Quantitative Analysis of Transaction Level Models for the AMBA Bus

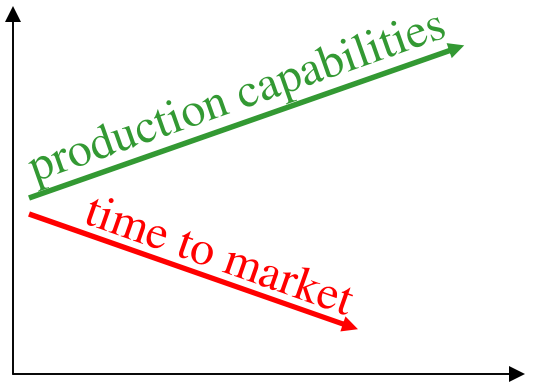
Gunar Schirner and Rainer Dömer

Center for Embedded Computer Systems
University of California, Irvine

UCIrvine
University of California, Irvine

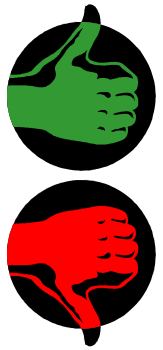
Motivation

- Higher productivity is needed for SoC design
  - Increased production capabilities
  - Shorter time-to-market
- Explore larger design space in less time
  - Requires fast simulation capabilities
- One approach: higher levels of abstraction
  - Transaction Level Modeling
    - Proposed to model communication [02 T. Grötker et. al, System C]
    - Widely used and accepted
    - Gains performance, but looses accuracy by abstraction
    - Exists a trade-off *speed* vs. *accuracy*
    - ➢ ***No detailed analysis yet!***
      - ➢ ***Designer: which features to abstract?***
      - ➢ ***Users: consequences of using an abstract model?***



Goals

- **Quantitatively analyze** trade-off in Transaction Level Model
  - How much speed improvement?
  - How much loss in accuracy?



- Identify model for a environment condition
  - Guidance for model developer
  - Guidance for model user

- Based on a case study:
  - AMBA AHB 2.0

Outline

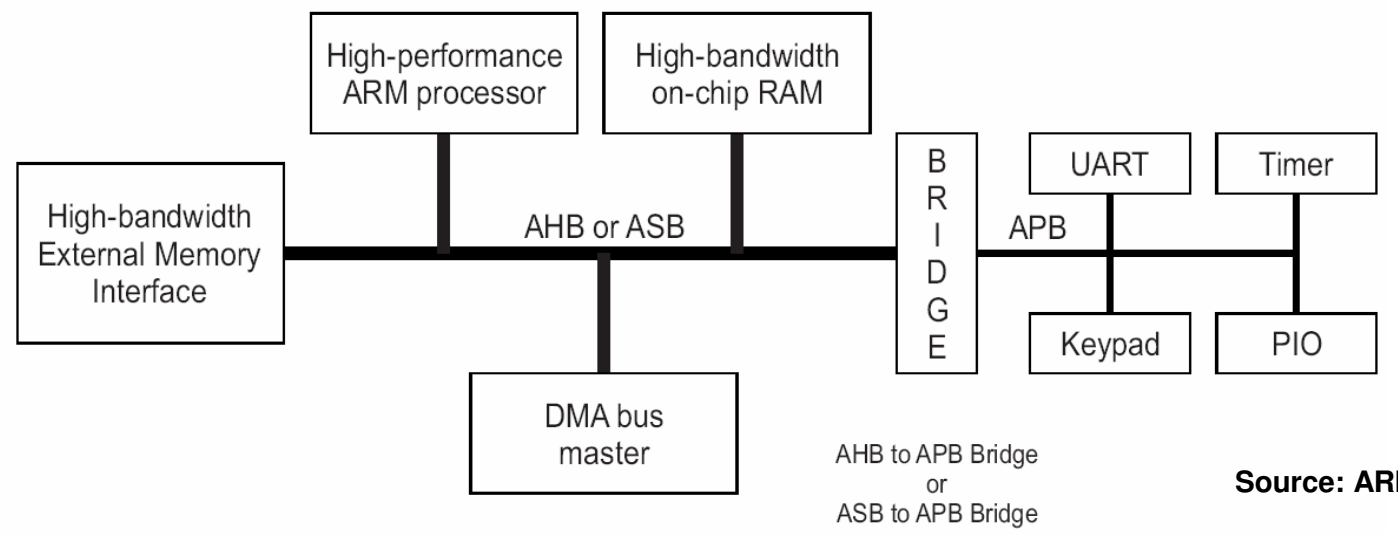
- Related Work
- Introduction of AMBA
- Modeling
  - Abstraction Levels
  - Bus Models
- Measurements and Quantitative Analysis
  - Performance
  - Accuracy
- Summary and Conclusions

Related Work

- T. Grötker et al., *System Design with SystemC*. Kluwer Academic Publishers, 2002
- M. Caldari et al., *Transaction-level models for AMBA bus architecture using SystemC 2.0*, DATE 2003
- M. Coppola et al., *IPSIM: SystemC 3.0 Enhancements for Communication Refinement, DATE* 2003
- S. Pasricha et al., *Fast exploration of bus-based on-chip communication architectures*, CODES + ISSS 2004
- ARM, *Amba AHB Cycle Level Interface specification,* ARM IHI 0011A

Introduction to AMBA

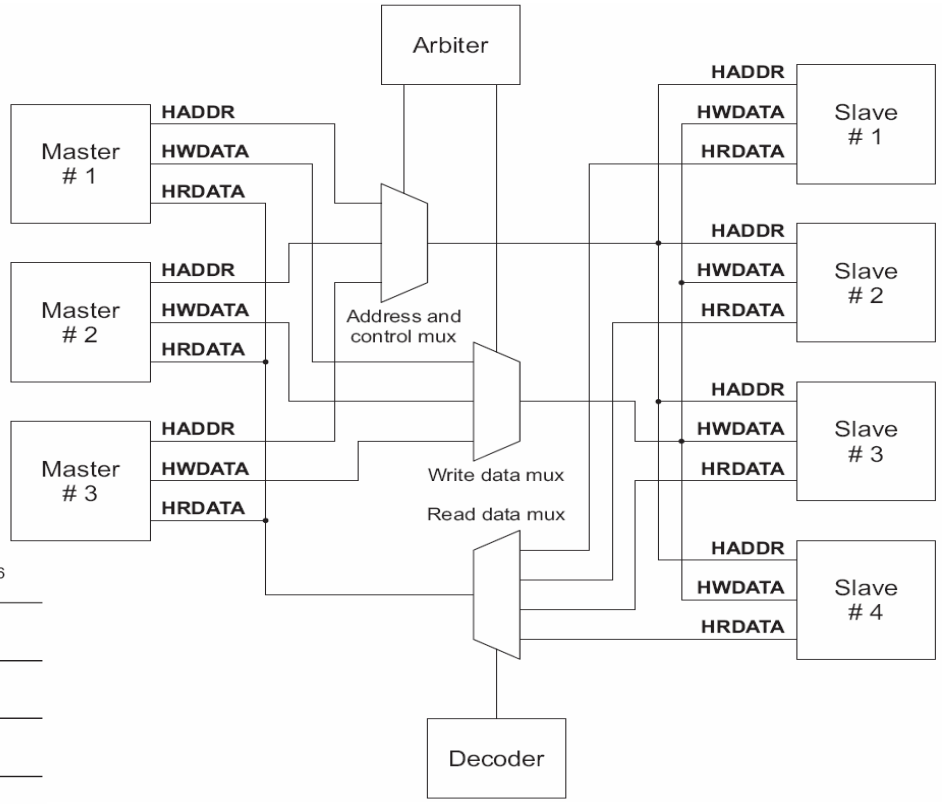
- Advanced Microprocessor Bus Architecture (AMBA)
  - By ARM
- De-facto standard for on-chip bus system
- Hierarchical structure:
  - System bus + Peripheral bus



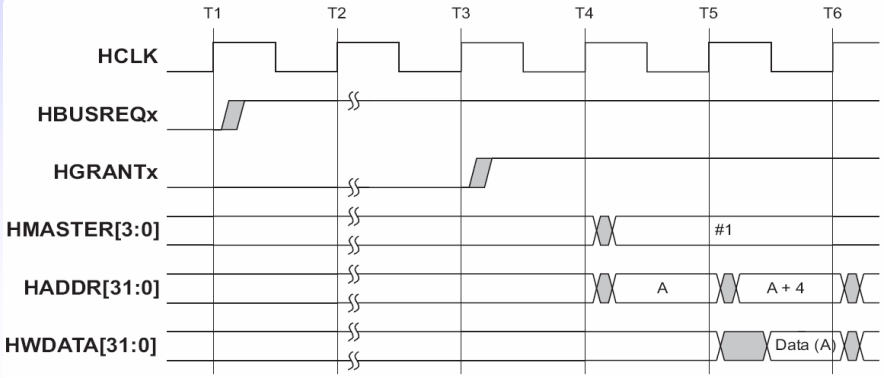
Source: ARM

Introduction to AMBA: AHB

- Advanced High-performance Bus (AHB)
  - Multi-master bus
  - Pipelined operation
  - Burst transfers
  - Retry and split transactions
  - Multiplexed interconnection
  - Locked, unlocked transfers



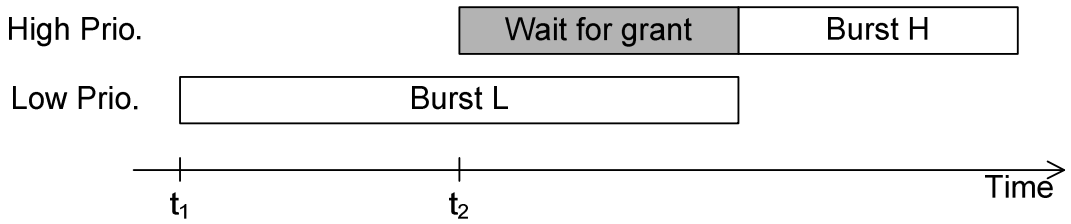
Source: ARM



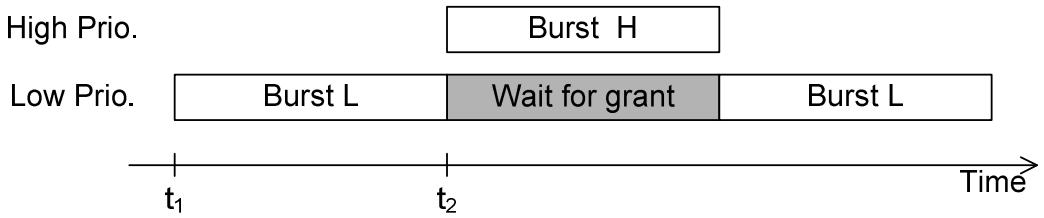
Source: ARM

Introduction to AMBA: Locking

- Bursts over multiple bus cycles (e.g. 4 beats, 8 beats)
  - Locked (non-preemptable):
    - Burst can not be preempted (even from higher priority master)



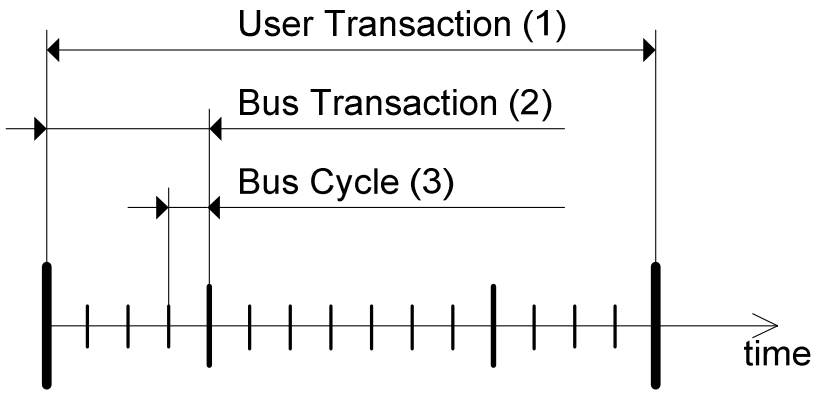
  - Unlocked (preemptable):
    - Burst may be preempted, resumed later



- Analyze both transfer types separately

Modeling: Abstraction Levels

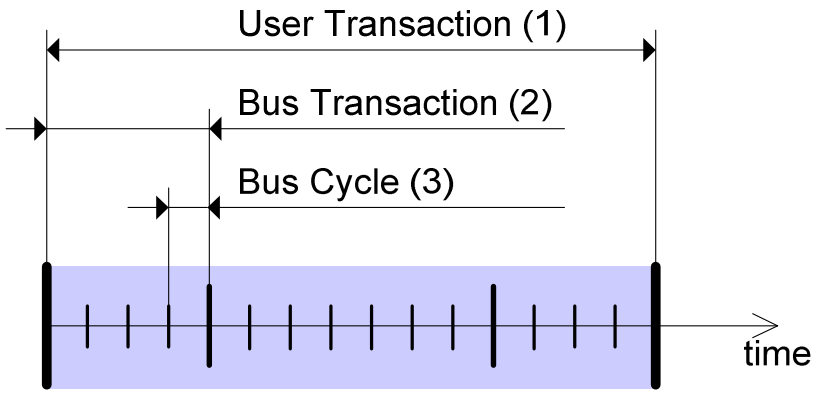
- What are possible abstraction levels?
- ISO/OSI reference layer-based architecture
  - Functionality
  - Granularity of data and arbitration handling
- Layers:
  1) Media Access Control (MAC)
     - User Transaction
       - Contiguous block of bytes
       - Arbitrary length, base address
  2) Protocol
     - Bus Transaction
       - Bus primitives (e.g. store word)
       - Observes bus address restrictions
  3) Physical
     - Bus Cycle
       - Drive or sample bus wires on bus cycle



- Models are composed of layers
  - Using fewer layers yields a more abstract model

Modeling: Abstraction Levels

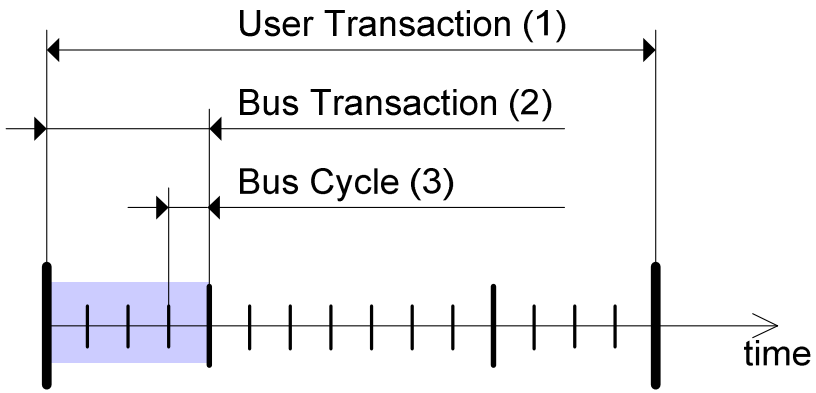
- What are possible abstraction levels?
- ISO/OSI reference layer-based architecture
  - Functionality
  - Granularity of data and arbitration handling
- Layers:
  1) Media Access Control (MAC)
     - User Transaction
       - Contiguous block of bytes
       - Arbitrary length, base address
  2) Protocol
     - Bus Transaction
       - Bus primitives (e.g. store word)
       - Observes bus address restrictions
  3) Physical
     - Bus Cycle
       - Drive or sample bus wires on bus cycle



- Models are composed of layers
  - Using fewer layers yields a more abstract model

Modeling: Abstraction Levels

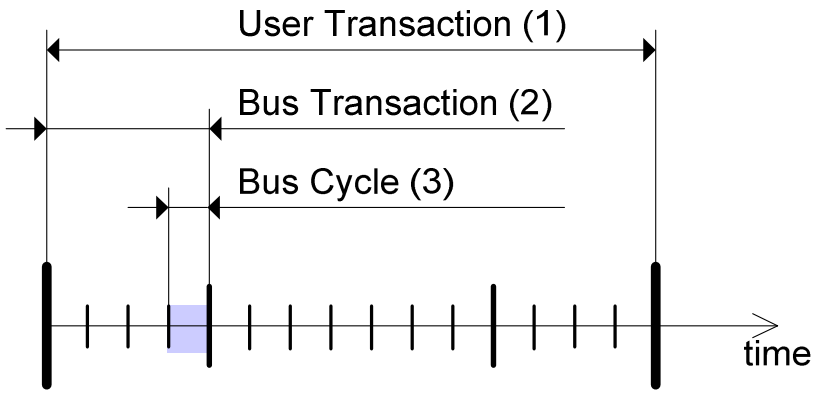
- What are possible abstraction levels?
- ISO/OSI reference layer-based architecture
  - Functionality
  - Granularity of data and arbitration handling
- Layers:
  1) Media Access Control (MAC)
     - User Transaction
       - Contiguous block of bytes
       - Arbitrary length, base address
  2) Protocol
     - Bus Transaction
       - Bus primitives (e.g. store word)
       - Observes bus address restrictions
  3) Physical
     - Bus Cycle
       - Drive or sample bus wires on bus cycle



- Models are composed of layers
  - Using fewer layers yields a more abstract model

Modeling: Abstraction Levels

- What are possible abstraction levels?
- ISO/OSI reference layer-based architecture
  - Functionality
  - Granularity of data and arbitration handling
- Layers:
  1) Media Access Control (MAC)
     - User Transaction
       - Contiguous block of bytes
       - Arbitrary length, base address
  2) Protocol
     - Bus Transaction
       - Bus primitives (e.g. store word)
       - Observes bus address restrictions
  3) Physical
     - Bus Cycle
       - Drive or sample bus wires on bus cycle



- Models are composed of layers
  - Using fewer layers yields a more abstract model

Modeling: Bus Functional (BFM)

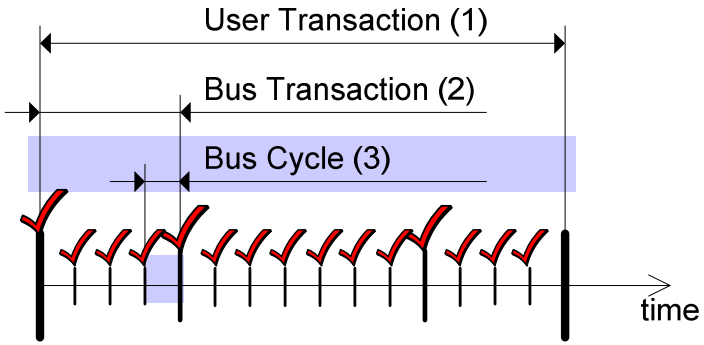
Implemented Layers:

MAC

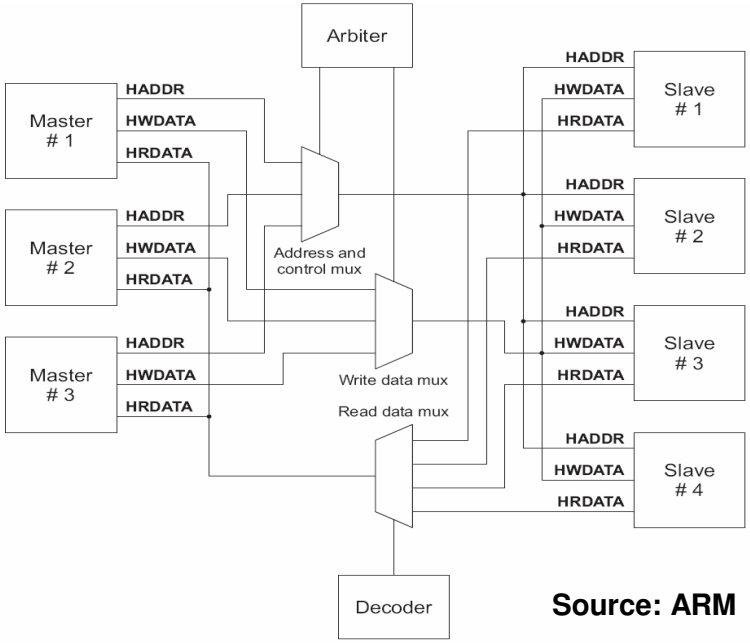
Protocol

Physical

Granularity:



- Pin accurate
- Bus cycle accurate
  - Arbitration check ✓ on each cycle
- Includes additional active components
  - Multiplexers (tri-state-free bus)
  - Arbiter
  - Address decoder
  - Clock generator



Source: ARM

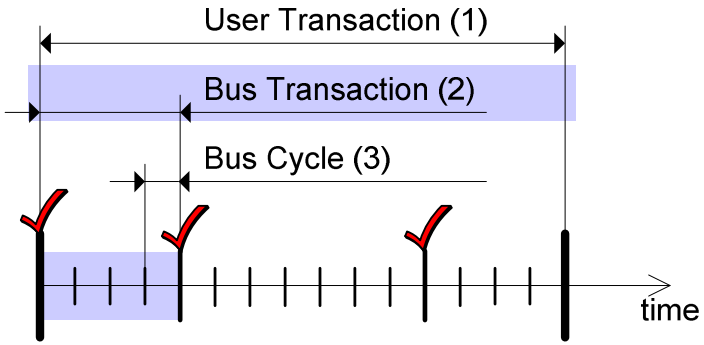
Modeling: Arbitrated TLM (ATLM)

Implemented Layers:

MAC

Protocol

Granularity:



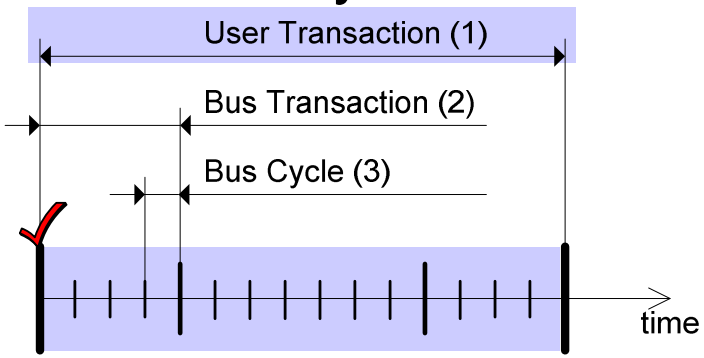
- Priority arbitration per bus transaction (e.g. StoreWord, StoreWordBurst4)
- Abstract model
  - Not pin accurate, not bus cycle accurate in all cases
- Variants:
  - ATLM (a): as above
  - ATLM (b): arbitration decision immediately
    - Arbitration requests not collected for one CLK cycle
    - ➢ May lead to wrong arbitration decision, depending on execution order

Modeling: Transaction Level (TLM)

Implemented Layers:

MAC

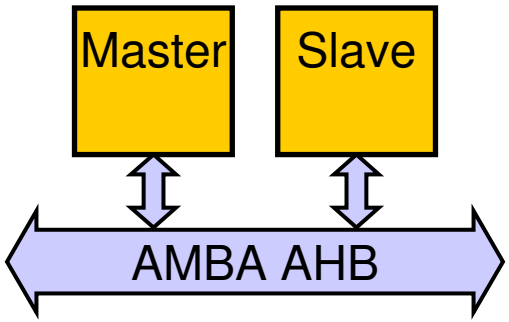
Granularity:



- No arbitration: Contention avoidance by semaphore
  - Resolution depends on simulator
- Expected to be the fastest model
  - Single **`memcpy`**, Single **`time wait`**
- Variants:
  - TLM (a): as above
  - TLM (b): no contention resolution at all
    - Multiple transfers at same time

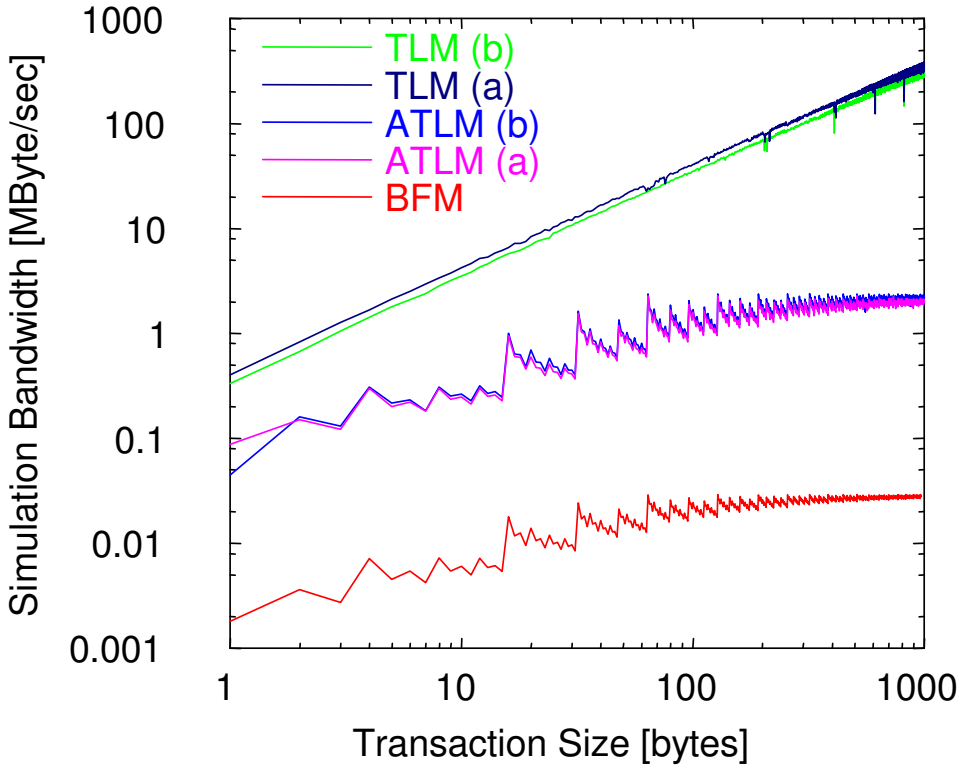
Performance Analysis: Test Setup

- Performance metrics
  - Simulation bandwidth for a model with one master and one slave
- Connection Setup
  - 1 master
  - 1 slave
- Repeatedly send user transaction
  - Measure simulation time for user transaction
  - Compute simulation bandwidth
- Platform
  - SpecC compiler and simulator
    - `scc` version 2.2.0, based on QuickThreads
  - Linux PC
    - Pentium 4, 2.8 GHz



Performance Analysis: Bandwidth

- Confirmation:
  Abstraction yields speedup
- Two orders of magnitude between major models
- No performance difference between variants
- Saw tooth shape due to bus transactions, e.g.:
  - 3 byte == 2 bus transactions (1 short + 1 byte)
  - 4 byte == 1 bus transaction (1 word)

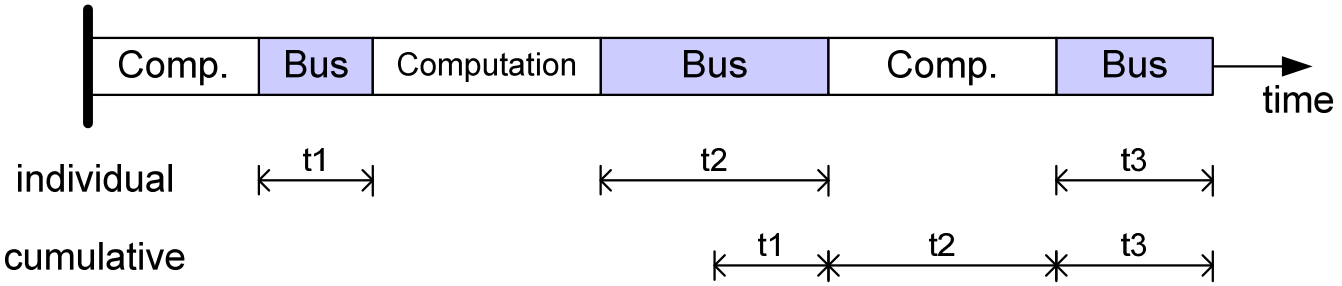


Feature	BFM	ATLM	TLM
Simulation Time [ms]	16.75	0.2137	0.00246
Sim. Bandwidth [MByte/s]	0.03	2.29	198
Rel. Speedup over BFM	1	78	6802

Performance for a 512 byte transfer

Accuracy Analysis

- What is accuracy?
  - Functionality / **Timing**
- What are the relevant measurements for a time accuracy?
  - Depends on prediction goal
    - Application latency due to bus accesses
      - Analyze ***individual*** transfer duration
    - Overall application delay due to all bus accesses
      - Analyze ***cumulative*** transfer duration



- Definition of error (for this work):

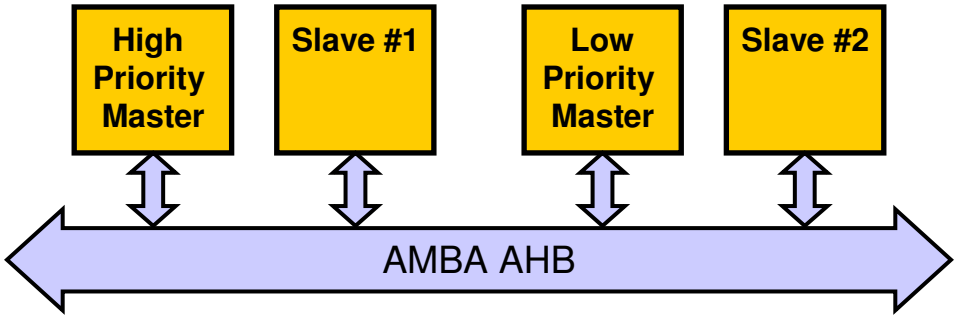
$$d_{std}: \quad \text{duration as per AHB standard}$$

$$d_{test}: \quad \text{duration in model under test}$$

$$error_i = 100 * \frac{|d_{test} - d_{std}|}{d_{std}}$$

Accuracy Analysis: Setup

- Connection setup
  - 2 masters
  - 2 slaves

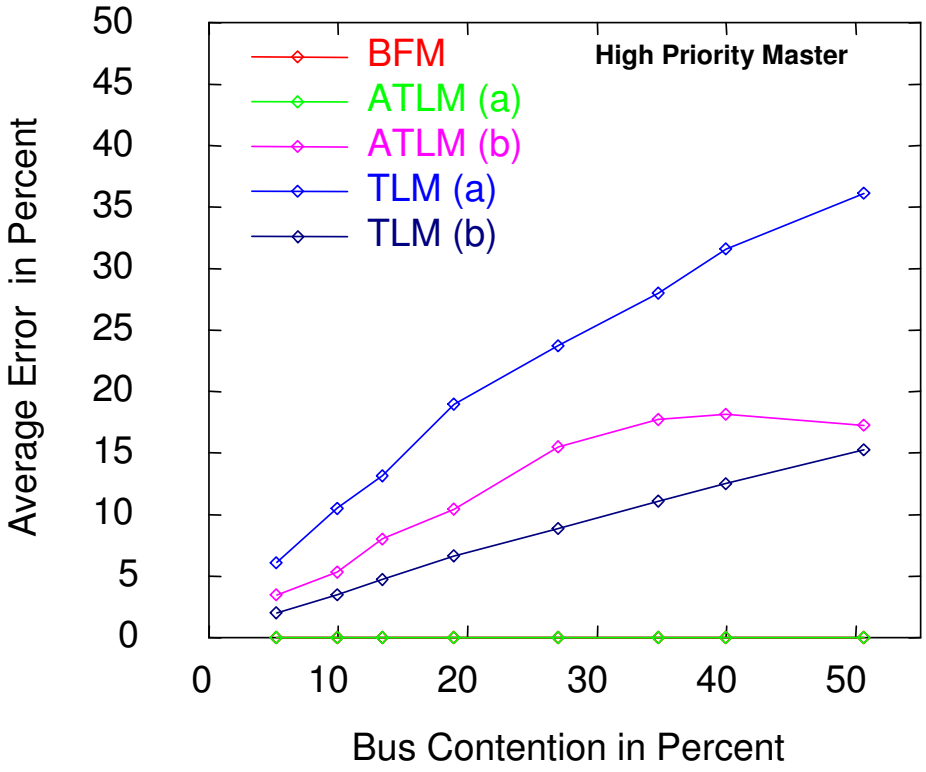


- Predefined set of 5000 transactions
  - Linear random distributed in:
    - Length (1 … 100 bytes) and content
    - Delay between transactions (simulates local computation time)
    - Destination address
  - Log for each transaction
    - Start time, Duration
- Repeat same set of transactions with each model
- Repeat for varying bus contentions

Accuracy Analysis: Locked Transfers (Individual)

- Average error of individual user transaction duration
  - Each point is avg. of 5000 transactions
- BFM
  - No error
- ATLM (a)
  - NO ERROR
  - Locked transfers only, additional features of BFM not exercised
- ATLM (b)
  - Up to 15% error
  - Immediate arbitration may make wrong prediction
- TLM (a)
  - Linear increasing worse decisions
  - Coarse grain arbitration simulation
- TLM (b)
  - Surprising close results
  - Assumes always available bus

➢ The more abstract the more inaccurate



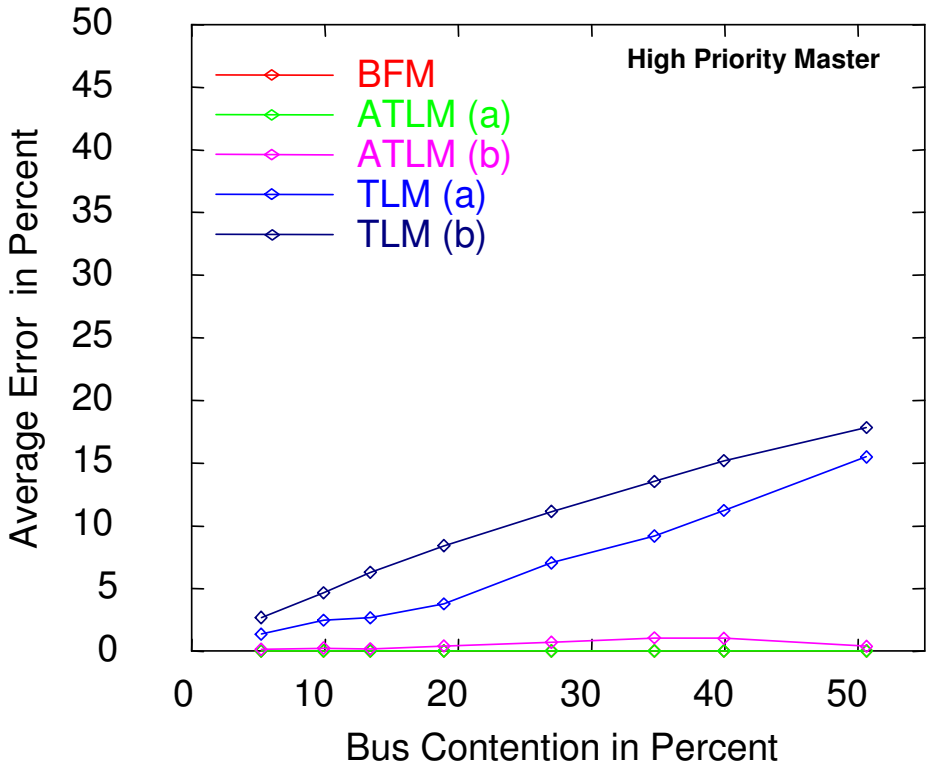
Accuracy Analysis: Locked Transfers (Cumulative)

- Error of sum of all user transaction durations
  - Each point is cumulative error of 5000 transactions

- ATLM (a):
  - NO ERROR
- ATLM (b):
  - Minimal error only
  - Miss predictions seen in duration analysis averages out!
- TLM (a):
  - Linear increasing worse decisions
  - Now better than the TLM (b)
- TLM (b):
  - Linear increasing worse decisions
  - Always too optimistic

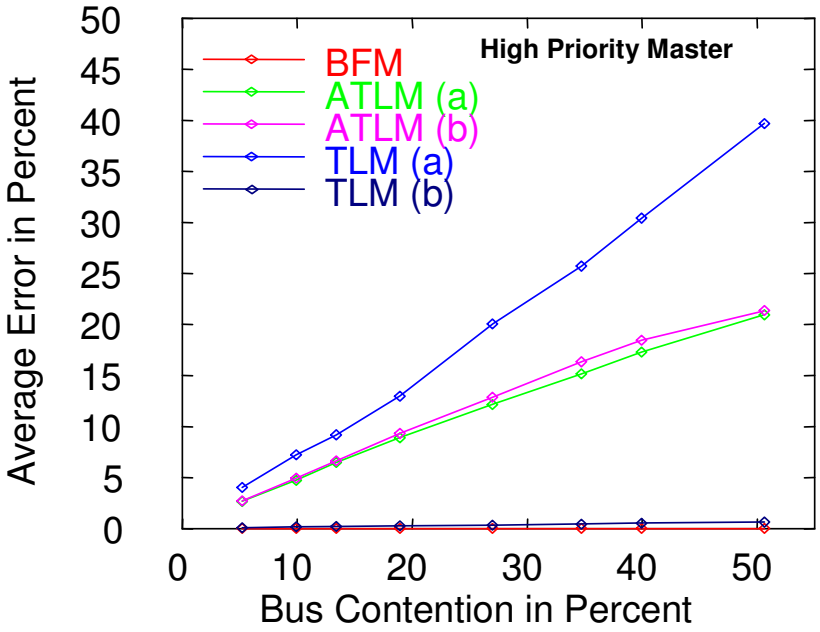
➢ Errors average out

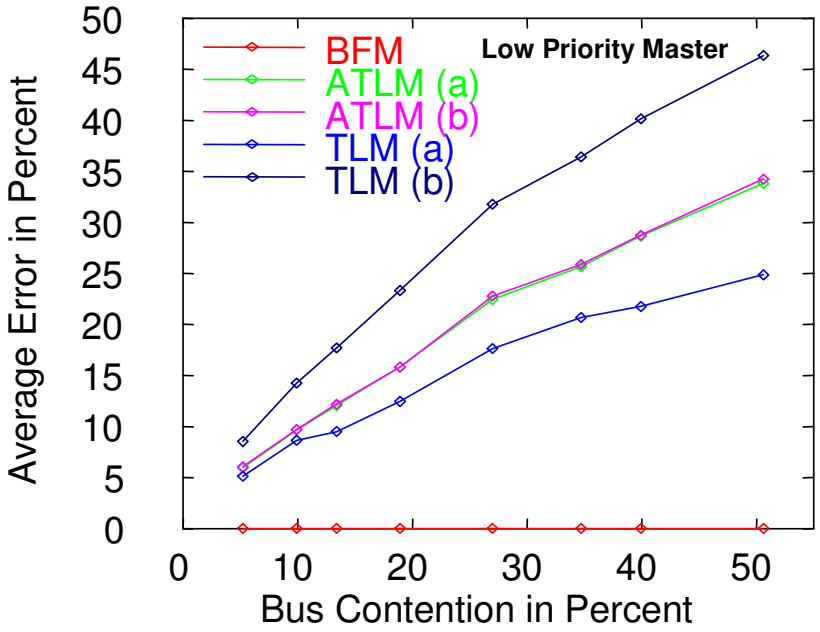
➢ Except for unrealistic TLM(b)



Accuracy Analysis: Unlocked Transfers (Cumulative)

- Unlocked transfers: low priority burst may be preempted





- BFM: the only accurate model
- ATLM (a): now shows error, arbitration check per bus cycle
- ATLM (b): similar to ATLM (a), additional arbitration error negligible
- TLM (a) + TLM (b): Inverse results between high prio. and low prio.
- TLM (b): error is less predictable

➢ Only BFM yields accurate results

➢ TLM(b) is unreliable

Summary

- Modeled AMBA AHB in 3 major models:
  - BFM, ATLM, TLM
  - Variants ATLM (b), TLM (b)
- Analyzed execution performance
  - 100x speedup per abstraction step
- Quantified error due to abstraction
  - ATLM: 0% (indiv., locked), 35% (cumul., unlocked)
  - TLM: 35% (indiv., locked), 40% (cumul., unlocked)

Conclusion

- Higher abstraction (decreasing accuracy) yields speedup
  - 100x speedup per abstraction step
- Variation at same abstraction level
  - No significant speed up
  - Accuracy loss
- Accuracy depends on
  - Abstraction level
  - Bus contention
  - Used bus features
- Guideline for model user:

Environment Condition	Model	Granularity	Speedup
• Single master • No bus contention	TLM	User Transaction	10000x
• Locked transfers only • Unlocked transfers with low contention	ATLM	Bus Transaction	100x
• Unlocked transfers with high contention	BFM	Bus Cycle	1