Performance Driven Reliable Link Design for Network on Chips

Rutuparna Tamhankar
Srinivasan Murali
Prof. Giovanni De Micheli

Stanford University

Outline

Introduction

- Wire delay contribution to total delay is increasing
  - 6 -10 cycles needed to diagonally cross the chip in 50 nm
- Faster clock cycles, scaling voltages and noisy environment make wires unreliable.
- Unpredictability in wire characteristics results in variation in wire delay
- Noise-induced wire delay consumes greater percentage of useful clock cycle.
- Conservative design approaches that consider worst case operating conditions, result in poor performance.

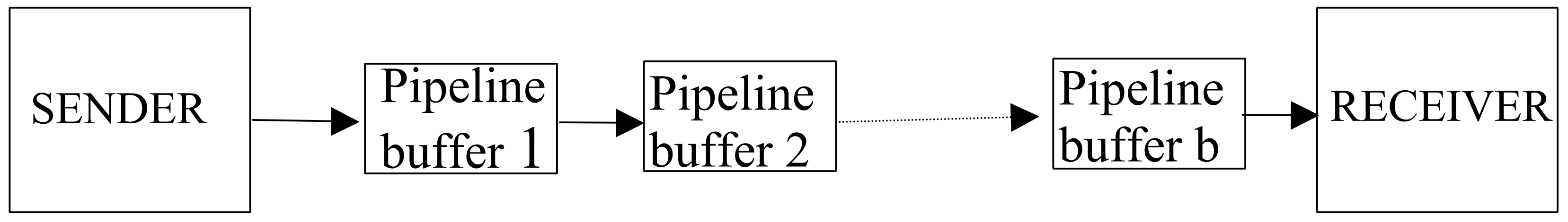
Objective of work

- Motivational example
  - ***A 3-bit adversarial switching pattern (101->010, 010->101) increases wire delay by 50%.***
  - ***Conservative approach results in larger clock cycle to account for such delay variations.***
- Use aggressive design approach
  - Designed for normal (ignoring noise) conditions, tolerate errors caused to noisy environment.
  - Use higher clock rate than conservative design or increase spacing between adjacent buffers.

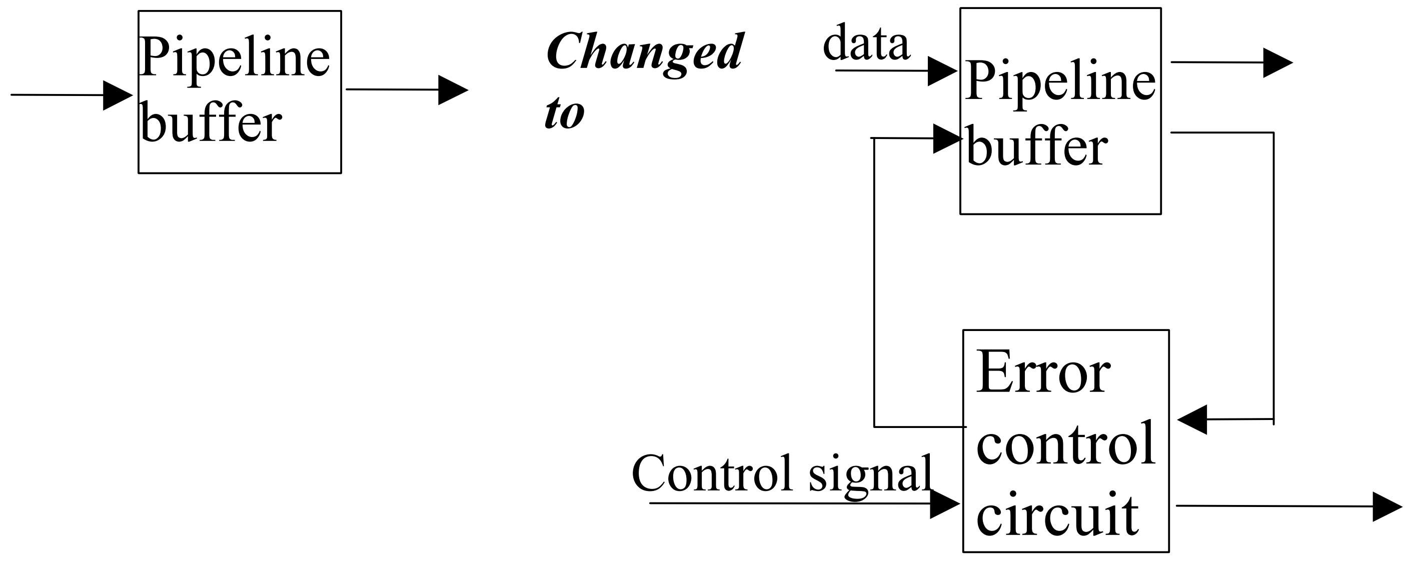
Timing Error-Tolerant System

- Timing error-tolerant (*Terror*) system
  - Detects and corrects timing errors with minimal impact on latency.
  - ***Terror systems can tolerate delay variations giving large latency savings (upto 35%) over traditional retransmission scheme.***
  - ***Only transient timing errors are corrected, static errors due to logic faults, soft errors etc. are not corrected.***

Buffered link design

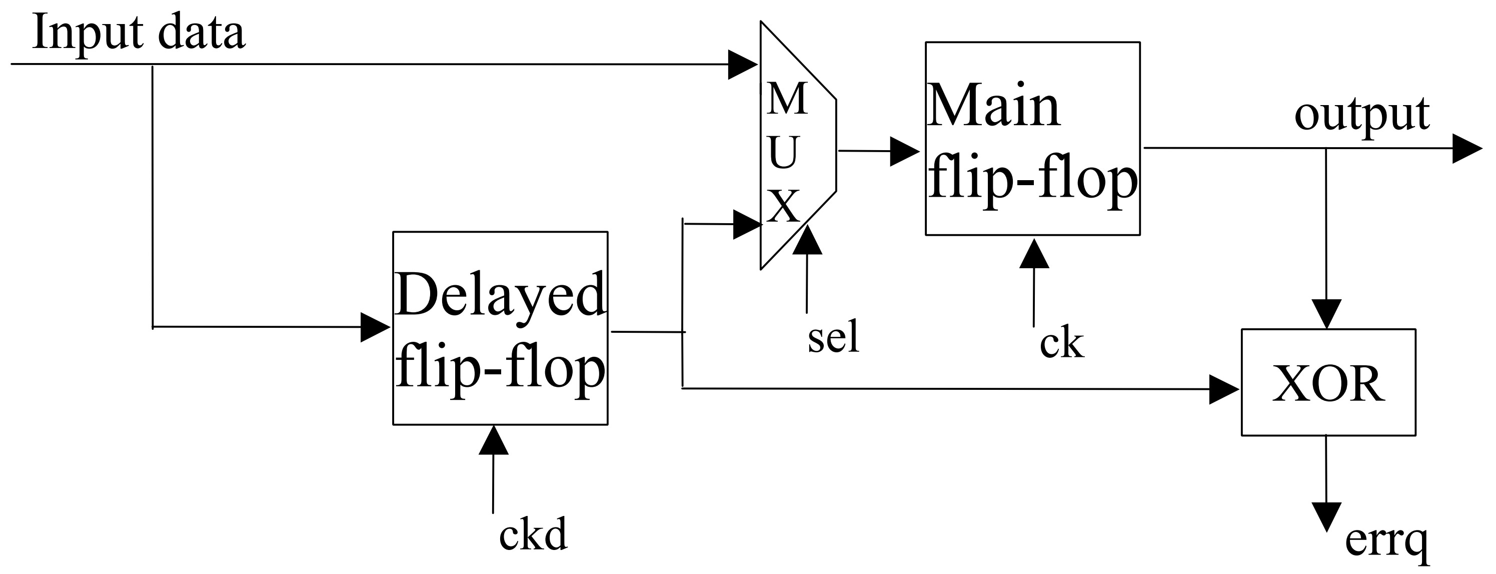


Typical pipelined link design with b stages



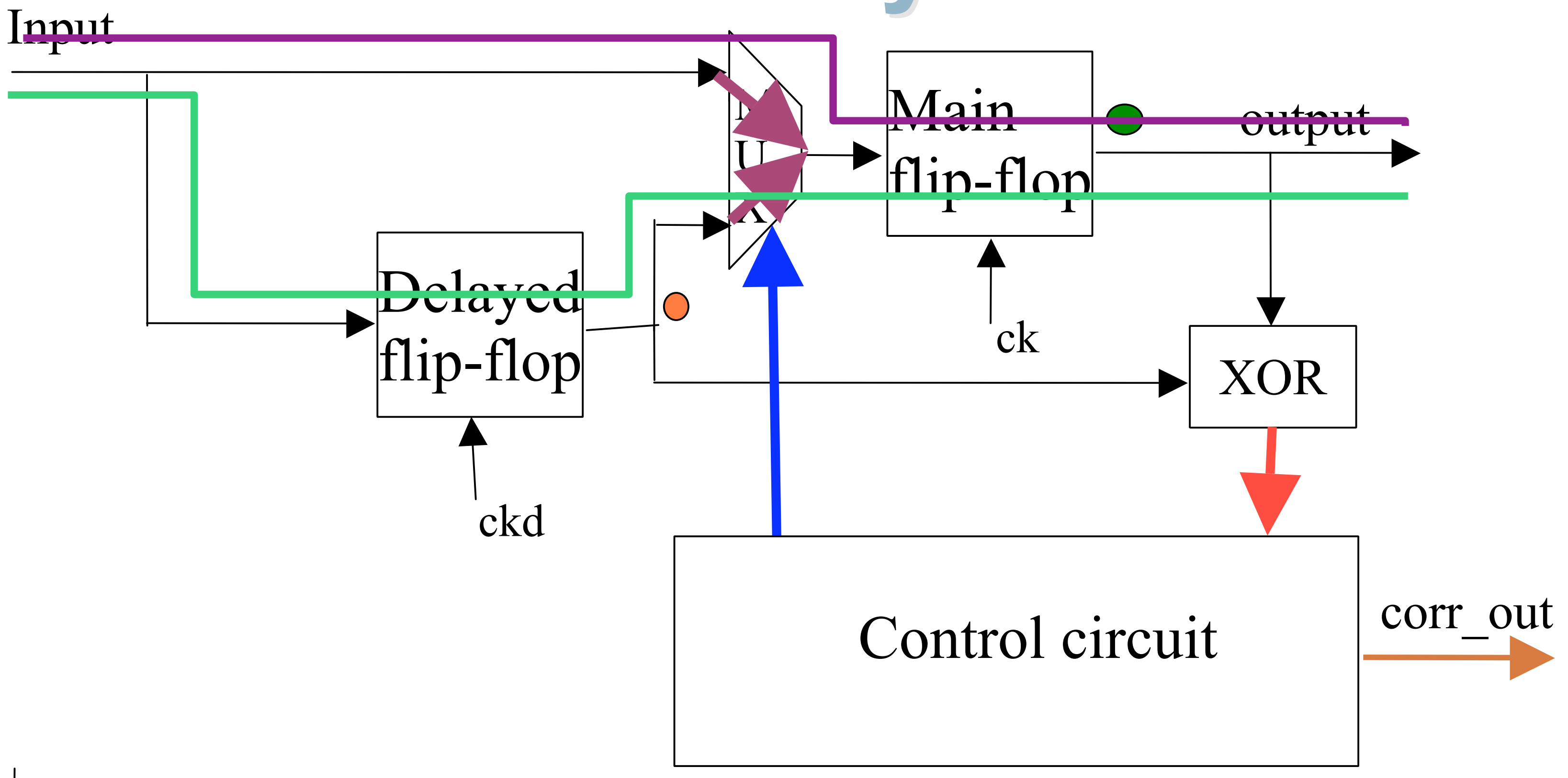
Proposed Design

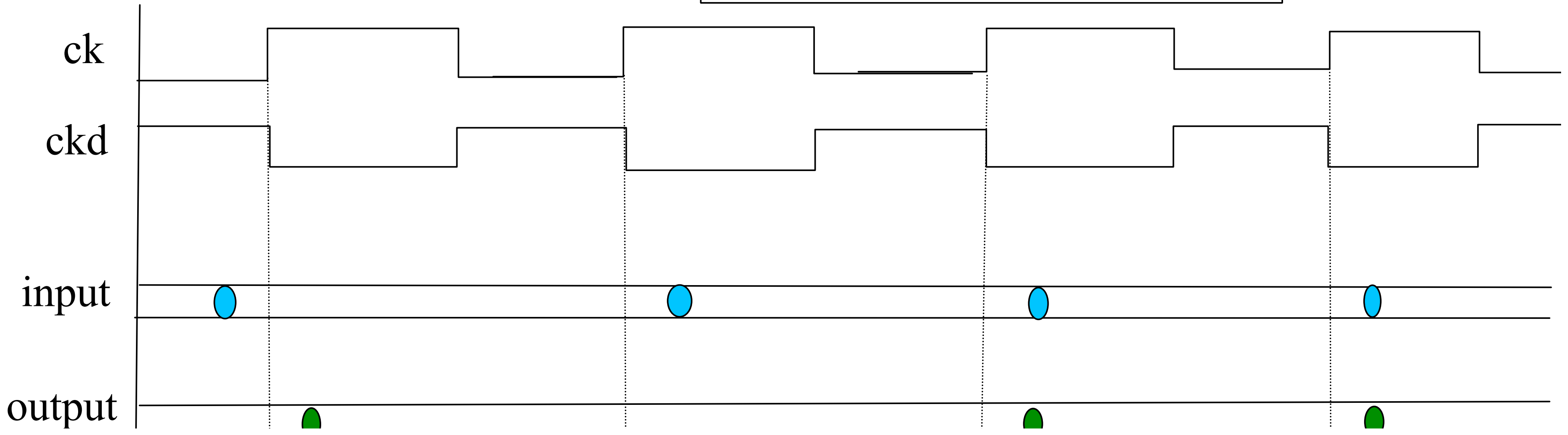
Terror Design Principle



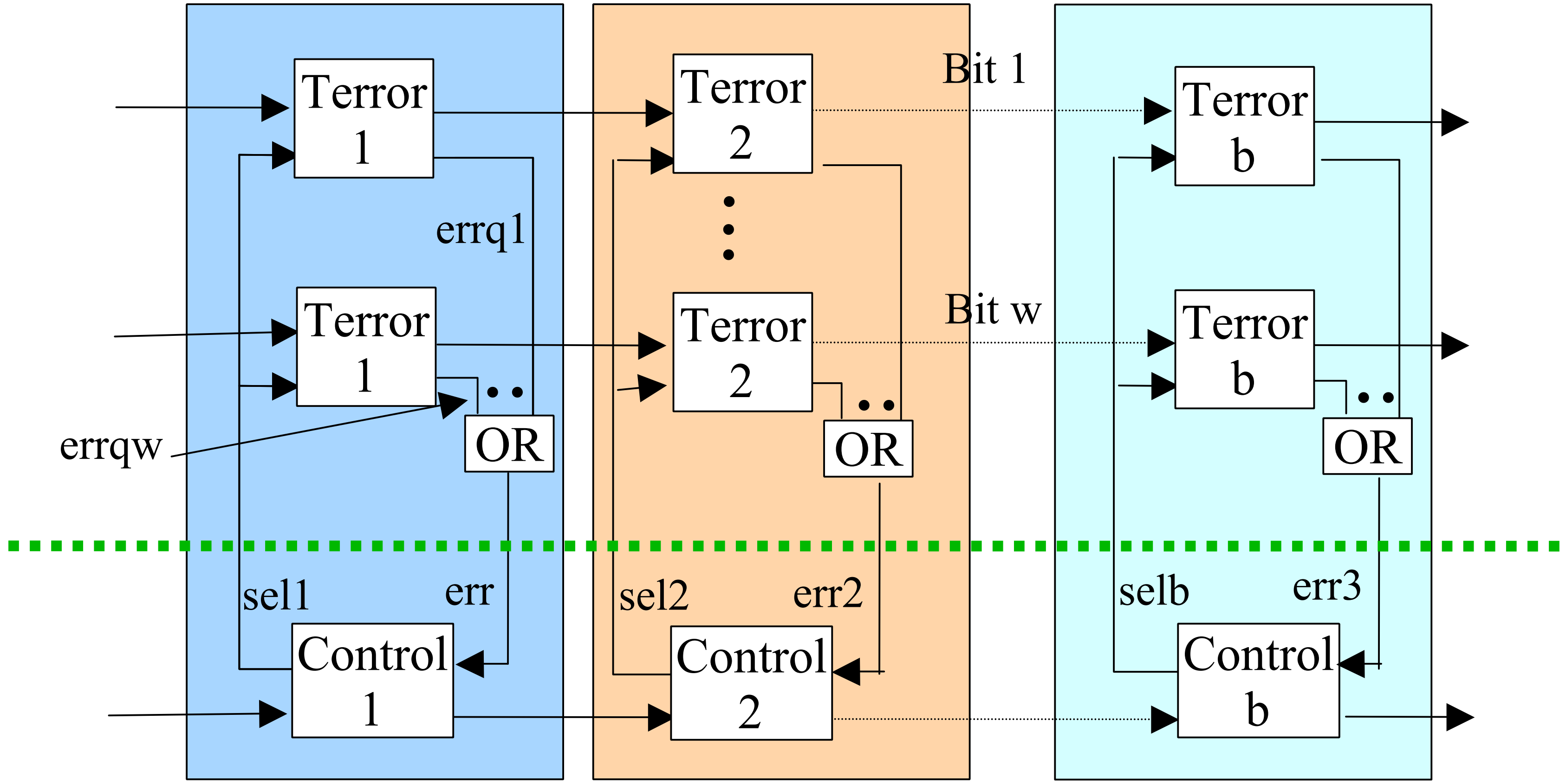
- In *normal* state, *data is captured and sent by main flip-flop*
- XOR detects difference in data captured by delayed flip-flop and main flip-flop
  - Delay between *ck and ckd ensures data bits get sufficient time to reach delayed flip-flop.*

Normal to Delayed state



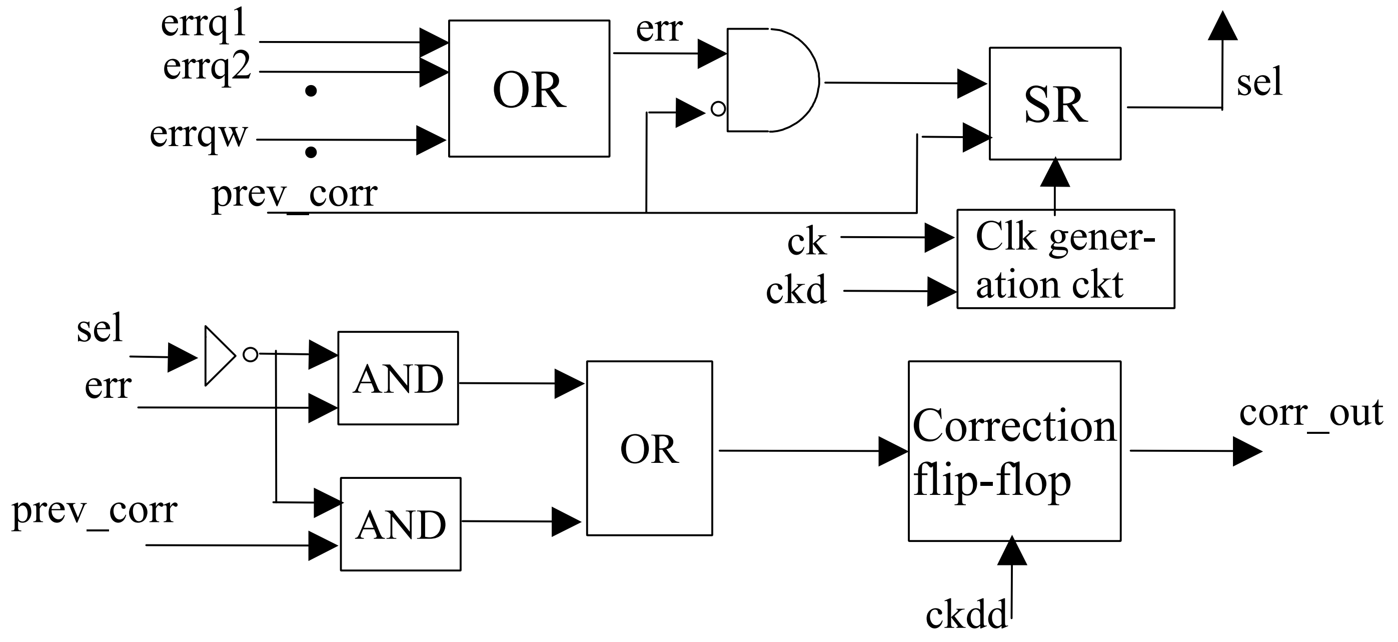


Terror enabled Link design



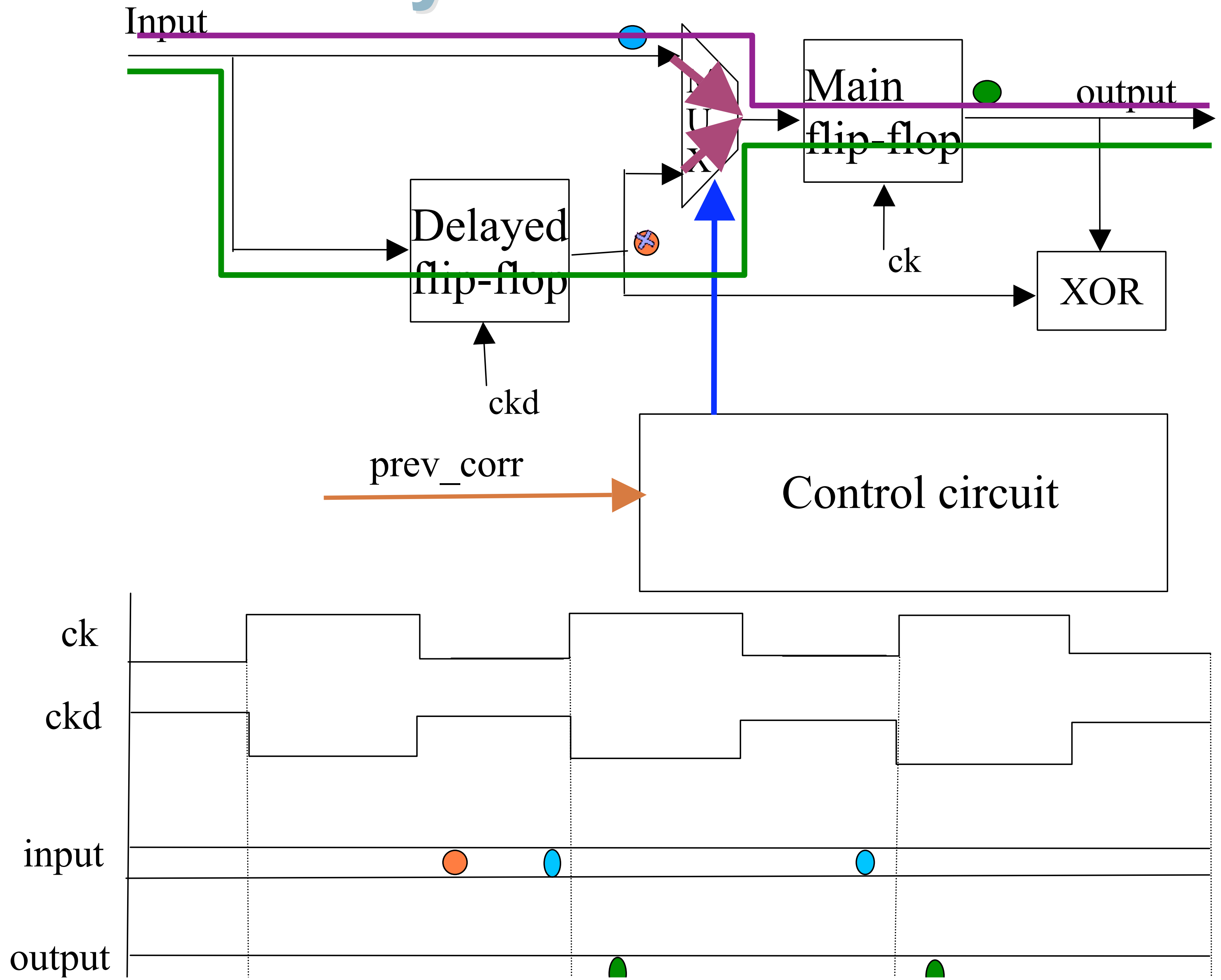
- One error correction circuit common to *w buffers*
  - *Decreases overall cost of control circuitry*
- *errq signals of all buffers (vertically) are Ored and fed to the control circuit.*
  - *Avoids synchronisation circuit at the end*

Error control circuit



- corr_out signal indicates data sent on previous cycle was incorrect (due to a timing error)
  - Clocks ckd and ckdd are locally generated by a delay chain (chain of inverters)
- SR latch is set when err =1, and is reset when prev_corr =1

Delayed to Normal state



Alternative approaches

- Teatime [Comp. '04] tracks logic delay and avoids errors by changing the clock frequency.
  - Requires complex frequency controller and tracking logic
- Razor [Micro '04] monitors error rate to control power.
  - Uses gated clock or flushes pipeline to correct error.
- Favalli et al. [DFT/VLSI '97] use encoded data and decoder at flip-flop input to detect errors.
  - Overhead of decoder at each flip-flop input.
- Mousetrap[ICCD '01] uses high speed asynhcro-nous pipeline.

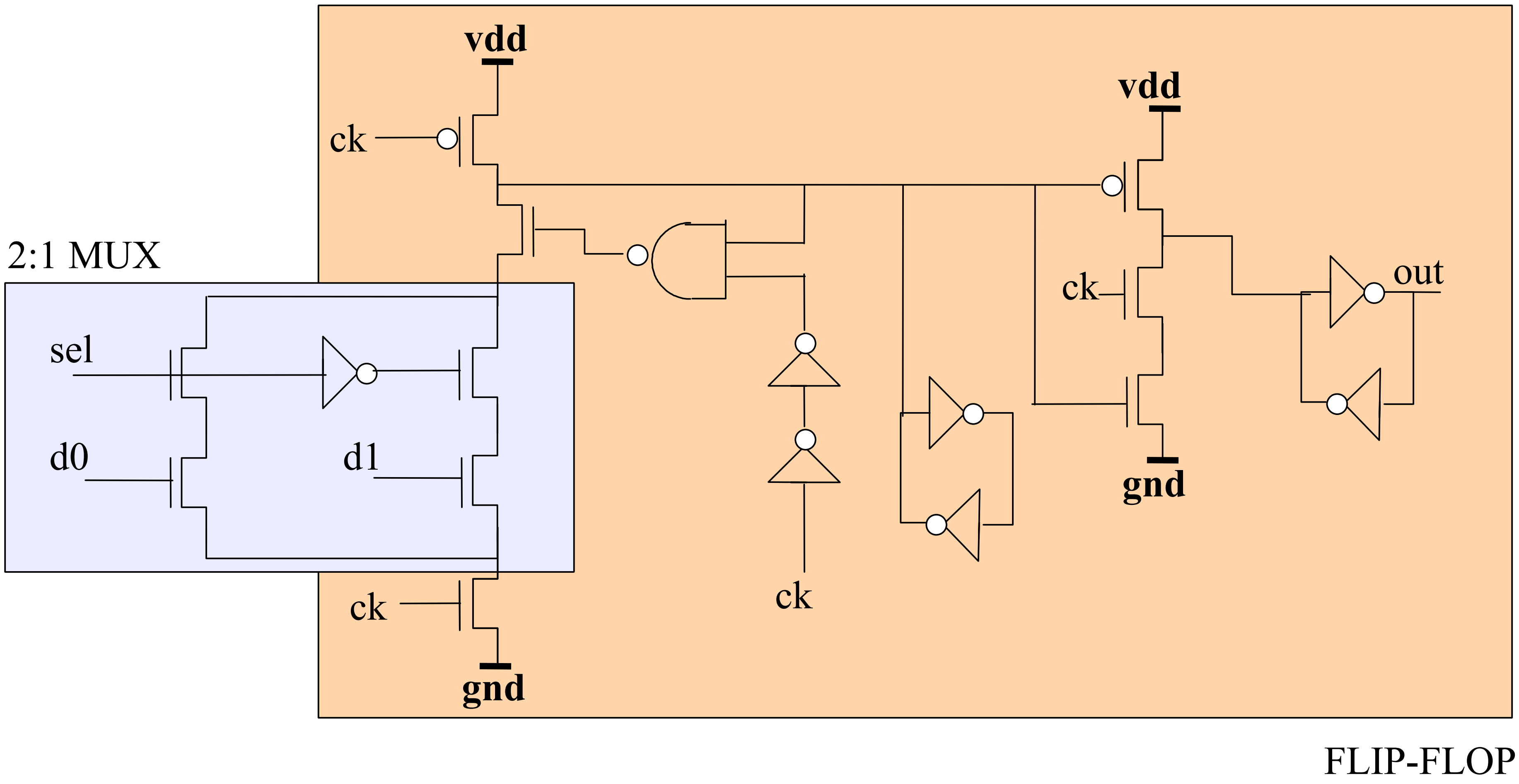
Comparison of latency

- Previous approaches give large latency overheads for high error rates and do not scale well bus widths.
  - Error penalty increases linearly with error rate.
  - Difficult to correct multiple errors.
- For example, if errors occur each cycle, for N bits
  - Razor [Micro '04] : % of useful cycles = N/2N = 50%
  - Terror : % of useful cycles = N / (N + b)
    where b = no. of buffers on the link
- For b << N, the benefits are substantial

Transistor level design

- Terror element was designed for 32 bit bus in 100 nm technology targeted at 1Ghz frequency.
- Transistor level optimisations were included
  - Include 2:1 Mux inside the main flip-flop.
  - Use domino OR instead of static OR for errq signals
  - Combining AND-OR gates into correction flip-flop
  - Using a simple SR latch design.
- Timing overheads were calculated based on SPICE simulations.

Transistor level design



Schematic showing 2:1 Mux inside a flip-flop

Timing Overheads

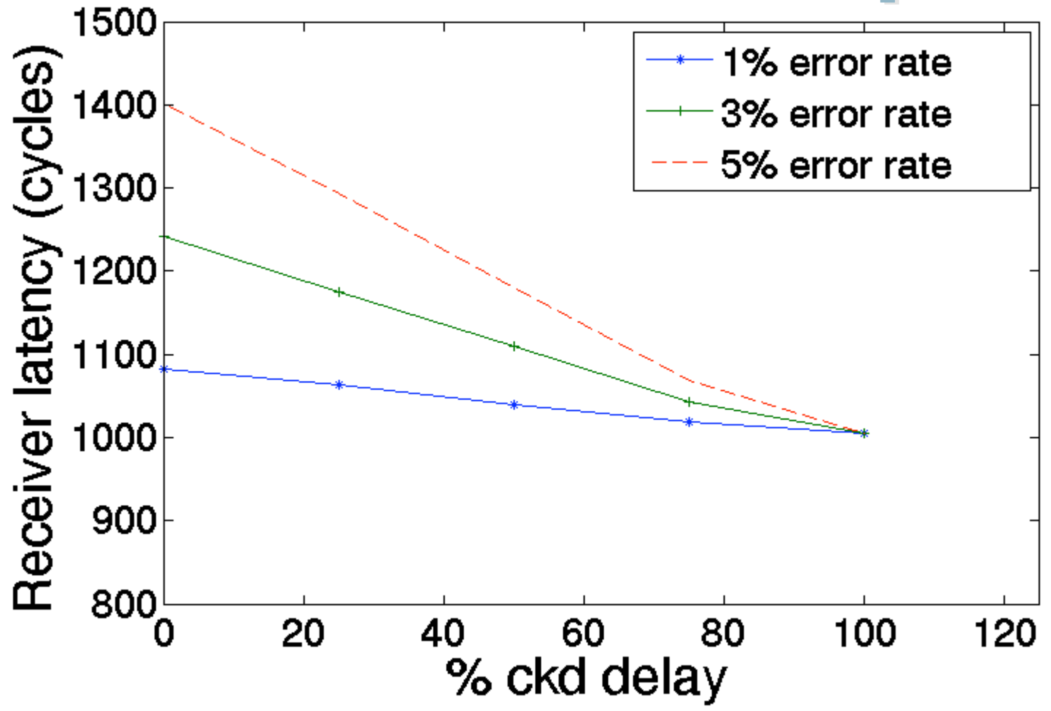
Parameter	% Overhead
Hold time	10.0
Normal->Delayed	27.0
Delayed->Normal	9.7

- Ideally ckd can be delayed by one cycle
- Practically, ckd delay limited by finite timing delays
  - Hold time flip-flop and logic delay involved in going from normal to delayed state and vice-versa.
  - Hence range of variation of ckd is
    100 – 9.7 – 27 – 10 = 53.3% of cycle
- To simplify, we use 50% delay variation for ckd
  - Clock ckd can be delayed by half clock cycle.

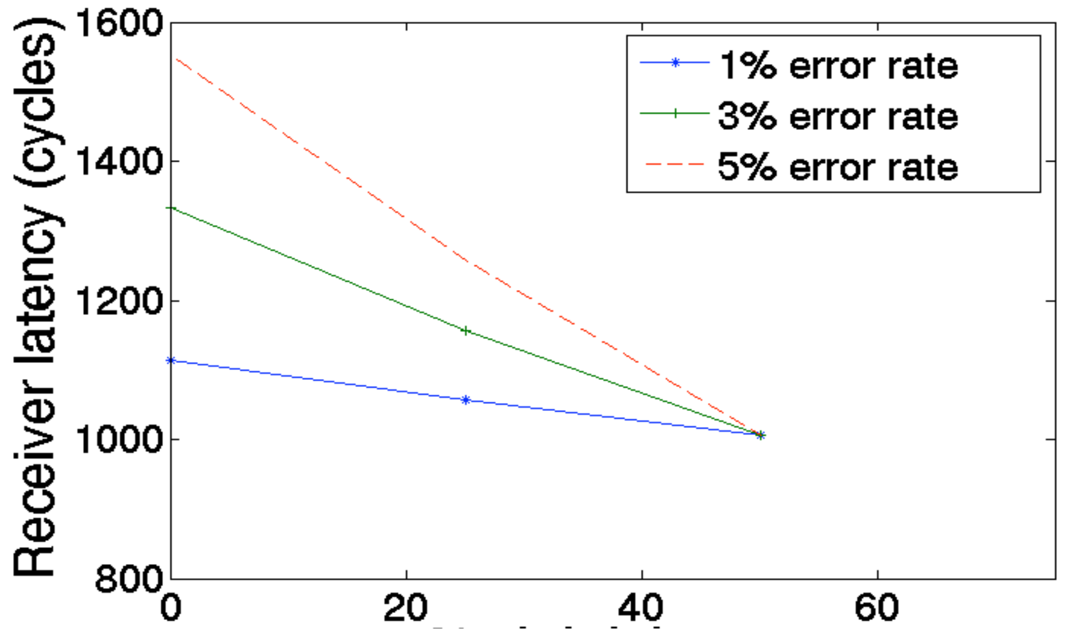
Simulation results

- We considered a SoC with on-chip link length of 12 mm operating at a frequency of 1Ghz
- With conservative design approach assume distance between successive stages as 2 mm.
- Thus number of stages required are 6.
- A Terror enable pipeline will have 3 stages, ideally (since ckd can be delayed by one cycle)
- Practically ckd can be delayed by 50%, so number of stages with Terror are 4.
- We plot the receiver latency for different error rates for 1000 data bits in both cases.
  - Errors not detected by Terror are corrected by retransmission scheme.

Simulation plots



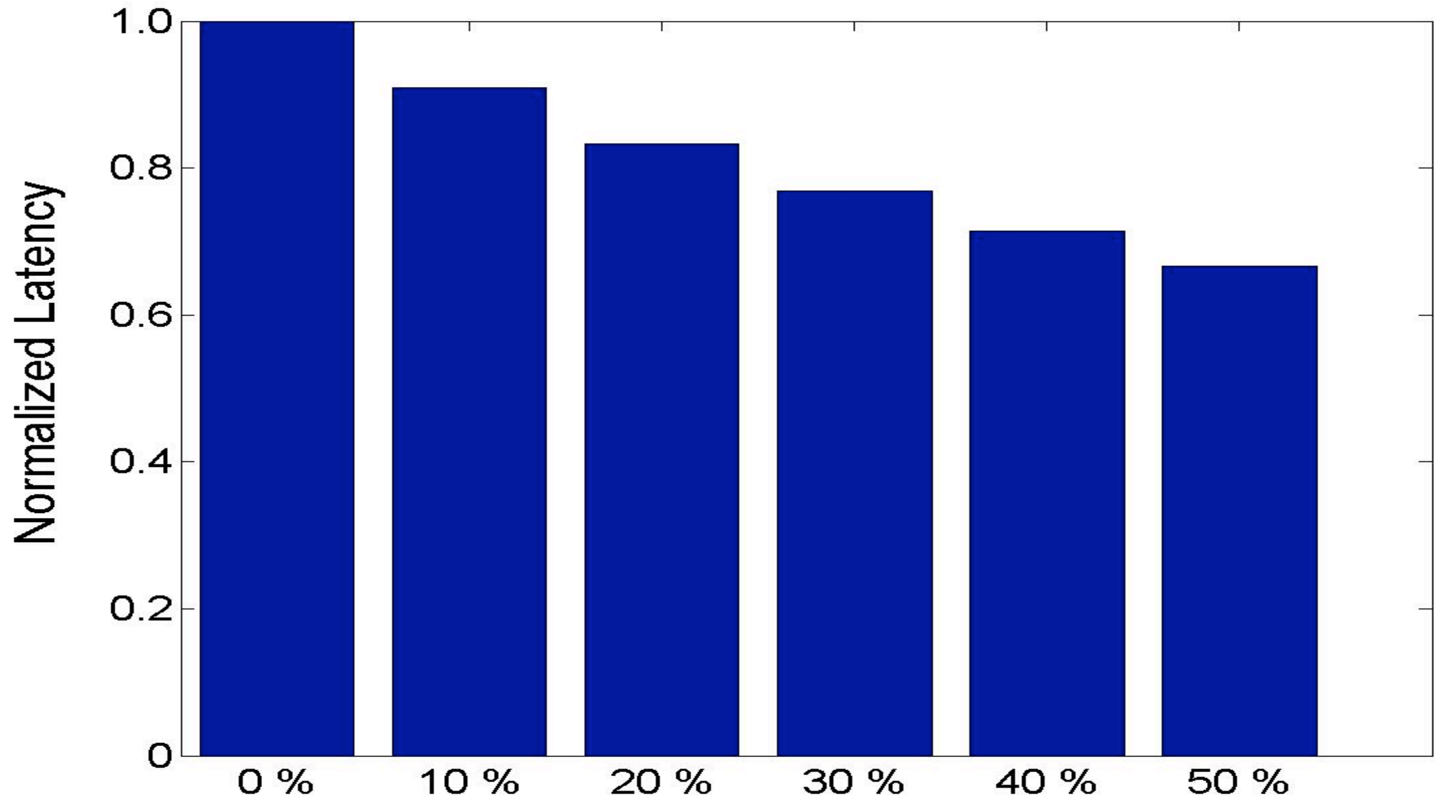
Receiver latency variation with delay between clocks *ck* and *ckd* for ideal case



Receiver latency for practical case

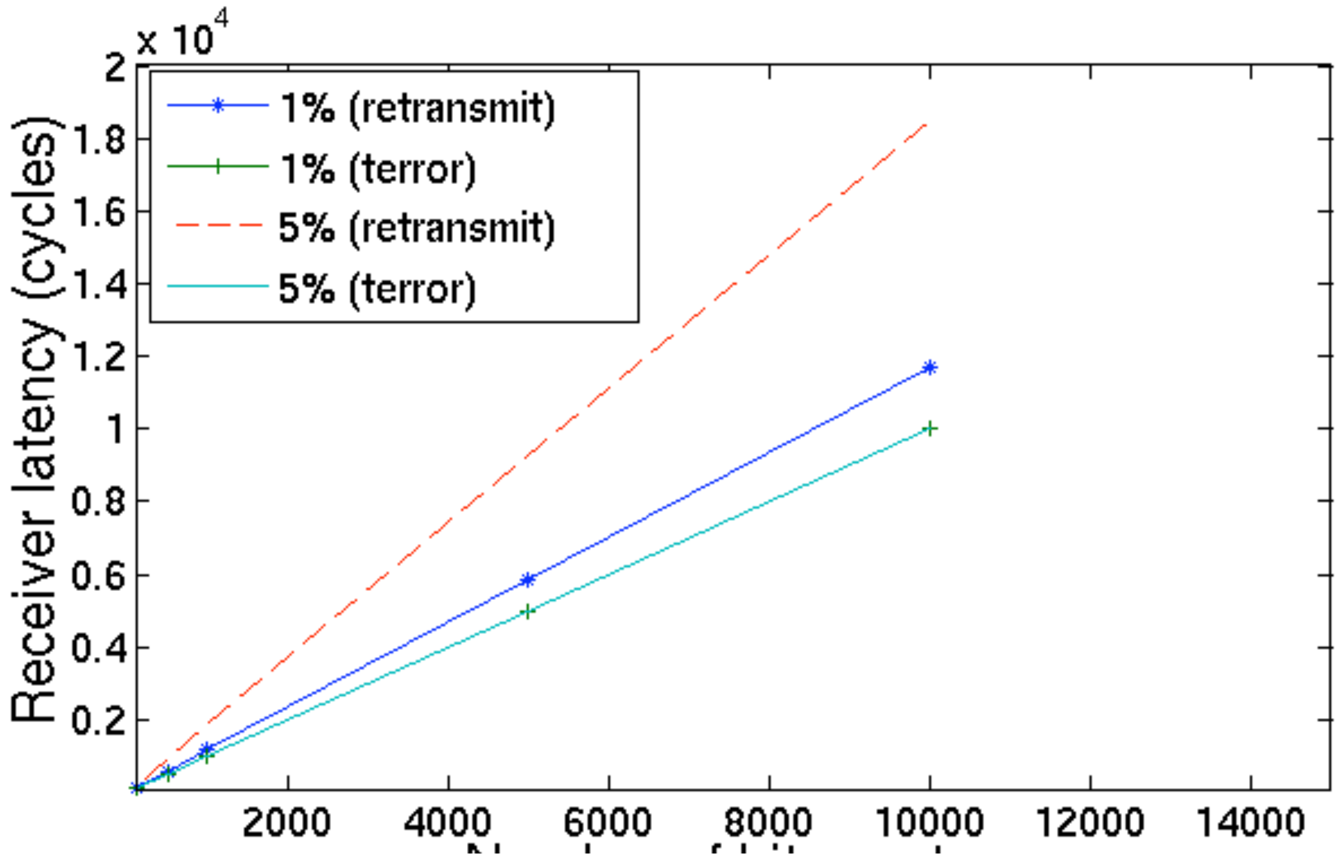
Aggressiveness Vs Latency

- We define aggressiveness = percentage increase in inter-buffer spacing over conservative design.
- Latency reduces by 33% for 50% aggressiveness.
  - Plot terminates at 50% since distance between pipeline stages can only be increased by 50%.



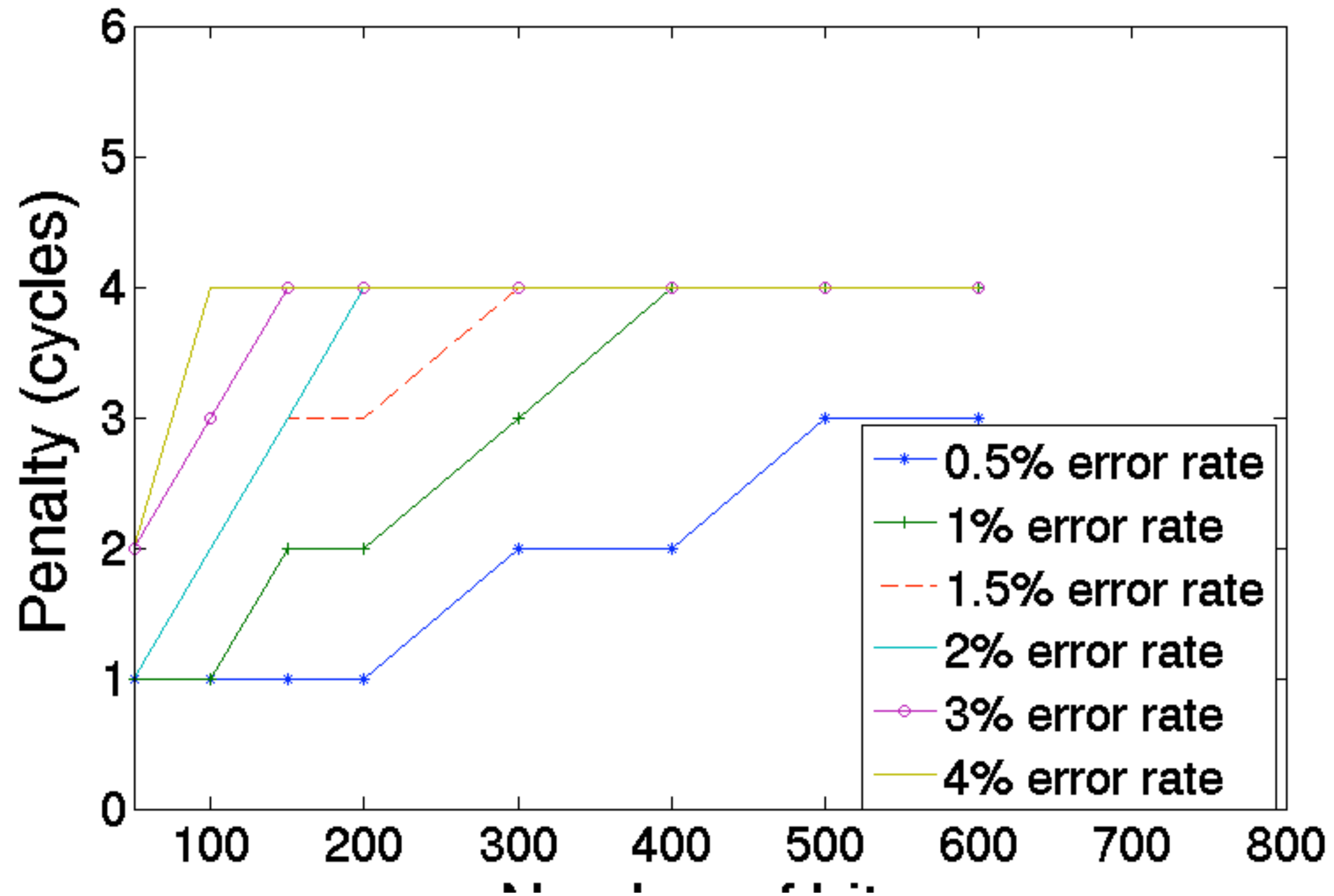
Retransmission Vs Terror Scheme

- Receiver latency for receiving 1000 bits at 1% and 5% error rate is plotted for both schemes.
- Terror scheme gives a 35% reduction in latency.



Maximum Penalty Vs Data size

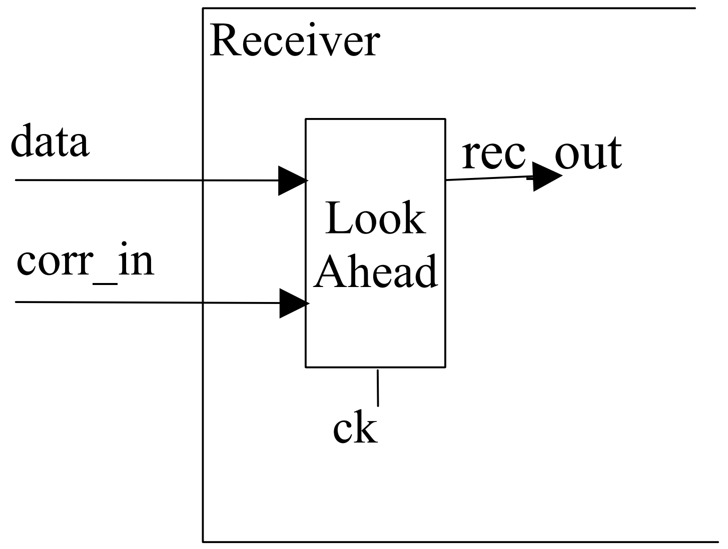
- Variation in penalty for different error rates and data sizes for a 4-stage pipeline is plotted.
- Maximum penalty is 4 cycles.



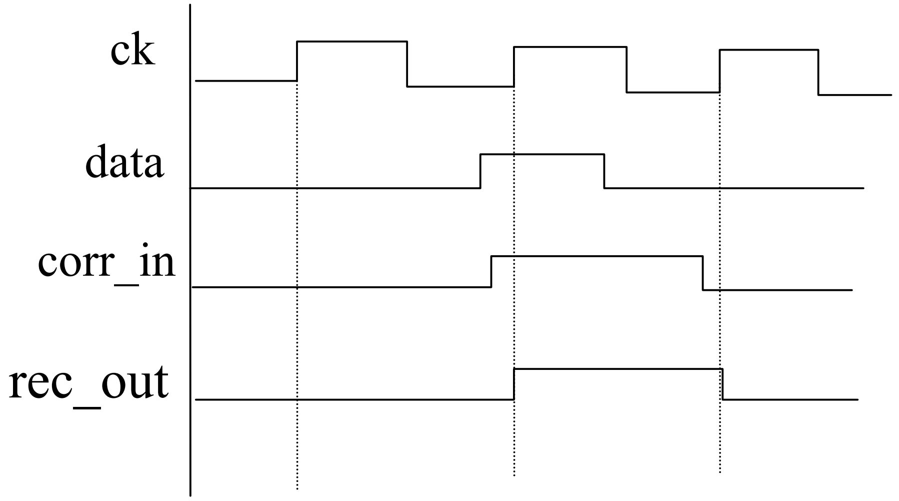
Error Penalty

- Maximum latency is bounded by the number of buffer stages and is not affected by the error rate.
- Latency overhead does not increase for large data sizes or high error rates.
  - Typical error correcting schemes degrade at high error rates.
- Terror enabled systems are suitable for high error rate designs.
  - This can happen in high noise environment or at low voltage levels.
  - Used in aggressive designs, where clock frequency is higher than conservative value or physical spacing between buffers is increased.

Receiver design



Receiver design



Look-ahead stage operation

- Receiver looks ahead 1 cycle for the data.
- Only first bit incurs 1 cycle penalty since other bits follow in a pipeline fashion.
  - 1 cycle penalty can be hidden in switch, it only occurs at the end receiver.

Area Overhead in typical SoCs

Design	Area overhead
Merlot	0.12%
DSP	0.6%
MIT RAW	0.86%
Alpha MP	0.9%
Average	0.62%

- We estimate area overhead in typical MPSoCs
  - Based on increase in gate count due to Terror.
- At a 0.6% increase in area, we get large latency savings (upto 35%)
  - Power overhead is negligible.

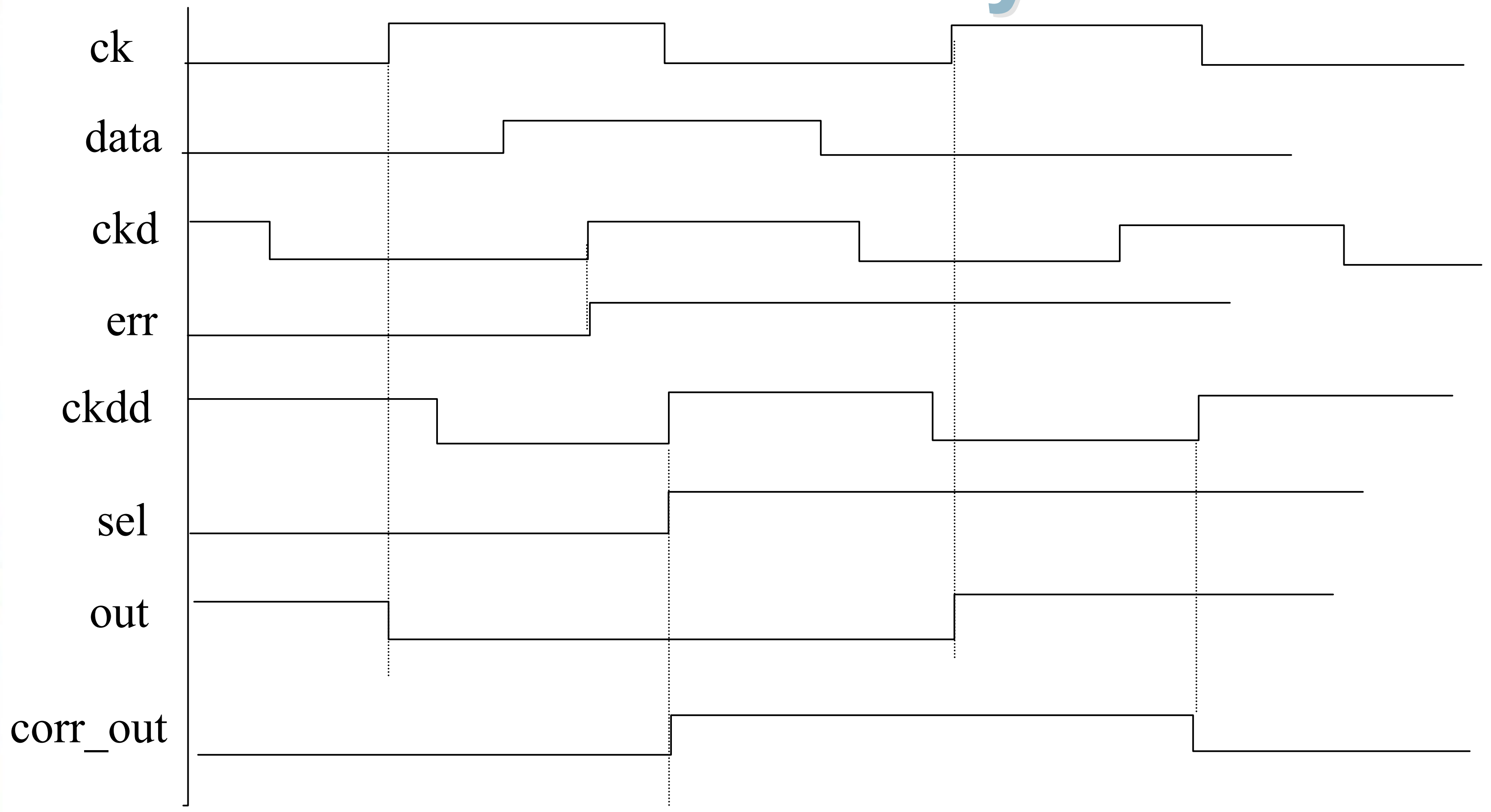
Conclusion

- Reliability, delay and throughput of link is affected by unpredictability in link characteristics.
  - Increasingly affected by cross-talk, other interferences.
- Terror enabled systems tolerate such variation in link delay and encourage an aggressive design approach.
  - Provide a 35% savings in latency over traditional approach
- Network on Chips (NoCs) a communication centric approach will be required for future SoCs
  - Provides scalability and reliability for efficient communication between cores.

THANK YOU

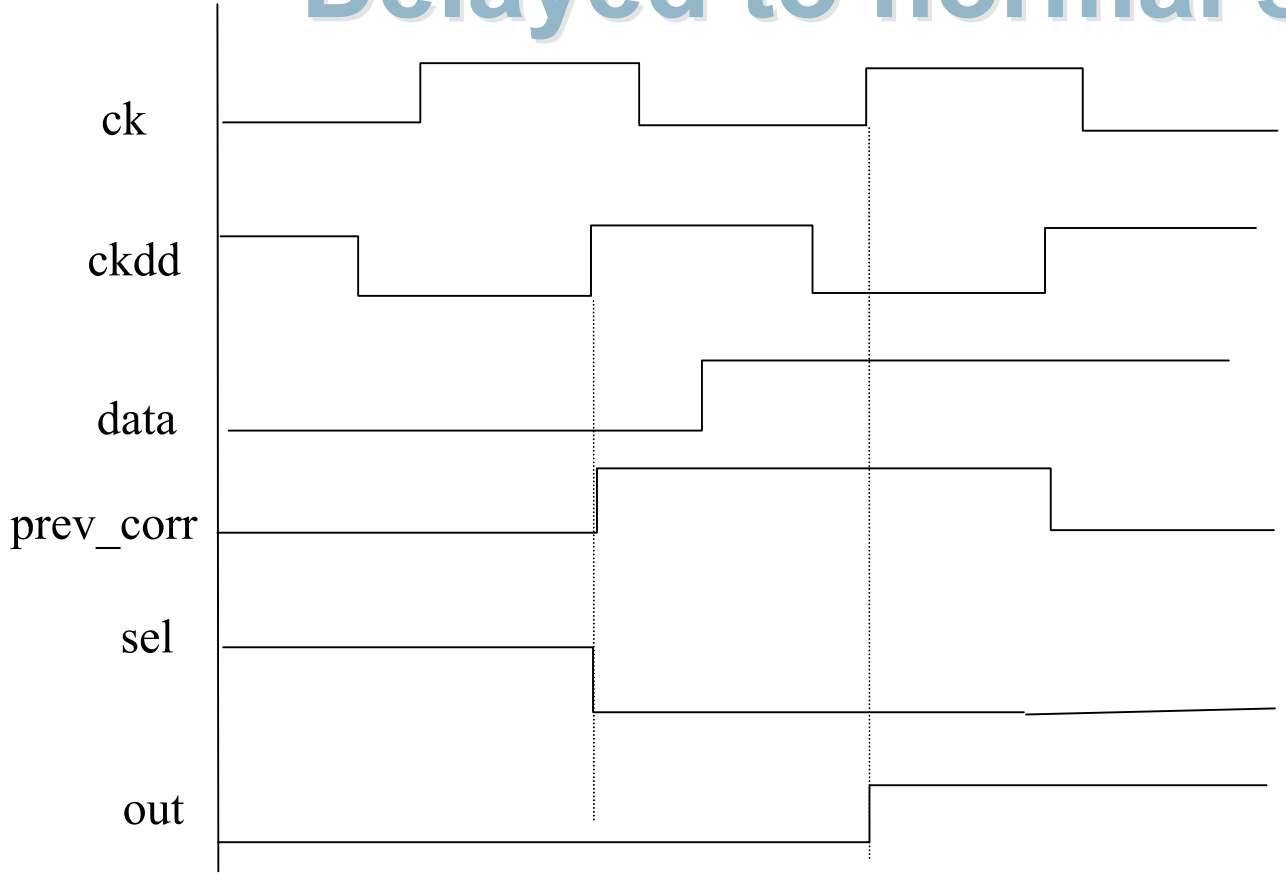
EXTRA

Normal to delayed state



- *Terror goes into delayed state when err =1.*
- *In delayed state, data is captured by delayed flip-flop and sent by main flip-flop.*
  - *One cycle penalty occurs for the first occurrence of*

Delayed to normal state



- Terror returns to normal state when prev_corr signal is received.
  - Incorrect data captured by delayed flip-flop is not sent.
  - *For proper operation, prev_corr signal is made error free by shielding from other wires, routing the*

Analysis of Penalty

- Maximum latency bounded by pipeline stages.
- Average penalty depends upon the *when* and *where* timing errors occur along the pipeline.

$$1 \leq \text{ Penalty} \leq b \text{ (for b link buffers)}$$

- One cycle penalty occurs when error in (b-1)th stage is absorbed by bth stage since bth goes from delayed to normal state
- Worst case (b cycle) penalty occurs when error occurs first in pipeline stage 1 , then in stage 2, up to bth stage .

Timing Analysis

- Setup time increases due to 2:1 Mux
  - $T_{setup} = t_{setup(nominal)} + t_{\mathbf{d(mux)}}$
- **Setup time of correction flip-flop**
  - $\mathbf{T_{setup} = t_{and} + t_{or} + t_{setup(nominal)}}$
- **Minimum ckd delay is path delay of err signal**
  - $\mathbf{T_{ckd} = t_{ck\text{-}q} + t_{xor} + t_{domino\text{-}or} + t_{and} + t_{or} + t_{setup(nominal)}}$
- **Minimum ckdd delay is path delay of sel signal**
  - $\mathbf{T_{ckdd} = t_{SRlatch} + t_{mux} - t_{setup(nominal)}}$
- **Hold time condition of correction flip-flop**
  - $\mathbf{T_{hold} < t_{SRlatch} + t_{and} + t_{or}}$