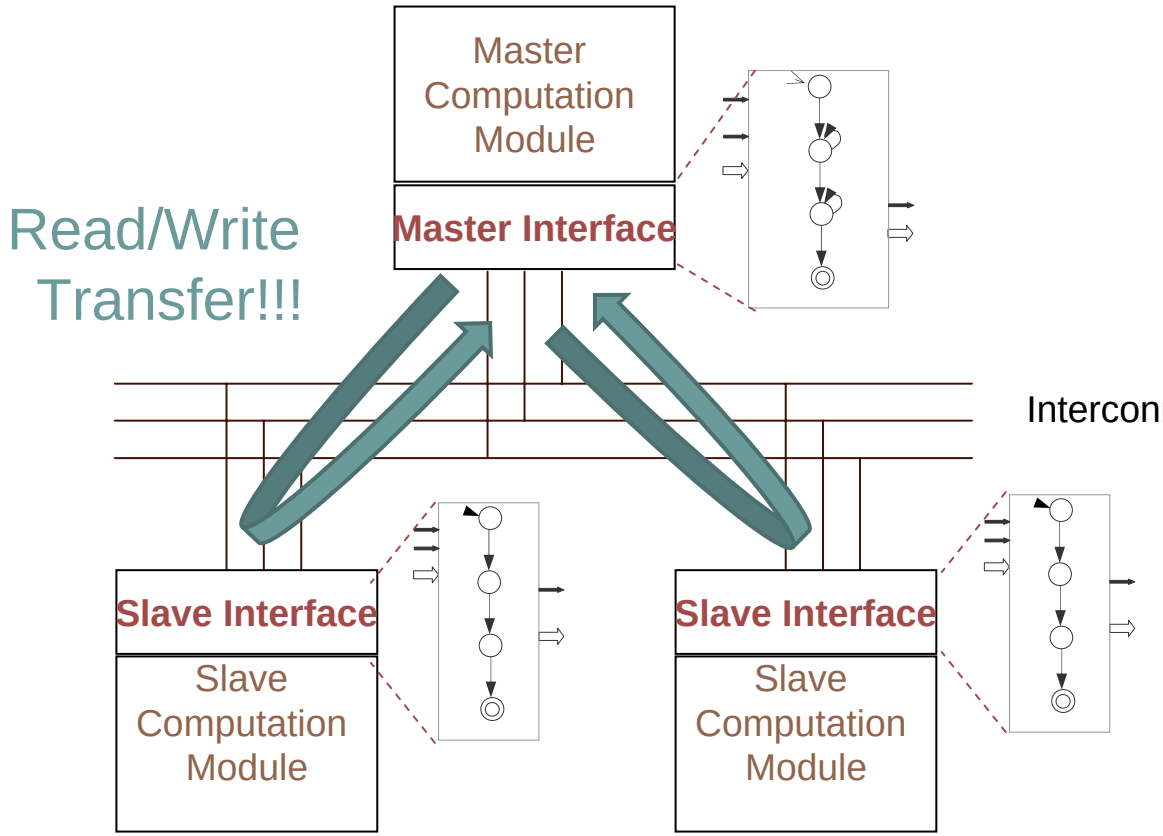
- Manual Modeling Techniques
  - M. Caldari, et al., "Transaction-level models for AMBA bus architecture using SystemC 2.0", *DATE'03*.
  - S. Pasricha, N. Dutt, M. Ben-Romdhane, "Extending the Transaction Level Modeling Approach for Fast Communication Architecture Exploration", *DAC'04*.
    - Cycle Count Accurate at Transaction Boundary (CCATB)
- Library Based Approaches
  - A. Harverinen, M. Leclercq, N. Weyrich, D. Wingard, "A SystemC™ OCP Transaction Level Communication Channel", Technical Report'07.

Outline

- Introduction
- Related Work
- Problem Formulation
- Transaction Level Bus Model Generation
- Experimental Results
- Conclusion and Future Work

Bus transaction formal modeling

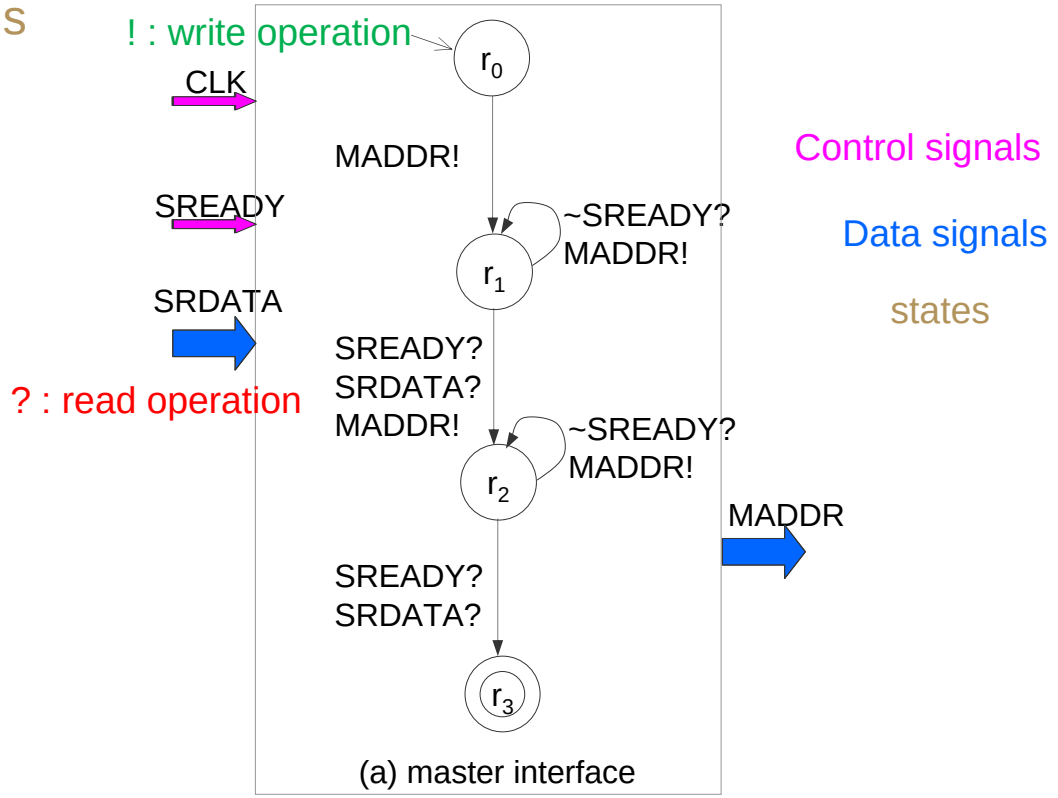
- A bus transaction is a read/write transfer between master and slave computation modules.



A Bus transaction Formal Model

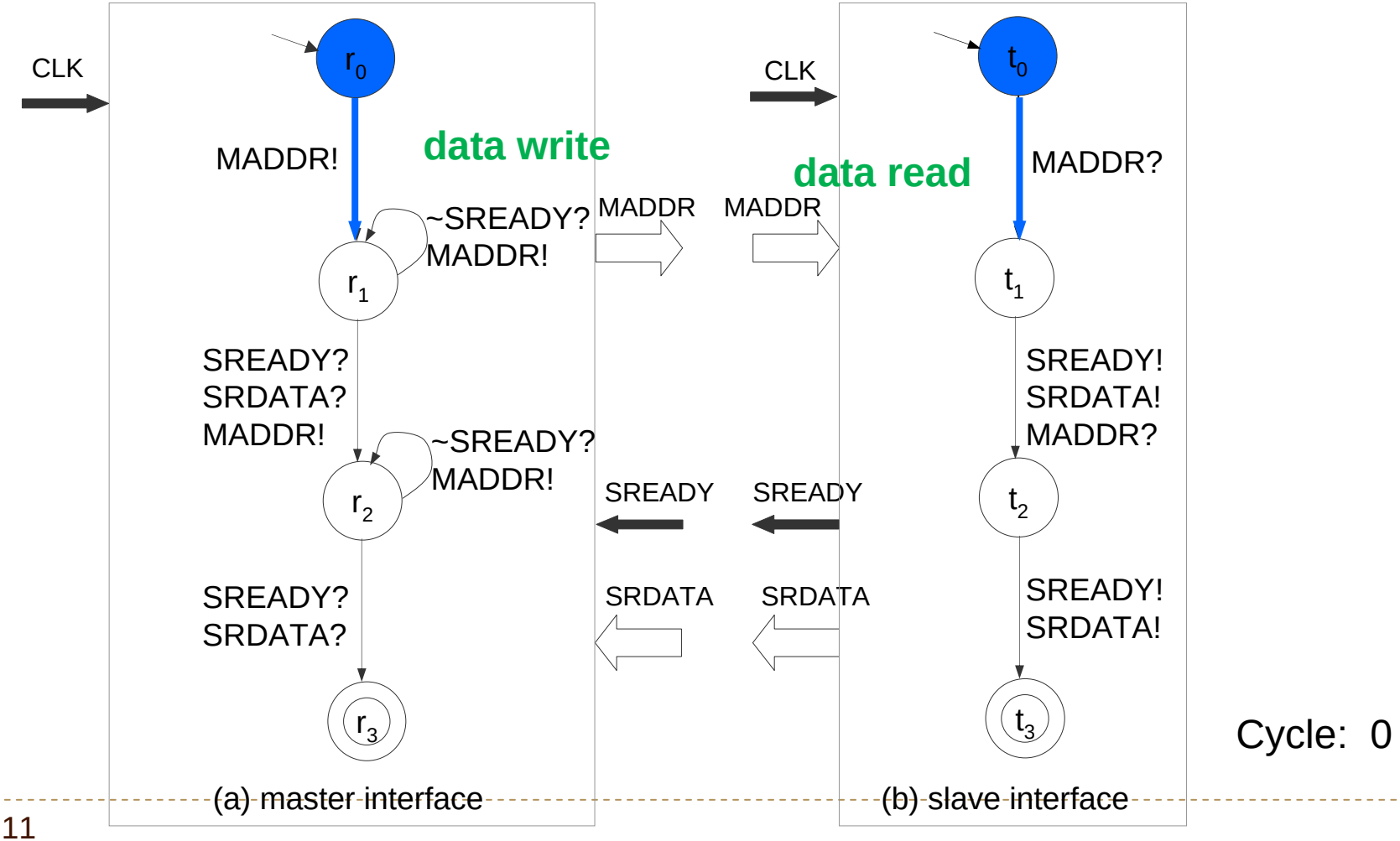
Synchronous Protocol Automaton

- A SPA is a state machine designed for bus modeling.
  - with data and control signals
  - whose state progressing is synchronous with clock



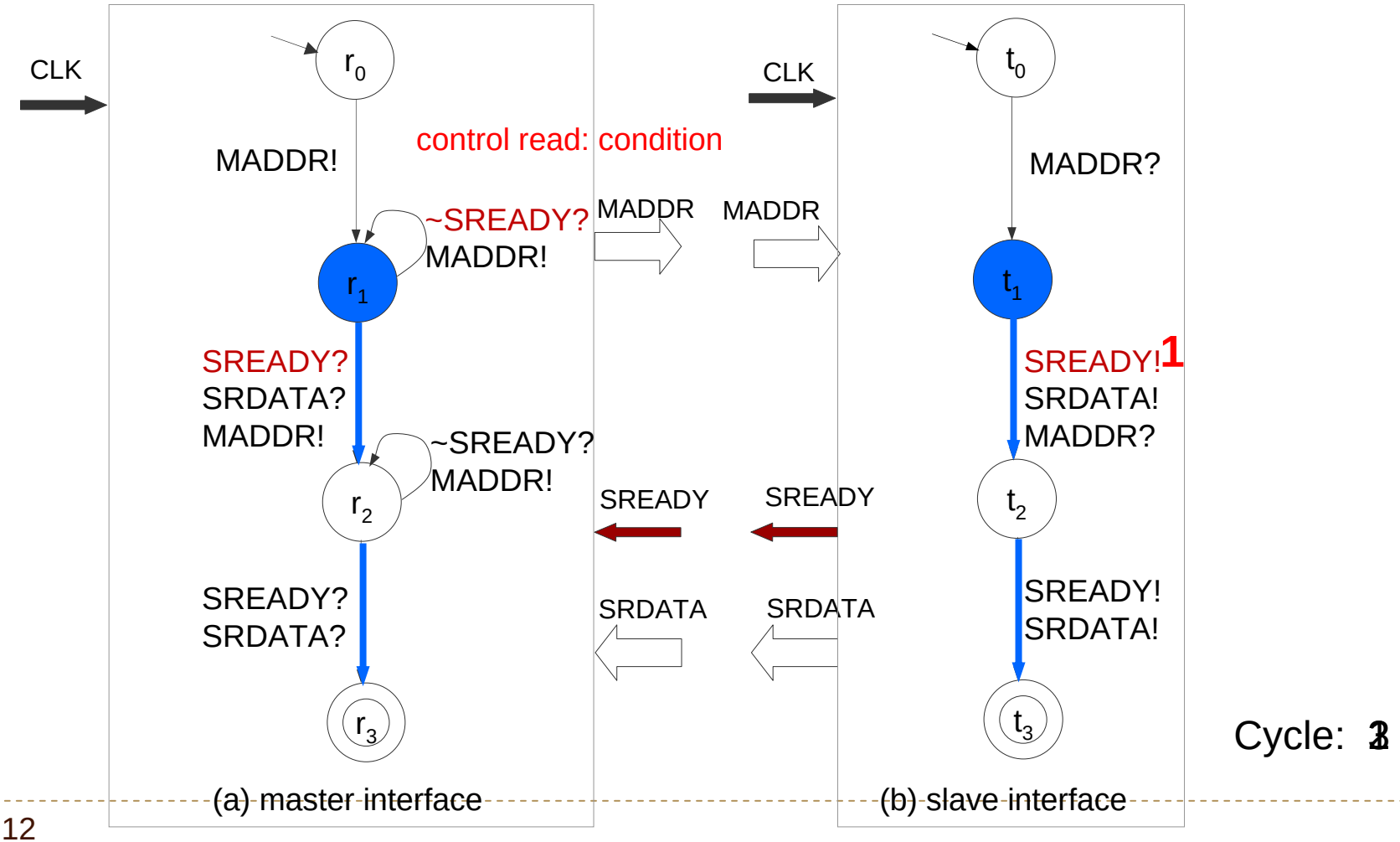
V. D'silva, et al., "Synchronous Protocol Automata: A Framework for Modeling and Verification of SoC Communication Architectures", *DATE, 2004*

A Burst Transaction Modeled by a SPA-pair



Cycle: 0

A Burst Transaction Modeled by a SPA-pair

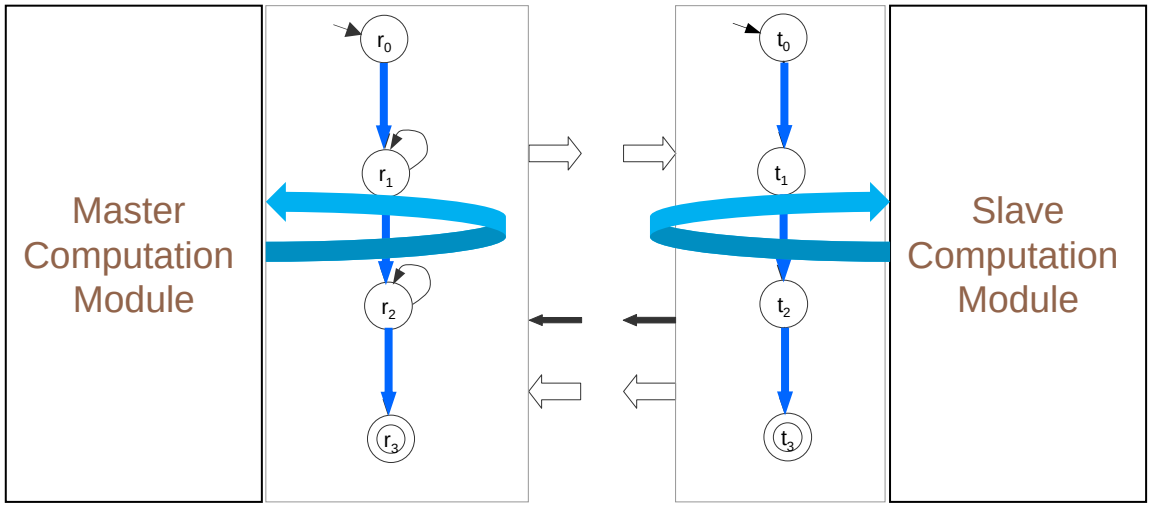


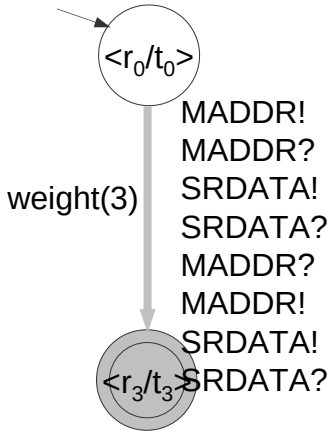
Cycle: 1 2 3

Observations for Abstracting CA to CCA Model

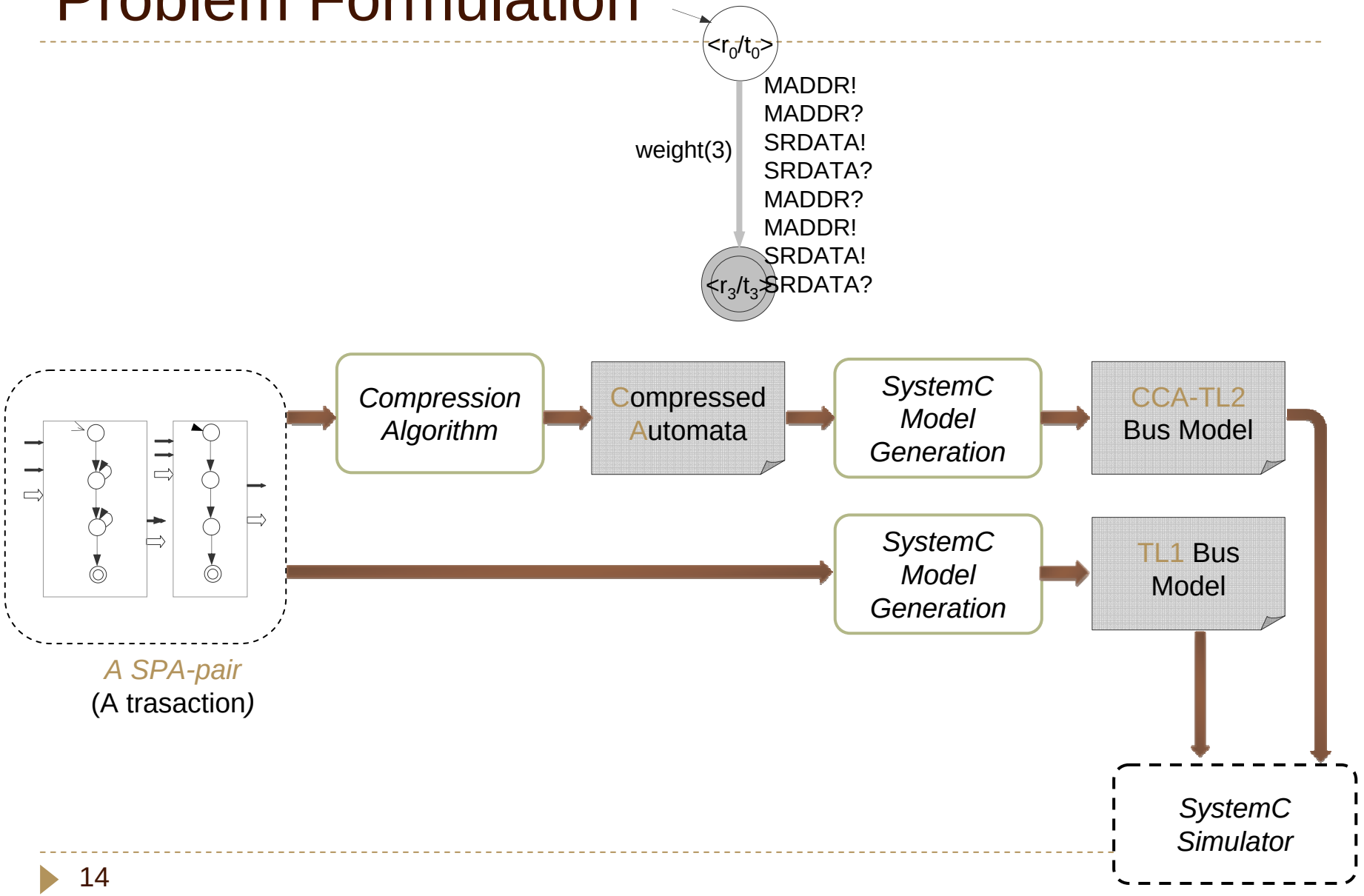
1. Most transitions of the SPA-pair can be *pre-determined* before simulation (at static time).
2. A computation module concerns only with data content transferred.

- Reduce the simulation overhead.





Problem Formulation

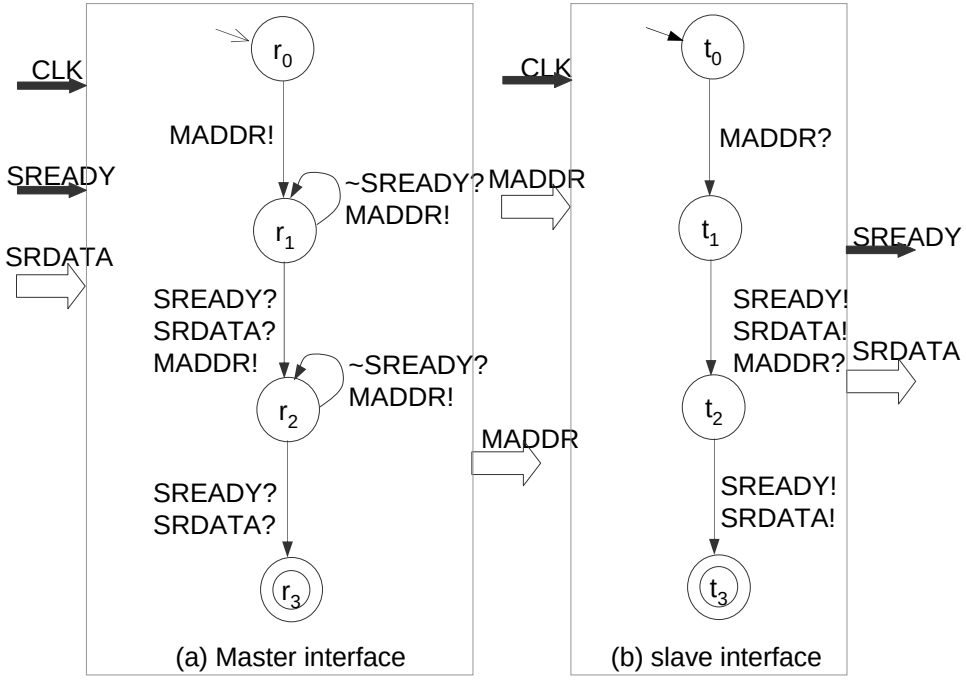


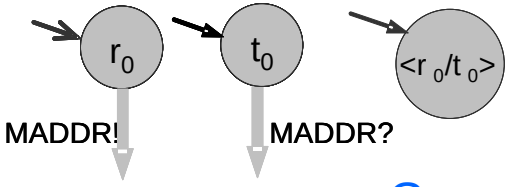
Outline

- Introduction
- Related Work
- Problem Formulation
- Transaction Level Bus Model Generation
  - Compression Algorithm
  - SystemC Bus Model Generation
- Experimental Result
- Conclusion and Future Work

Compression Algorithm

- Traces of the compression algorithm with an example with predetermined transitions:

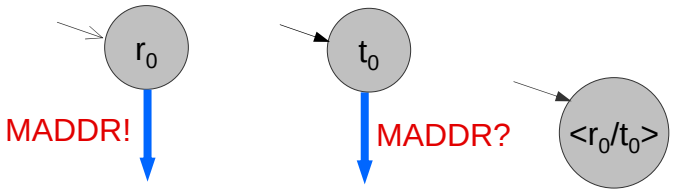




Generated Compressed Automaton

Compression Algorithm

- Traces of the compression algorithm with an example with predetermined transitions:



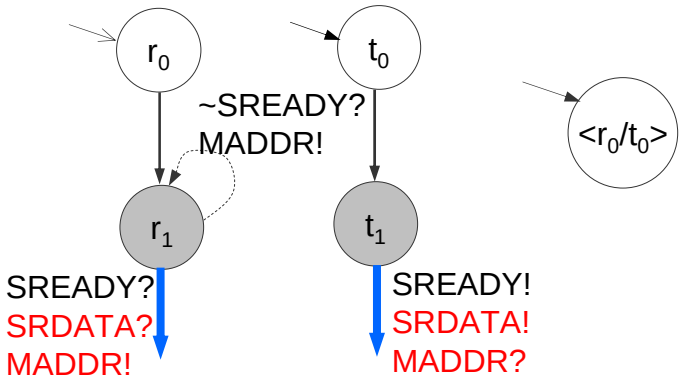
Collected data operation:

MADDR!
MADDR?

Weight: 1

Compression Algorithm

- Traces of the compression algorithm with an example with predetermined transitions:



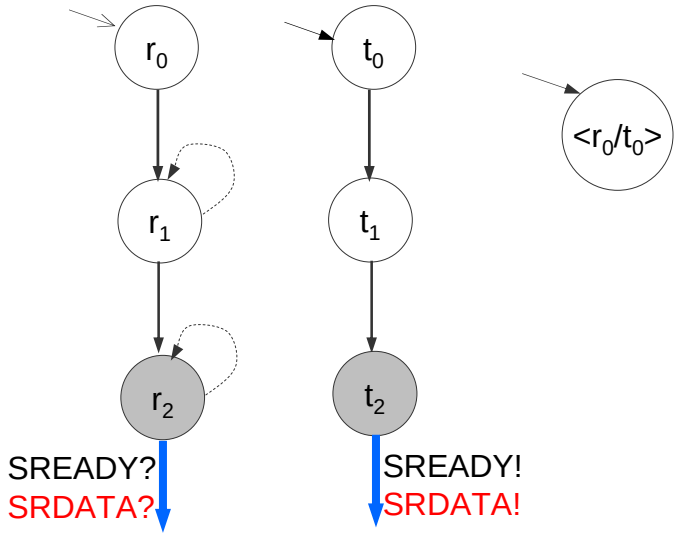
Collected data operation:

MADDR!
MADDR?
SRDATA!
SRDATA?
MADDR!
MADDR?

Weight: 2

Compression Algorithm

- Traces of the compression algorithm with an example with predetermined transitions:



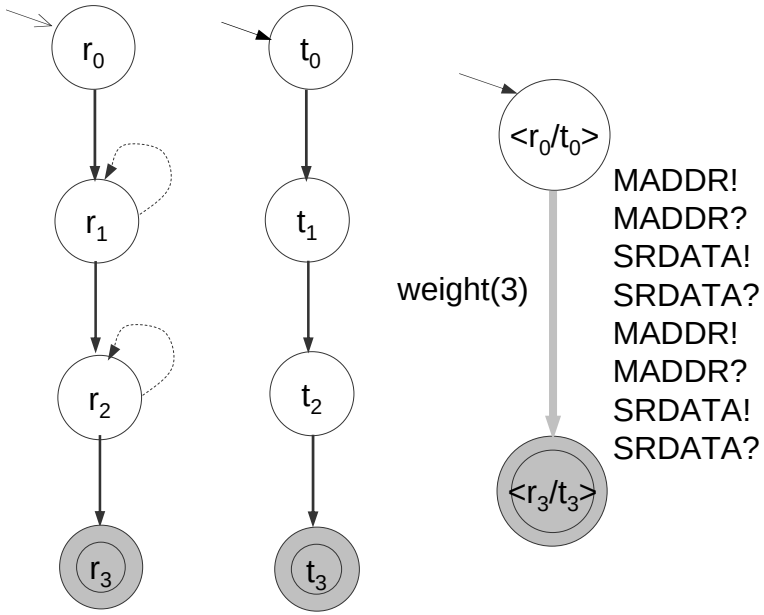
Collected data operation:

MADDR!
MADDR?
SRDATA!
SRDATA?
MADDR!
MADDR?
SRDATA!
SRDATA?

Weight: 3

Compression Algorithm

- Traces of the compression algorithm with an example with predetermined transitions:



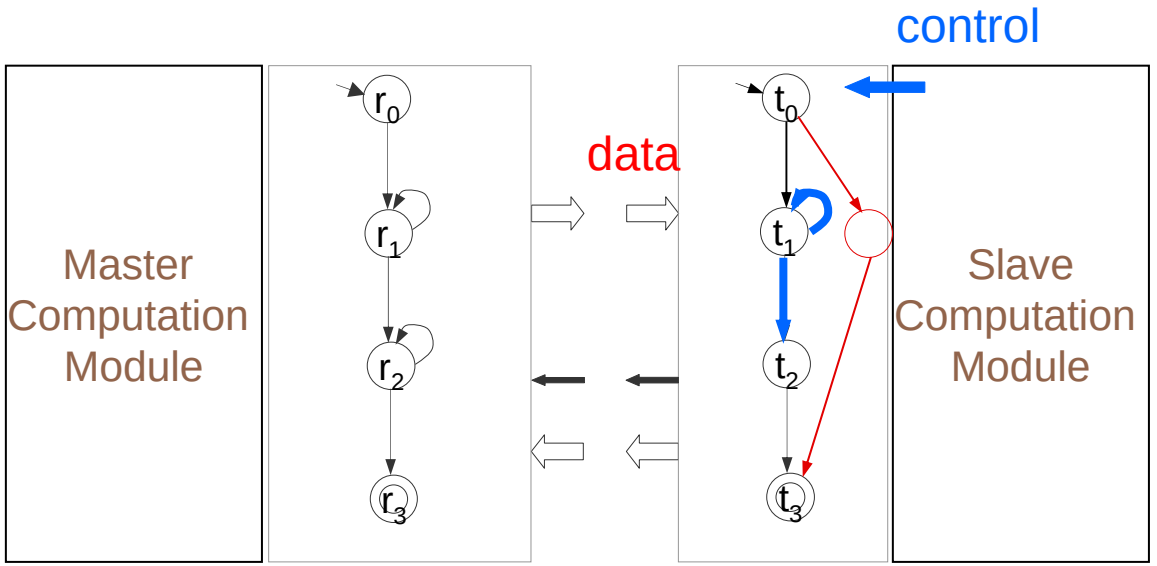
Collected data operation:

MADDR!
MADDR?
SRDATA!
SRDATA?
MADDR!
MADDR?
SRDATA!
SRDATA?

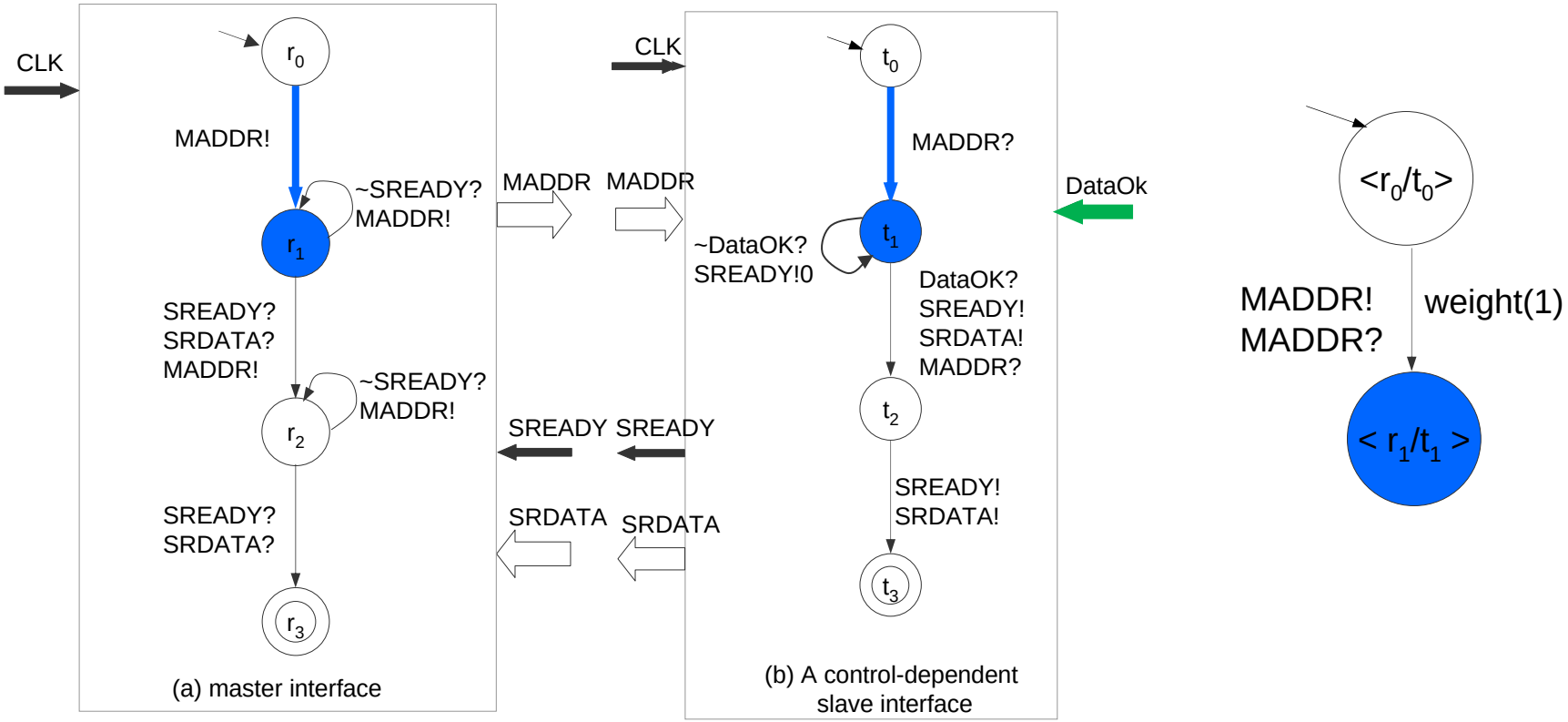
Weight: 3

Non-predetermined transition

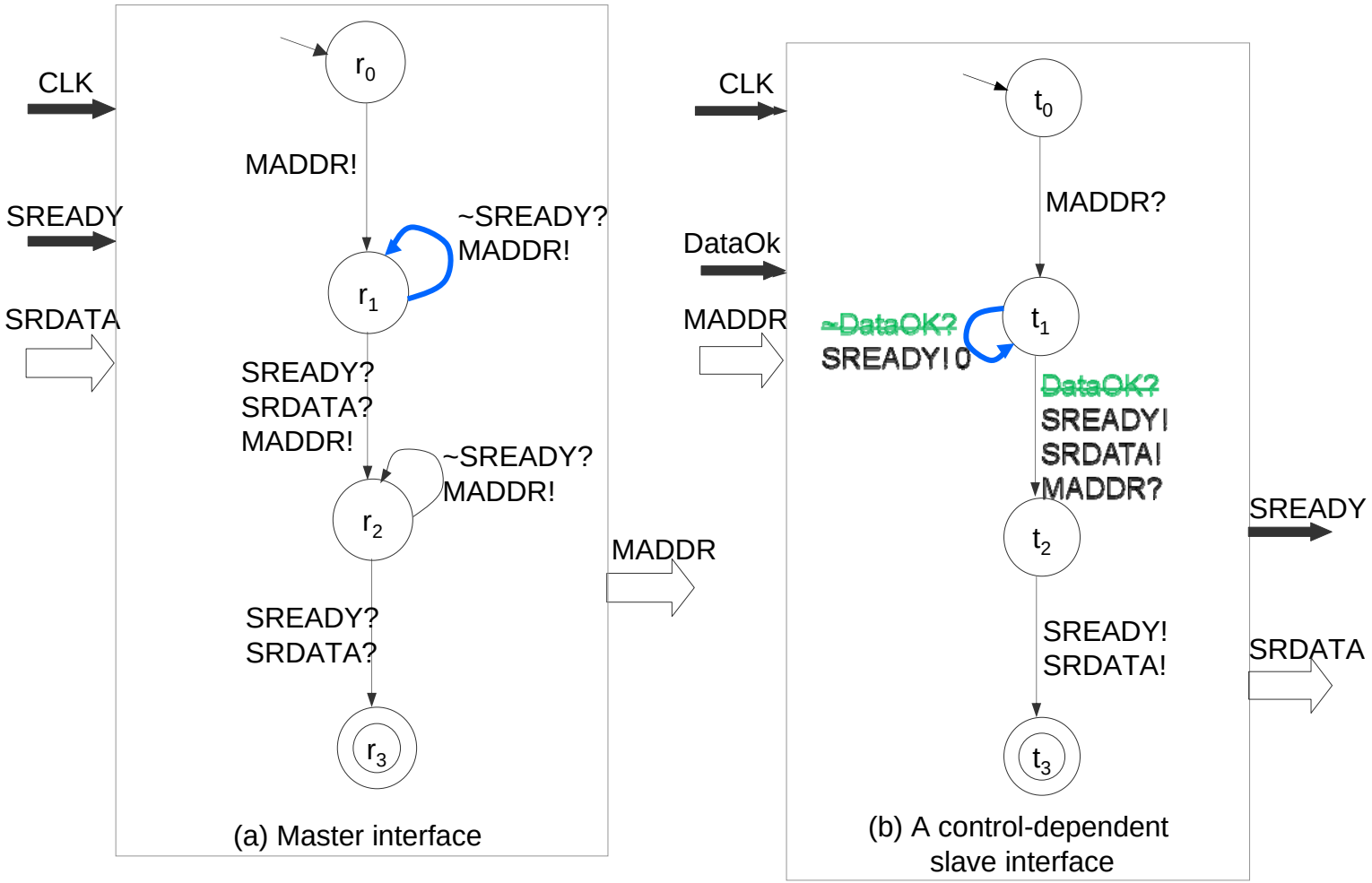
- Two kinds of non-predetermined transitions:
  - control dependent
  - data dependent
- Traverses and compresses each possible path.



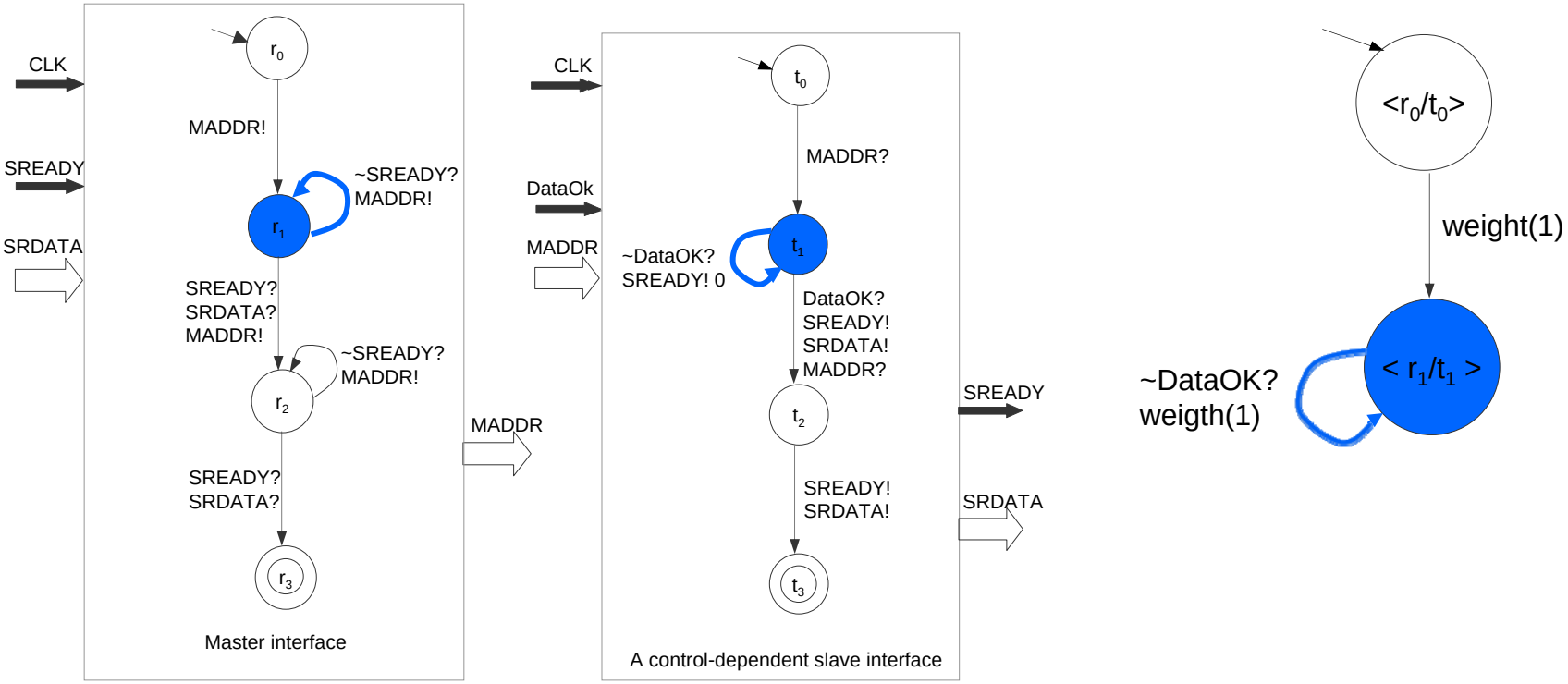
Control-dependent Case



1st Branch

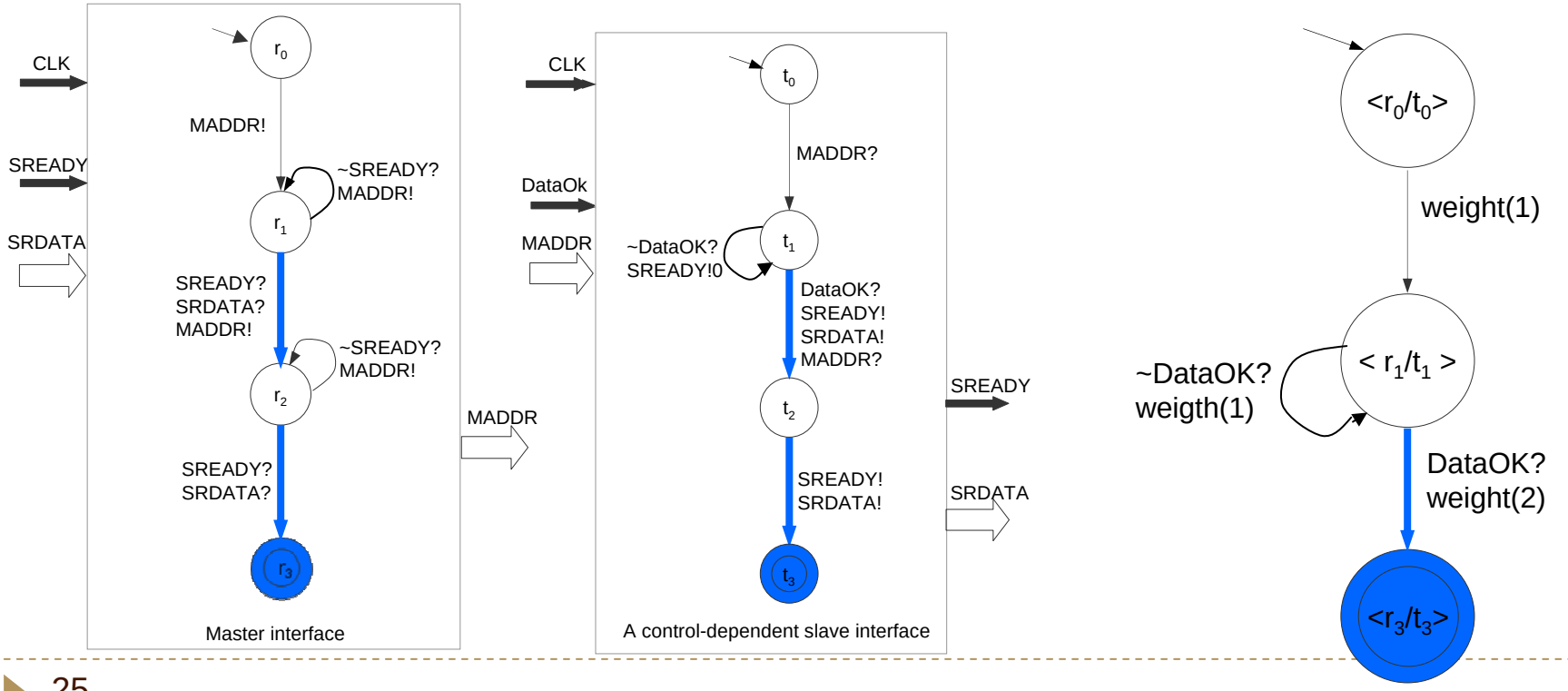

CLK
SREADY
SRDATA
MADDR
r0
MADDR!
r1
~SREADY?
MADDR!
SREADY?
SRDATA?
MADDR!
r2
~SREADY?
MADDR!
SREADY?
SRDATA?
r3
(a) Master interface
CLK
DataOk
MADDR
t0
MADDR?
t1
~DataOK?
SREADY! 0
DataOK?
SREADY!
SRDATA!
MADDR?
t2
SREADY!
SRDATA!
t3
SREADY
SRDATA
(b) A control-dependent slave interface

1st Branch (cont'd)



2nd Branch

- The compression is finished.
  - Cycle Count Timing is preserved.

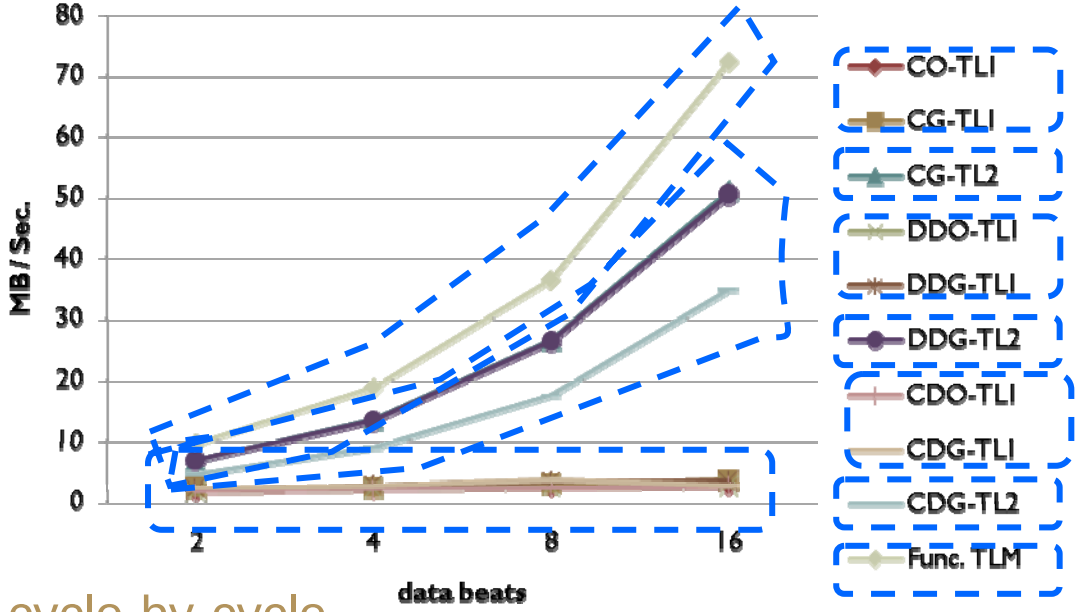


SystemC Bus Model Generation

- Is implemented in SystemC interface and channel pattern
  - the signals in SPA are translated into
    - variables
    - read/write events
  - data operations implemented as IMC (Interface Method Call).
- Is scheduled
  - in the clock-driven style for the TL1 bus models
  - in the event-driven style for the CCA-TL2 bus models

Experimental Results

- The core protocol, *burst write with handshake*, from OCP-IP is chosen.
  - Intel 3.40 GHz Xeon CPU
- For speed comparison:



- For accuracy:
  - Compare TL1 bus model cycle-by-cycle.
  - Compare CCA-TL2 bus model at transaction boundaries.

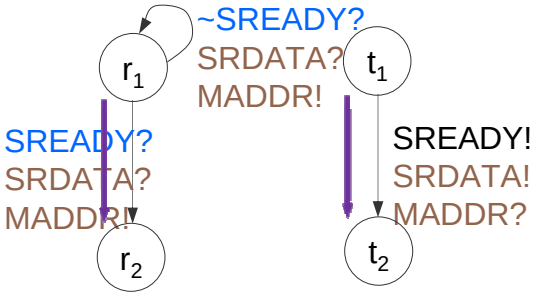
Conclusion & Future Work

- Formally discusses what information between different transaction levels can be simplified.
- Proposes the first automatic approach.
- Considers an automatic approach for multiple masters and multiple slaves with an arbiter in the future.

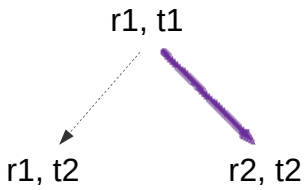
Thanks for your attention!!

Synchronous Protocol Automata (SPA)

- A synchronous protocol automaton is a tuple (Q,D,C,A, V, →, clk, q0, qf), where:
  - Q: a finite set of control states
  - q0, qf: initial state and final state
  - D,C: a set of input or output of data and control signals
  - V: a set of internal variables
  - A: a set of actions
  - → ⊂ QxQxclk?xA : transition relations



(a) A segment of SPA



(b) Corresponding State chart