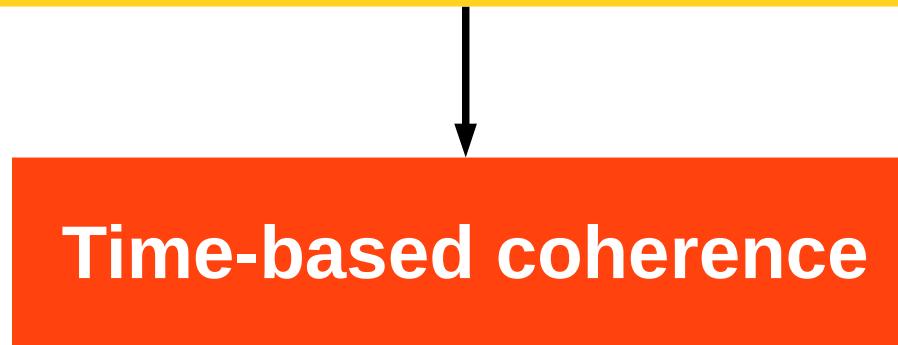


Time-Based Memory Coherence

Alan Mujumdar

Hypotheses

Cache coherence without coherence messaging



Runs existing software

Competitive performance

Complies with
RMO★ consistency

Offers cache
side-channel masking

Memory Consistency

Dual 386



ARM



Intel x86



SPARC TSO



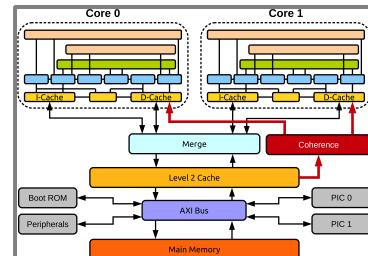
PowerPC



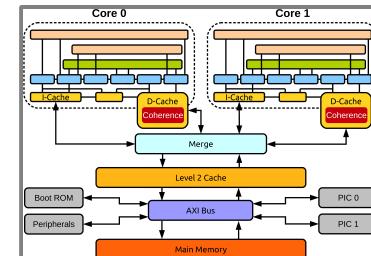
DEC Alpha



BERI: Directory



BERI: Time-Based



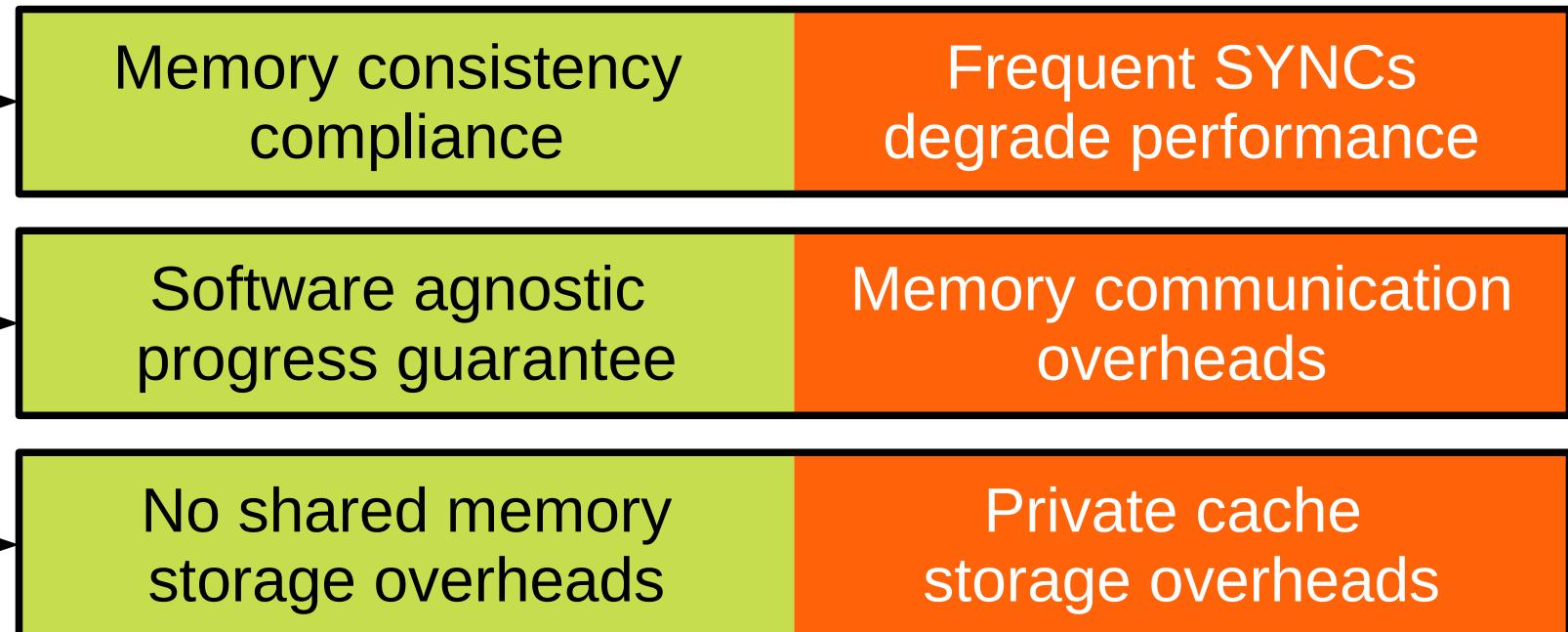
BERI Deliverables



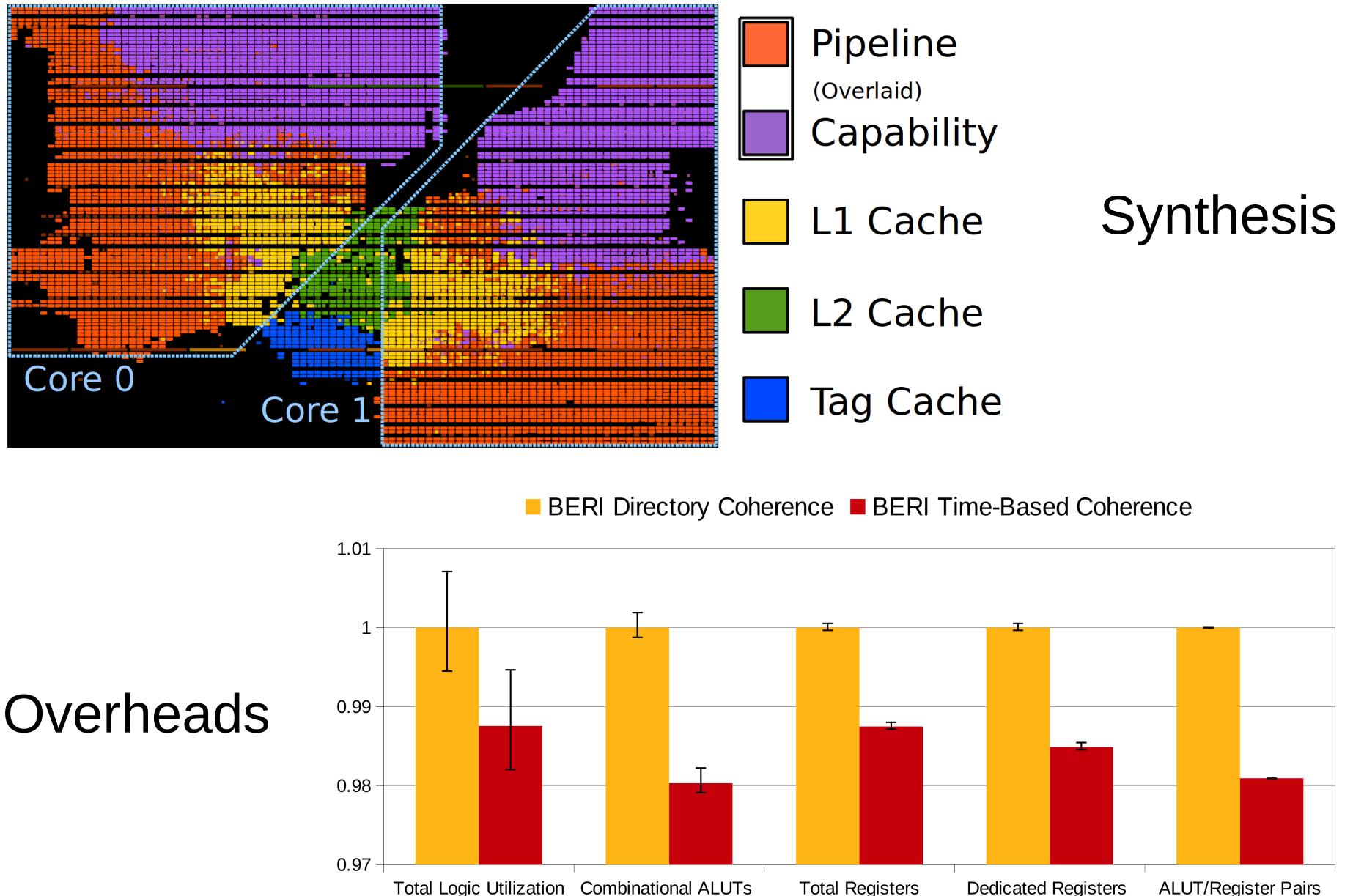
- Testing: CHERI-Test, AXE, CHERI Litmus
- Evaluation: FreeBSD, Splash-2, CHERI Litmus
- Additional evaluation: side-channel attacks

Time-Based Coherence

- Each cache line has a **lifespan** – timer imposed
- Reading an expired line causes a **reload**
- Synchronisation instructions soft-**flush** the cache

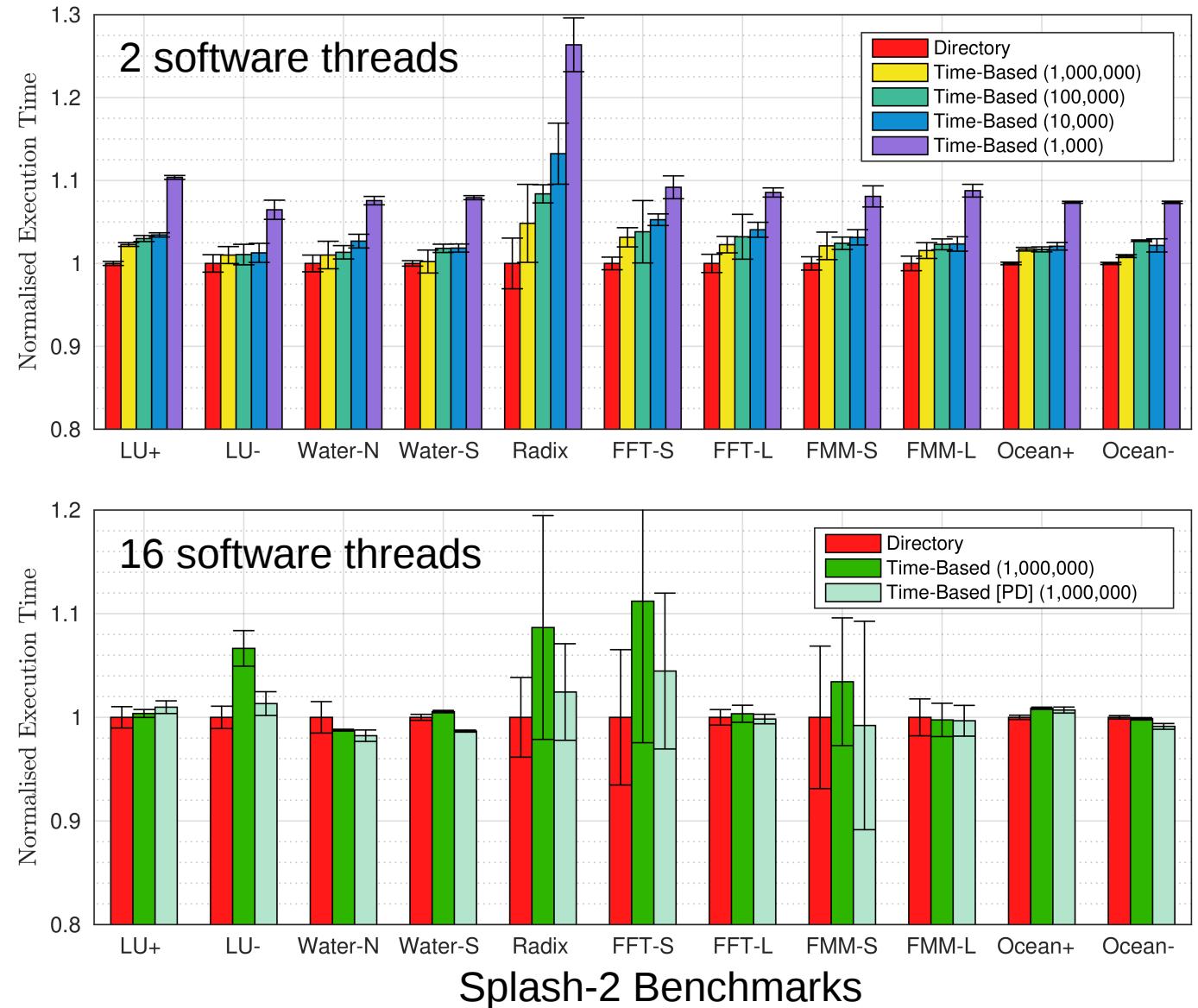


FPGA Complexity



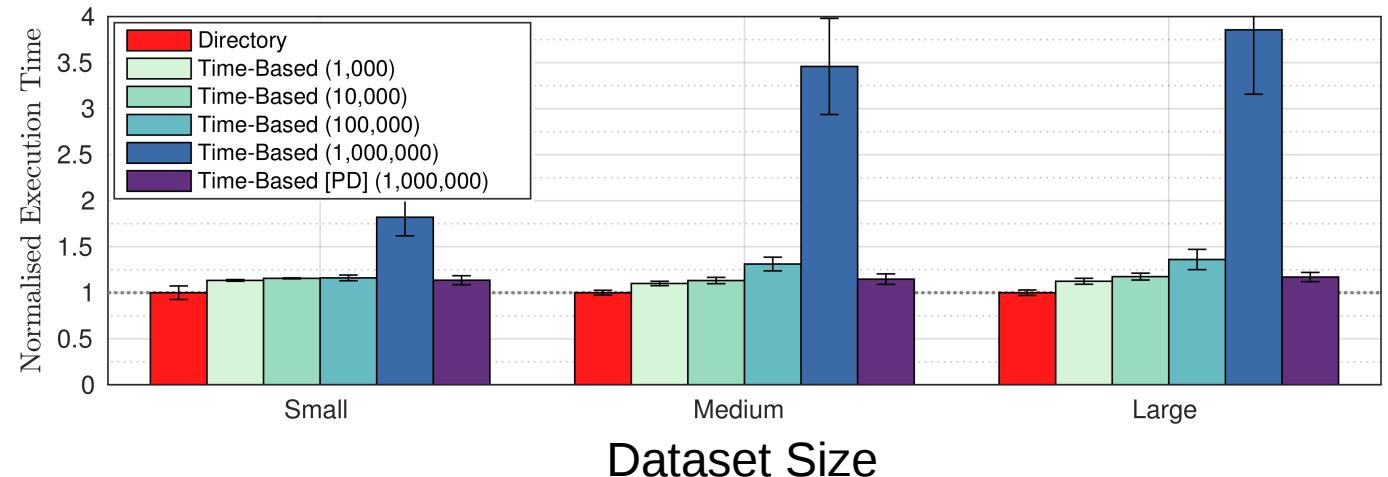
Results

- Performance
- Optimisation
- Workload
- Cache size
- Scalability
- Energy
- Simplicity

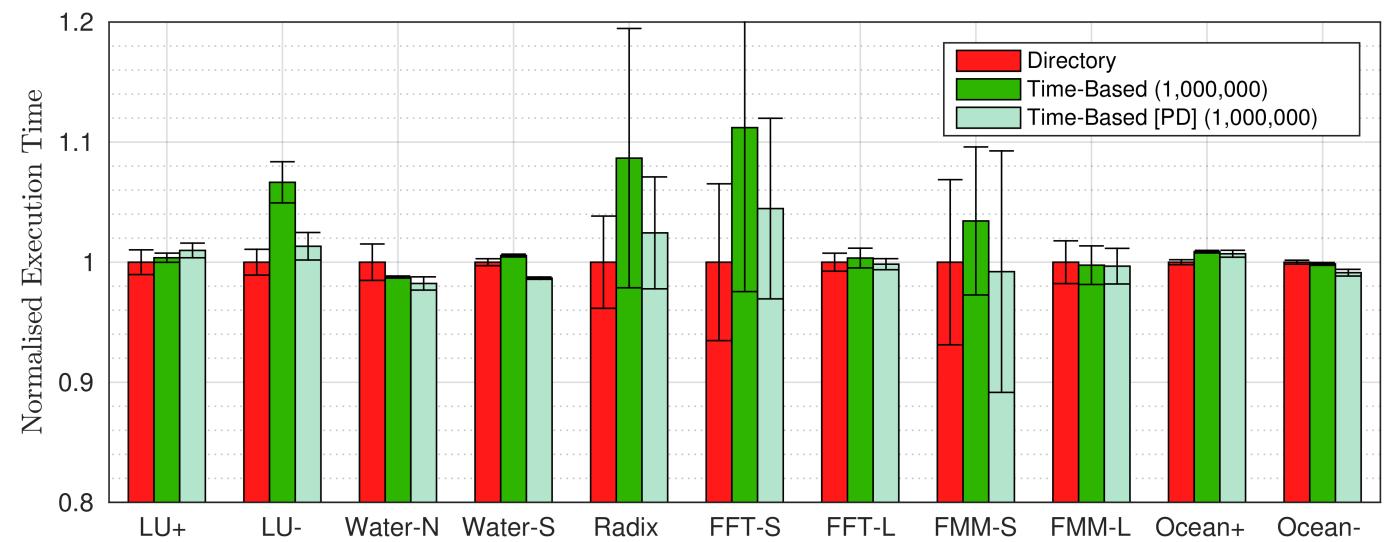


Results

CP on FreeBSD
Polling hampers the time-based model



Splash-2
Polling detection is still beneficial



Time-Based – Polling Detection

No polling detector

LL/SC loop

```
1: lld      $8, 0(%0) ←  
2: daddu   $8, $8, %1  
3: scd     $8, 0(%0)  
4: beqz    $8, 1b      {Branch to 1}
```

Polling loop

```
6: ld       $8, 0(%0) ←  
7: bne    $8, %2, 6b {Branch to 6}  
8: nop  
9: sync
```

No issues for hardware supporting **strong memory consistency**

Time-Based – Polling Detection

No polling detector

LL/SC loop

```
1: lld      $8, 0(%0)
2: daddu   $8, $8, %1
3: scd      $8, 0(%0)
4: beqz    $8, 1b      {Branch to 1}
5: nop
```

Polling loop

```
6: sync*
7: ld       $8, 0(%0)
8: bne    $8, %2, 6b {Branch to 6}
9: nop
10: sync
```

Software solution for time-based coherence polling issue

Time-Based – Polling Detection

Polling detector used

LL/SC loop

```
1: lld      $8, 0(%0)
2: daddu   $8, $8, %1
3: scd      $8, 0(%0)
4: beqz    $8, 1b      {Branch to 1}
5: nop
```

Polling loop

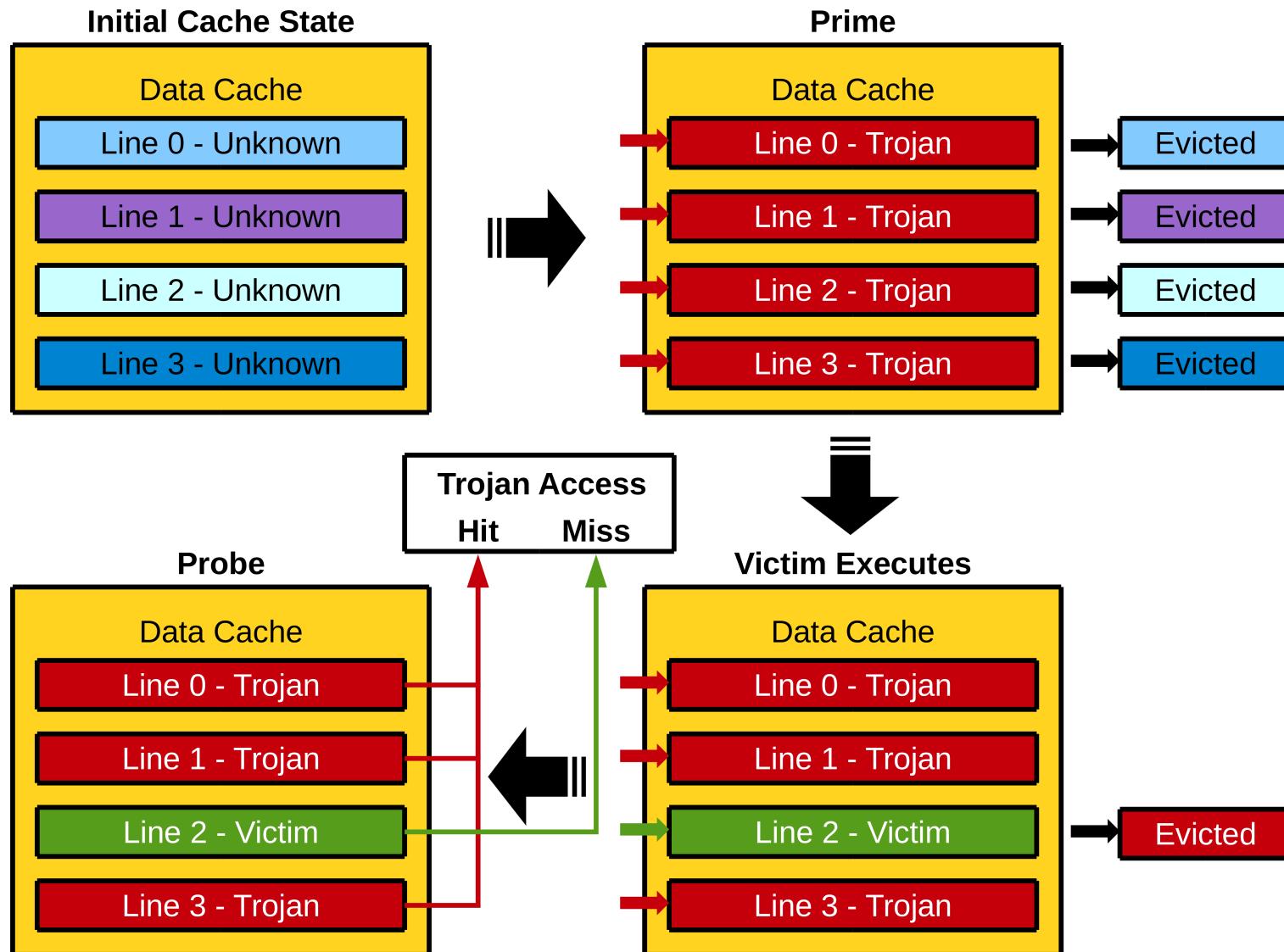
```
6: ld       $8, 0(%0)
7: bne    $8, %2, 6b {Branch to 6}
8: nop
9: sync
```

L2 cache access

Read to unmodified
data detected

Hardware solution for time-based coherence polling issue

Cache Side-Channel Attacks



Cache Side-Channel Attacks

- Directory-base coherence

Displays ideal leakage

- Time-base coherence

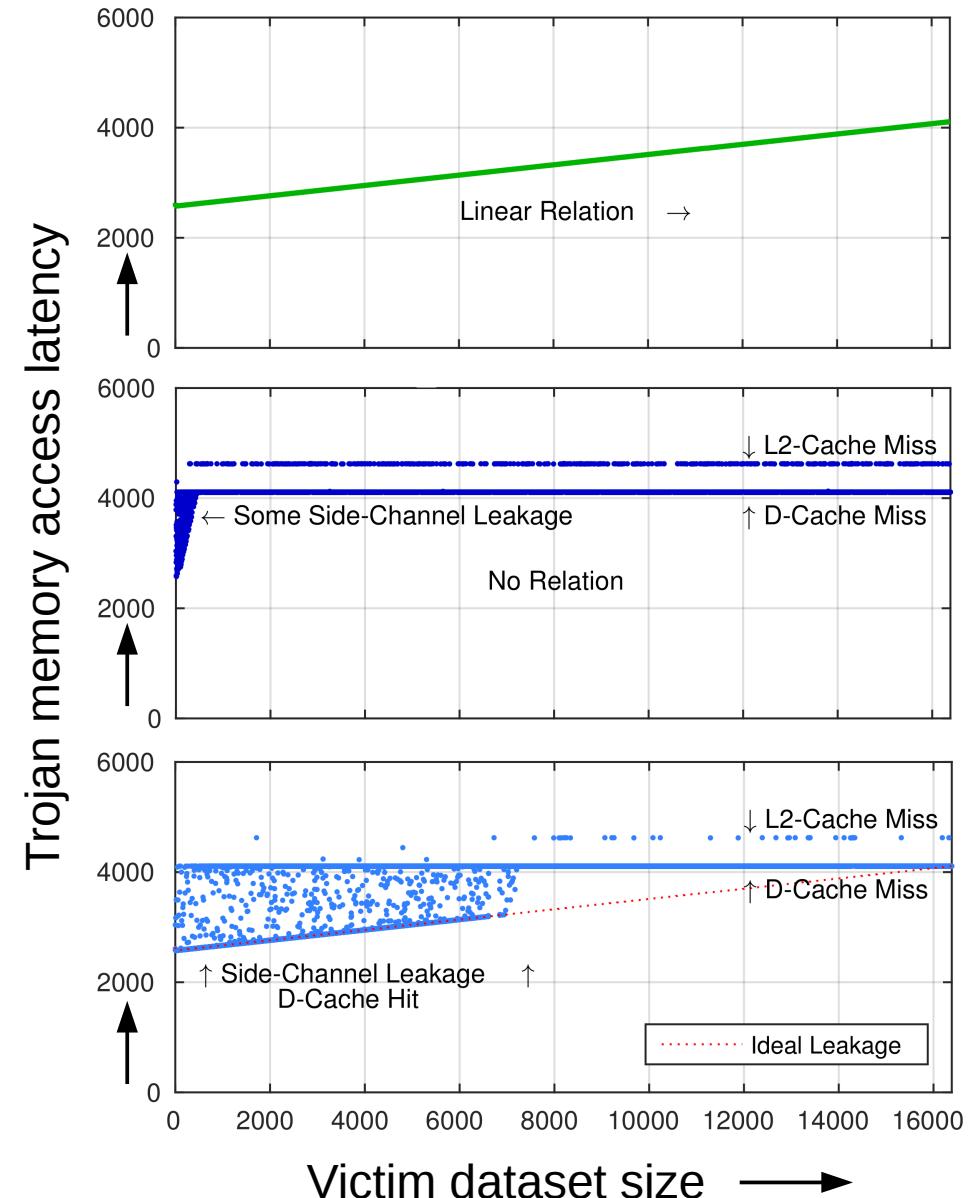
Time-out: 10,000 cycles

Frequent self-invalidates

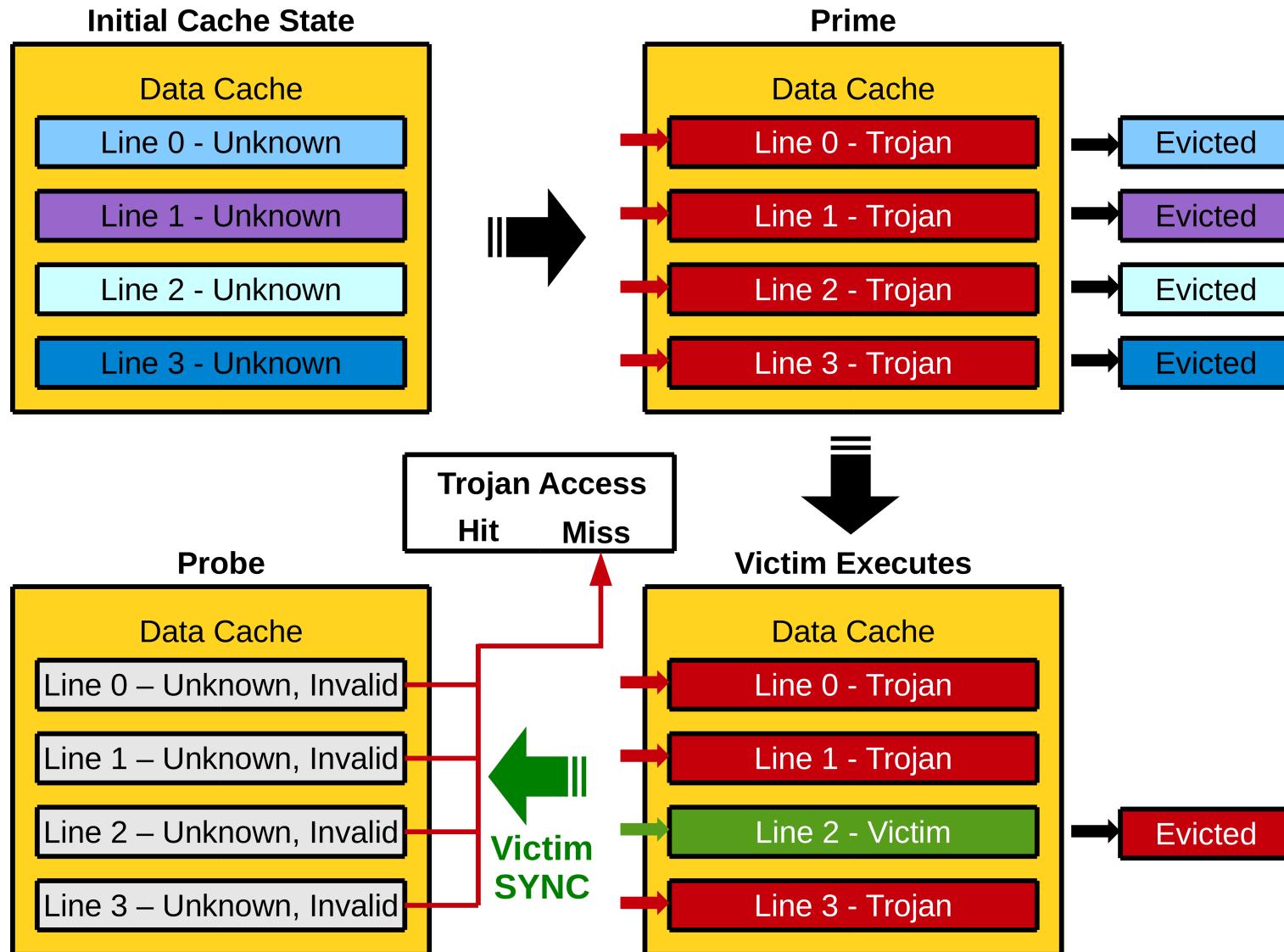
- Time-base coherence

Time-out: 100,000 cycles

Less frequent self-invalidates



Cache Side-Channel Attacks



Conclusion

- Coherence without messaging
- Time-based self-invalidation works
- FreeBSD runs time-based coherence
- Performance can be competitive
- Provides side-channel masking

Two BERI
multiprocessor
designs

Bringing up
FreeBSD on
multi-BERI

BERI
testing

Additional Slides

Time-Based Coherence

