A Specification Methodology by a Collection of Compact Properties

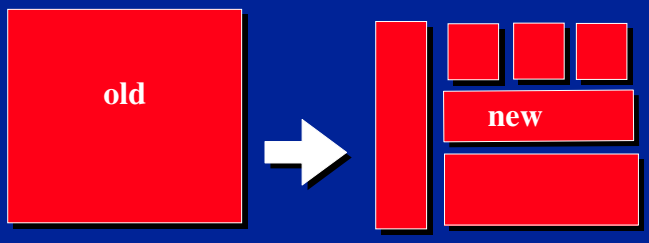
as Applied to the Intel® Itanium™ Processor Bus Protocol

Kanna Shimizu, David L. Dill,
Ching-Tsun Chou

Stanford University
Intel® Corporation

Background

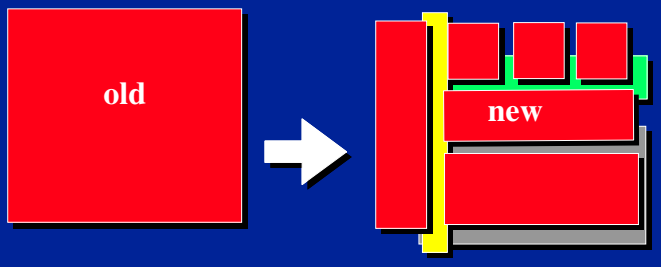
As digital circuit designs become more complex…



©Copyright by Kanna Shimizu 2002

Background

As digital circuit designs become more complex…



Interfaces must be clearly defined.

- Avoid communication problems between designers, IP buyer and seller, …

©Copyright by Kanna Shimizu 2002

Background

Problem

Target subsequent latency is the number of clocks from the assertion of IRDY# and TRDY# for one data phase to the assertion of TRDY# or STOP# for the next data phase in a burst transfer. The target is required to complete a subsequent data phase within eight clocks from the completion of the previous data phase. This requires the target to complete the data phase either by transferring data (TRDY# asserted), by

Current State

©Copyright by Kanna Shimizu 2002

Background

Problem

Target subsequent latency is the number of clocks from the assertion of IRDY# and TRDY# for one data phase to the assertion of TRDY# or STOP# for the next data phase in a burst transfer. The target is required to complete a subsequent data phase within eight clocks from the completion of the previous data phase. This requires the target to complete the data phase either by transferring data (TRDY# asserted), by

```
prev(!final_dphase_done & !idle)
        IMPLIES cbe_e;
prev(frame) IMPLIES (frame | irdy);
prev(stop & irdy) IMPLIES !stop;
prev(stop & irdy) IMPLIES !irdy;
prev(!frame) IMPLIES (frame | ad);
prev((counter = 3) & a) IMPLIES !a;
prev((m_initial = 7) & !irdy)
        IMPLIES irdy;
prev((m_subseq =7) & !irdy)
        IMPLIES irdy;
```

Current State

Why not this?

5

©Copyright by Kanna Shimizu 2002

Background

Goal

- **The cost-value problem**
  - **Too much time and expertise needed**
  - **Not very valuable**
- **Two-pronged solution**
  - **Make the specification process *easier***
  - **increase the *usefulness* of formal documents**

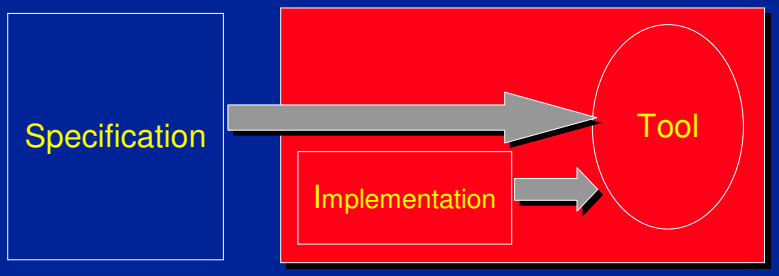
6

©Copyright by Kanna Shimizu 2002

Background

Goal

- *Methodology* is the key
  - As opposed to *tool* or *language* development
- Little work on methodology so far

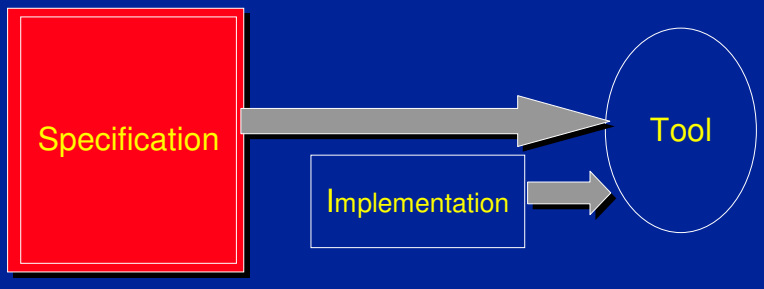


©Copyright by Kanna Shimizu 2002

Background

Goal

- *Methodology* is the key
  - As opposed to *tool* or *language* development
- Little work on methodology



©Copyright by Kanna Shimizu 2002

Background

- In [SDH00] (Shimizu, Dill, Hu, FMCAD '00)
  - Using PCI as an example
  - Specification style developed
  - Two specification debugging methods described

Methodology

©Copyright by Kanna Shimizu 2002

Contribution

- Shows the method applies to more advanced bus protocols
  - Intel® Itanium™ Processor bus protocol
  - Example : pipelining
- Reports interesting issues revealed during the protocol development
  - Worked with Intel® to prepare a standard

©Copyright by Kanna Shimizu 2002

Outline

1. Methodology
2. Specifying the Itanium™ processor bus protocol
3. Checking the protocol

©Copyright by Kanna Shimizu 2002

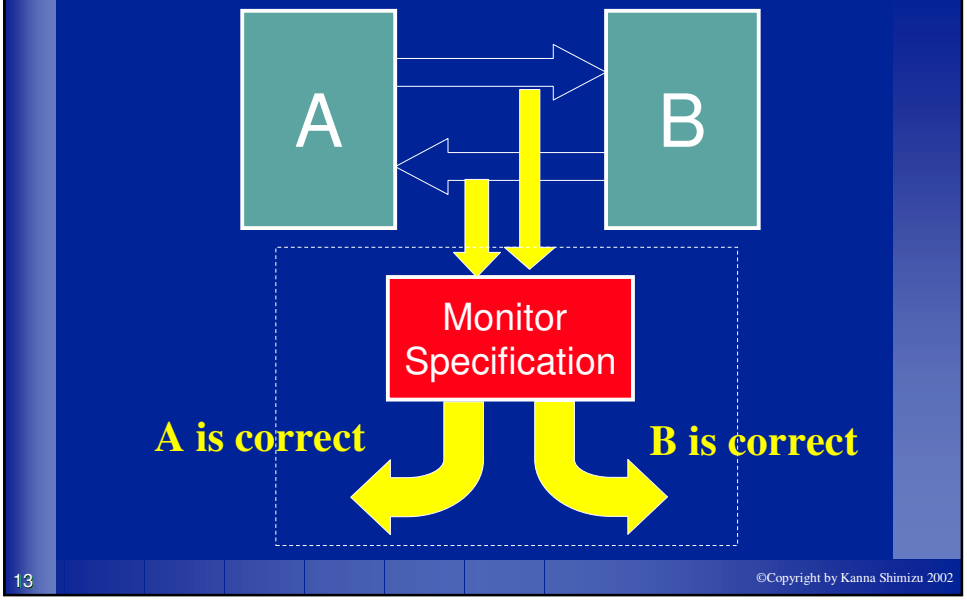
Methodology

1.Monitor-Based Specification

©Copyright by Kanna Shimizu 2002

Methodology

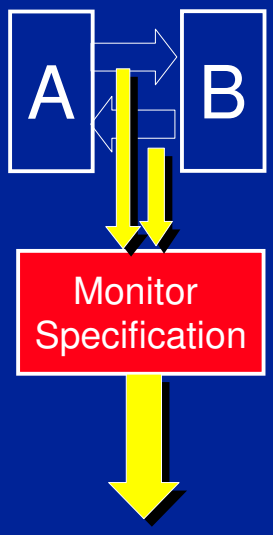
Specification (1)



©Copyright by Kanna Shimizu 2002

Methodology

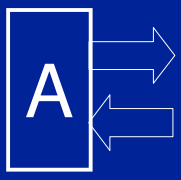
Specification (1)



©Copyright by Kanna Shimizu 2002

Methodology

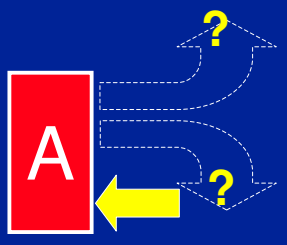
Specification (1)



©Copyright by Kanna Shimizu 2002

Methodology

Specification (1)

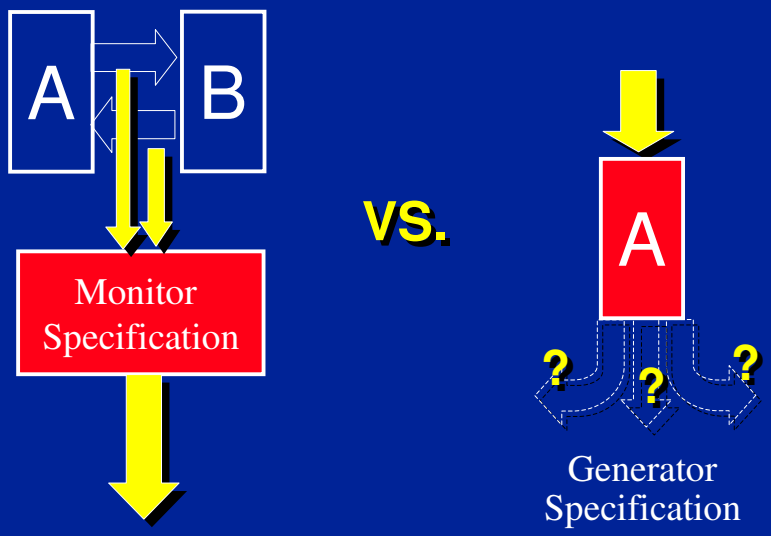


Generator
Specification

©Copyright by Kanna Shimizu 2002

Methodology

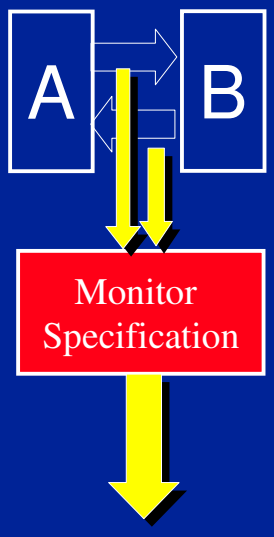
Specification (1)



©Copyright by Kanna Shimizu 2002

Methodology

Specification (1)



1. Monitor specification is deterministic
   - even if behaviour specified is non-deterministic
2. A list of simple constraints

©Copyright by Kanna Shimizu 2002

Methodology

2.Collection of Compact Properties

©Copyright by Kanna Shimizu 2002

Methodology

Specification (2)

Specify using a conjunction of simple constraints

prev(input0 & input1 & … | output0) IMPLIES

current(output1 | output2 & … & outputn)

©Copyright by Kanna Shimizu 2002

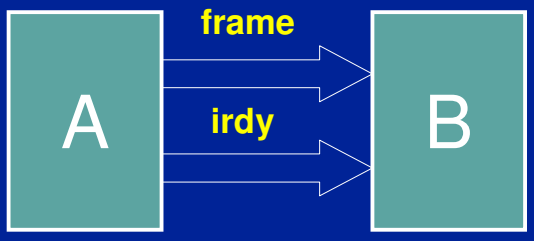
Methodology

Specification (2)

Example

"Only when IRDY# is asserted can FRAME# be deasserted"

prev(frame) IMPLIES frame | irdy

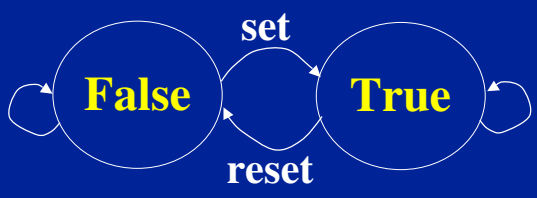


©Copyright by Kanna Shimizu 2002

Methodology

Specification (2)

- Simple state machines each track one thread of information
- Use many *generic* state machines instead of a few custom ones
  - Example : set-and-reset state machine



©Copyright by Kanna Shimizu 2002

Methodology

Summary

The specification *style* requires

1. Observe; then, deem actions "correct" or "incorrect"
2. Many small properties and many generic state machines

©Copyright by Kanna Shimizu 2002

Outline

1. Methodology
2. Specifying the Itanium™ processor bus protocol
3. Checking the protocol

©Copyright by Kanna Shimizu 2002

Itanium™

Pipelining

©Copyright by Kanna Shimizu 2002

Itanium™

Major Phases of a Transaction

- **Arbitration Phase (ARB)**
  - **Arbitrate to use the bus**
- **Request Phase (REQ)**
  - **Start transaction: put the request (write, read) on the bus**
- **Snoop Phase (SNP)**
  - **Snoop results are put out**
- **Response Phase (RESP)**
  - **Transaction is concluded with a data transfer**

©Copyright by Kanna Shimizu 2002

Itanium™

Major Phases of a Transaction

- Arbitration Phase (ARB)
  - Arbitrate to use the bus
- Request Phase (REQ)
  - Start transaction: put the request (write, read) on the bus
- Snoop Phase (SNP)
  - Snoop results are put out
- Response Phase (RESP)
  - Transaction is concluded with a data transfer

©Copyright by Kanna Shimizu 2002

Itanium™

Major Phases of a Transaction

- Arbitration Phase (ARB)
  - Arbitrate to use the bus
- Request Phase (REQ)
  - Start transaction: put the request (write, read) on the bus
- Snoop Phase (SNP)
  - Snoop results are put out
- Response Phase (RESP)
  - Transaction is concluded with a data transfer

©Copyright by Kanna Shimizu 2002

Itanium™

Major Phases of a Transaction

- Arbitration Phase (ARB)
  - Arbitrate to use the bus
- Request Phase (REQ)
  - Start transaction: put the request (write, read) on the bus
- Snoop Phase (SNP)
  - Snoop results are put out
- Response Phase (RESP)
  - Transaction is concluded with a data transfer

©Copyright by Kanna Shimizu 2002

Itanium™

Pipelining

Time	1	2	3	4	5	6
transaction0	ARB					
transaction1						
transaction2						
transaction3						

©Copyright by Kanna Shimizu 2002

Itanium™

Pipelining

Time	1	2	3	4	5	6
transaction0	ARB	REQ				
transaction1		ARB				
transaction2						
transaction3						

31

©Copyright by Kanna Shimizu 2002

Itanium™

Pipelining

Time	1	2	3	4	5	6
transaction0	ARB	REQ	SNP	RESP		
transaction1		ARB	REQ	SNP	RESP	
transaction2			ARB	REQ	SNP	RESP
transaction3				ARB	REQ	SNP

32

©Copyright by Kanna Shimizu 2002

Itanium™ — Pipelining

Time	1	2	3	4	5	6
transaction0	ARB	REQ	SNP	RESP		
transaction1		ARB	REQ	SNP	RESP	
transaction2			ARB	REQ	SNP	RESP
transaction3				ARB	REQ	SNP

33

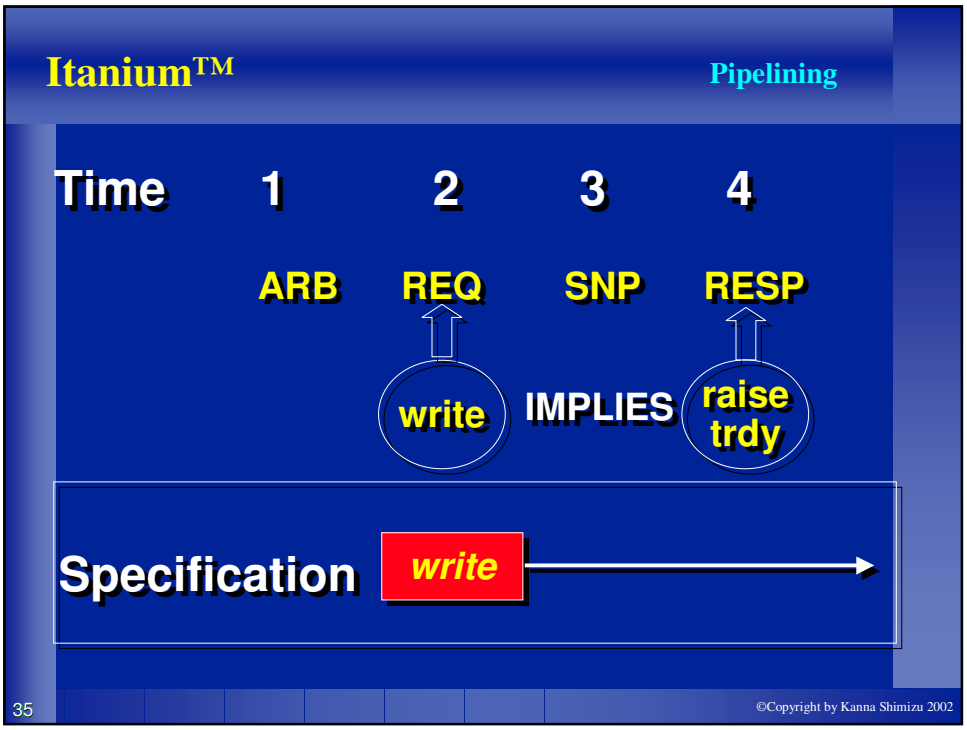
©Copyright by Kanna Shimizu 2002

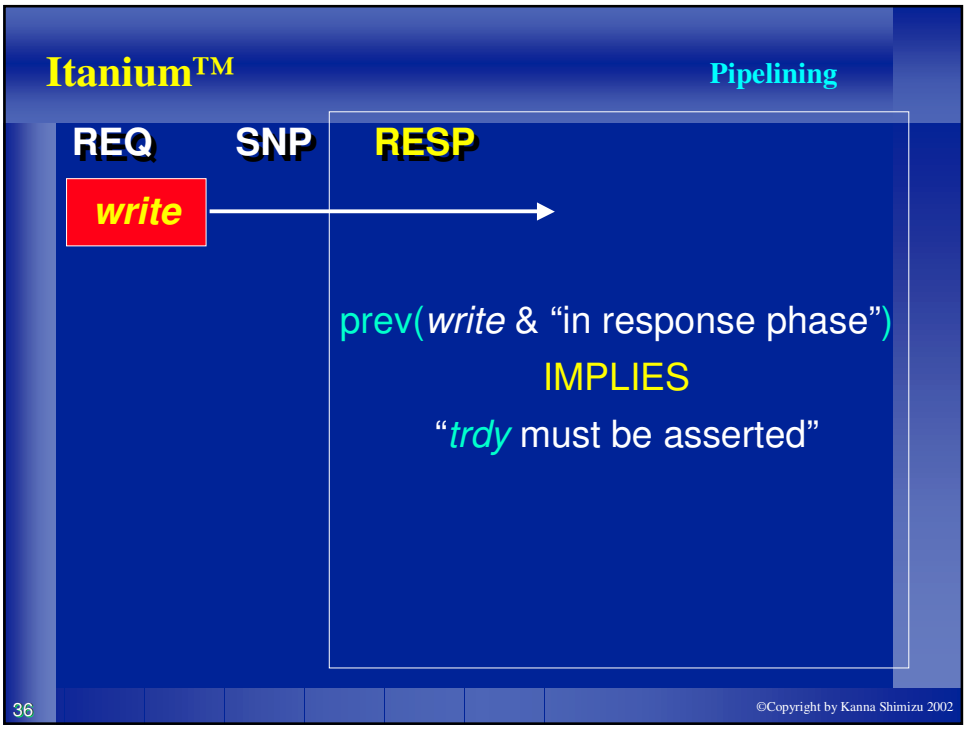
Itanium™ — Pipelining

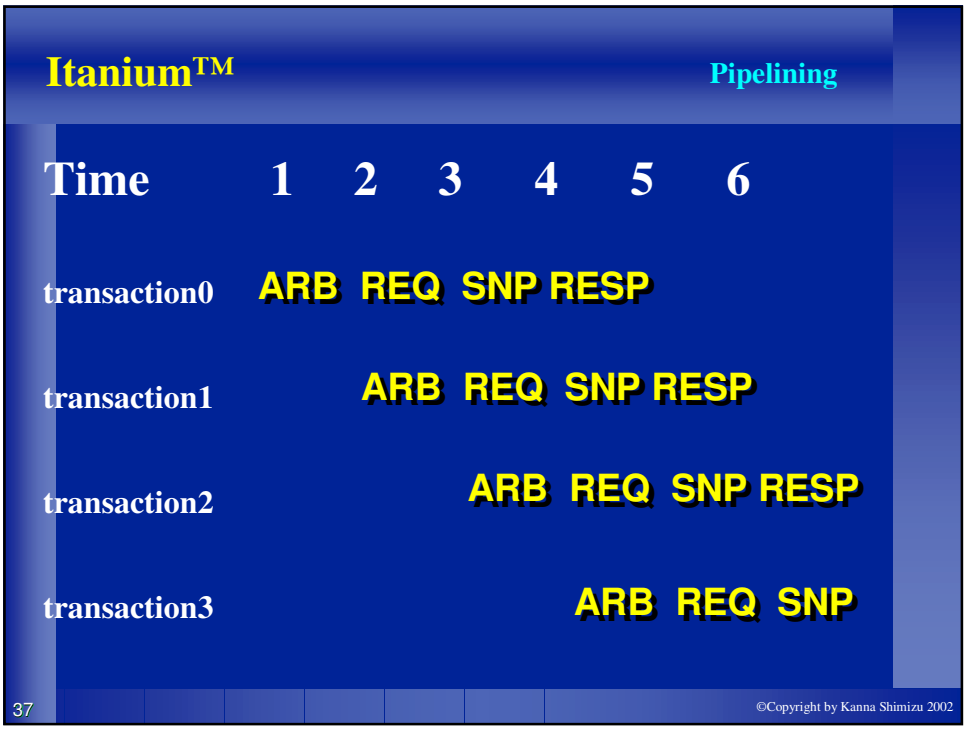
Time	1	2	3	4
	ARB	REQ	SNP	RESP

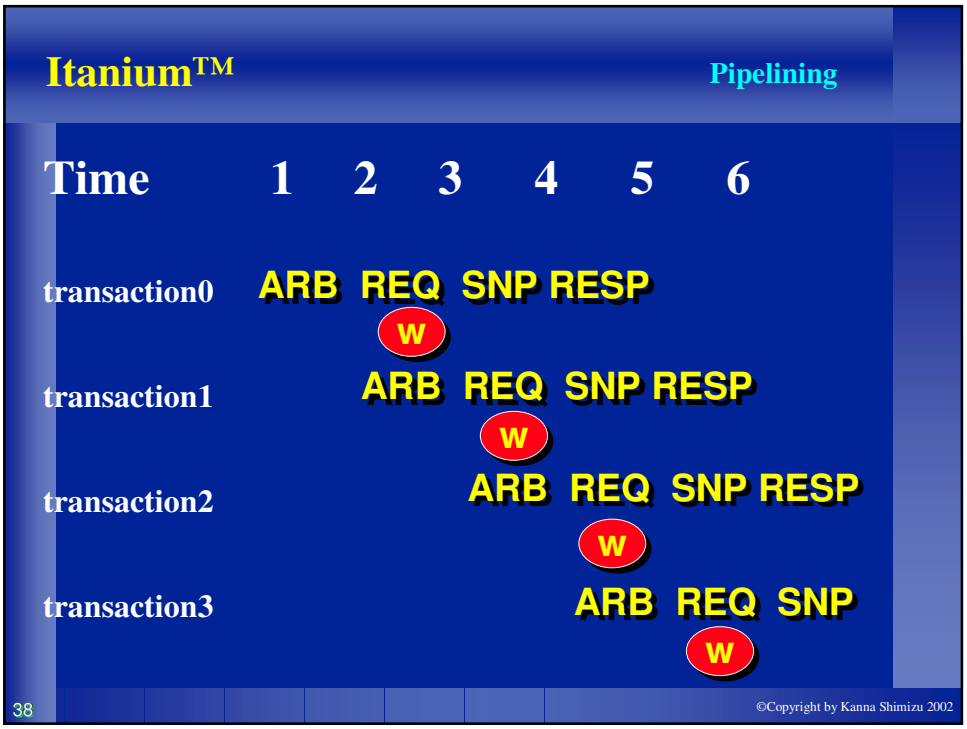
34

©Copyright by Kanna Shimizu 2002


Itanium™
Pipelining
Time 1 2 3 4
ARB REQ SNP RESP
write
IMPLIES
raise trdy
Specification
write
35
©Copyright by Kanna Shimizu 2002


Itanium™
Pipelining
REQ SNP RESP
write
prev(write & "in response phase")
IMPLIES
"trdy must be asserted"
36
©Copyright by Kanna Shimizu 2002


Itanium™
Pipelining
Time 1 2 3 4 5 6
transaction0 ARB REQ SNP RESP
transaction1 ARB REQ SNP RESP
transaction2 ARB REQ SNP RESP
transaction3 ARB REQ SNP
37
©Copyright by Kanna Shimizu 2002


Itanium™
Pipelining
Time 1 2 3 4 5 6
transaction0 ARB REQ SNP RESP
W
transaction1 ARB REQ SNP RESP
W
transaction2 ARB REQ SNP RESP
W
transaction3 ARB REQ SNP
W
38
©Copyright by Kanna Shimizu 2002

Itanium™

Pipelining

©Copyright by Kanna Shimizu 2002

Itanium™

Pipelining

©Copyright by Kanna Shimizu 2002

Itanium™

Pipelining

Multiple Instantiations

prev(*write* & "in response phase") IMPLIES "*trdy* must be asserted"

prev((RESP = 0) & write_*0*) IMPLIES "*trdy* must be asserted"

prev((RESP = 1) & write_*1*) IMPLIES "*trdy* must be asserted"

prev((RESP = 2) & write_*2*) IMPLIES "*trdy* must be asserted"

©Copyright by Kanna Shimizu 2002

Outline

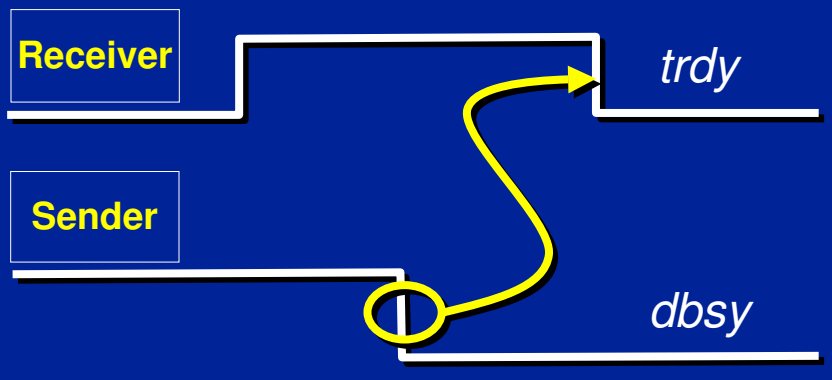
1. Methodology
2. Specifying the Itanium™ processor bus protocol
3. Checking the protocol

©Copyright by Kanna Shimizu 2002

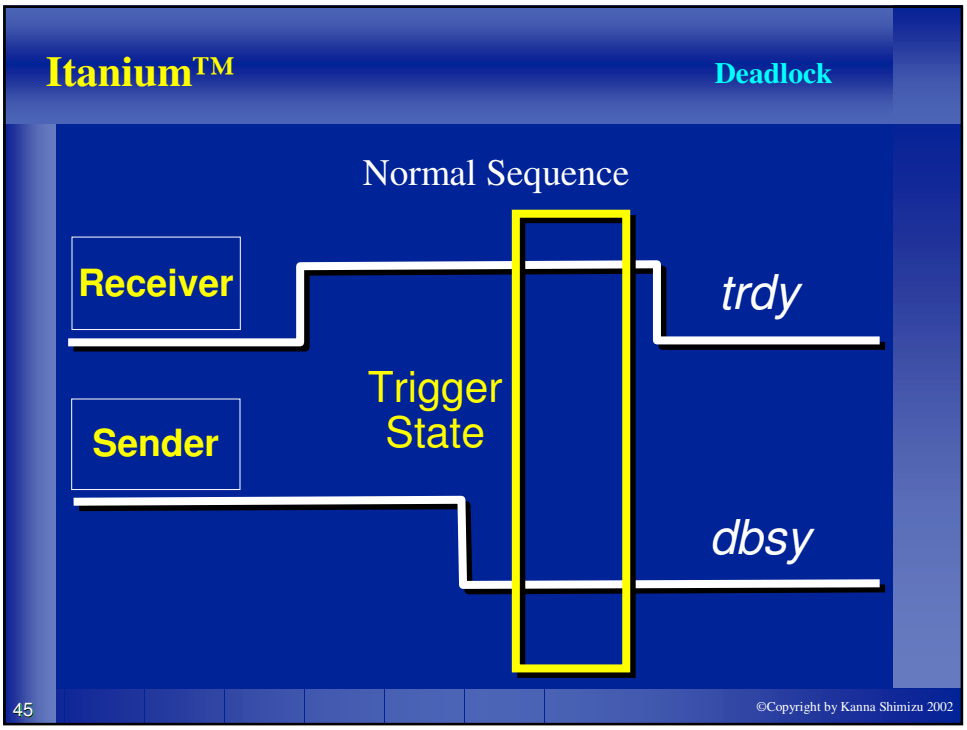
Deadlock Found in Protocol

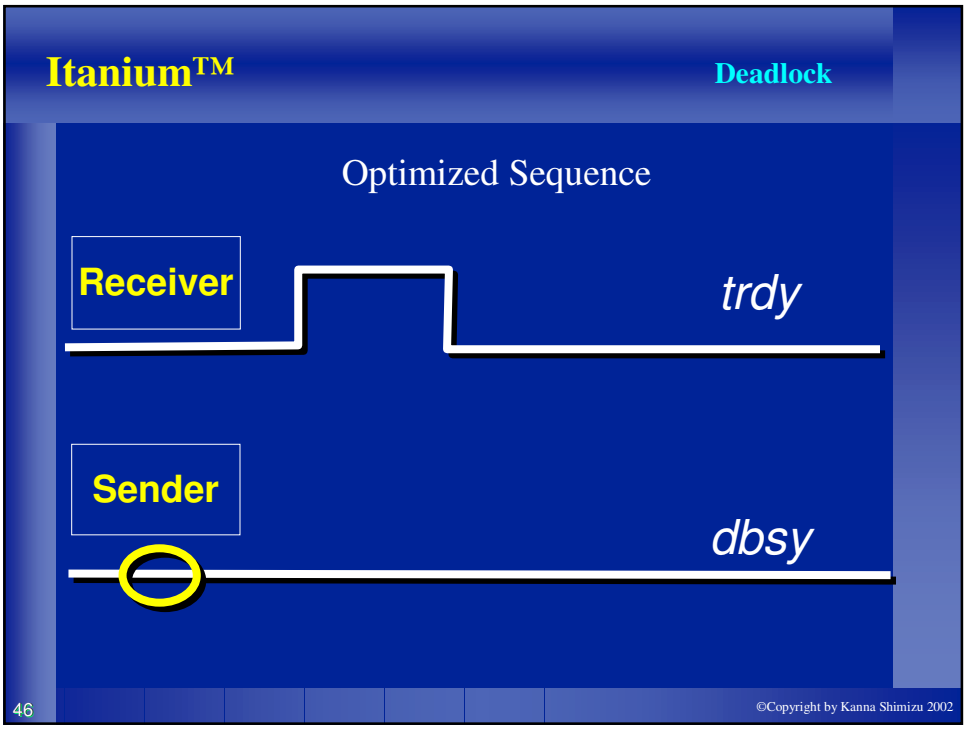
©Copyright by Kanna Shimizu 2002

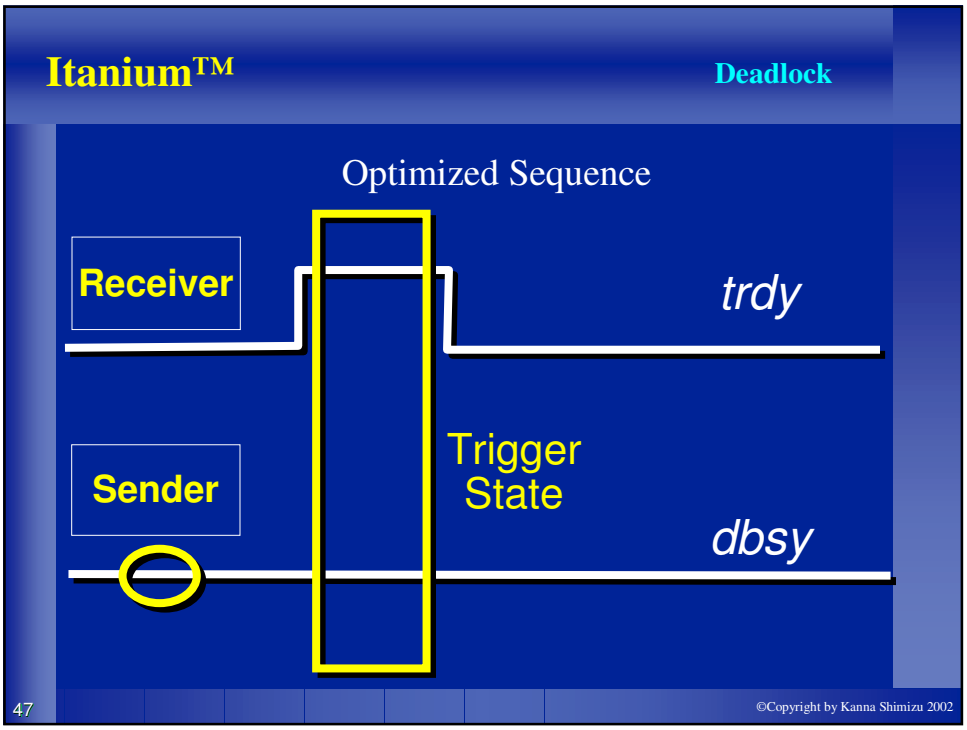
Normal Sequence

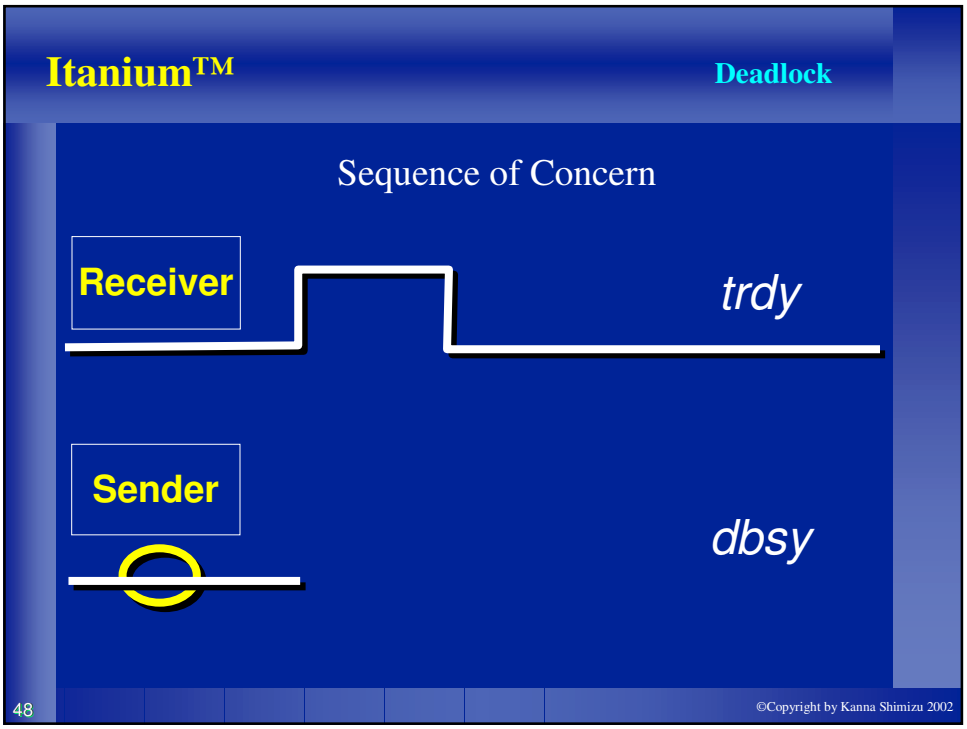


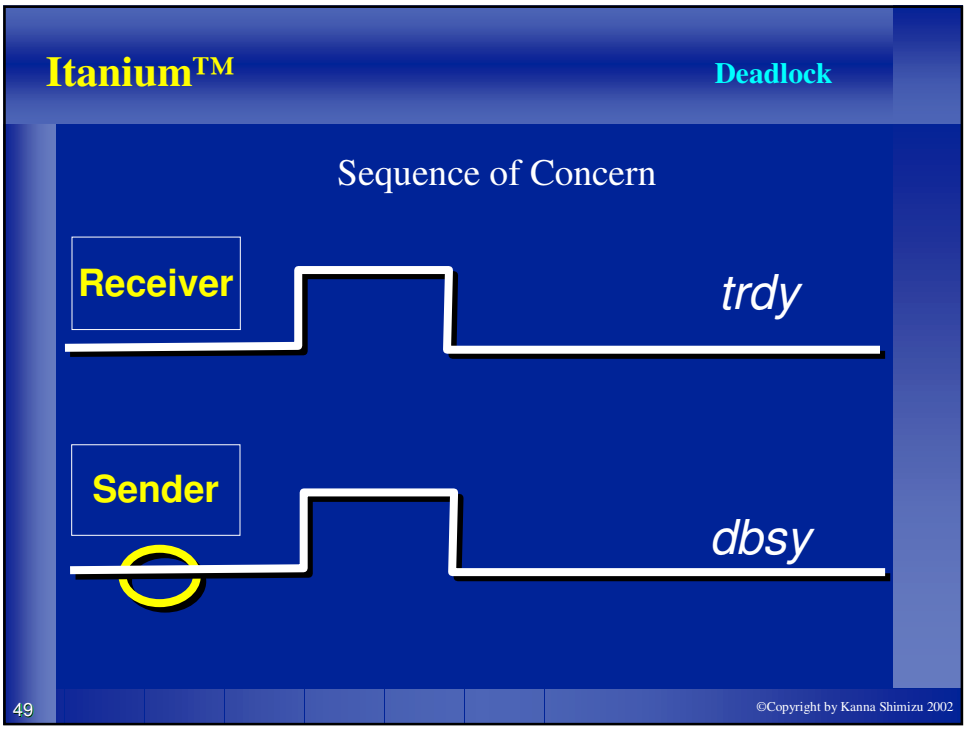
©Copyright by Kanna Shimizu 2002

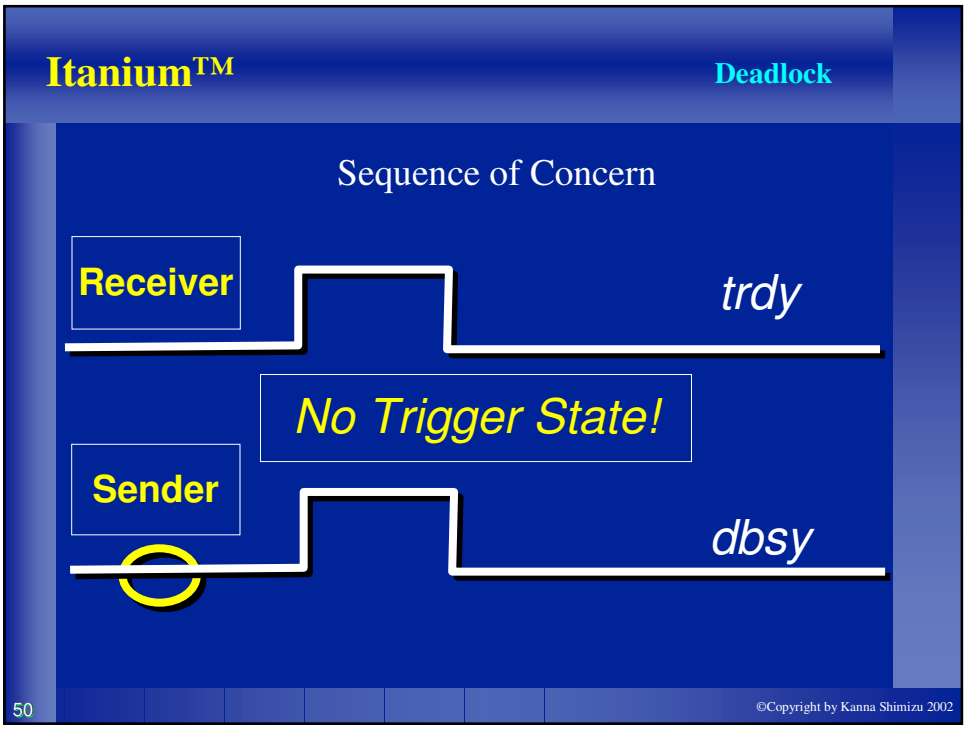

Itanium™
Deadlock
Normal Sequence
Receiver
Sender
Trigger State
trdy
dbsy
45
©Copyright by Kanna Shimizu 2002

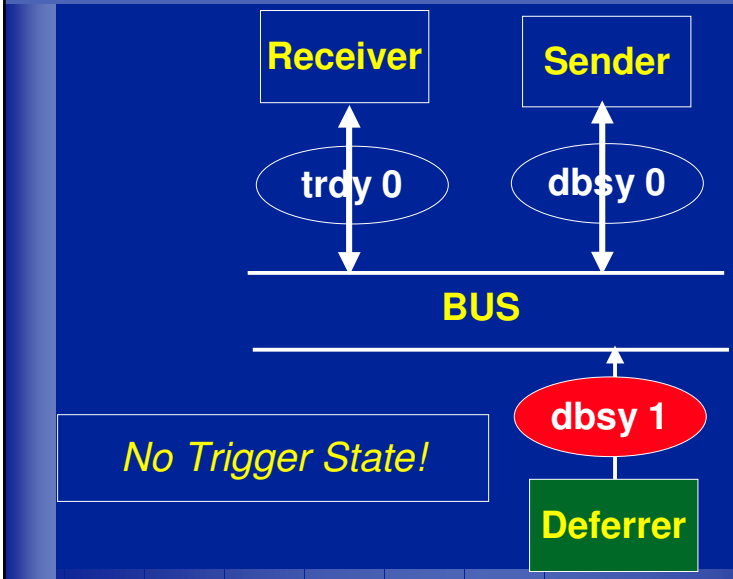

Itanium™
Deadlock
Optimized Sequence
Receiver
Sender
trdy
dbsy
46
©Copyright by Kanna Shimizu 2002


Itanium™
Deadlock
Optimized Sequence
Receiver
trdy
Trigger
State
Sender
dbsy
47
©Copyright by Kanna Shimizu 2002


Itanium™
Deadlock
Sequence of Concern
Receiver
trdy
Sender
dbsy
48
©Copyright by Kanna Shimizu 2002


Itanium™
Deadlock
Sequence of Concern
Receiver
trdy
Sender
dbsy
49
©Copyright by Kanna Shimizu 2002


Itanium™
Deadlock
Sequence of Concern
Receiver
trdy
No Trigger State!
Sender
dbsy
50
©Copyright by Kanna Shimizu 2002



©Copyright by Kanna Shimizu 2002

- Consequences
  - Data transfer permanently blocked
  - Deadlock in some cases

Fixed

©Copyright by Kanna Shimizu 2002

Itanium™ Summary

- Experience specifying and checking
  - Two issues found with the protocol
    - Resulted in changes to the protocol
  - Formal specification written and debugged in two man-months
    - Partial description
    - Core protocol in 46 independent constraints

©Copyright by Kanna Shimizu 2002

Conclusion

1. Methodology
   - Simple yet effective
2. Specifying the Itanium™ bus protocol
   - Advanced, with pipelining
3. Checking the protocol
   - Issues discovered
   - Formal work fed back to standard rectification

©Copyright by Kanna Shimizu 2002

Future Directions

- Complete Specification
- Extract Useful Information
  - For synthesis
  - "Don't Care"s

©Copyright by Kanna Shimizu 2002