bittide: Control Time, Not Flows

First Time Hardware Implementation and Validation

Martijn Bastiaan[1] ⟨martijn@qbaylogic.com⟩
Christiaan Baaij[1] ⟨christiaan@qbaylogic.com⟩
Martin Izzard[2] ⟨izzard@google.com⟩
Felix Klein[1] ⟨felix@qbaylogic.com⟩
Sanjay Lall[2,3] ⟨sanjaylall@google.com⟩
Tammo Spalink[2] ⟨tammo@google.com⟩

Hot Interconnects, 2025

[1]QBayLogic
[2]Google DeepMind
[3]Department of Electrical Engineering, Stanford University

This talk

- Introduction to bittide
- What hardware we built
- What we measured

Introduction

bittide

Clock cycle accurate synchronization at datacenter scales

Promises

Ahead of time scheduling

Promises

Ahead of time scheduling

Zero in-band overhead

Promises

Ahead of time scheduling

Zero in-band overhead

Decentralized

Promises

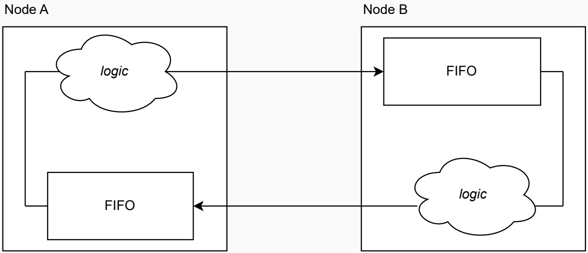
Ahead of time scheduling

Zero in-band overhead

Decentralized

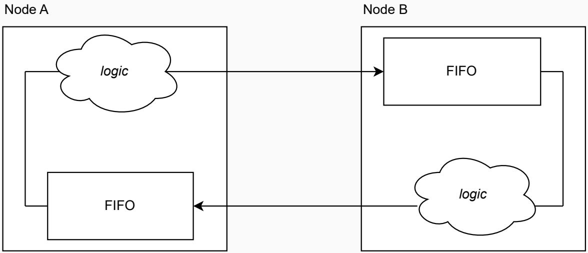
Simple

How



FIFO on each incoming link

How



FIFO on each incoming link

Must not under or overflow

Clock control

Start with FIFOs half-full

Clock control

Start with FIFOs half-full

Speed up when buffers fill

Clock control

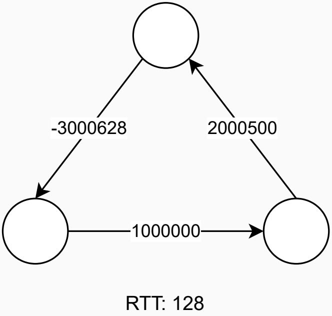
Start with FIFOs half-full

Speed up when buffers fill
Slow down when buffers drain

Clock control

Start with FIFOs half-full

Speed up when buffers fill
Slow down when buffers drain



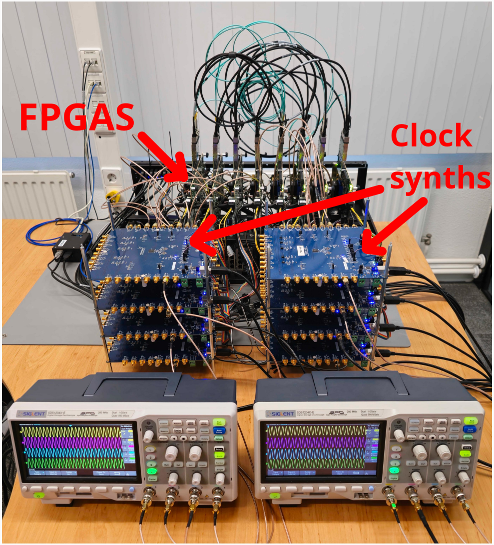
Implementation

System architecture

8 fully connected FPGAs

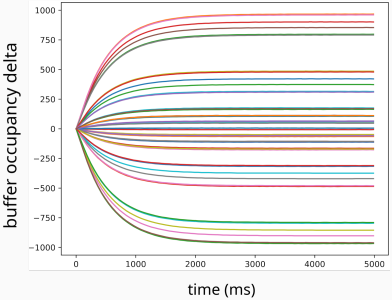
8 clock synthesizers

External measuring equipment

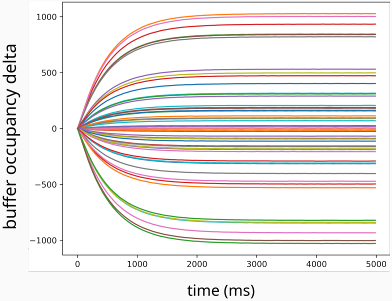


Measurements and validation

Fully connected

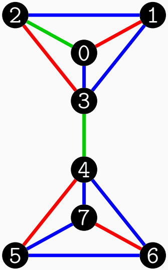


2m

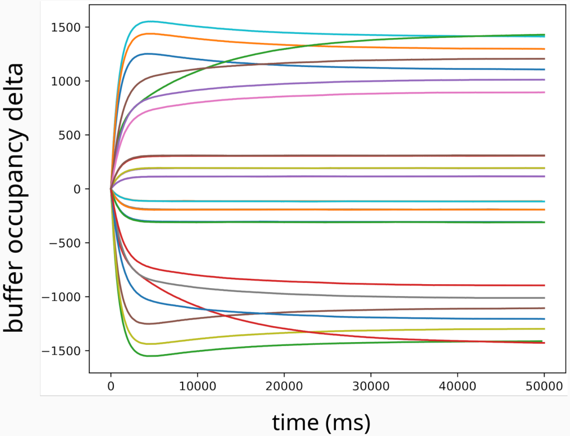


2km

Hourglass



2
1
0
3
4
7
5
6

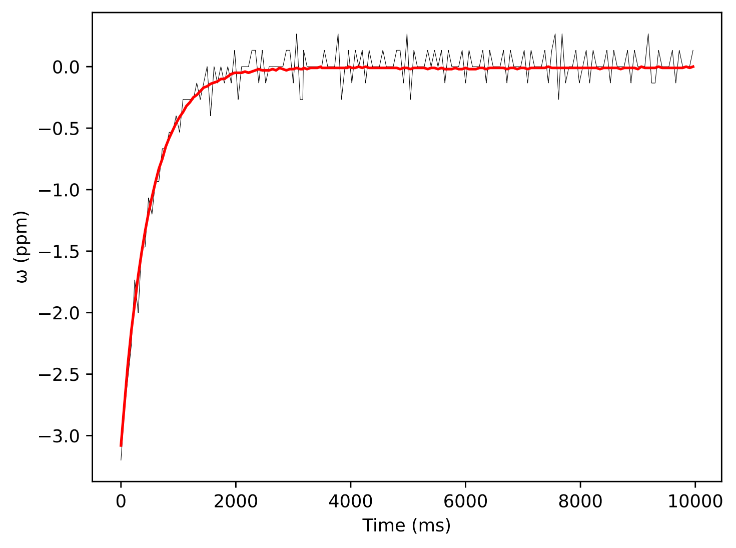


1500
1000
500
0
−500
−1000
−1500
0
10000
20000
30000
40000
50000
buffer occupancy delta
time (ms)

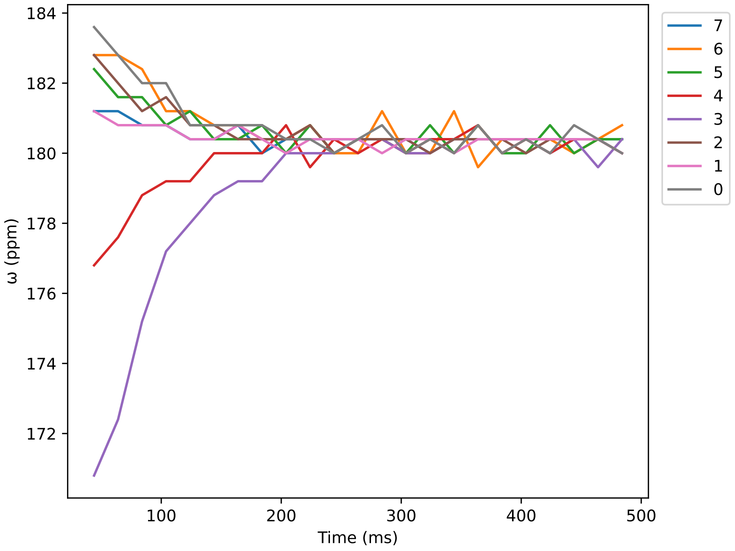
RTTs in clock cycles

FPGA	1	2	3	4	5	6	7	8
1								
2	68							
3	**1299**	68						
4	67	68	68					
5	68	68	68	70				
6	69	67	68	69	69			
7	69	70	69	68	69	68		
8	69	69	69	70	69	70	68	

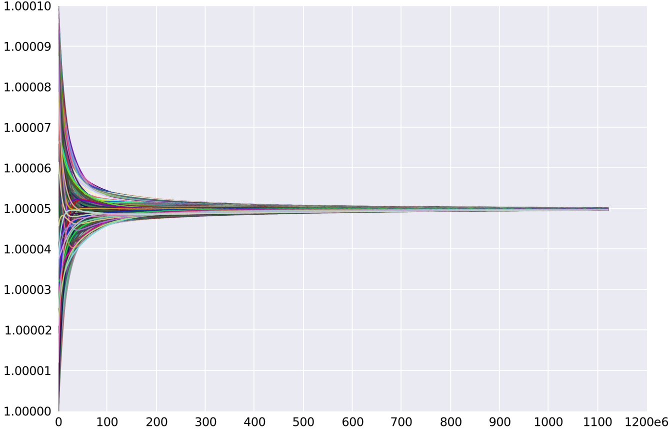
Comparison against model



Aggressive gain



Scalability



Conclusion

Summary and contributions

bittide synchronization scheme:

- ahead of time clock cycle scheduling
- zero in-band overhead
- decentralized
- simple

Summary and contributions

bittide synchronization scheme:

- ahead of time clock cycle scheduling
- zero in-band overhead
- decentralized
- simple

First hardware implementation

- only off the shelve components
- model very closely matches measurements
- clock control incredibly stable

Thank you

Open source resources

- **Project website:** `www.bittide.io`
- **Hardware repository:** `github.com/bittide/bittide-hardware`
- **Clock control simulation:** `github.com/bittide/Callisto.jl`

Questions?