innovations
for high
performance
microelectronics

Performance Analysis of GALS Datalink Based on Pausible Clocking

Xin Fan, Milos Krstic, Eckhard Grass

fan@ihp-microelectronics.com

IHP
Im Technologiepark 25
15236 Frankfurt (Oder)
Germany

Outline

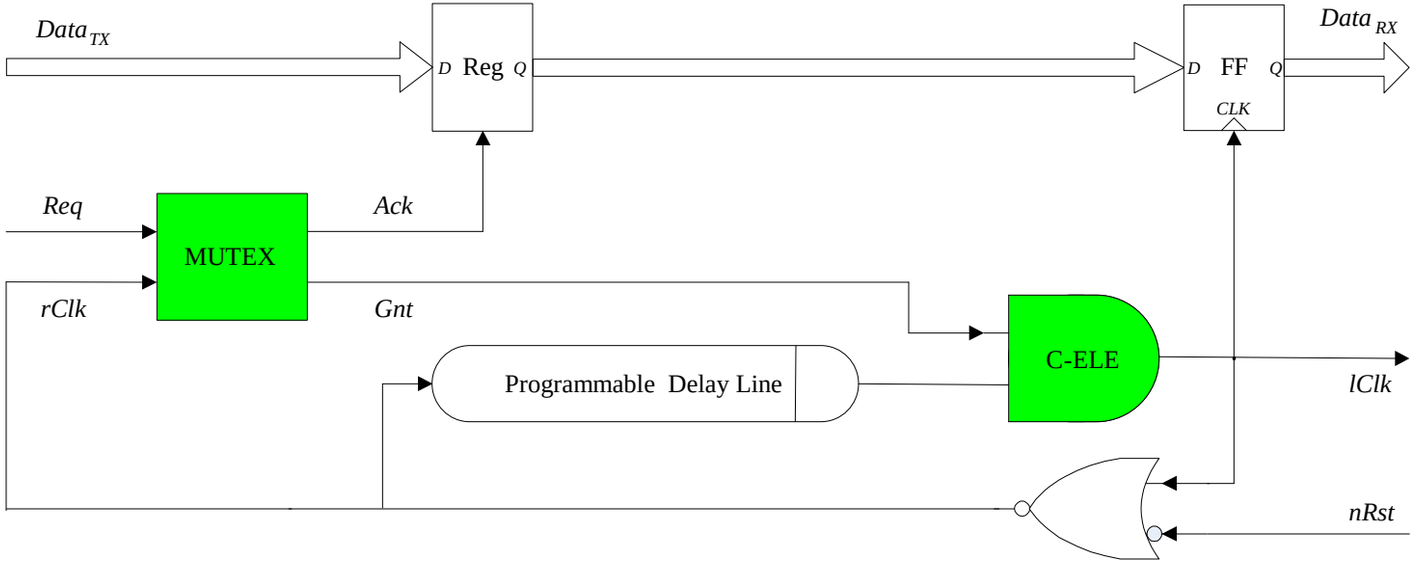
- **Pausible clocking based GALS design – an overview**
- **Performance modeling of typical GALS datalinks**
- **Upper bound of throughput-tolerant clock/interconnect delays**
- ***Moonrake*: a SYNC/GALS OFDM BB TX chip in the 40-nm process**
- **Conclusions**

Pausible clocking based GALS design – an overview

- **What's pausible clocking?**

  Temporarily stop the clock if its rising edge is too close to the input data, i.e., synchronize the clock activity according to the arrival time of data.

  It was once expected to be a *high reliability*, *high performance* while *low overhead* solution for GALS system design.



Pausible clocking based GALS design – an overview

- **Fundamental limitations of applying pausible clocking:**

  Taking into consideration the clock tree distribution delay nullifies all the assumptions for source clock scheduling to avoid metastability.

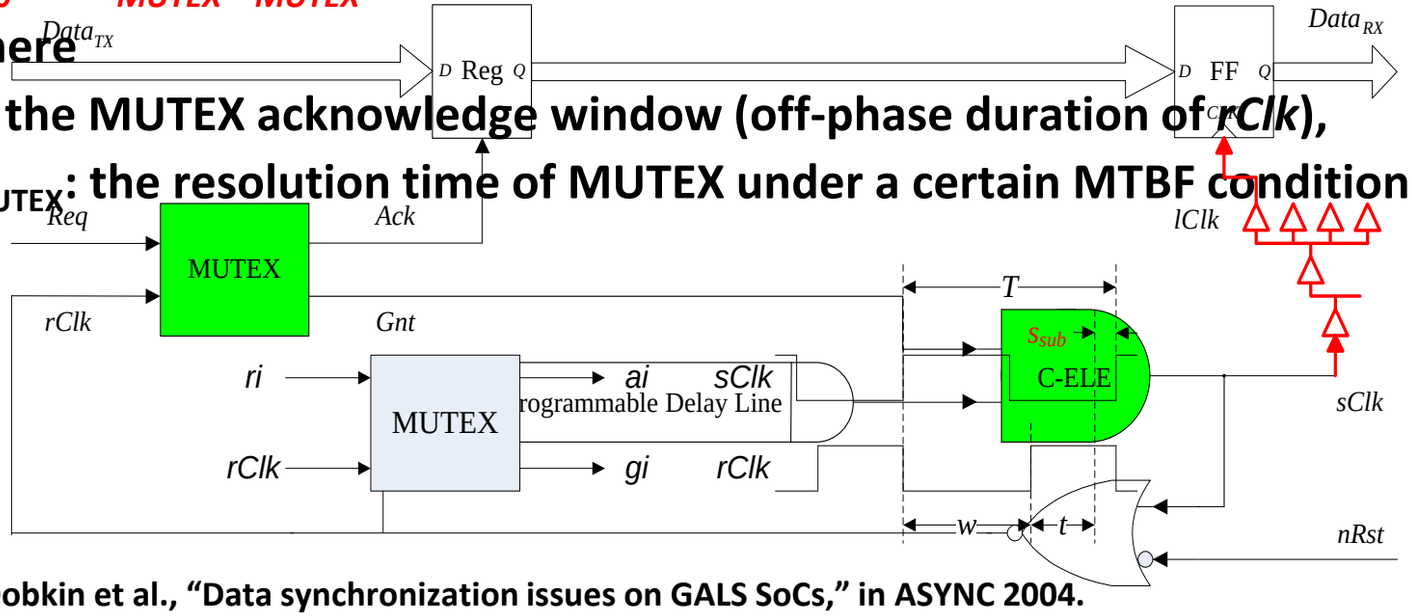
  Multi-cycle vs. sub-cycle clock tree delay.

  $s_{sub}=T-w_{MUTEX}-t_{MUTEX}$

  where

  w: the MUTEX acknowledge window (off-phase duration of *rClk*),

  t_{MUTEX}: the resolution time of MUTEX under a certain MTBF condition.

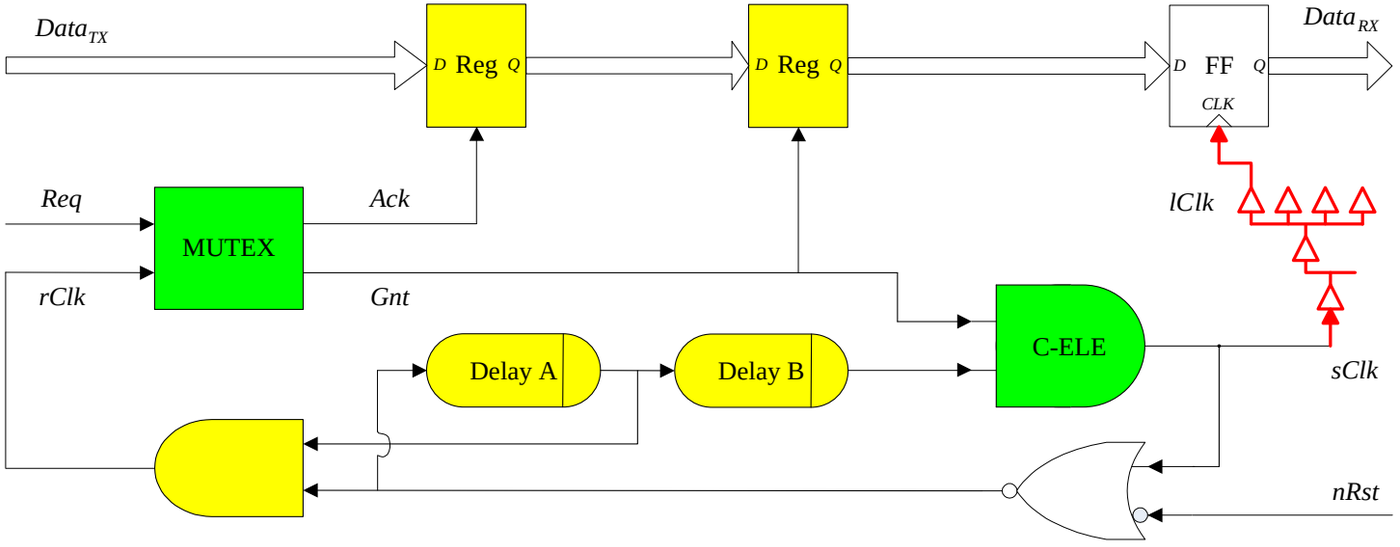


R. Dobkin et al., "Data synchronization issues on GALS SoCs," in ASYNC 2004.

Pausible clocking based GALS design – an overview

- **Optimization for safe synchronization**

  For sub-cycle clock tree delay: maximize the safe timing region.
  Double stages of register timed by the output of MUTEX, $s_{sub}=w$.
  Two cascaded delay lines with a feedback loop to maximize *w*.



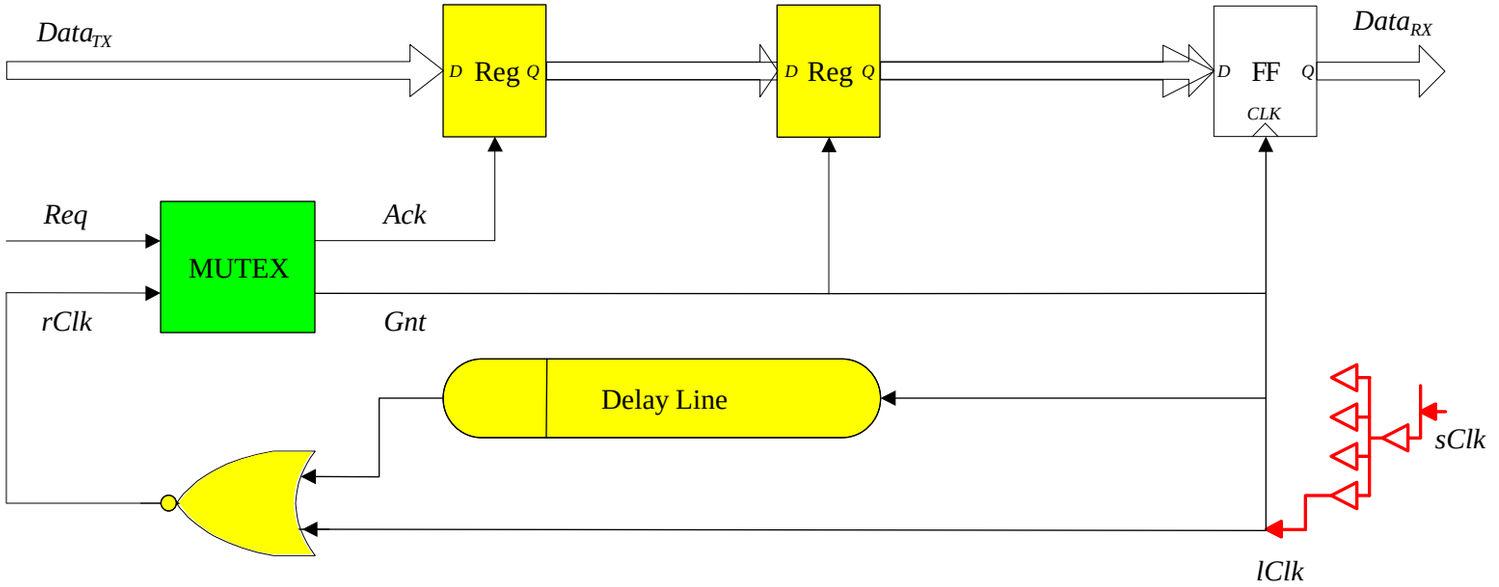
X. Fan et al., "Analysis and optimization of pausible clocking based GALS design," in ICCD 2009.

Pausible clocking based GALS design – an overview

- **Optimization of pausible clocking scheme**

  For multi-cycle clock tree delay: locally data latching (*LDL*) scheme from *Technion,* arbitrating on the leaf registers instead of on the source clock.

  Integrate double-register scheme with *LDL* to maximize *w*.



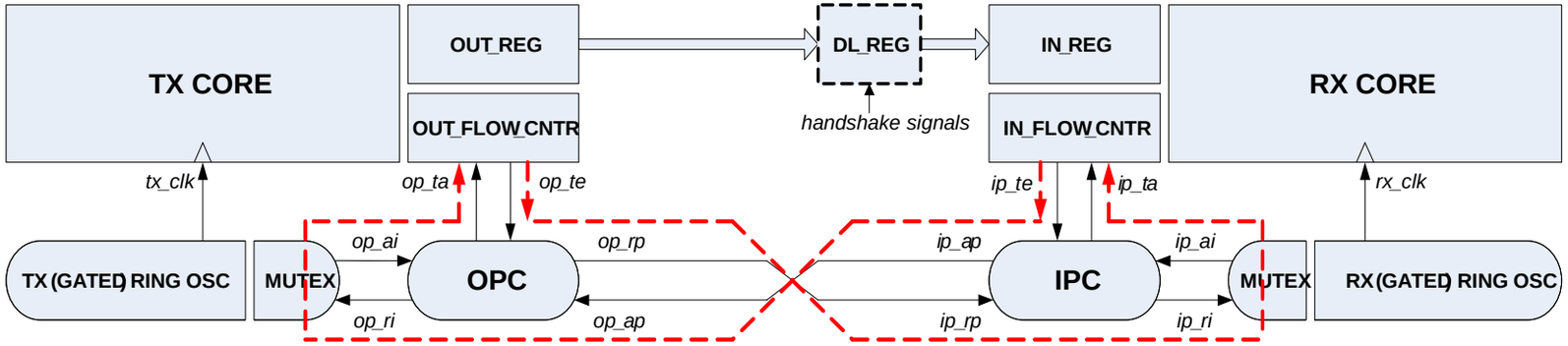
R. Dobkin et al., "High rate data synchronization in GALS SoCs," in TVLSI 2006

Pausible clocking based GALS design – an overview

- **Architecture of pausible clocking based GALS design**

  Communication across clock domains is initiated by the output/input flow control logic of TX/RX synchronous functional blocks.

  Data transfer between TX and RX is scheduled by the asynchronous handshake signals.

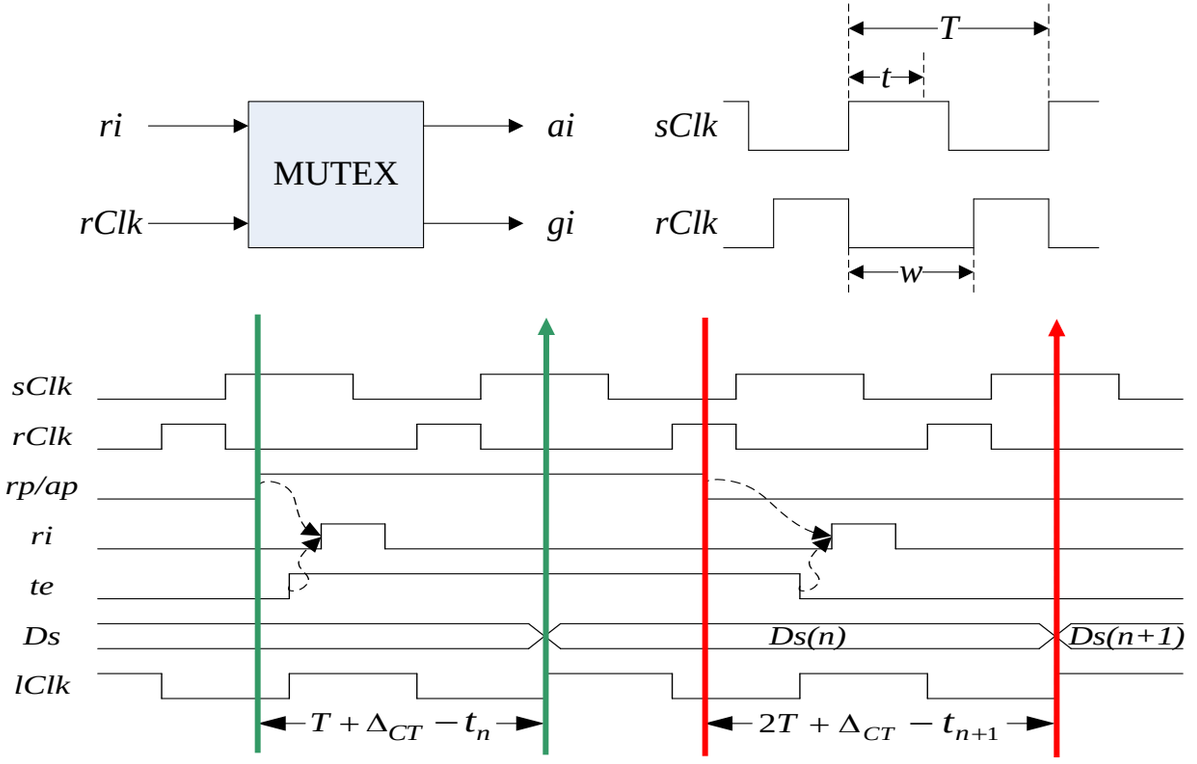


- **Previous studies show that the pausible clocking based GALS datalink could reach a throughput of one data item every second clock cycle!?**

Performance modeling of typical GALS datalinks

- **Data synchronization latency**

The timing interval from receiving an input data, indicated by the input request, to sampling the data by the local clock.



Performance modeling of typical GALS datalinks

- **Synchronization latency function *L (t, w,* Δ_{CT}*)***

$$L(t, w, \Delta_{CT}) = \begin{cases} T + \Delta_{CT} - t, t \in [0, w - \Omega); \\ 3T/2 + \Delta_{CT} - w, t \in [w - \Omega, w + \Omega]; \\ 2T + \Delta_{CT} - t, t \in (w + \Omega, T). \end{cases}$$

For the uniform distribution of *t* within [0, T), we have:

$$l = \int_0^T \frac{1}{T} L(t, w, \Delta_{CT}) dt = (\frac{3}{2} - \frac{w - \Delta_{CT}}{T})T.$$

***l* is determined by the effective acknowledge window of MUTEX** $(w - \Delta_{CT})$**.**

If $(w - \Delta_{CT}) > T/2$**, a sub-cycle latency can be achieved on average.**

Given *w*, the minimum latency is obtained when $\Delta_{CT} = 0$**.**

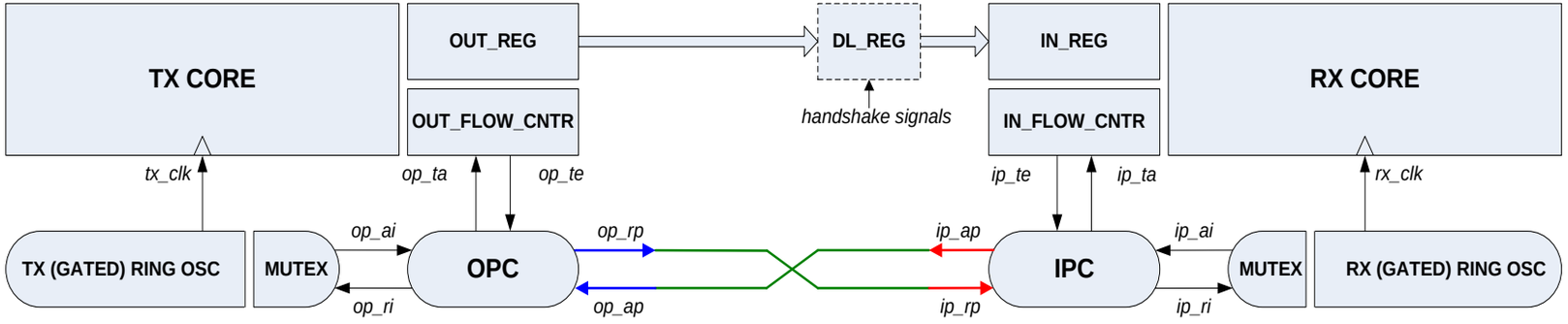
Performance modeling of typical GALS datalinks

- **Data throughput in burst-mode communication**

  The handshake loop delay of asynchronous datalink consists of 3 parts:

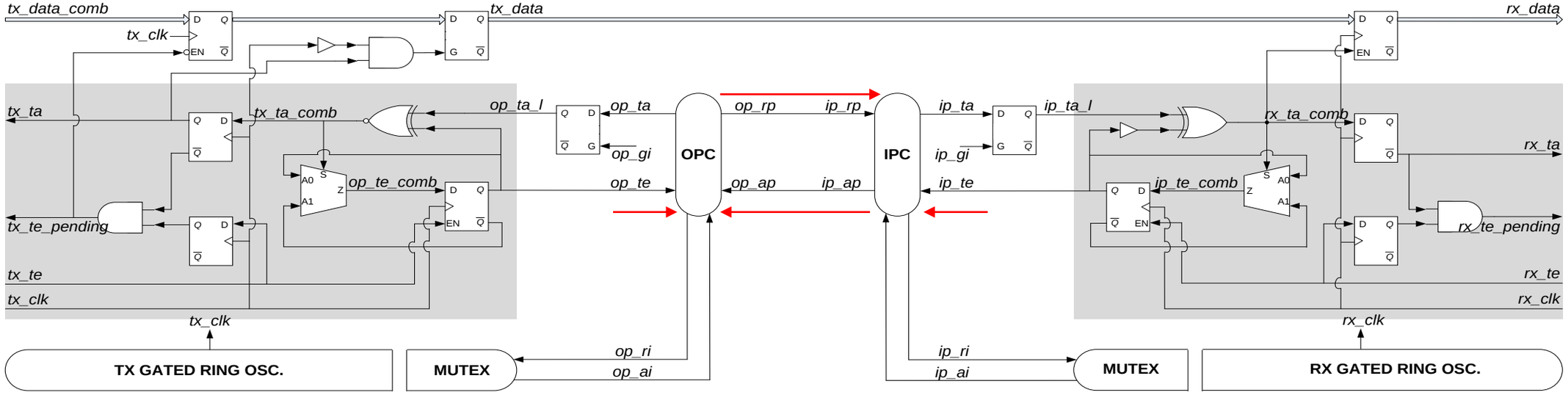
$$T_{Loop}^{avg} = d_{IPC}^{avg} + d_{OPC}^{avg} + d_{INT}.$$

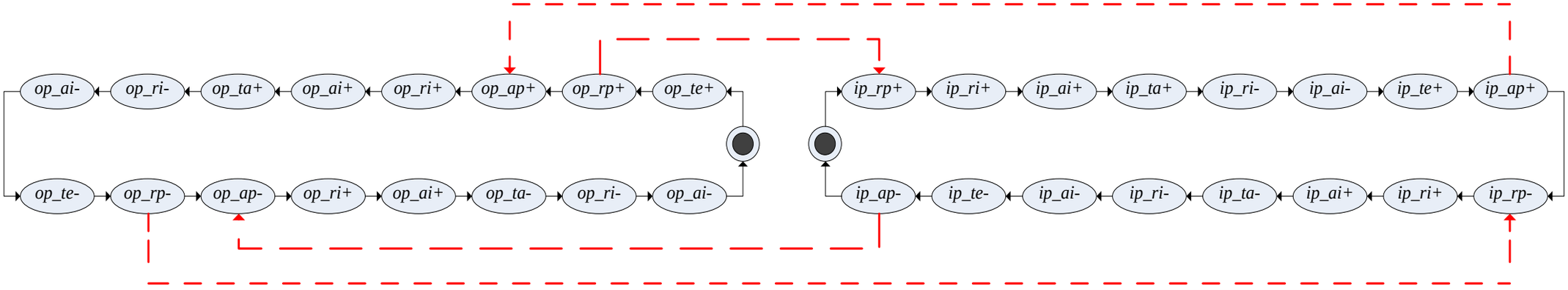
- **The optimization target:** $T_{Loop}^{avg} \le \max(T_{TX}, T_{RX}).$



Performance modeling of typical GALS datalinks

- **An example: tightly coupled GALS datalink**





Performance modeling of typical GALS datalinks

- **Average loop period of the tightly coupled datalink**

  It is the sum of the average synchronization latencies on both TX and RX sides and the interconnect delays between OPC and IPC:

$$T_{Loop}^{avg} = (\frac{3}{2} - \frac{w_{RX} - \Delta_{CT}^{RX}}{T_{RX}})T_{RX} + (\frac{3}{2} - \frac{w_{TX} - \Delta_{CT}^{TX}}{T_{TX}})T_{TX} + d_{INT}.$$

  Given T_{TX}/T_{RX} close to 1, we can derive the throughput-tolerant condition on clock-tree and interconnect delays:

$$(w_{TX} - \Delta_{CT}^{TX}) + (w_{RX} - \Delta_{CT}^{RX}) \geq 2T + d_{INT}.$$

  Note that, due to *w<T*, above requirement could never be satisfied, even with zero clock-tree/interconnect delay.

  It means throughput drop is unavoidable in the tightly coupled datalink, even if TX and RX have only a tiny mismatch on operating frequency!

Upper bound of throughput-tolerant clock-tree/inter-connect delay

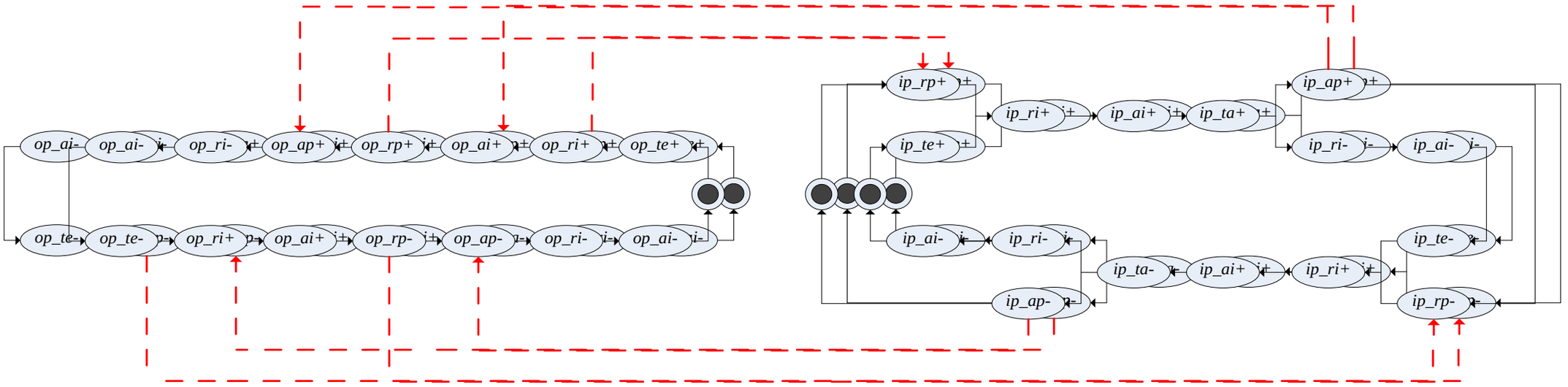
- **Loosely coupled GALS datalinks**

(a) Reduce d_{IPC} by introducing concurrency in IPC:

$$(w_{TX} - \Delta_{CT}^{TX}) + (w_{RX} - \Delta_{CT}^{RX}) \geq T + d_{INT}.$$

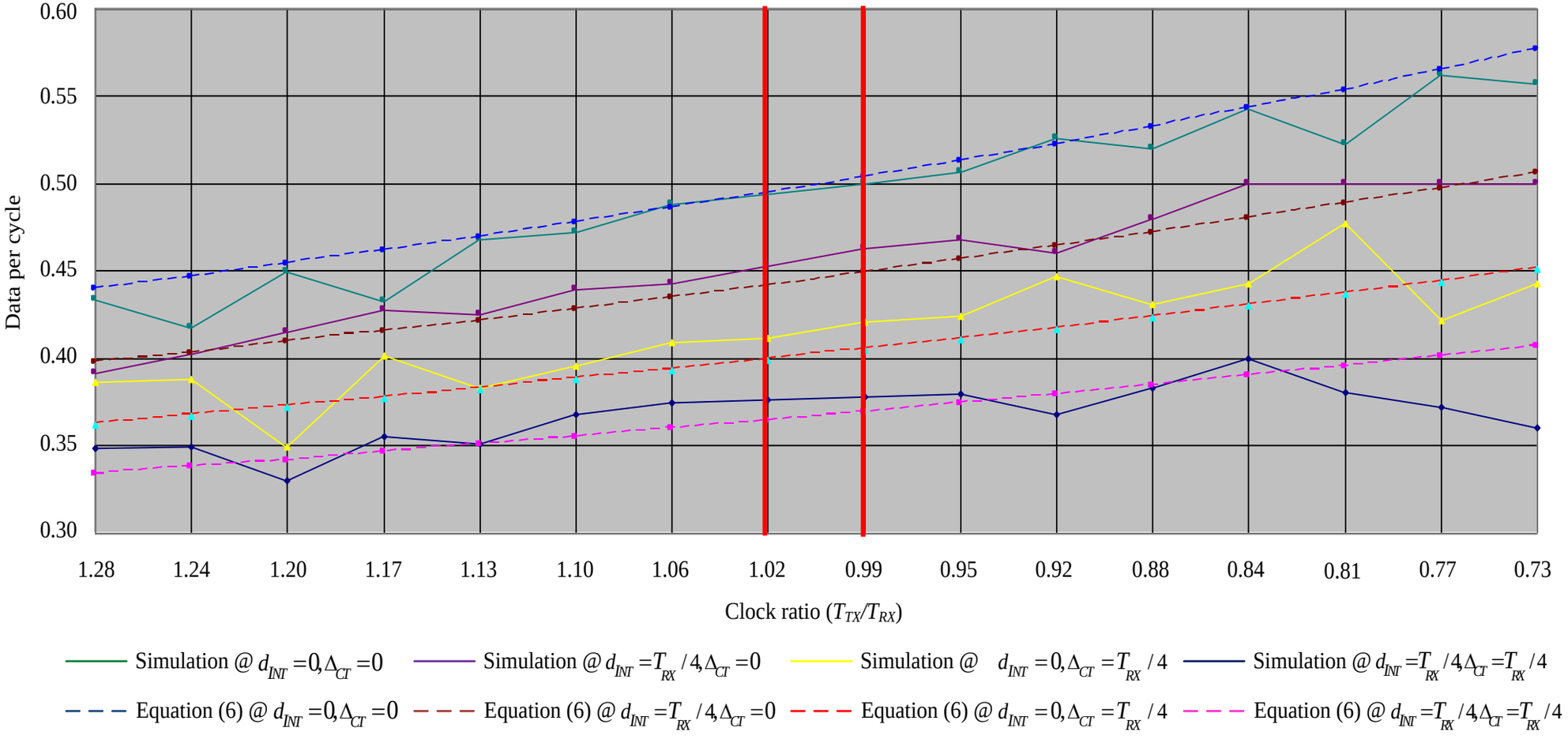
(b) Reduce d_{OPC} by stopping TX local clock if possible:

$$(-\Delta_{CT}^{TX}) + (w_{RX} - \Delta_{CT}^{RX}) \geq d_{INT}.$$



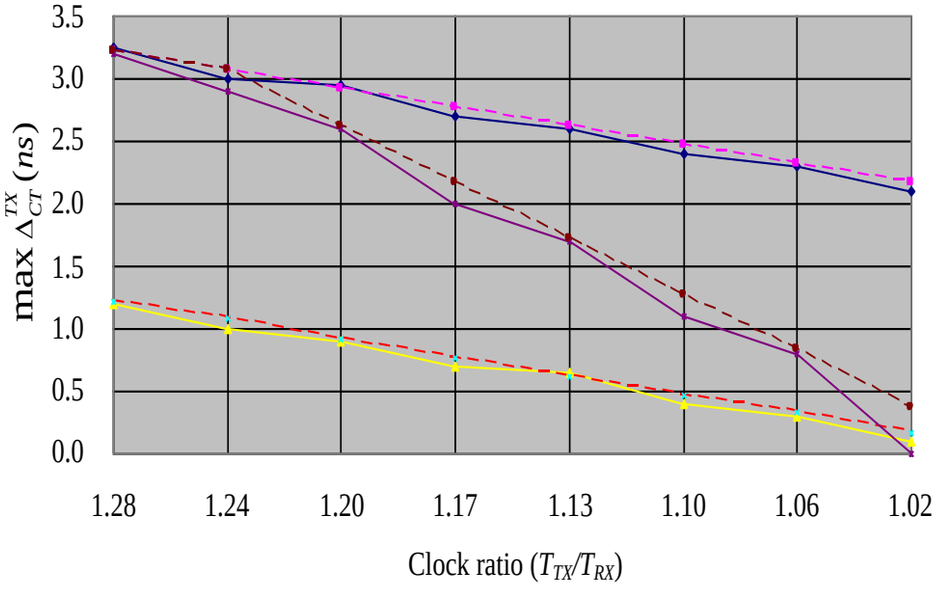
Upper bound of throughput-tolerant clock-tree/inter-connect delay

- **Comparison in data throughput of tightly coupled GALS datalink**

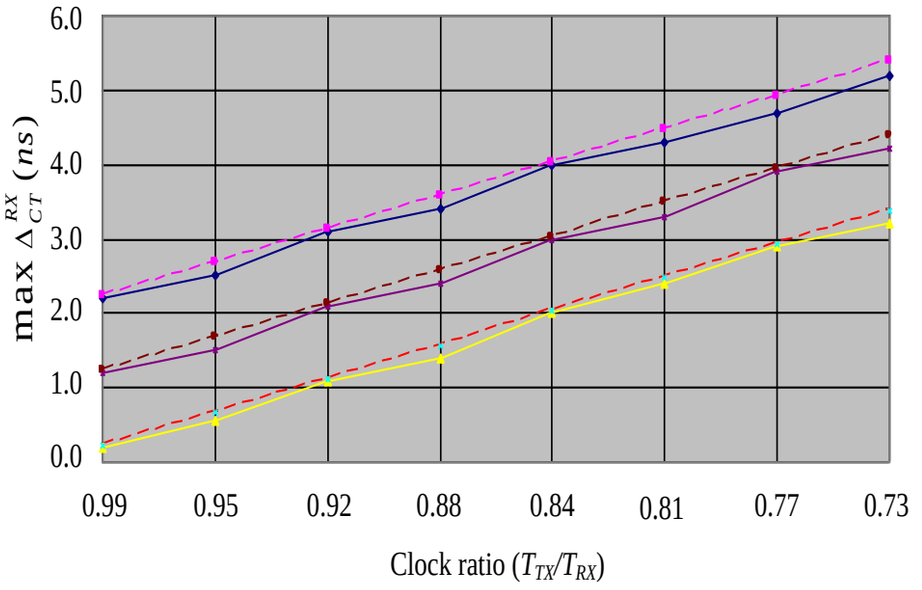


Upper bound of throughput-tolerant clock-tree/inter-connect delay

- **Comparison in throughput-tolerant delay in loosely coupled datalinks**



(a)



(b)

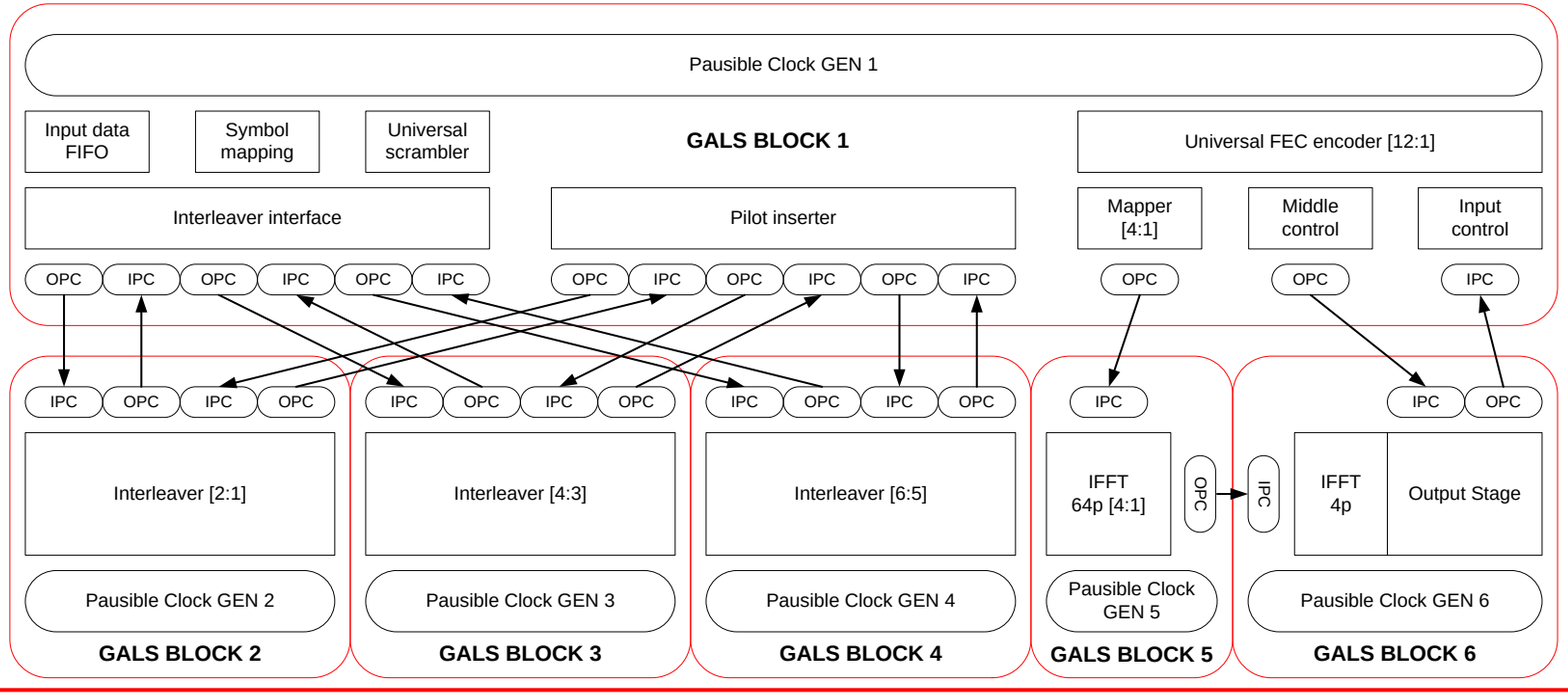
Moonrake: a 40-nm SYNC/GALS OFDM BB TX chip

- **Architecture of GALS design**

System partition accounting both function and physical criteria;

6 GALS blocks balanced in area and power dissipation;

16 asynchronous datalinks (32 IPC/OPC).

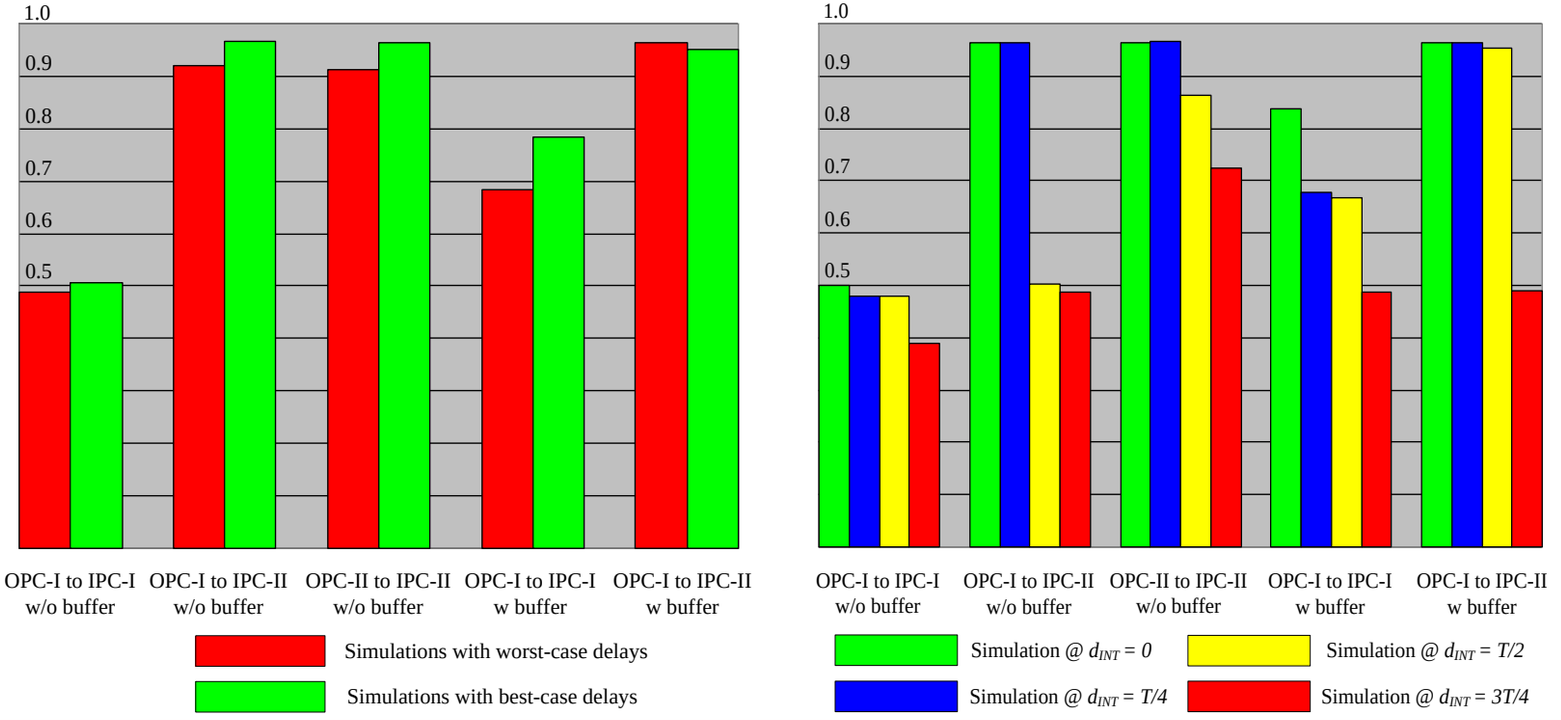


Moonrake: a 40-nm SYNC/GALS OFDM BB TX chip

- **Comparison in GALS TX throughput fabricated by different datalinks**

A data frame with QPSK modulation was processed by the GALS TX.

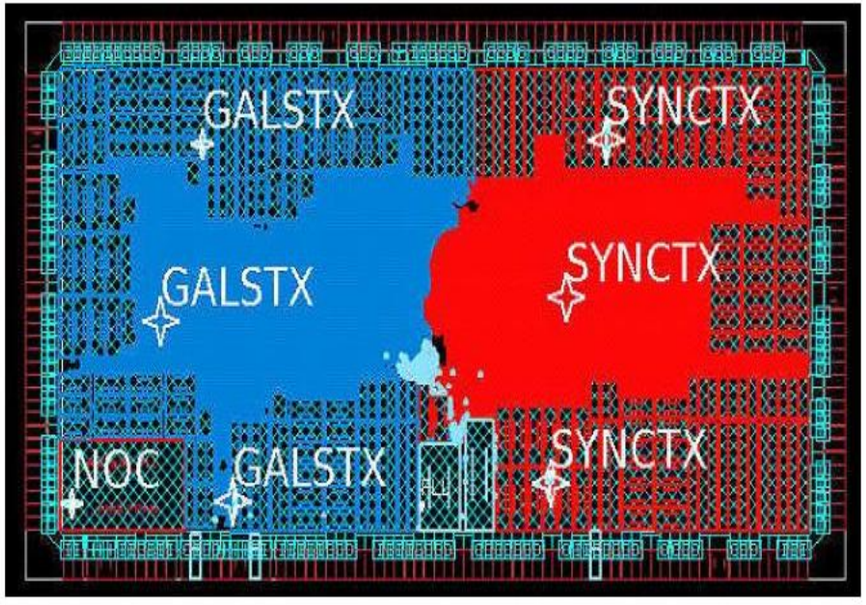
RTL of TX functional blocks, netlist of IPC/OPC and clock generators, with back-annotated clock-tree/interconnect delays after layout.

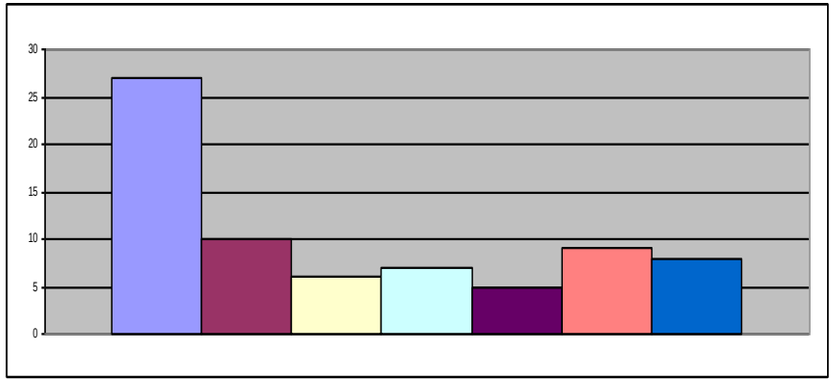


Moonrake: a 40-nm SYNC/GALS OFDM BB TX chip

- **Power and area comparison of SYNC/GALS TXs**

	GALS_BLK 1	GALS_BLK 2	GALS_BLK 3	GALS_BLK 4	GALS_BLK 5	GALS_BLK 6	TOTAL
AREA	19%	18%	18%	18%	10%	17%	100%
POWER	12%	17%	17%	17%	18%	19%	100%





	Cell area (μm^2)		Power dissipation (mW)	
	Post-synthesis	Post-layout	Post-layout	Measurement
SYNC w/o PLL	2206895	2234712	234	252
GALS TX	2225823	2220080	225	237

Conclusions

- **The performance (synchronization latency and data throughput) of pausible clocking based GALS datalinks is dominated by *w*, which is in turn determined by the target resolution time of MUTEX:**

$$MTBF \rightarrow t_{MUTEX} \rightarrow w = T - t_{MUTEX}.$$

- **It can cause large performance penalty in speed aggressive design, such as high-speed micro-processors. On the other hand, it is very suitable for complicated system integration with moderate speed.**

- **By the optimization of IPC/OPC, it is possible to tolerate clock and interconnect delays, to some extent, with little performance drop.**

- **The marginal hardware overhead caused by the GALS infrastructure based on pausible clocking can be compensated at the system level.**

Thank you!
Question?