2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION

UNITED STATES

SAN JOSE, CA, USA
FEBRUARY 28 - MARCH 3, 2022

Modeling Memory Coherency for concurrent/parallel accesses

Subramoni Parameswaran

XILINX®

accellera SYSTEMS INITIATIVE

Agenda

- Problem Statement
- Current approaches and pitfalls
- Window of Uncertainty (WoU)
- Multiple outstanding writes
- Simultaneous outstanding reads
- Conclusion

Problem Statement

- Modeling memory coherency
  - Predict coherent read value based on writes
  - Read must return most recently written value
- Issues
  - Different latencies
  - Multiple outstanding writes
  - Simultaneously outstanding reads

Current approaches and pitfalls

- Cycle accurate modeling
  - Expensive
  - Error prone leading to lots of debug
- Observation of internal design variables
  - Potentially incomplete verification and missed bugs
  - High maintenance costs when design changes

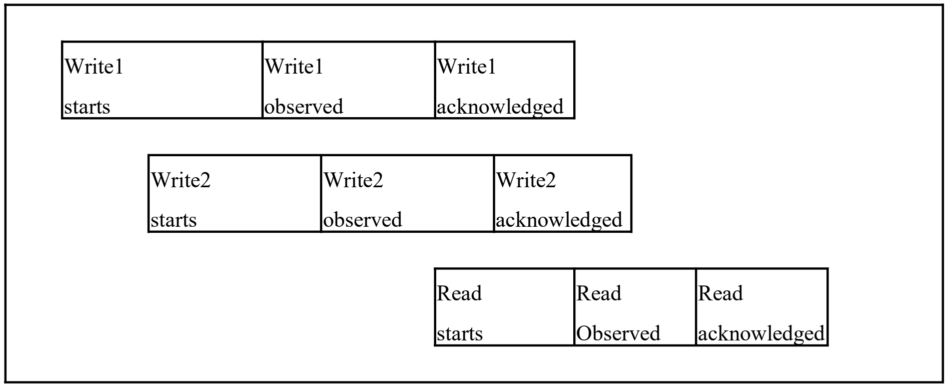
Window of Uncertainty (WoU)

Definitions

- Issue
  - Start of the write or read transaction
- Acknowledgment
  - Read or write - Response to transaction from the design
  - Write - Unique transaction value is visible to a subsequent read
- Outstanding
  - From the time of transaction issue until its acknowledgment

Uncertainty of outcomes for a read

- Issue of write introduces uncertainty for reads
- Example - Read issued while two writes are outstanding
  - Read can coherently return either write1 value or write2 value
- Deterministic result prediction is expensive/error prone



Write1 starts | Write1 observed | Write1 acknowledged

Write2 starts | Write2 observed | Write2 acknowledged

Read starts | Read Observed | Read acknowledged

Window of uncertainty (WoU)

Time window where the value of a write "persists"

Start of uncertainty

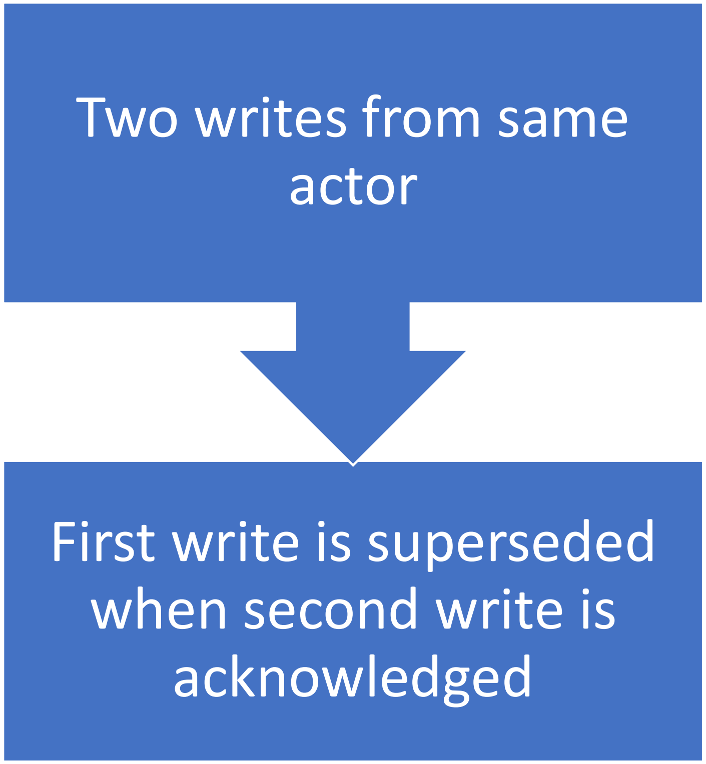
Issue of the write

End of uncertainty

Write value must be superseded by other writes (Coherency rules)

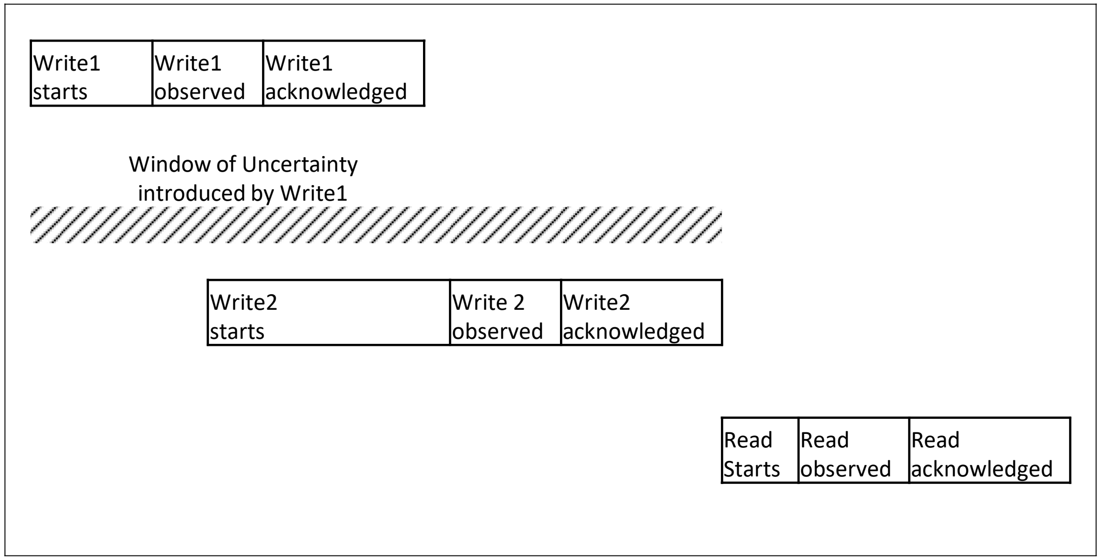
Multiple outstanding writes

WoU – same actor, multiple writes outstanding



Two writes from same actor

First write is superseded when second write is acknowledged



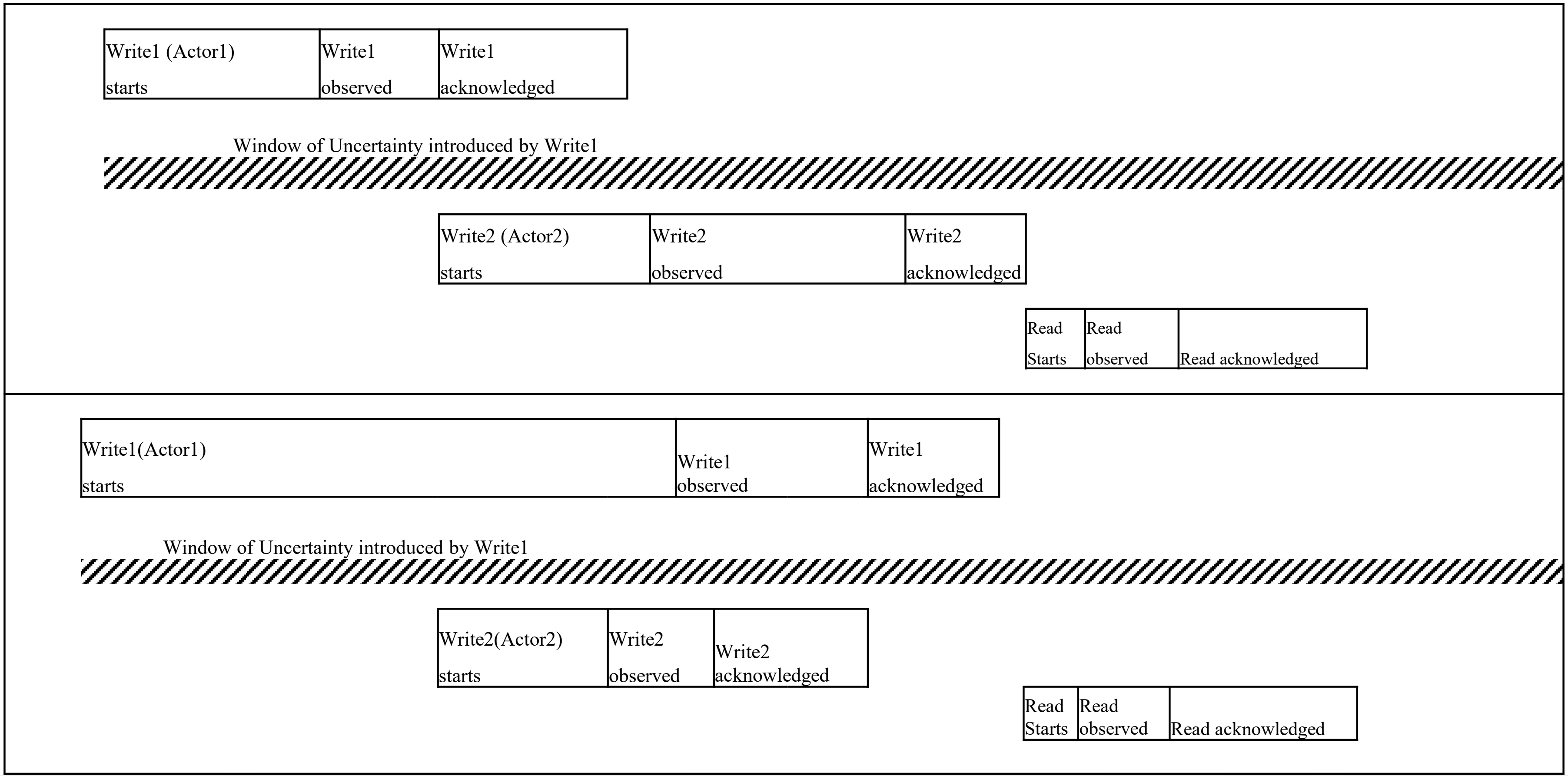
Write1 starts | Write1 observed | Write1 acknowledged

Window of Uncertainty introduced by Write1

Write2 starts | Write 2 observed | Write2 acknowledged

Read Starts | Read observed | Read acknowledged

WoU - Different actors, multiple outstanding



Write1 (Actor1) starts | Write1 observed | Write1 acknowledged

Window of Uncertainty introduced by Write1

Write2 (Actor2) starts | Write2 observed | Write2 acknowledged

Read Starts | Read observed | Read acknowledged

Write1(Actor1) starts | Write1 observed | Write1 acknowledged

Window of Uncertainty introduced by Write1

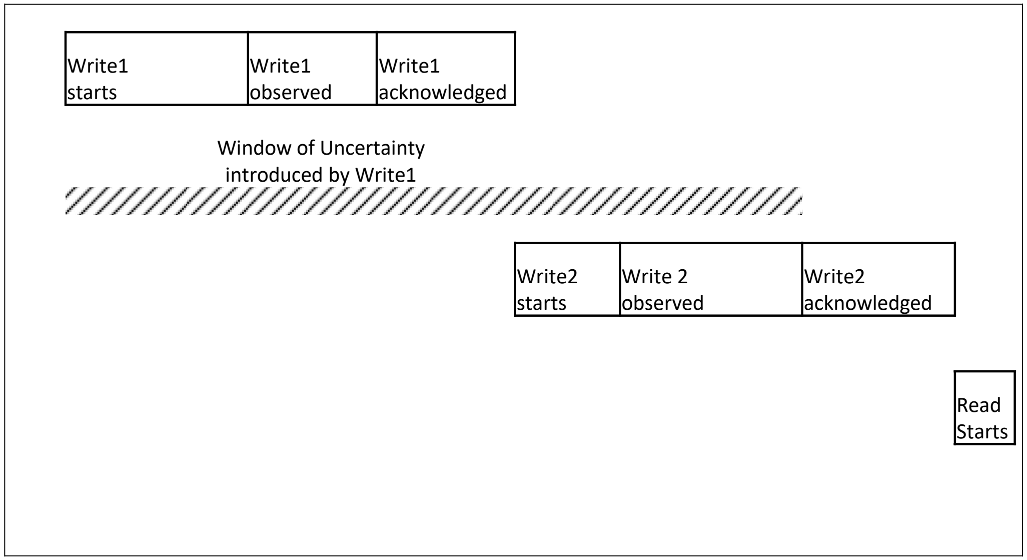
Write2(Actor2) starts | Write2 observed | Write2 acknowledged

Read Starts | Read observed | Read acknowledged

Different actor WoU
Special case – Serialized write issue

Two writes from different actors

First write is superseded after it completes, and a subsequent write is completed



Write1 starts

Write1 observed

Write1 acknowledged

Window of Uncertainty introduced by Write1

Write2 starts

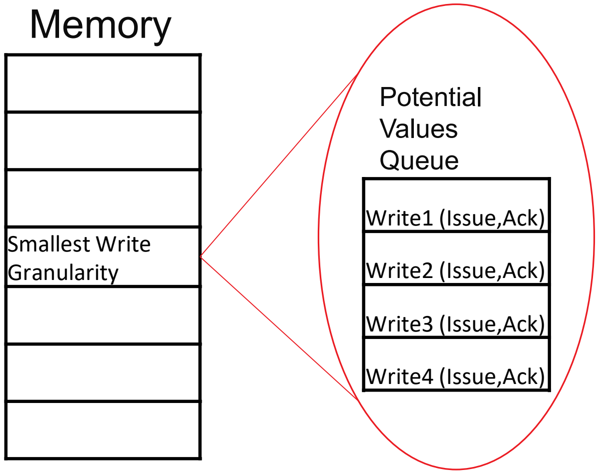
Write 2 observed

Write2 acknowledged

Read Starts

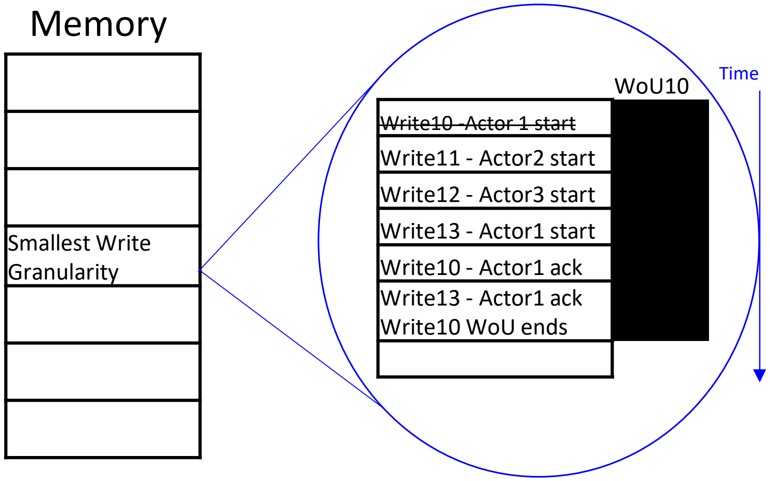
Potential Values Queue

Choose	Granularity of Model = smallest write size
Store	Multiple data values that might be the winner
Order	Potential data values based on issue of the writes
Remove	Superseded writes based on window of uncertainty



Potential Values Queue Examples

- Single actor superseding



Memory

Smallest Write Granularity

WoU10

Time

~~Write10 -Actor 1 start~~

Write11 - Actor2 start

Write12 - Actor3 start

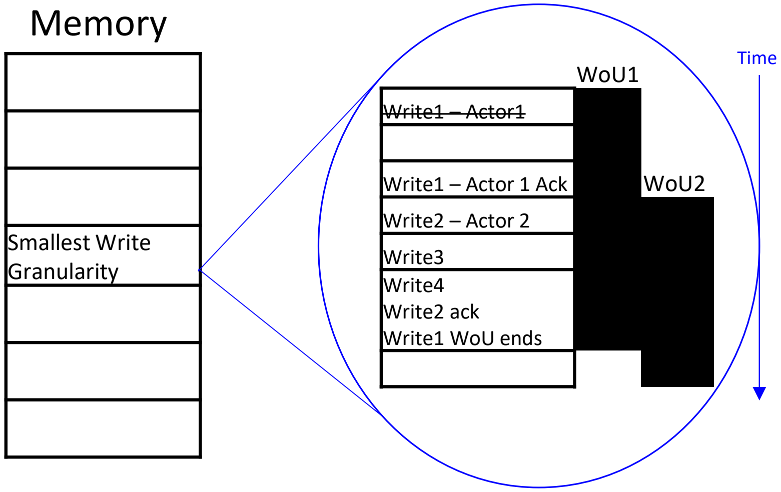
Write13 - Actor1 start

Write10 - Actor1 ack

Write13 - Actor1 ack

Write10 WoU ends

- Multiple actor superseding



Memory

Smallest Write Granularity

WoU1

WoU2

Time

~~Write1 – Actor1~~

Write1 – Actor 1 Ack

Write2 – Actor 2

Write3

Write4

Write2 ack

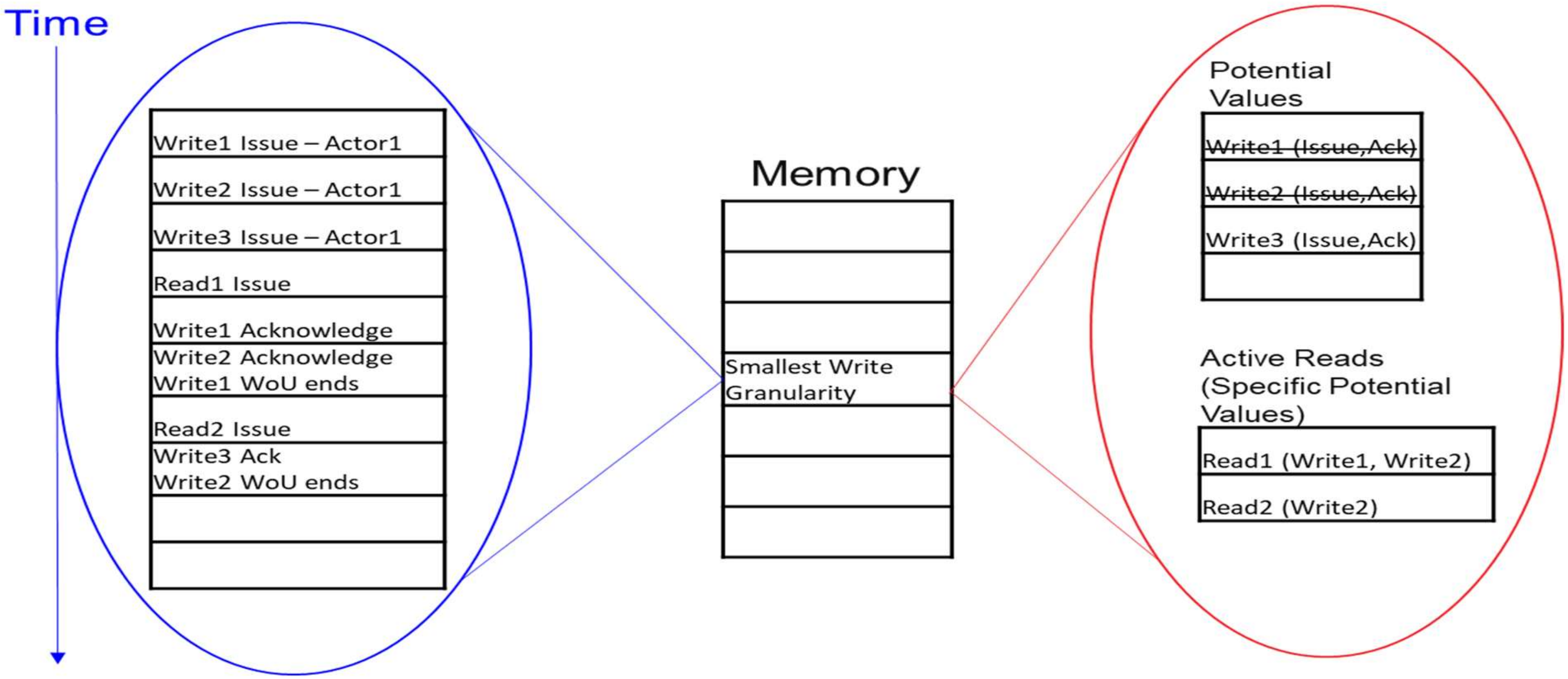
Write1 WoU ends

Simultaneous outstanding reads

Active Reads Queue

- Writes can be superseded while a read is outstanding
- Read can return:
  - Potential values from Potential values queue at its time of completion
  - Superseded writes after its issue, but before its completion
- Need a per-read active reads queue
  - Deal with simultaneous reads
  - Store write values superseded while that specific read was outstanding

Active Reads Queue Examples



Write1 Issue – Actor1
Write2 Issue – Actor1
Write3 Issue – Actor1
Read1 Issue
Write1 Acknowledge
Write2 Acknowledge
Write1 WoU ends
Read2 Issue
Write3 Ack
Write2 WoU ends

Potential Values

~~Write1 (Issue,Ack)~~
~~Write2 (Issue,Ack)~~
Write3 (Issue,Ack)

Active Reads (Specific Potential Values)

Read1 (Write1, Write2)
Read2 (Write2)

Conclusion

- Coherent generic memory model for uncertain read outcomes
  - Concurrent reads and writes
  - Multiple actors with different latencies
- Model seamlessly scales for any stimulus
  - Simultaneous reads and writes
  - No simultaneous reads and writes
  - Single outstanding transaction
- Use of model allows extensive concurrent memory verification
  - Improved verification through expanded stimulus
  - Reduced cost of modeling and more testbench flexibility

Questions