Relative Timing Driven Multi-Synchronous Design: Enabling Order-of-Magnitude Energy Reduction

Kenneth S. Stevens

University of Utah

Granite Mountain Technologies

Learn from Prof. Kajitana

- Think differently and **deeply**
- Apply thought to current challenges

Then **collaborate**

Goals of Presentation:

1. Define and propose “rule breaker” idea
2. Request support from physical design community

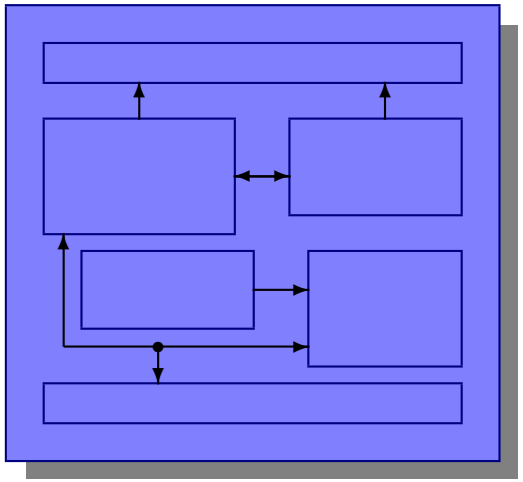
Multi-Synchronous Advantage

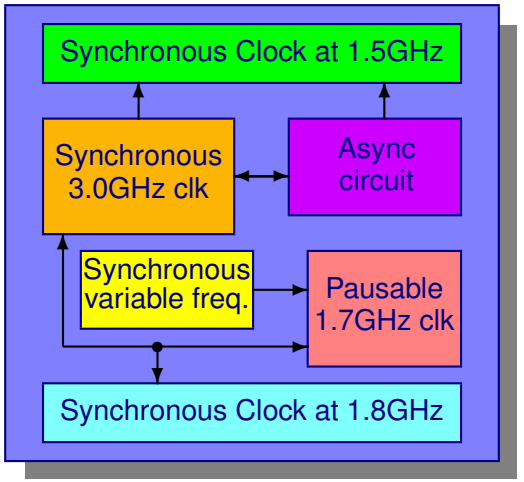
1. **Efficiency** in **power** and **performance** is new game in town
2. Multi-synchronous design provides optimization opportunity
3. New (asynchronous) timing model is one excellent path
4. Produces average 10× $e\tau^2$ improvement
   - Pentium: $e\tau^2 = 17.5\times$
   - FFT: $e\tau^2 = 16.9\times$
5. But ... ***need improved physical design support***

Design	Energy	Area	Freq.	Latency	Aggregate
Pentium F.E.	2.05	*0.85*	2.92	2.38	12.11×
64-pt FFT	3.95	2.83	2.07	3.37	77.98×

Timing is a Key Issue

Multi-synchronous design produces best results





Single frequency, low skew
(small blocks, standard CAD)

1. global block frequencies
2. higher clock power
3. clock design, distribution

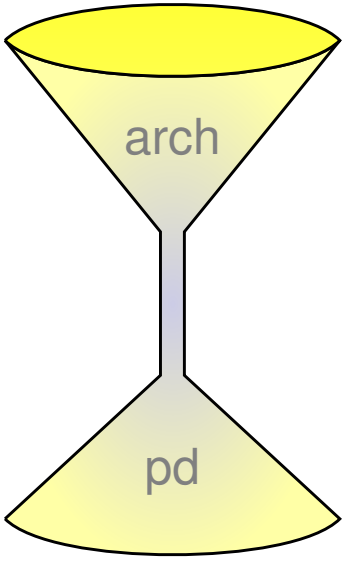
Multiple frequencies
(SoC reality – localization)

1. blocks operate at best frequency
2. network not synchronized
3. synchronizing FIFOs

Energy Efficient Design

Wine goblet model:

- Energy efficiency has two primary sources
  - System architecture
  - Physical design
- Methodology and CAD unify sources

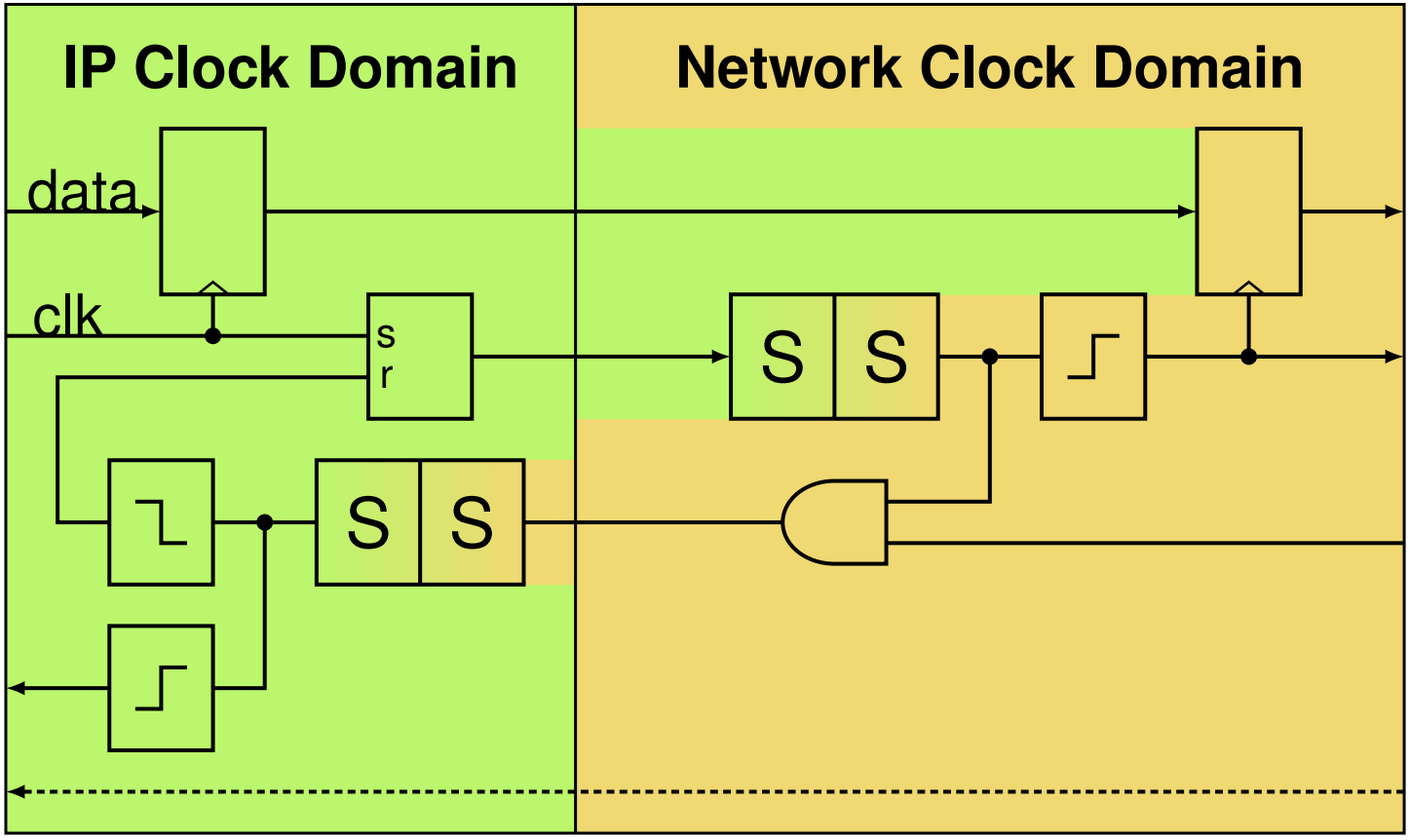


Best realization:

- Multi-synchronous
  - Defined by system's critical path
  - Then optimal local power-delay
  - Asynchronous best methodology:
    - no synchronization cost

Interface Matters!

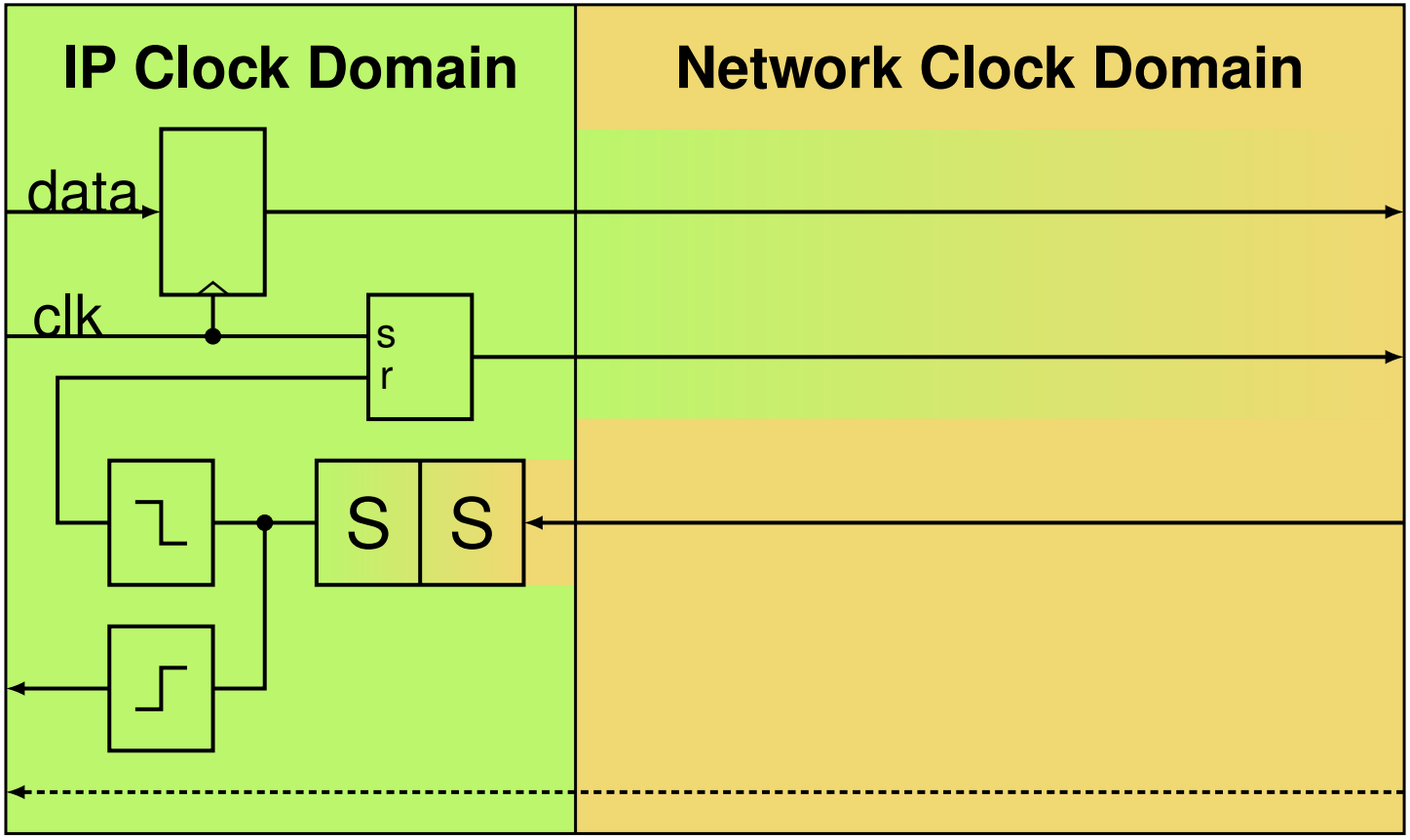
Clocked design requires synchronizers when crossing all domains.



Major location for buffering in a design.

Interface Matters!

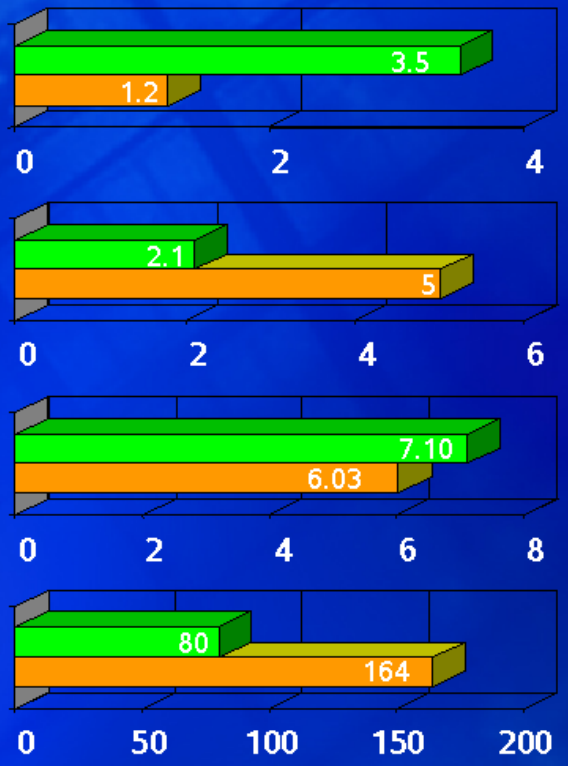
No synchronization required into async domain.



Improves power, performance, and modularity

Timed Asynchronous Designs

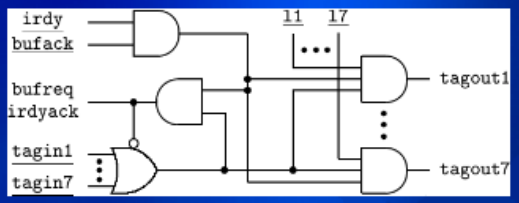
- 1997 RAPPID Si: 0.25µ, 1.8V, 35°C, for common instr
- Comparing to 400MHz Deschutes Processor



RAPPID

Clocked

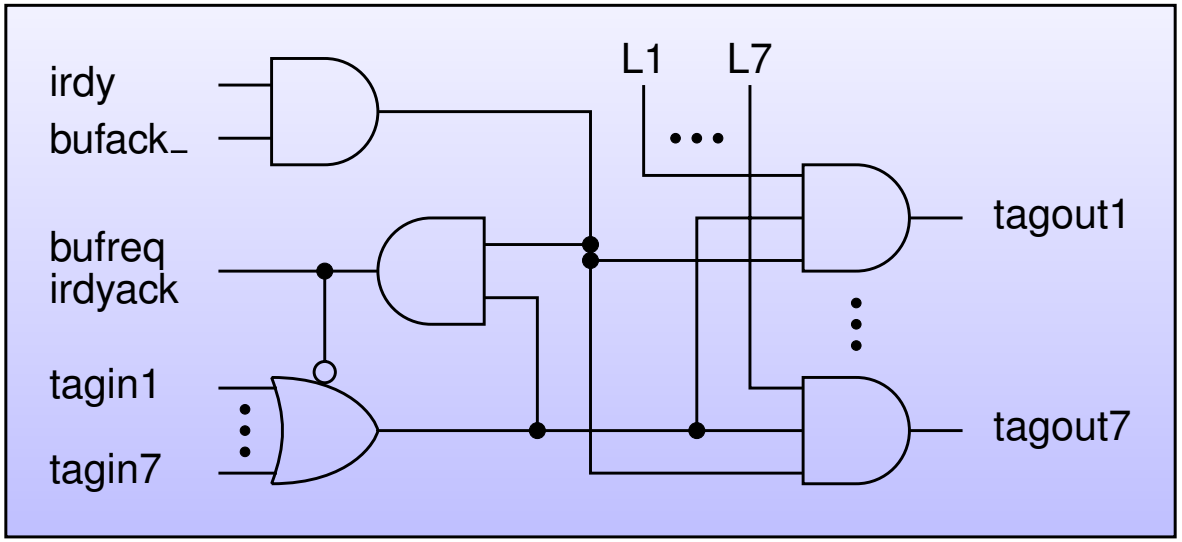
Testability: 95.9% (BIST stuck-at)



Key pipeline circuit

Multi-Synchronous Architecture

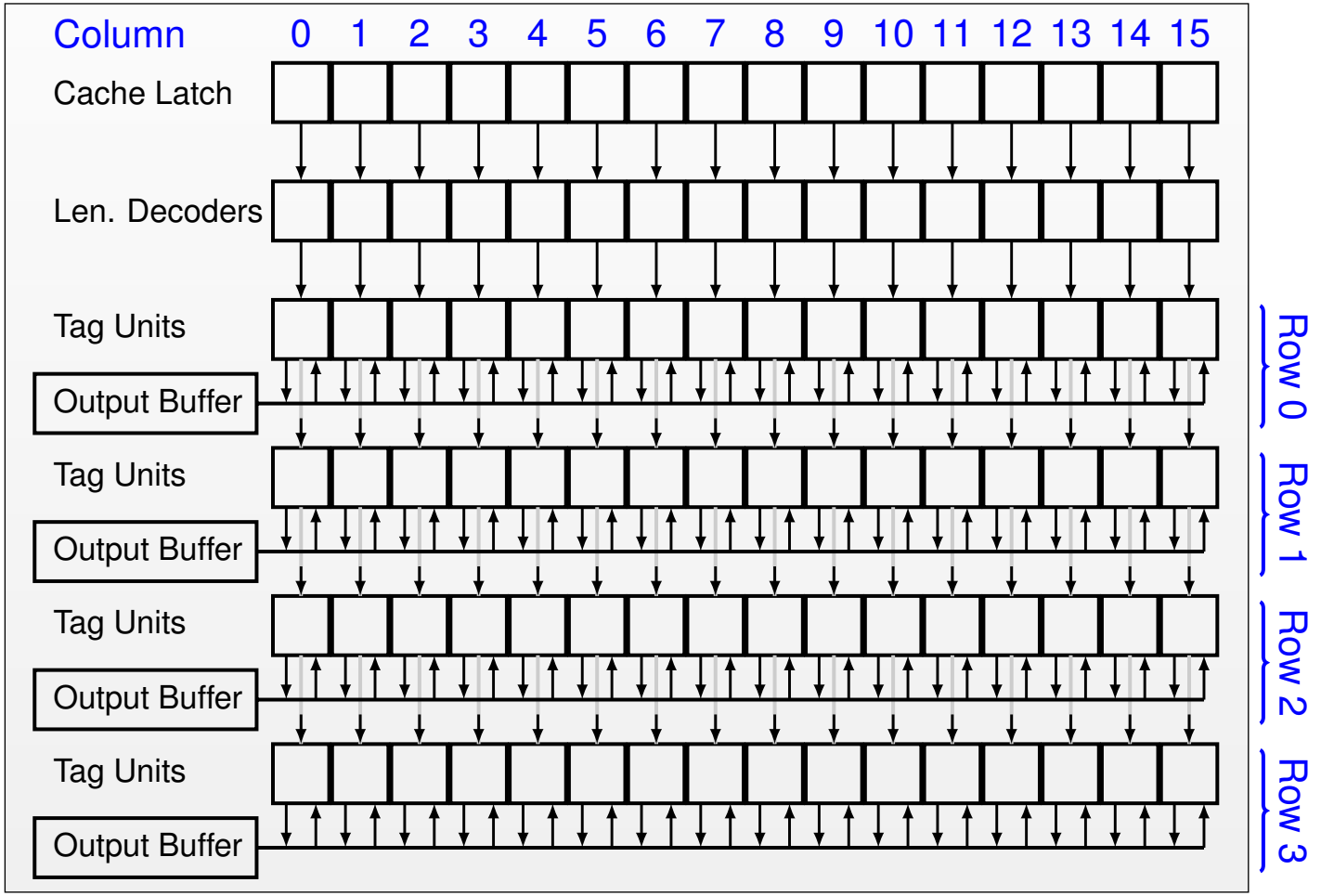
1. Make architectural bottleneck as fast as possible.
2. Make the rest of the design match bottleneck
   - . . . normally as slow as possible
3. Optimize locally for power/performance.



Asynchronous Pentium bottleneck circuit

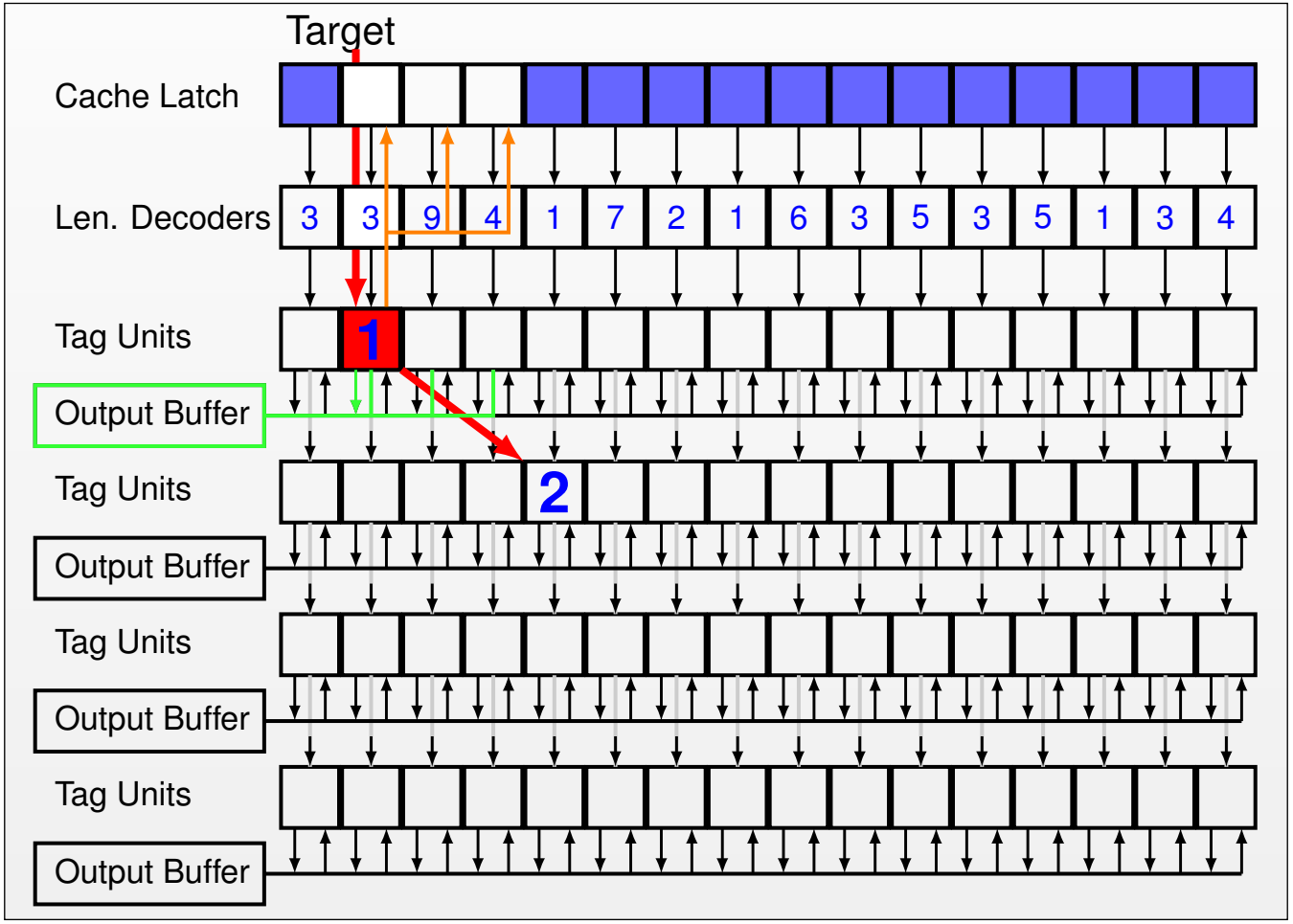
Concurrency and Time

Architectural level timing experiment: Pentium front end



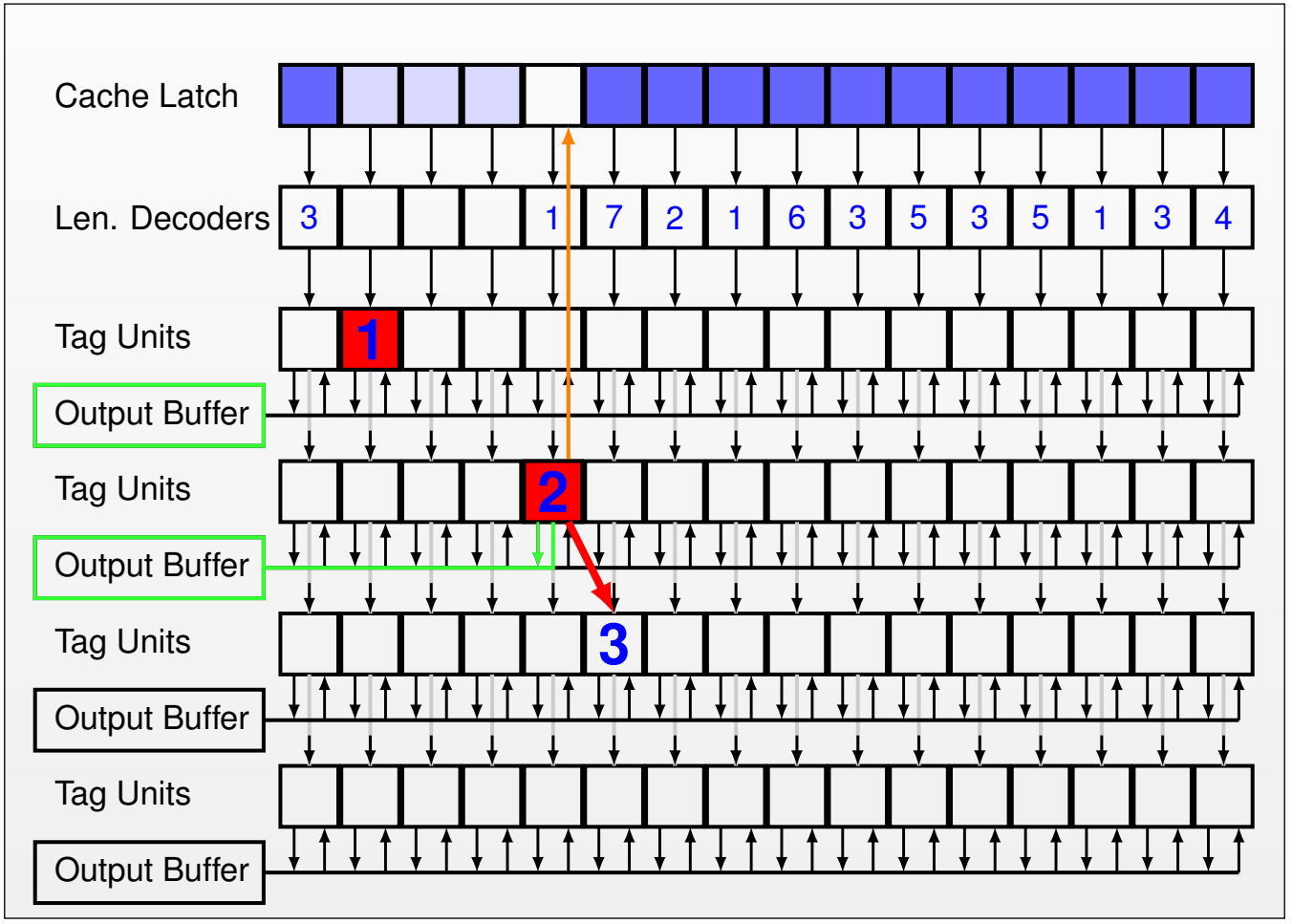
Concurrency and Time

Architectural level timing experiment: Pentium front end



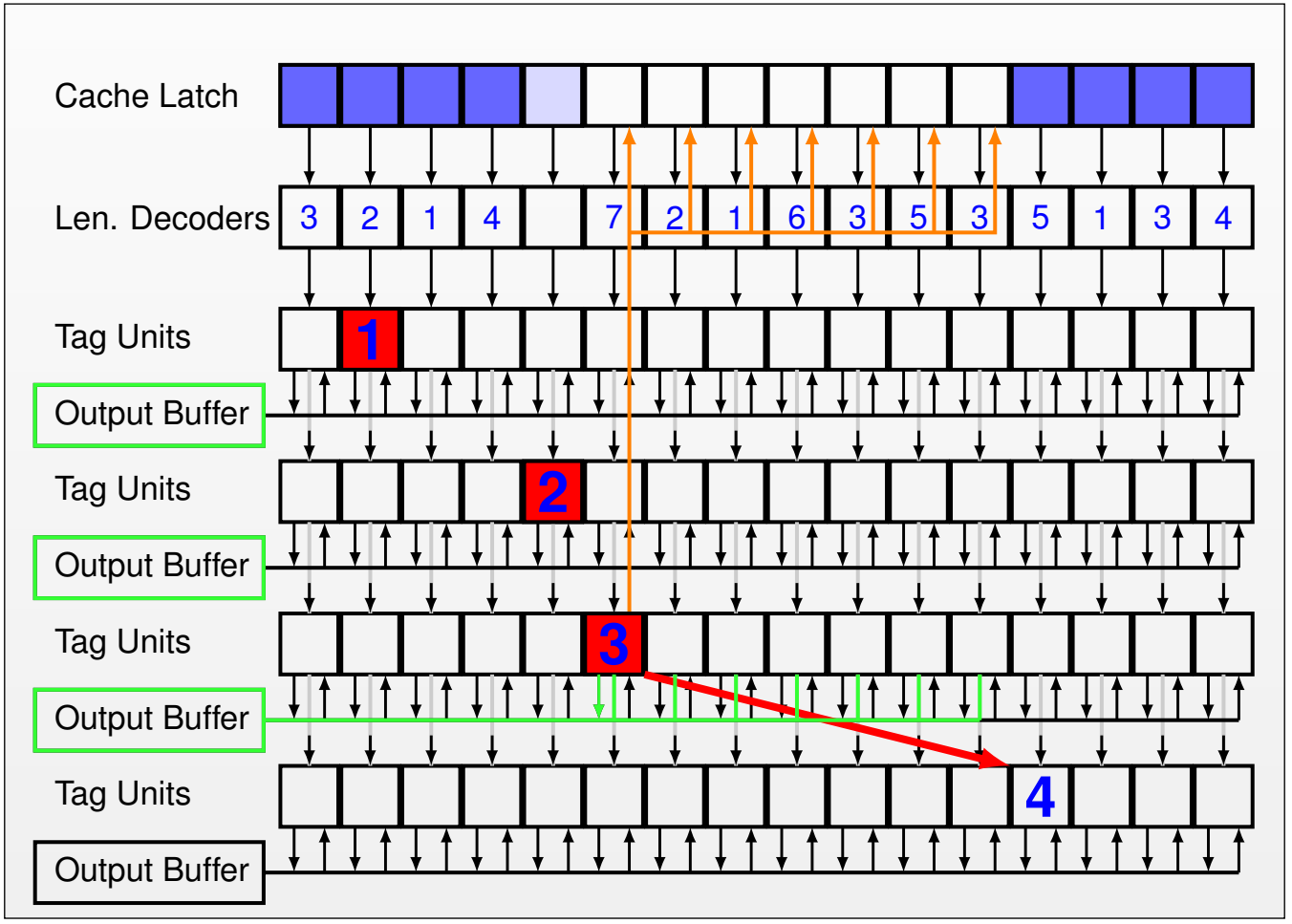
Concurrency and Time

Architectural level timing experiment: Pentium front end



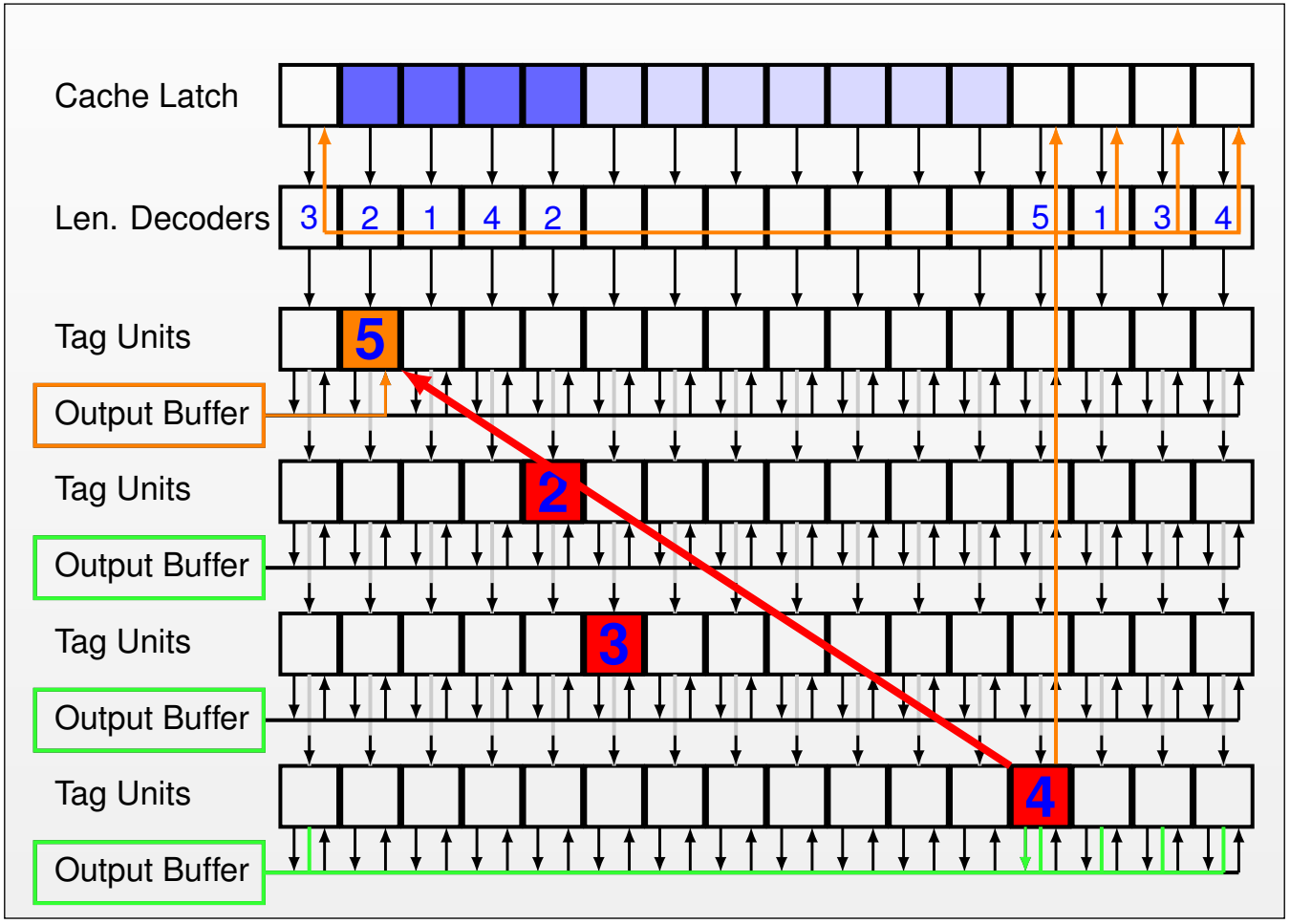
Concurrency and Time

Architectural level timing experiment: Pentium front end



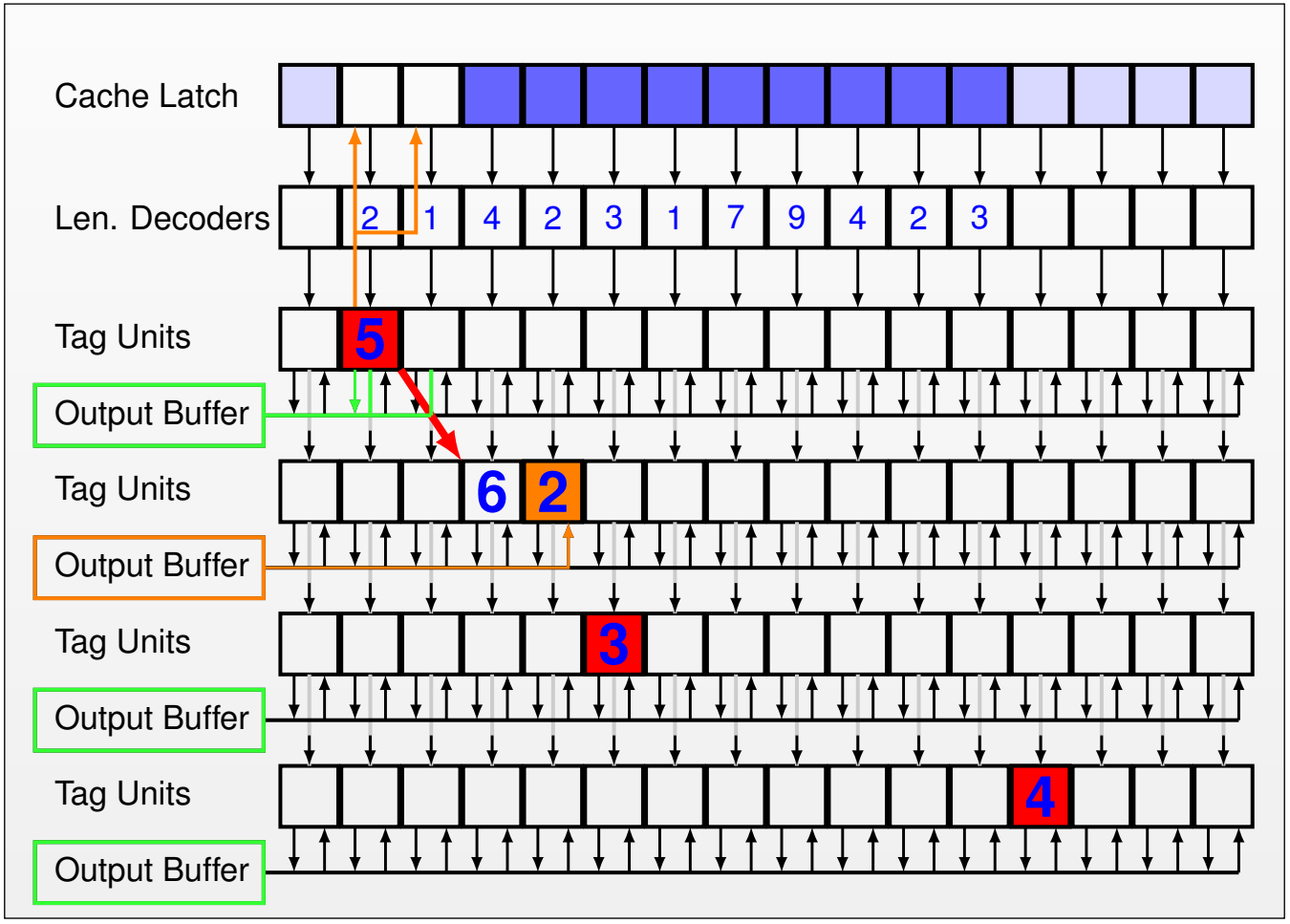
Concurrency and Time

Architectural level timing experiment: Pentium front end



Concurrency and Time

Architectural level timing experiment: Pentium front end



Timing and Sequencing

Traditional representation of timing:

- Metric values
  - On an IC we measure it to picoseconds
  - In track and ski racing, we measure it to milliseconds

But what do we really care about?

- *it isn't the number on the stop watch. . .*

Timing and Sequencing

Traditional representation of timing:

- Metric values
  - On an IC we measure it to picoseconds
  - In track and ski racing, we measure it to milliseconds

But what do we really care about?

- *it isn't the number on the stop watch. . .*

We care about who wins!!

The key: **Timing results in sequencing**

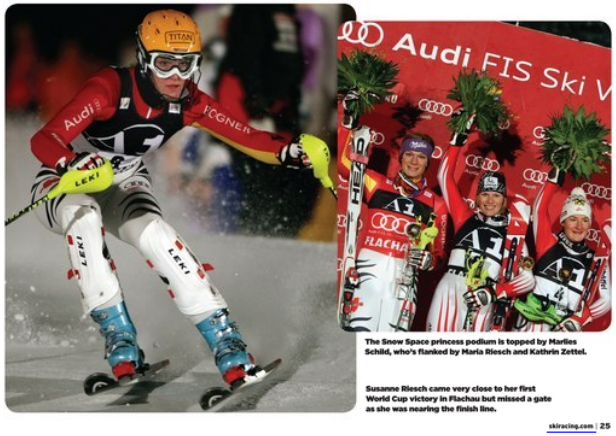
Relative Timing formally represents the signal sequencing *produced by circuit timing*

New *Formal* Abstract Model: Relative Timing

- **Timing** is both the technology differentiator and barrier
- **Relative Timing** is the generalized solution
- The **key property** of time is the **sequencing** it imposes

Sequence gives winner, performance, etc.

- true in semiconductors as well as sports
- absolute stopwatch value is auxiliary



Novel relativistic formal logic

representation of time (relative timing):

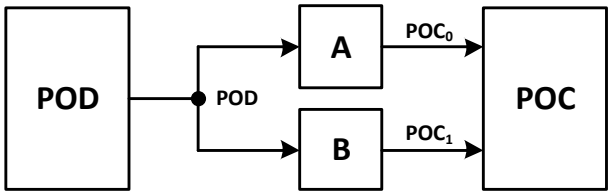
$$\text{pod} \mapsto \text{poc}_1 \prec \text{poc}_2$$

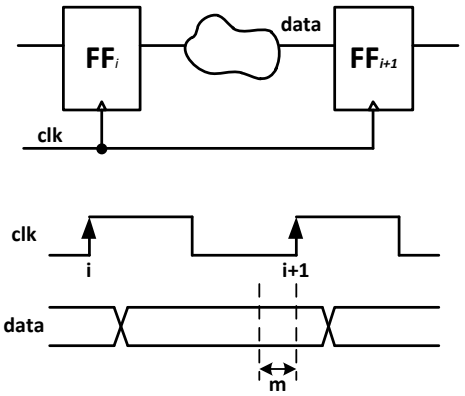
Sequencing relative to common reference

- can now *evaluate* sequencing
- can now *control* sequencing

Relative Timing

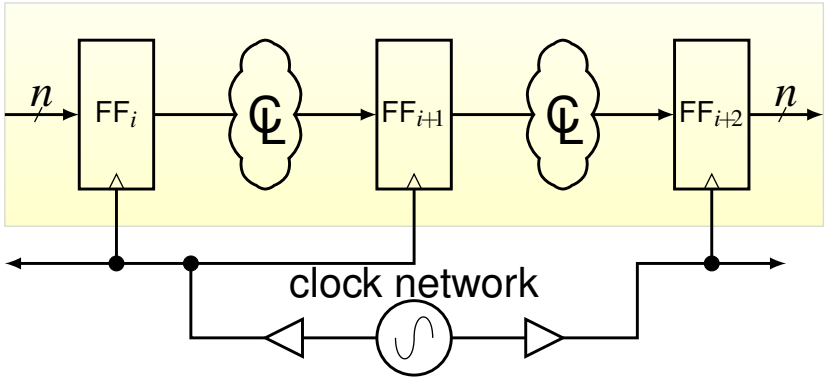
1. Relative Timing
   - Sequences signals at poc (*point of convergence*)
   - Requires a common timing reference: pod (*point of divergence*)
2. Formal representation: pod $\mapsto$ poc$_1$ + margin $\prec$ poc$_2$
3. RT models timing in ALL systems
   - Clocked: pod = clock poc = flops
   - Async: pod = request poc = latches
4. RT enables direct commercial CAD support of general timing requirements
   - formal RT constraints mapped to sdc constraints

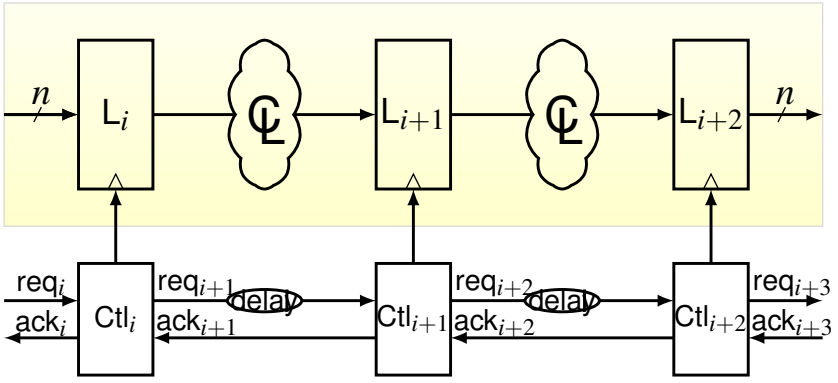




Relative Timed Design: Bundled Data

Bundled data design is much like clocked.





Frequency based (clocked) design. Clock frequency and datapath delay of first pipeline stage is constrained by

$L_i/clk\uparrow_i \mapsto L_{i+1}/d+s \prec L_{i+1}/clk\uparrow_{i+1}$

Timed (bundled data) handshake design. Delay element sized by RT constraint:

$req_i\uparrow \mapsto L_{i+1}/d+s \prec L_{i+1}/clk\uparrow$

*Clocked **physical design** directly supports the clocked Relative Timing constraints. The asynchronous circuit constraints must be provided as min and max constraints, and are **not well supported***

Relative Timing Driven Flow

```
set d0_fdel 0.600
set d0_fdel_margin [expr $d0_fdel + 0.050]
set d0_bdel 0.060

set_size_only -all_instances [find -hier cell lc1]
set_size_only -all_instances [find -hier cell lc3]
set_size_only -all_instances [find -hier cell lc4]

set_disable_timing -from A2 -to Y [find -hier cell lc1]
set_disable_timing -from B1 -to Y [find -hier cell lc1]
set_disable_timing -from A2 -to Y [find -hier cell lc3]
set_disable_timing -from B1 -to Y [find -hier cell lc3]

set_max_delay $d0_fdel -from a -to l0/d
set_max_delay $d0_fdel -from b -to l0/d
set_min_delay $d0_fdel_margin -from lr -to l0/clk
set_max_delay $d0_bdel -from lr -to la
#margin 0.050 -from a -to l0/d -from lr -to l0/clk
#margin 0.050 -from b -to l0/d -from lr -to l0/clk
```

Multi-rate 64-Point FFT Architecture

Initial design target: high performance military applications

- Mathematically based on $W_N = e^{-j\frac{2\pi}{N}}$ notation
- Hierarchical multi-rate design: $N = N_1 N_2$
- Decimate frequency ($\downarrow$) by N_2
  - operate on N_2 low frequency streams
- Transmute data & frequency to N_1 low frequency streams
- Expand ($\uparrow$) by N_1 to reconstruct original frequency stream

Design Models

Hierarchical derivation of multi-frequency design:

$$X_{m_1}(m_2) = \sum_{n_2=0}^{N_2-1} \left[W_N^{m_1 n_2} \sum_{n_1=0}^{N_1-1} x_{n_2}(n_1) W_{N_1}^{m_1 n_1} \right] W_{N_2}^{m_2 n_2}$$

- N_2 FFTs using N_1 values as the inner summation
- Scaled and used to produce N_1 FFTs of N_2 values

Hierarchically scale design

- Base case when $N = 4$, $X(m) = W^4 x(n)$
- 4-point FFT performed without multiplication
  - Multiplication constants W^4 become ± 1

FFT-64

Implemented on IBM's 65nm 10sf process, Artisan academic library

Three design blocks:

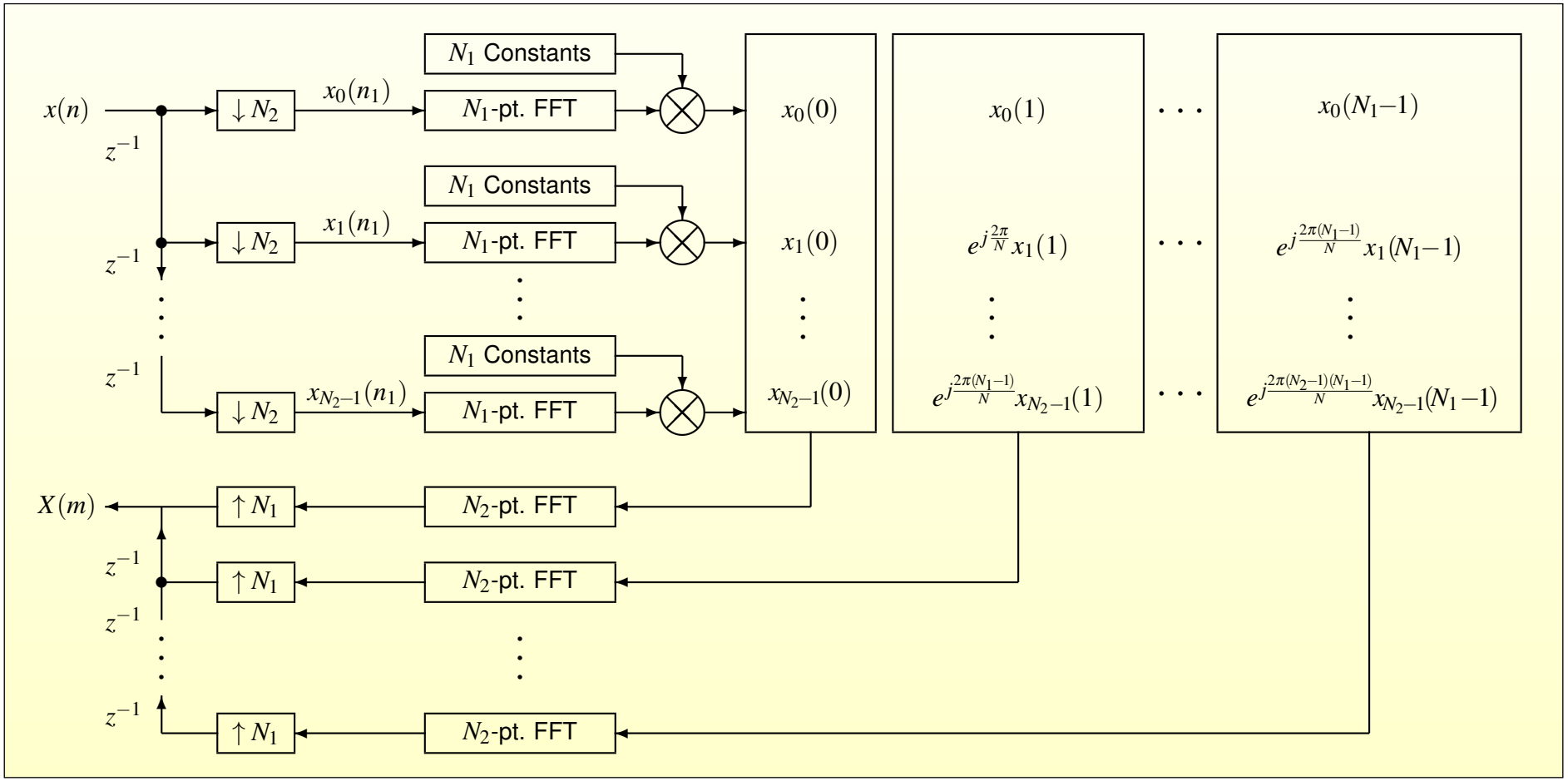
- FFT-4
- FFT-16 $N_1, N_2 = 4$
- FFT-64 $N_1 = 16, \quad N_2 = 4$

Two designs:

- Clocked Multi-Synchronous
- Relative Timed Multi-Synchronous
  - near identical architectures
  - additional RT area / pipeline optimized version for FFT-64

General Multi-rate FFT Architecture

1.25GHz 313MHz 313MHz to 78MHz

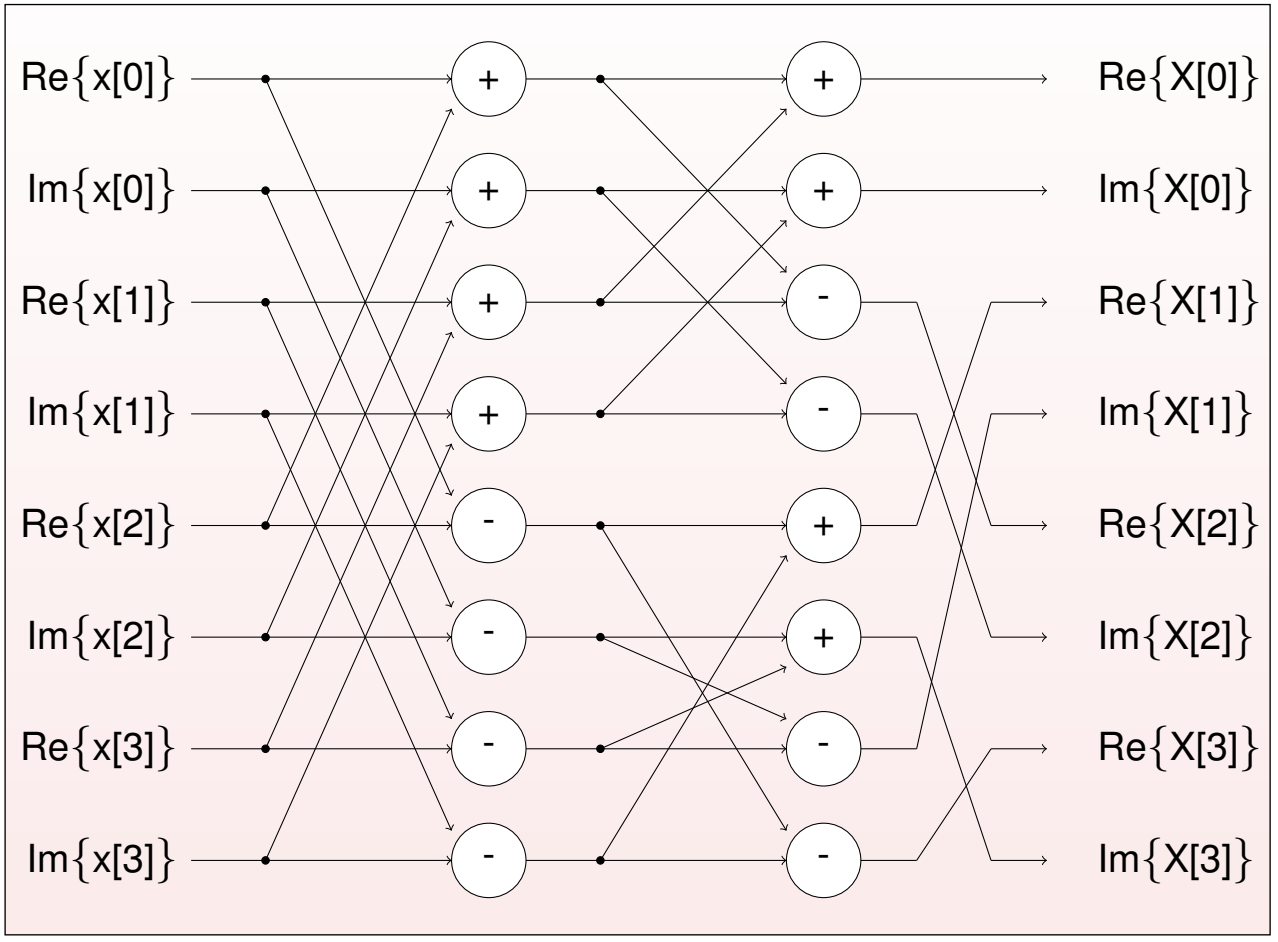


1.25GHz 78MHz

ASIC tool flow, 65nm technology

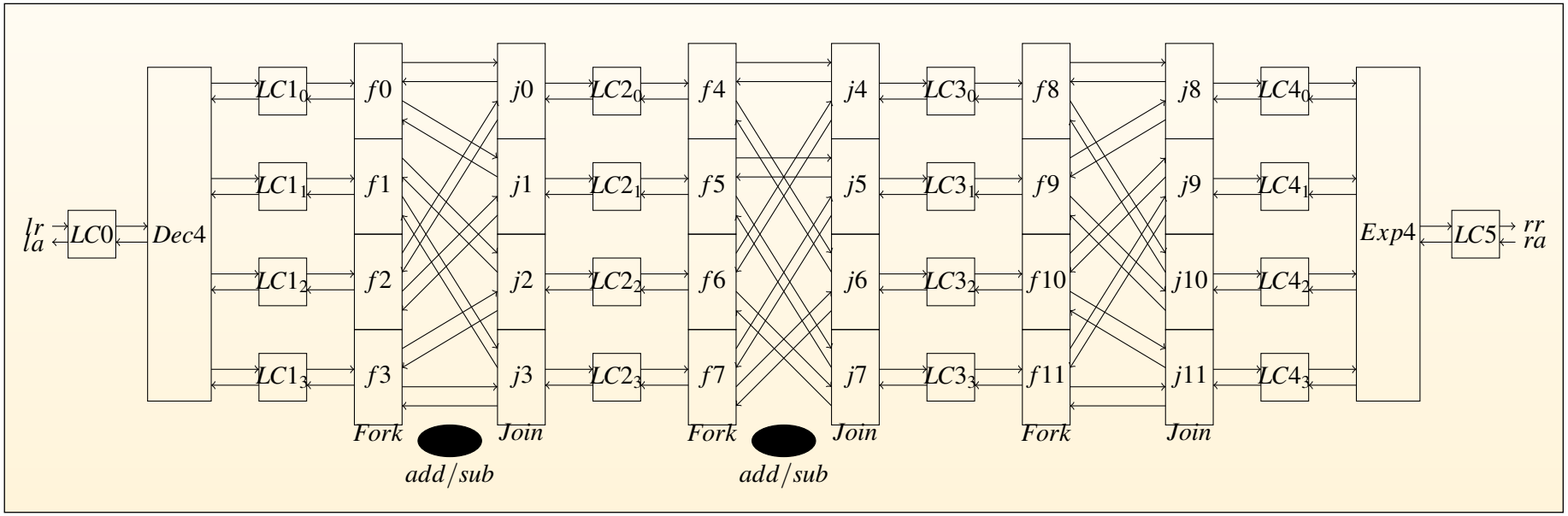
FFT-4 Building Block

Data flow graph of pipelined 4-Point FFT design:



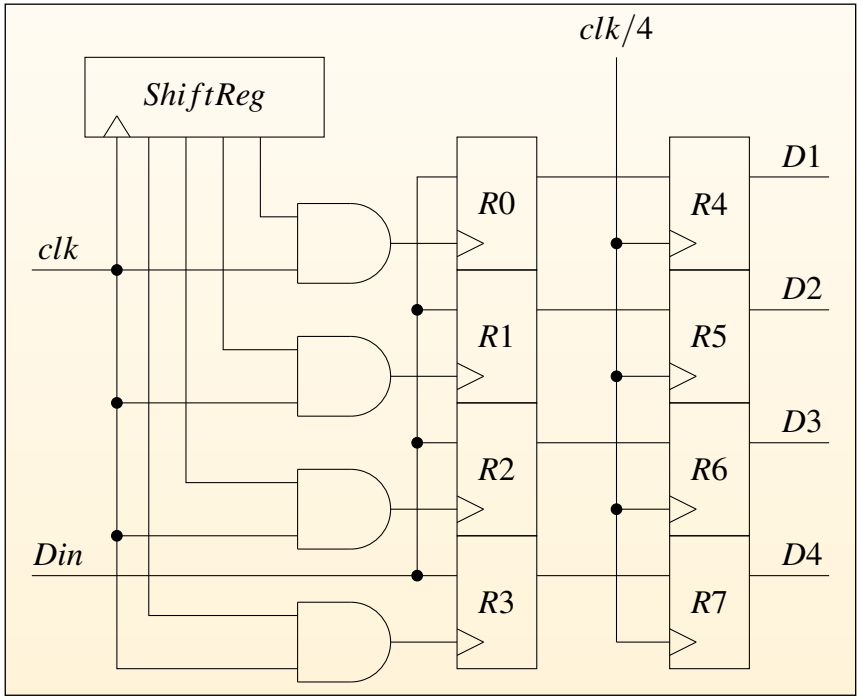
Pipelined Asynchronous 4-Point Architecture

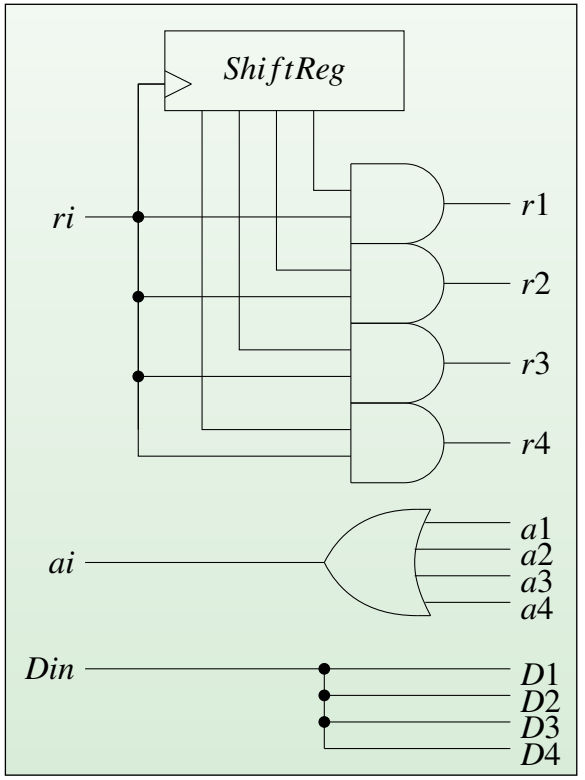
- Operates at 1/4 the input frequency
- Synchronization occurs between decimated rows
  - Fast internal pipeline stages essential



Decimator-4 Design Comparison

- Clocked block requires pipeline to change frequency
- Async block latency combinational and concurrent





Multi-Synchronous asynchronous design smaller, faster, lower power

Results

The 16-point FFT Comparison Result (* values are scaled ideally to 65 nm technology)

	Points	Word $bits$	Time for 1K-point μs	Clock MHz	Tech. nm	Energy/point $pJ/data-point$	Area	Power mW	Energy Benefit	Area Benefit	Throughput Benefit
Our Design(Async)	16-1024	32	0.83	1274	65	25.05	54 Kgates	30.9	8.01	2.77	8.32
Our Design(clock)	16-1024	32	1.73	588	65	41.83	71 Kgates	24.7	4.8	2.07	3.98
Guan [1]	16-1024	16	6.91*	653*	130	200.68	147 Kgates	29.7*	1	1	1

The 64-point FFT Comparison Result (* values are scaled ideally to 65 nm technology)

	Points	Word $bits$	Time for 1K-point μs	Clock MHz	Tech. nm	Energy/point $pJ/data-point$	Area	Power mW	Energy Benefit	Area Benefit	Throughput Benefit
Our Design(Async-opt)	64-1024	32	0.93	1284	65	62.41	0.41 mm^2	68.5	6.1	0.46	30.16
Our Design(Async)	64-1024	32	0.84	1366	65	59.94	0.50 mm^2	72.9	6.35	0.38	33.42
Our Design(clock)	64-1024	32	3.13	588	65	246.75	1.16 mm^2	80.7	1.54	0.16	8.99
Baireddy [2]	64-4096	-	28.14*	514*	90	380.88	0.19 mm^{2*}	13.86*	1	1	1

The 64-point async-opt design contains 229k gates, our clocked 454k.

* For comparison, these designs were scaled to a 65nm process by scaling frequency, power, and area in the 130nm technology by 2.0, 0.5, 0.25$\times$, and in the 90nm design by 1.43, 0.7, and 0.49$\times$ respectively.

[1] X. Guan, Y. Fei, and H. Lin, "Hierarchical Design of an Application-Specific Instruction Set Processor for High-Throughput and Scalable FFT Processing" in IEEE *Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 20, No. 3, pp. 551–563, march 2012.

[2] V. Baireddy, H. Khasnis, and R. Mundhada, "A 64-4096 point FFT/IFFT/Windowing Processor for Multi Standard ADSL/VDSL Applications", in IEEE *International symposium on Signals, Systems and Electronics (ISSSE'07)*, pp. 403–405, 2007.

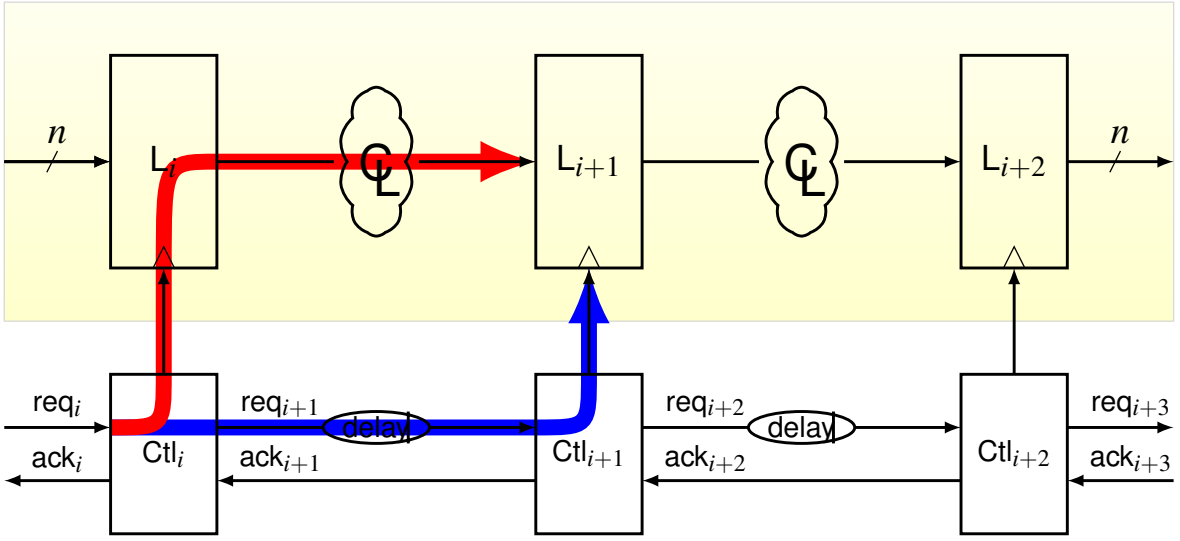
Multi-Synchronous Advantage

1. **Efficiency** in **power** and **performance** is new game in town
2. Multi-synchronous design provides optimization opprotunity
3. New (asynchronous) timing model is one excellent path
4. Produces average 10× $e\tau^2$ improvement
   - Pentium: $e\tau^2 = 17.5\times$
   - FFT: $e\tau^2 = 16.9\times$
5. But ... ***need improved physical design support***

Design	Energy	Area	Freq.	Latency	Aggregate
Pentium F.E.	2.05	*0.85*	2.92	2.38	12.11×
64-pt FFT	3.95	2.83	2.07	3.37	77.98×

RT Physical Design Optimization

Timing, power, and performance optimizations driven by relative timing constriants.

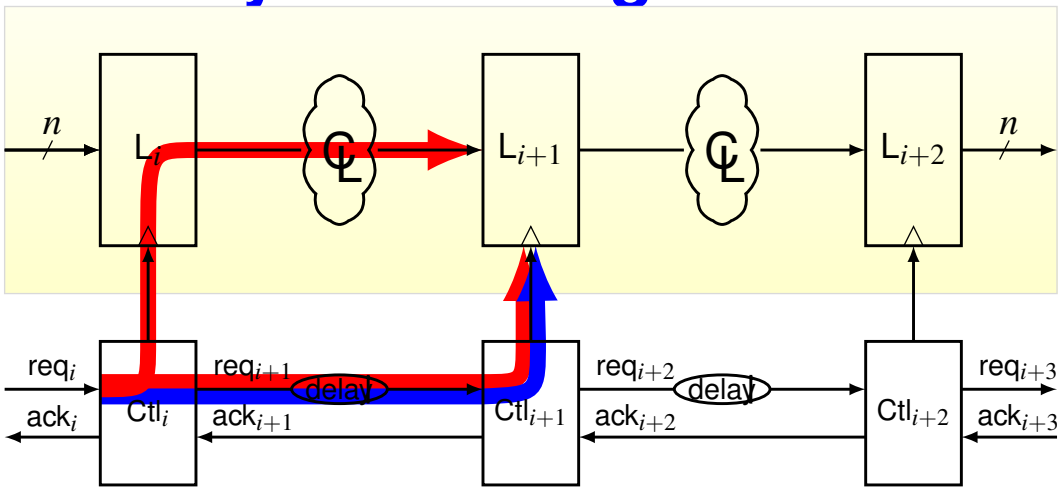


$\text{req}_i\uparrow \mapsto \mathbf{L_{i+1}/d} + m \prec \mathbf{L_{i+1}/clk}\uparrow$

Mapped to `set_max_delay` and `set_min_delay` constraints

Clock frequency determines min delay, async adds “hold time”

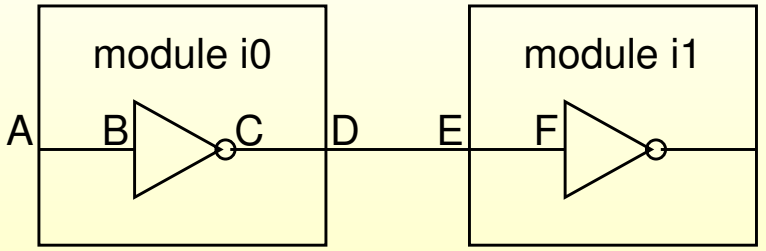
RT Physical Design Problems



1. Inconsistency between operation and results
   - supported pins & formats, synthesis vs place and route, etc.
2. Min-delay constraints not well supported
   - Treated as “hold time fixing”
   - Create arbitrarily large delays
     - Degrades performance
     - Required matching max-delay constraint to bound delay
3. Poor job of optimizing competing constraints
4. Placement can be substantially improved

RT Physical Design Problems

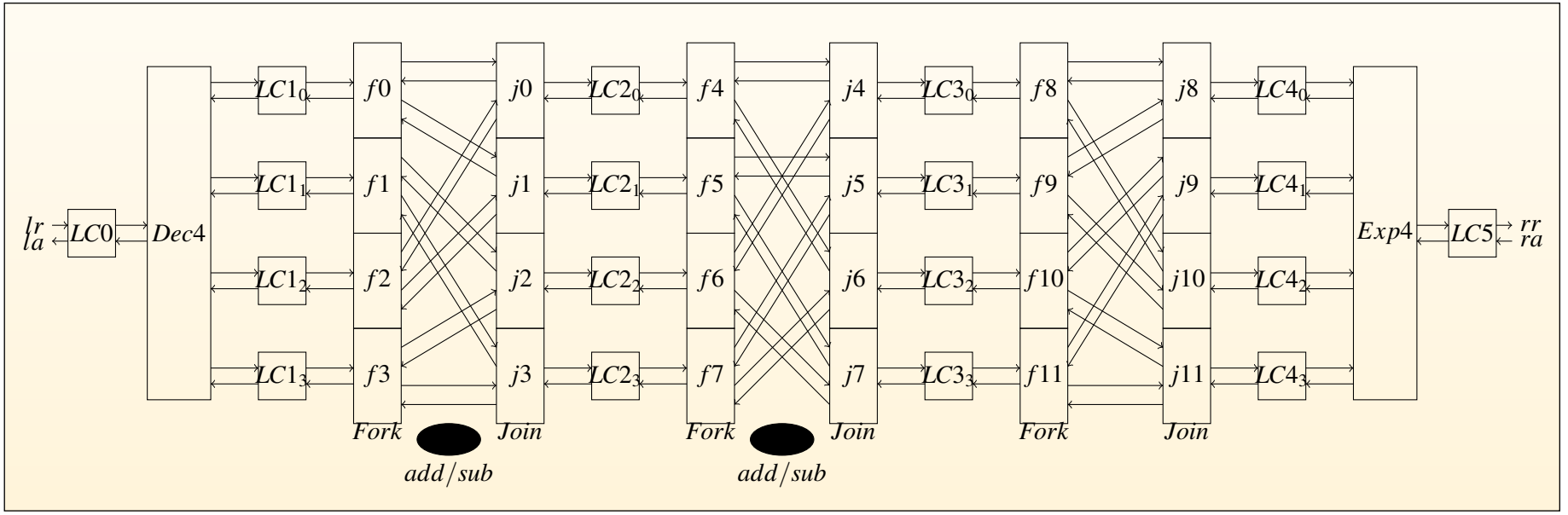
Simple experiment with inverters with endpoints mapping either to module pin or library gate pin:



	Design Compiler			SoC Encounter	
Path	**Result**	**Iterations**	**Type**	**Result**	**type**
A $\rightarrow$ E	Yes	5	buffers	No	–
A $\rightarrow$ F	Yes	5	buffers	No	–
B $\rightarrow$ E	Yes	1	Dly Elts	No	–
B $\rightarrow$ F	Yes	1	Dly Elts	Yes	Dly Elts
C $\rightarrow$ E	Yes	1	Dly Elts	No	–
C $\rightarrow$ F	Yes	1	Dly Elts	Yes	Dly Elts
D $\rightarrow$ E	No	–	–	No	–
D $\rightarrow$ F	No	–	–	No	–

Paths use both max and min delay constraints

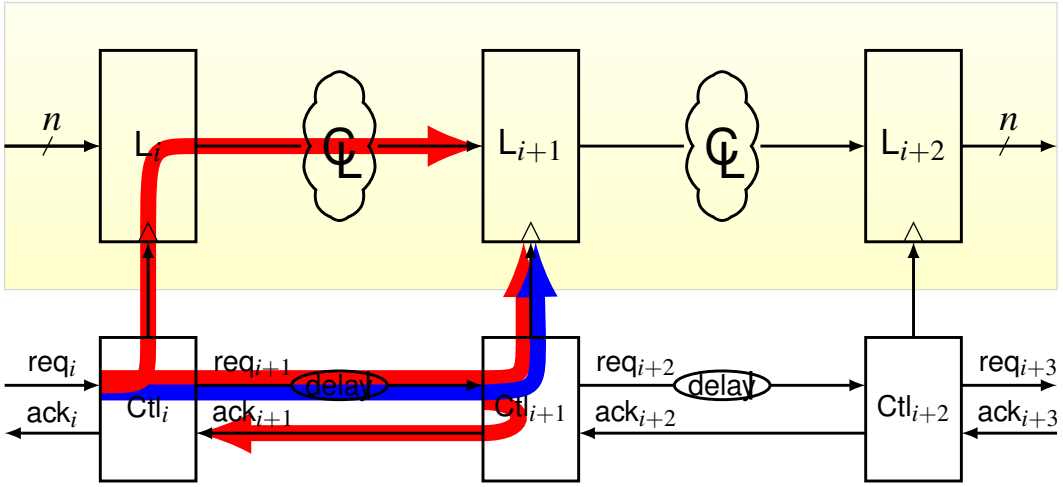
RT Physical Design Problems



Min-delay constraints get dropped, even in relatively small design!

	Design Compiler		SoC			SoC - timing closure		
Model	#iter	cyc. time	#iter	cyc. time	energy/op	#iter	cyc. time	energy/op
wl0.5	9	738ps	1	728ps	5.16pJ	*70*	*785ps*	*4.85pJ*
wl0	7	666ps	1	764ps	5.07pJ	16	763ps	4.87pJ

RT Physical Design Potential



1. Low hanging fruit for performance improvements
2. Force directed algorithms
   - Combine power/placement optimizations
   - Drive cell clustering
   - Drive pipeline/repeater placement and wire optimization
3. Tool performance: Convergence and run-time

Multi-Synchronous Advantage

1. **Efficiency** in **power** and **performance** is new game in town
2. Multi-synchronous design provides optimization opprotunity
3. New (asynchronous) timing model is one excellent path
4. Produces average 10× $e\tau^2$ improvement
   - Pentium: $e\tau^2 = 17.5\times$
   - FFT: $e\tau^2 = 16.9\times$
5. But ... ***need improved physical design support***

Design	Energy	Area	Freq.	Latency	Aggregate
Pentium F.E.	2.05	*0.85*	2.92	2.38	12.11×
64-pt FFT	3.95	2.83	2.07	3.37	77.98×