2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Clock Domain Crossing Challenges in Latch Based Designs

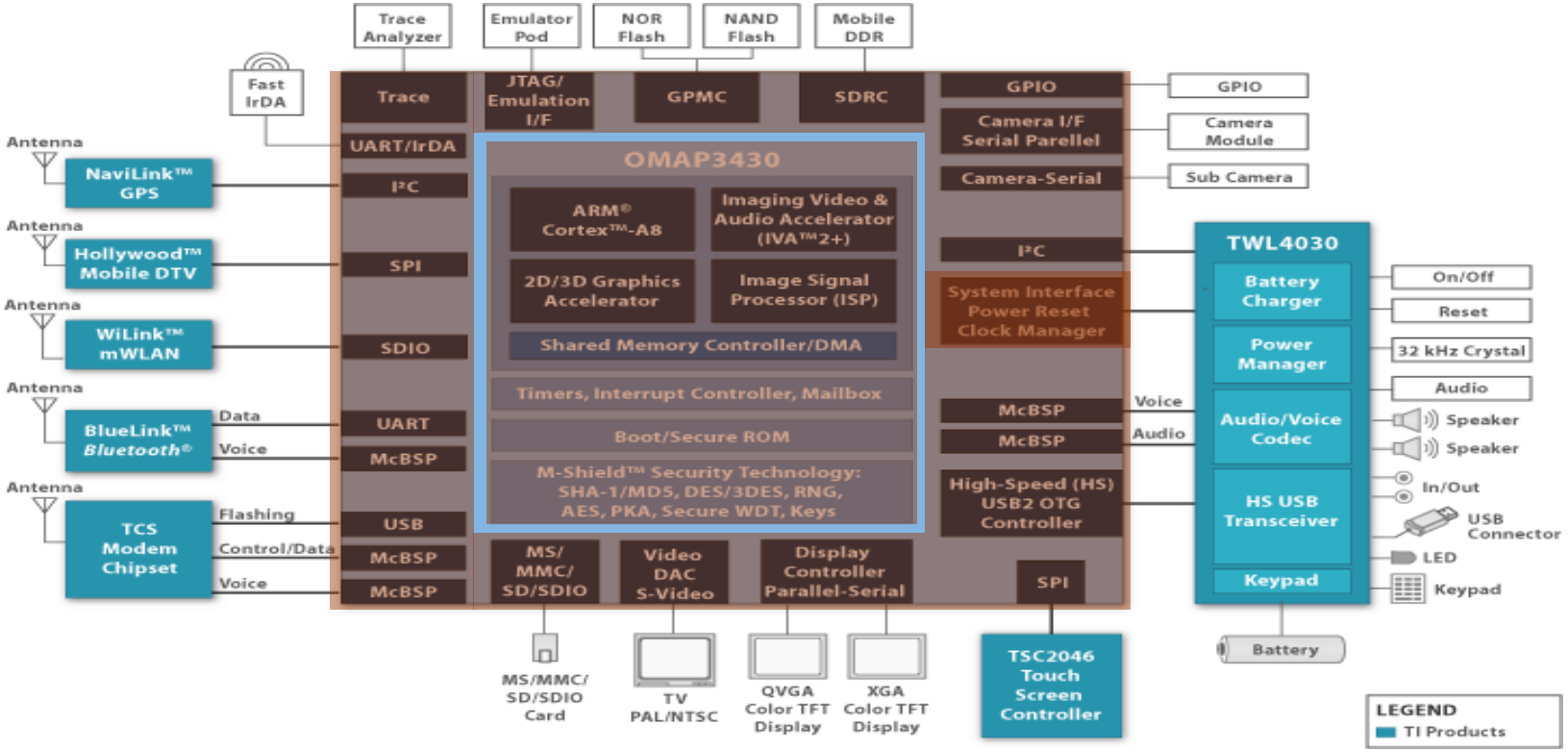
Madan Das, Chris Kwok, Kurt Takara

Mentor, A Siemens Business

Mentor®
A Siemens Business

Today's Devices Need Multiple Clocks

- Increasing System Integration
- Increasing Peripherals and External Interfaces
- Complex Power Management for Low Power



What is Clock-Domain Crossing (CDC)?

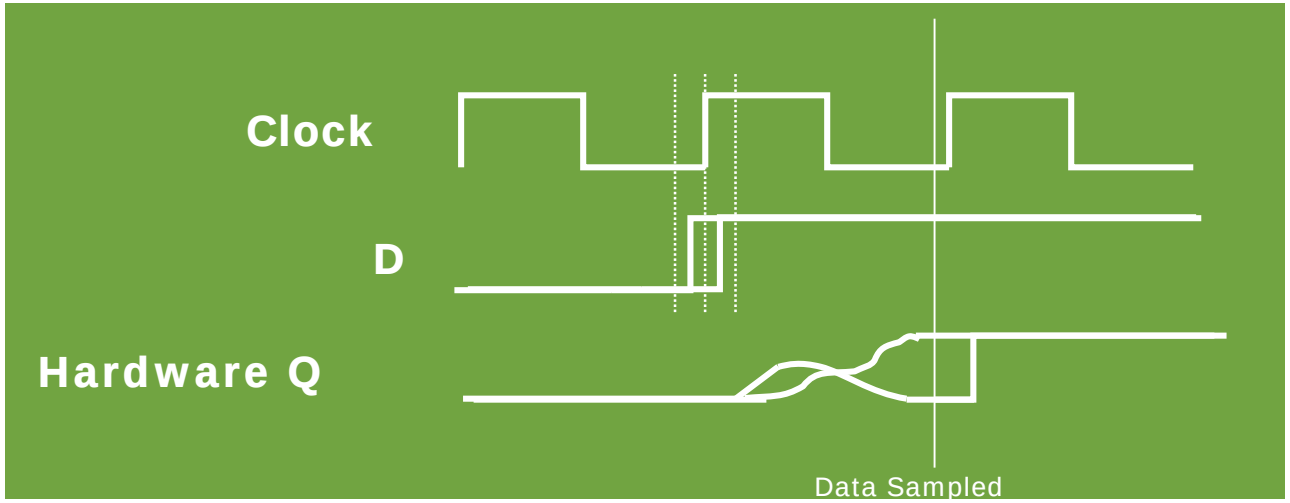
- Asynchronous clock domains
  - Contain registers or latches whose clocks have variable or unpredictable phase relationships vs. other domains
- CDC signals
  - Originate in one clock domain
  - Sampled by sequential logic in a different clock domain
- Today's SoCs can have >10^5 CDC signals on a single chip!

Why CDC Errors are Painful

- Usually not caught by simulation
- Found late in the development cycle
  - Often only in silicon
- Failures can be intermittent
  - Difficult to reproduce
  - Difficult to debug

CDCs Cause Metastability

- When a CDC signal changes values between the setup or hold time of a receiving domain's register
- Receiving register becomes *metastable*
  - Settles to a random value, after unknown amount of time



—Happens even with proper synchronization & protocols

—Can cause significant functional problems in the design

Metastability Happens

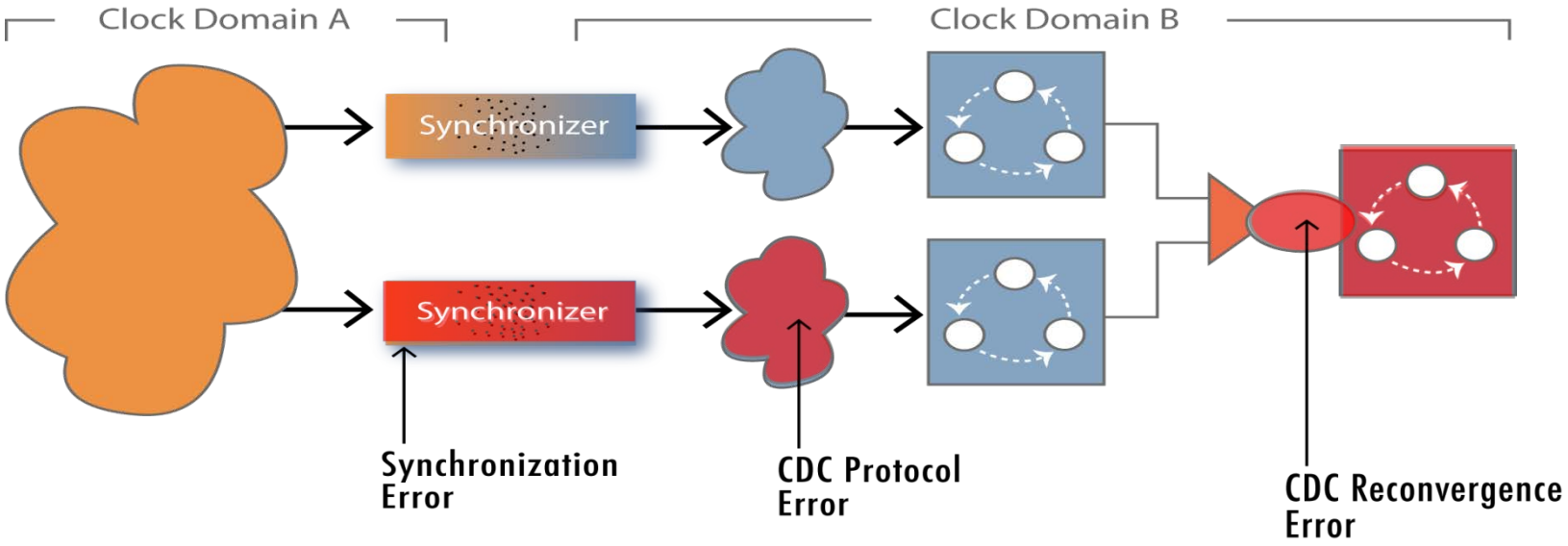
- Every type of storage element is susceptible to metastability
  - It's just a matter of time…

f_{clk} = Clock Frequency
f_{in} = Input Signal Frequency
t_d = Duration of critical time window

$$MTBF = \frac{1}{f_{clk} \times f_{in} \times t_d}$$

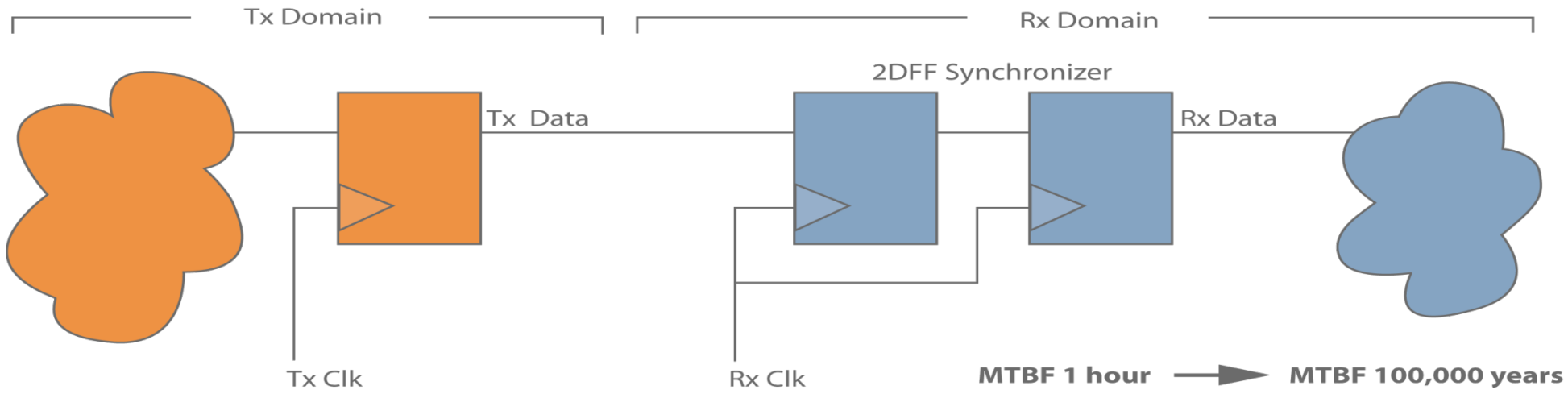
- Can your design handle it?

Common Types of CDC Errors



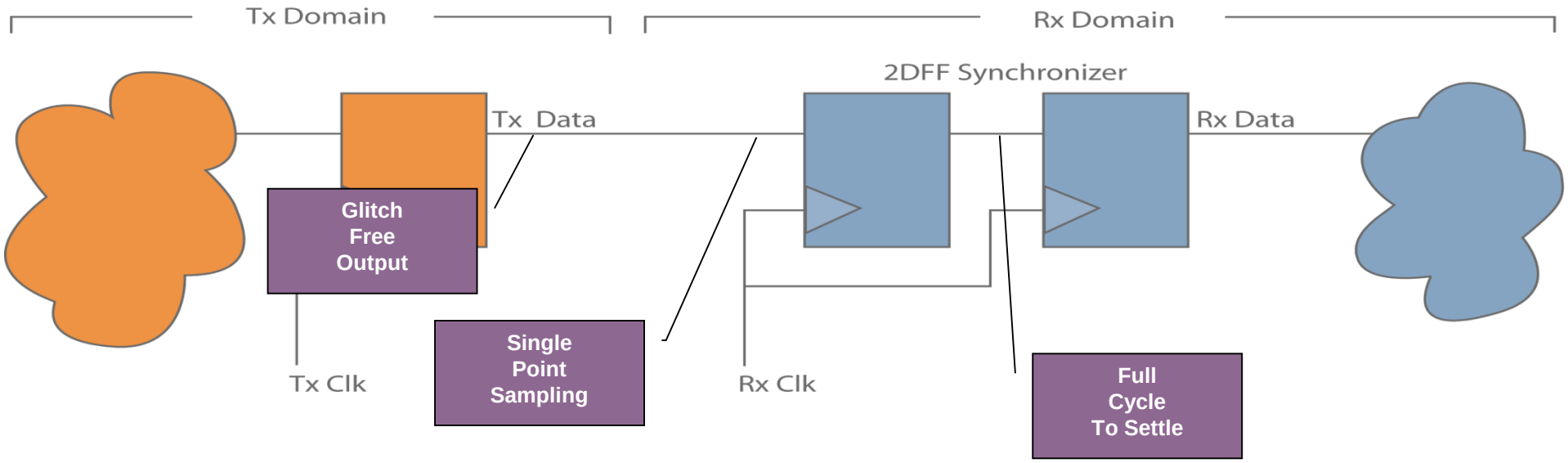
- Synchronization errors
- CDC protocol errors
- CDC reconvergence errors

CDC Synchronizers



- Designers add synchronizers to reduce the probability of metastable signals
- Synchronizers are sub-circuits that can prevent metastable values from being sampled across clock domains
  - Take unpredictable metastable signals and create predictable behavior

CDC Synchronizers (continued)

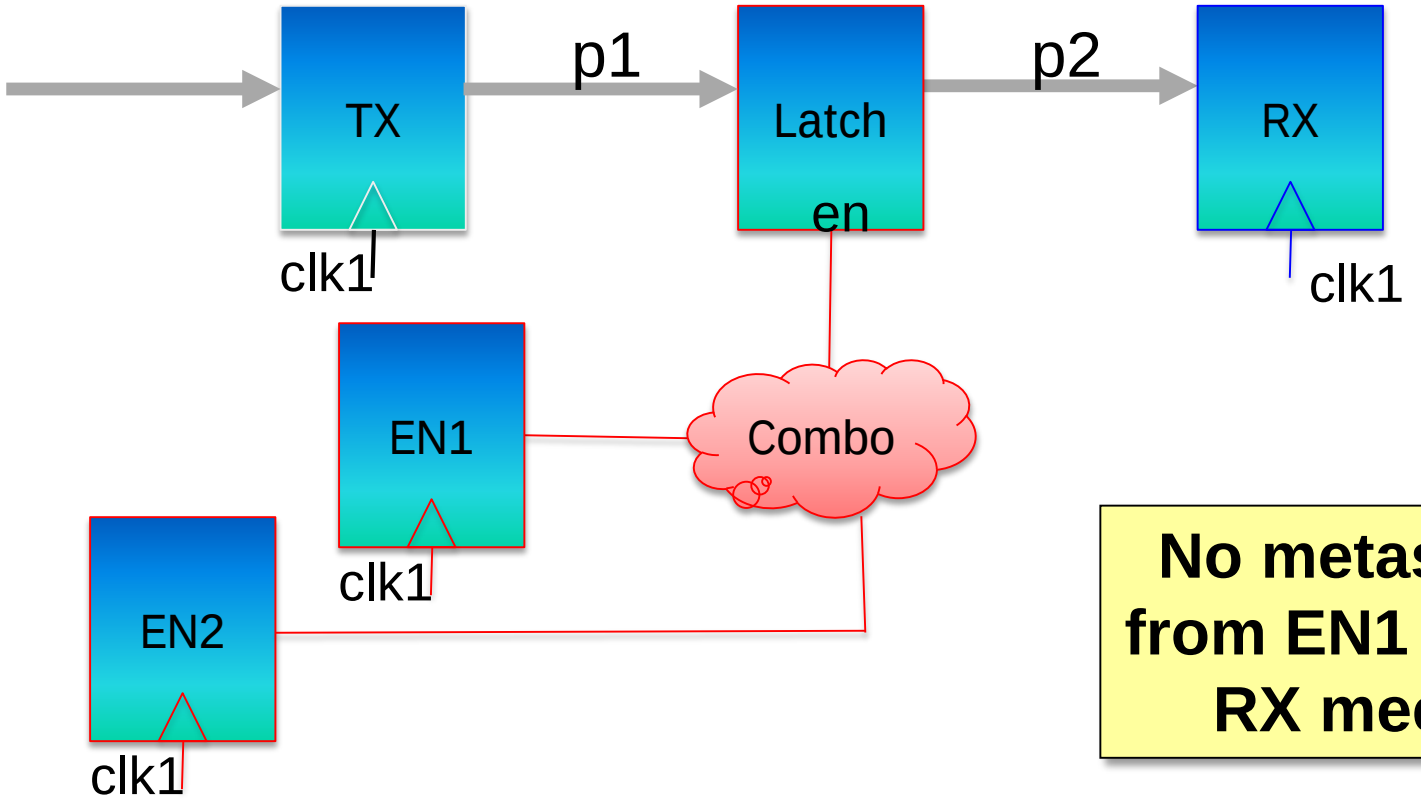


- Many different synchronizer types
  - "*Structured*" and "*Ad hoc*"
- Each structured synchronizer follows strict rules for *structure* that must be verified
- But…
  - Synchronizers can only resolve *unpredictable values*
  - Synchronizers cannot resolve *unpredictable delays*

CDC Paths for Latches

- Latches are transparent for a phase
- Latches have enables instead of clock
- Enable signals are typically clocks, but not always
- Unclocked latches may have the enable logic formed of control signals
  - A combinational logic output can drive the latch enable
- Clock network properties do not apply to such signals
- Treating the signals as clocks create additional clock domains
- To detect CDC paths, such latches must also be checked

Scenarios of Latch Enable: Same domain

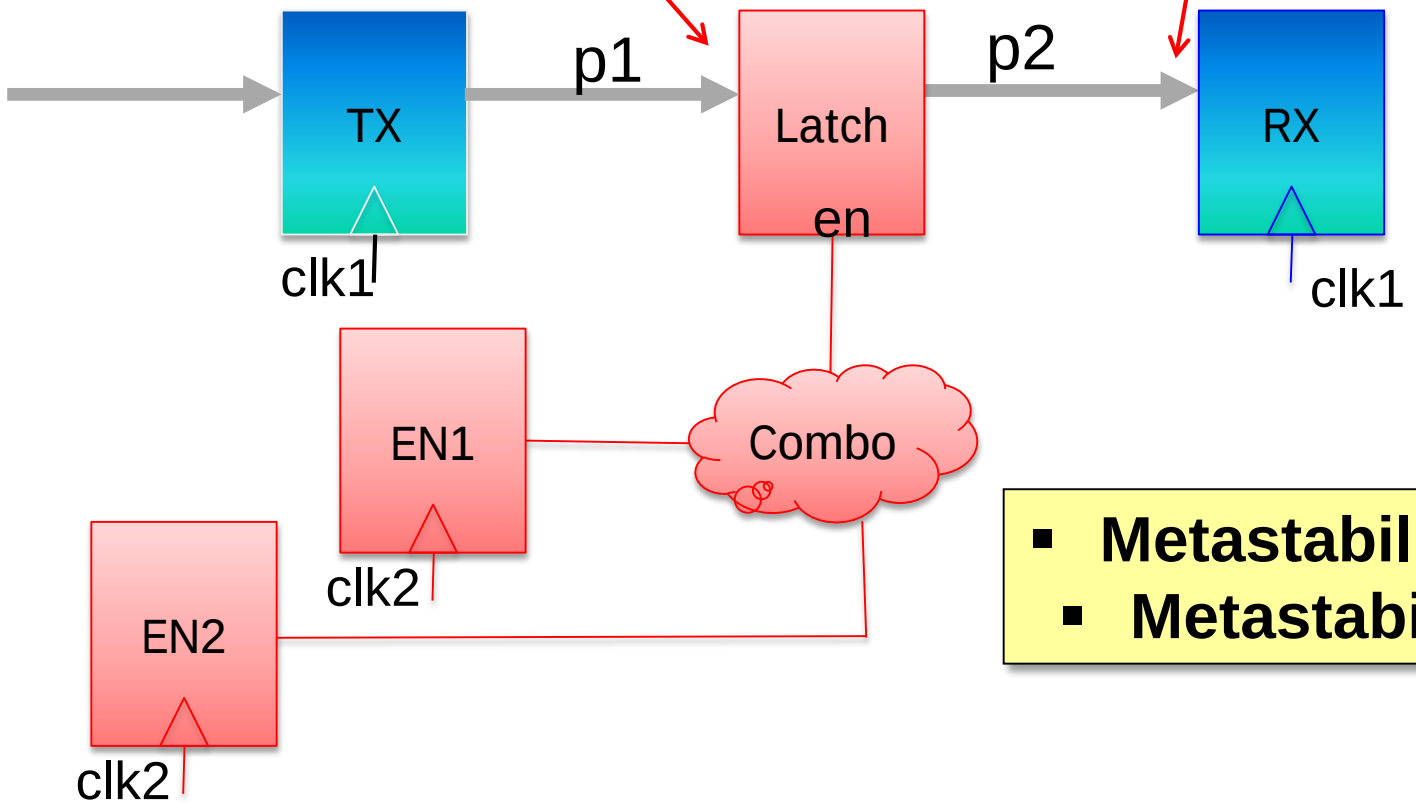


No metastability issue if path from EN1 and EN2 to D input of RX meet timing constraint

Scenarios of Latch Enable:Different domain

Latch enable can latch data while D input is changing

Latch enable can change D input of RX in its setup window

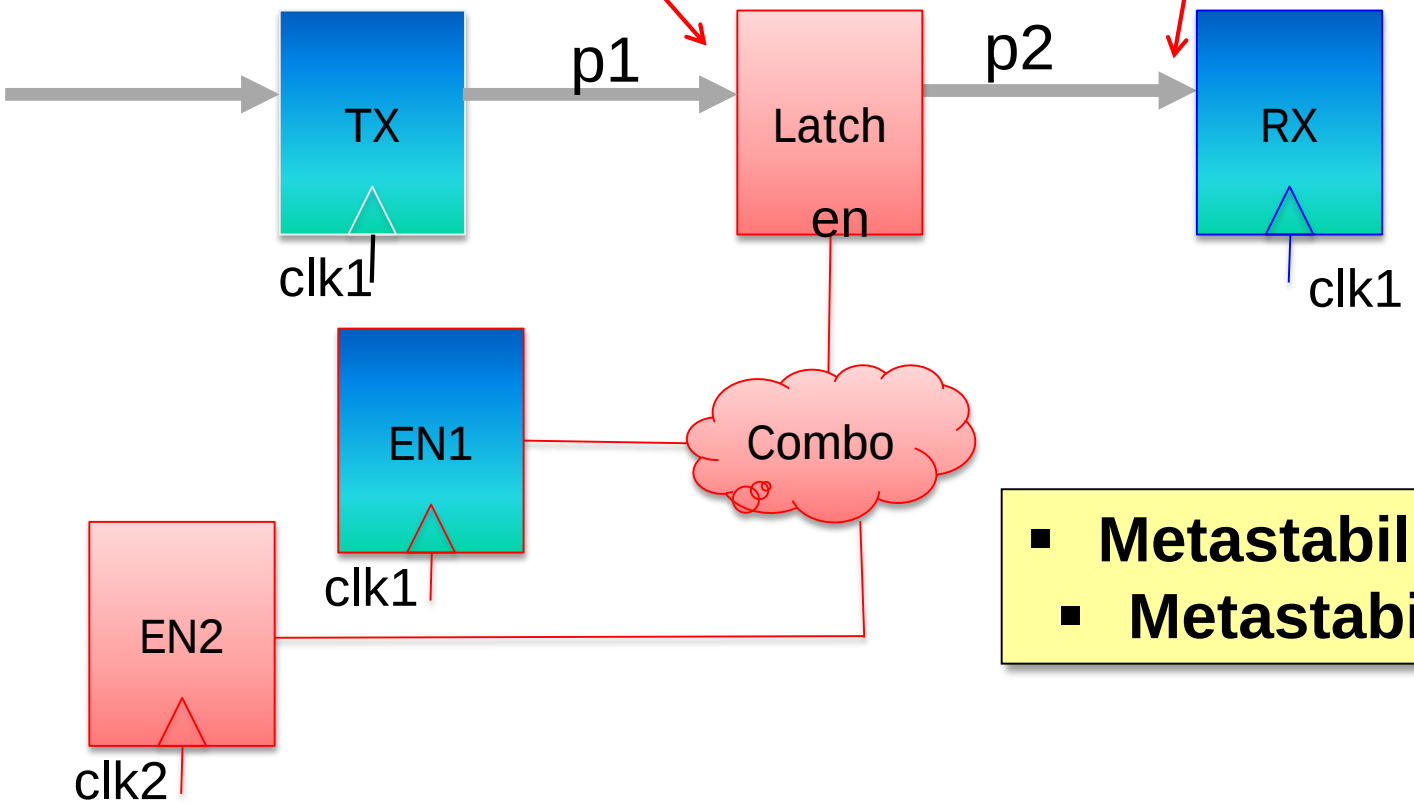


- **Metastability at latch data input**
- **Metastability at RX data input**

Scenarios of Latch Enable:Mixed domain

Latch enable can latch data while D input is changing

Latch enable can change D input of RX in its setup window



- **Metastability at latch data input**
- **Metastability at RX data input**

Traditional way of handling latch enables

- Treat latch enables as clock tree fanouts
- Foreach clock fanout
  - Traverse backwards till user specified clocks, primary portals, clock gating cells or clock muxes are found
  - Create the clock tree by linking the different parts of clock network
- Iterate till pieces of the clock network can be related to each other
- Check for CDC paths and schemes with clock domain assignments of registers, latches, memories and primary ports

Proposed way to handle latch enables

- Detect all clocks and divided clocks in the design
- For each latch *L* that is not enabled by a clock
  - Find set of registers and primary inputs driving the enable logic *Tr*
  - If all elements in *Tr* have same clock domain *C*, assign *L* to *C*
  - If elements in Tr belong to unknown clock domain, assign *L* to unknown
  - If elements in *Tr* belong to multiple clock domains *C1, C2, ..Cn*, then
    - Create a virtual clock domain *V(C1,C2,,Cn)* if it doesn't exist
    - Assign latch *L* to this virtual domain
- Check for CDC paths with this new domain assignment

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Benefits vs. creating new clocks

- More accurate analysis
- Fewer derived clocks
- Fewer CDC paths
- False clocks checks not triggered on control paths

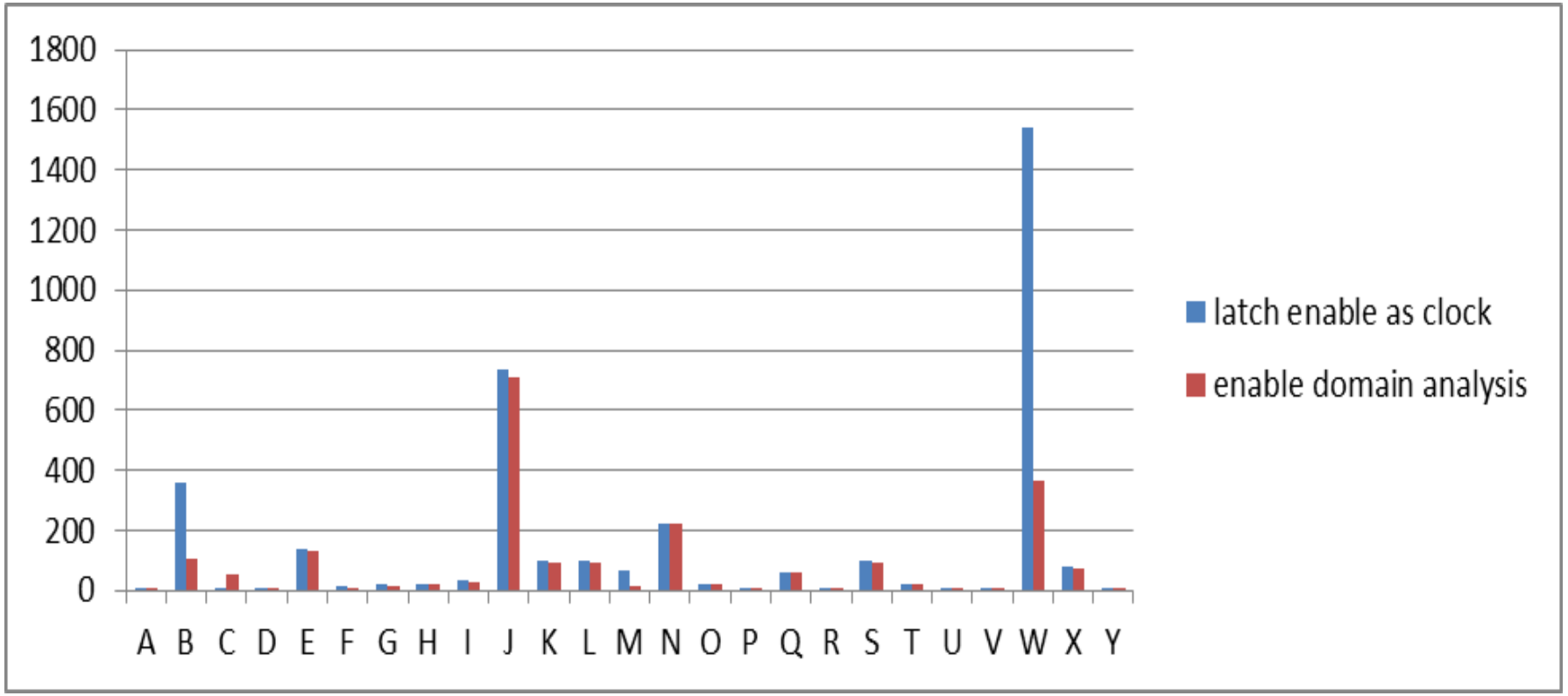
2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Results

Design	Total Latch Bits	Latch CDC Paths	Latch in single domain	Latch in mixed domain
1	455190	2460094	18492	436253
2	5382093	637841	4465438	514991
3	404494	92160	0	98304
4	311255	82741	1516	55549
5	367043	73450	76	10387
6	53792	34662	0	2401
7	52813	18523	133	10280
8	14873	14376	9	14596

Eight designs among 82 with most latch CDC paths in our evaluation

Number of clocks in two methods



Number of identified clocks when enables of latches are treated as clocks Vs. usign the latch domain based analysis in 25 designs

Reviewing Latch CDC Paths

- Check the clock inferencing for the enable signals
  - Is the clock domain of driving logic correct/assigned?
  - Should some constants be defined for proper clock domain inferencing?
  - Is this a design bug?
  - Is a synchronizer missing?
- For mixed domain latches
  - Most likely the different clocks drivign the enable need to be grouped together
  - There could be a design error of a wrong domain signal driving the enable
  - Some signals driving the enable may be stable even though it comes from a different clock domain

2018 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Summary

- Latches are key components in high speed logic
  - Latches are also prone to metastability
- Handling latches in CDC is tricky due to transparent phase
- Treating latch enables as control signal reduces unnecessary clock tree segments and CDC path counts
  - Yet does not miss CDC paths