Side-Channel Analysis For Detecting Hard-IP Trojans

Side-channels refer to access and measurement techniques that **bypass** the designer-intended input-output mechanisms, e.g., the digital I/O pins of an IC

Side-channels, as the name implies, refer to auxiliary electrical and/or electromagnetic (EM) access mechanisms, such as

- V_{DD} and GND (power supply) pins
- Top-layer metal connections in the physical layout of the IC

Side-channel attacks utilize these auxiliary electrical paths to

- Create a fault while the IC is operational as a means of learning internal secrets
- Measure signals, in an attempt to steal internal secrets, e.g., encryption keys

Side-channels can also be leveraged by the **trusted authority** to obtain information regarding the integrity of the IC

For example, leakage current (I_{DDQ}) and transient current (I_{DDT}) measurements have been widely used to detect manufacturing defects

Side-Channel Analysis For Detecting Hard-IP Trojans

On-chip **design-for-testability** (DFT) and other types of specialized instruments can be utilized that allow access to *embedded* side-channels, e.g., path delays

DFT components are designed to improve visibility of the internal and localized behavior of the IC, and include mechanisms to measure

- Internal logic states
- Path delays
- Localized quiescent and transients currents
- Localized temperature profiles

Care must be taken however b/c DFT added by the trusted authority can also be leveraged by adversaries as 'backdoor' access mechanisms to internal secrets, e.g., keys

Therefore, security features such as fuses must be included to disable DFT after manufacturing and testing

Side-Channel Analysis For Detecting Hard-IP Trojans

Side-channel signals are typically ***analog*** in nature, and can provide detailed, high resolution information about internal timing and regional signal behavior of the IC

For example, I_{DDT} measurements reflect performance characteristics of individual gates

This type of *temporal* information can be reverse-engineered and compared with simulation-generated data to validate the IC's structural characteristics

We refer to comparisons of this type as ***golden model-based*** analysis

Path delays, if measured at high resolutions, can also provide structural information about the chip

Unlike I_{DDx} measurements which provide a large-area regional observation, path delays are influenced by only components on the ***sensitized*** path

(defined as a path that propagates a logic signal transition)

Therefore, **path delay testing** can potentially provide a high resolution HT detection methodology

Side-Channel Analysis For Detecting Hard-IP Trojans

Unfortunately, path delays are also affected by variations which occur in fabrication processing conditions, commonly referred to as ***process variations***

Delay variations introduced by process variation effects are unavoidable and **must be** distinguished from delay variations introduced by an HT

Failing to distinguish is costly b/c:

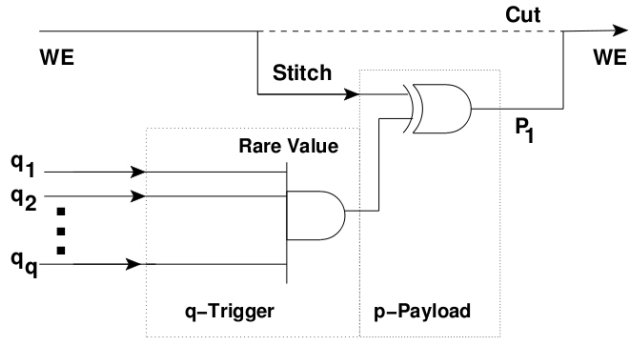
- Of the time and effort involved in verifying false alarms
- The damage caused by HT escapes, which leave fielded systems vulnerable to attacks

Side-Channel Analysis For Detecting Hard-IP Trojans

The underlying basis of parametric methods can be characterized by the Heisenberg principle or *observer effect*

Any attempt to measure or monitor a system **changes its behavior**

Parametric methods attempt to determine if an adversary has inserted an HT that is 'observing' the evolving state of the IC as a trigger mechanism



F. Wolff, C. Papachristou, S. Bhunia and R. S. Chakraborty, "Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme", DATE, 2008.

Trigger signals q_1 through q_q typically connect to nodes in the existing design and therefore add capacitive load to these signals, creating an *observer effect*

Note that both the trigger signals and payload add **delay** to paths in the design

Therefore, HT detection based on ***precise delay analysis*** may provide an effective solution

Technical Domains of Side-Channel-based Detection Methods

There are three fundamental **technical domains** that need to be considered by path delay-based methodologies

- The test vector generation strategy
- The technique employed for measuring path delays
- The statistical detection method for distinguishing between process variation effects and HT anomalies

Any commercially viable HT detection method **must** address ALL of these in a cost-effective manner

Many of the proposed methods only address a subset of these technical domains and therefore must be combined with other techniques to be fully operational in practice

Path Delay Test Methods

When technology scaling entered the deep submicron era circa 2000, higher frequency operation, within-die variations, etc. ushered in an emphasis on statistical modeling

This era also renewed interest in delay fault models, namely *transition fault*, *gate delay fault* and *path delay fault* models, which were introduced earlier

Driven by test cost issues, the VLSI test community developed *short-cuts* to allow the 2-vector sequences which define a delay fault test to be applied

The work-arounds became known as *launch-on-shift* (**LOS**) and *launch-on-capture* (**LOC**)

LOS and LOC allow 2-vector delay tests to be applied while minimizing the amount of additional on-chip logic needed to support this type of manufacturing test

Path Delay Test Methods

Unfortunately, LOS and LOC delay test mechanisms also create **constraints** on the form of the 2-vector sequences

- For example, they do not allow the 2 vectors that define a sequence to be independently specified

These constraints **reduce** the level of fault coverage that can be attained for delay defects

More elaborate design-for-testability (DFT) structures have been proposed that allow both vectors to be independently specified

- But are difficult to justify because of the negative impact they have on area and performance

These constraints continue to hold for modern day SoCs

The hope is that increasing awareness of hardware trust concerns may provide the impetus for a ***paradigm shift*** which would justify additional on chip support

- We will discuss several proposed on-chip solutions and corresponding benefits

Path Delay Test Methods

Path delay tests are defined as a 2-vector sequence $<V_1,V_2>$, with the initialization vector V_1 applied to the inputs of a circuit at time t_0

The circuit is allowed to stabilize under V_1

At time t_1, vector V_2 is applied and the outputs are *sampled* at time t_2

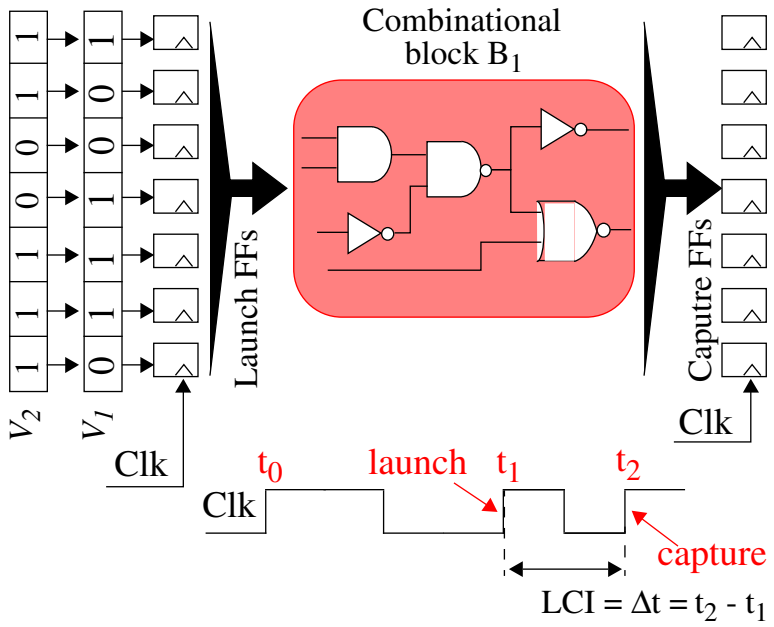
The *Clk* signal is used to drive both

- The **launch FFs**, which apply V_1 and V_2 to the combinational block inputs
- The **capture FFs** which sample the new functional values produced by V_2

The time interval (t_2 - t_1) is referred to as the ***launch-capture interval*** (LCI), and is typically set to the operational clock period for the chip

Path Delay Test Methods

Note that the ***standard form*** of path delay testing places no constraints on the values used for V_1 and V_2 as shown below



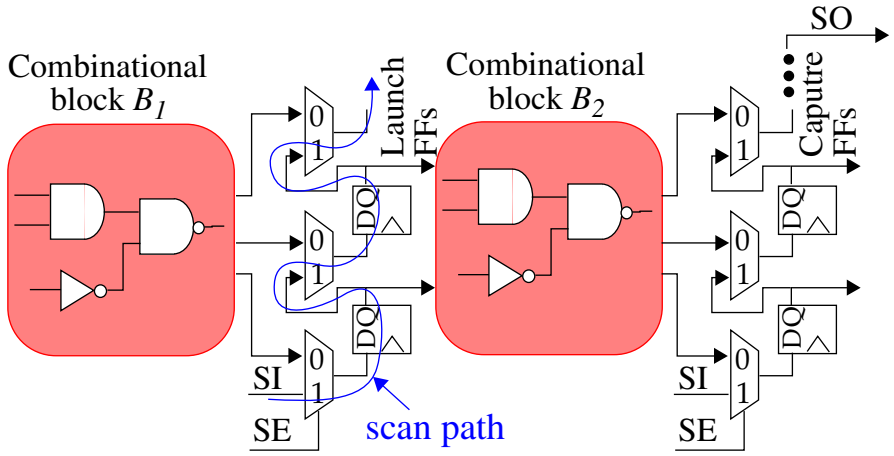
Unfortunately, external, off-chip access to the Launch and Capture FFs which connect to the combinational blocks within an IC is not possible

Therefore, complex *sequential testing methods* must be applied to obtain *visibility* to the internal states of the FFs

Path Delay Test Methods

A design-for-testability (DFT) feature called **scan** addresses this problem by enabling direct control and observability to embedded combinational blocks

Scan provides a **second**, **serial path** through all (or most) of the FFs in the IC



Launch-on-shift (LOS): V_2 defined as 1-bit shift of V_1
Launch-on-capture (LOC): V_2 defined as output of block B_1

The figure shows a typical circuit configuration with several cascaded combinational blocks B_1 and B_2, with interleaved FFs

The second path is commonly implemented by adding 2-to-1 MUXs

A *scan-enable* (SE) control signal is added as an I/O pin on the chip to allow test engineers to enable the serial path

Path Delay Test Methods

The scan architecture allows only a single vector V_1 to be applied

Manufacturing tests that **target defects** which prevent circuit nodes from switching (called stuck-at faults) can be applied directly using scan

Only a single vector is needed for these tests

Stuck-at fault testing is referred to as a DC test because no timing requirements exist, i.e., delays are irrelevant

The 2-vector requirement for delay testing can be solved in two ways, LOS and LOC

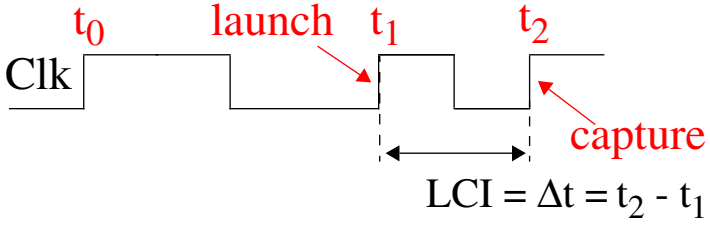
- Launch-on-shift (LOS) derives V_2 by shifting the scanned in vector V_1 by 1 bit position using the scan chain.
- Launch-on-capture (LOC) derives V_2 from the outputs of the previous combinational block, shown as B_1 in the previous figure for testing paths in B_2

In both cases, it is not possible to choose V_2 arbitrarily, as is often assumed in proposed HT methods

Path Delay Test Methods

Another issue that is often ignored deals with obtaining **accurate** timing information for paths

The timing diagram shown earlier suggests that it should be possible to set the **launch-capture interval (LCI)** to any arbitrary value



Unfortunately, this is not the case

The external tester (ATE) driving the clock pin on the chip is limited in how close consecutive edges of *Clk* can be placed

Moreover, most applications of delay tests for manufacturing defects only need to determine if the chip runs *at the operational clock frequency*

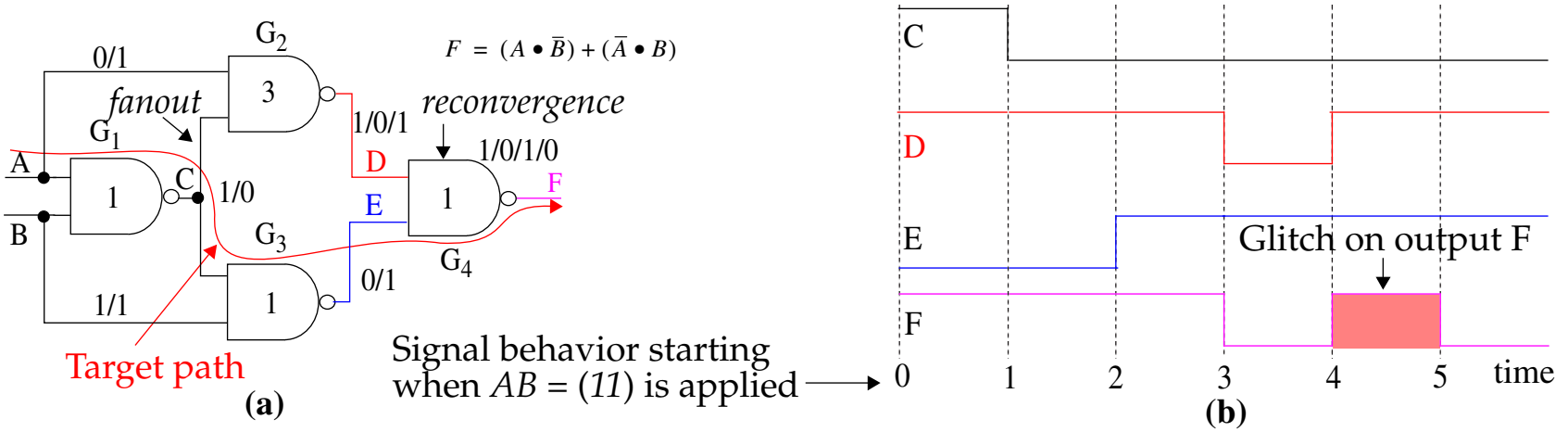
As a consequence, the LCI is typically **fixed** for all tests and **only upper bounds** on the delays of paths within the chip can be obtained

Path Delay Test Methods

Therefore, HT detection methods that require *picosecond* resolutions for individual path delays will require alternative clocking strategies and/or additional DFT

A last important issue regarding path delay testing is related to **circuit hazards**

Combinational logic blocks often possess instances of *reconvergent fanout*



The integers inside the NAND gates represent one possible assignment of gate delays

The test sequence $AB = \{01,11\}$ is designed to test the highlighted path but in fact propagates logic transitions along **both** branches of the *fanout* point *C*

The timing diagram shown on the right identifies a 'glitch' on the output *F* that is created by differences in the relative delays of these two paths

Path Delay Test Methods

This test is classified by the manufacturing test community as *robust*

However, the **glitch** introduces ***uncertainty*** for the security community in cases where the precise delay of the highlighted path is needed

The three transitions that occur on F each represent the delay of a *subpath* in the circuit, with the first, left-most edge in this case corresponding to the highlighted path

Subpath information might prove useful in providing additional HT coverage

Unfortunately, process variations render this information *challenging to leverage*

This is true because it is difficult to decide which edge corresponds to which subpath

In other words, the same test applied to a different chip with different assignments of delays to the NAND gates may **reorder** the edges

Or may in fact result in only single transition, i.e., the **glitch disappears** altogether

Path Delay Test Methods

All major synthesis tools **are oblivious to hazards**, making them very common in synthesized implementations of functional units

Special logic synthesis algorithms are needed to construct circuits that are *hazard-free*

But hazard-free implementations usually have **large area overheads** and therefore are rarely used

Unfortunately, hazards are *largely ignored* in many proposed HT test generation strategies even though they can invalidate tests and raise false alarms

Important Similarities/Distinctions of Delay Test for Manufacturing Defects and HT

Unlike logic-based testing, the goals of testing for defects and testing for HT using path delay tests are very similar

Path delay tests **for defects** are designed to determine if an imperfection causes a signal propagating along a path to emerge later than designed

Similarly, path delay tests for HT are designed to determine if an adversary has added fanout to logic gate inputs and outputs

As discussed, HT circuitry monitors the state of the IC (**trigger**) and modifies its function (**payload**) using series inserted gates

Both of these scenarios also cause the delay of paths to increase

An important distinguishing characteristic between defects and HT relates to *false positives*

False positives are situations in which a test for an HT indicates **it is present when in fact it is not**

Important Similarities/Distinctions of Delay Test for Manufacturing Defects and HT

This issue is less important for defects, and can be minimized using modern automatic test pattern generation (ATPG) tool flows

False positives can occur for HT when the detection method does not adequately account for normal delay variations introduced by process variations

Unfortunately, the cost associated with false positive detection decisions is very different for defects and HT

- A false positive in manufacturing test results in a defect-free chip being falsely discarded
- A false positive HT detection can initiate a *very expensive and time consuming* reverse engineering process of the IC

False negatives, on the other hand, need to be handled by both manufacturing defect and HT testing communities

False negatives are situations in which a **defect or HT exists and it is not detected by the applied tests**

Important Similarities/Distinctions of Delay Test for Manufacturing Defects and HT

False negatives can occur in either application either because

- The measurement technique does not provide sufficient resolution
- The applied tests do not provide adequate coverage

The cost associated with false negatives **can be high** in either case, resulting in system failure once the IC is installed in a customer application

Tech Area #1: High Resolution Path Delay Measurement Techniques

On-chip clock generation for digital ICs can be accomplished using:

- Delay-locked loop (DLL)
- Phase-locked loops (PLLs)
- Digital clock managers (DCM)

These clock generation modules typically use the a reference clock generated by an off-chip temperature-stable oscillator

They are responsible for

- Maintaining *phase alignment* with the off-chip oscillators
- Creating **multiple internal clocks** at different frequencies and with specific phase shifts

Clocks can also be generated directly by **automatic test equipment** (ATE) for path delay testing

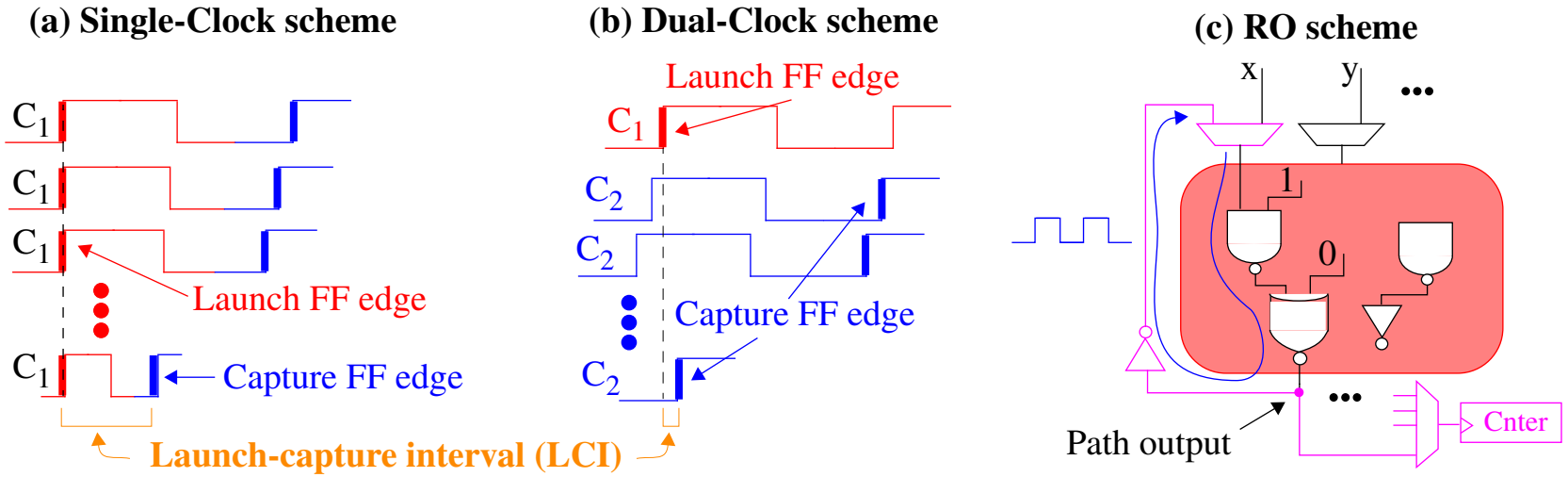
However, on-chip clock and phase shift mechanisms generally provide higher accuracy and resolution

This is true b/c off-chip parasitic components are eliminated

Tech Area #1: High Resolution Path Delay Measurement Techniques

Many HT detection techniques depend on high resolution timing measurements, making on-chip techniques better suited

Examples of on-chip measurement techniques



The first, called *Single-Clock scheme* (or ***clock sweeping***), requires repeated application of a 2-vector sequence

On each iteration, the *frequency* of C_1 is increased, which moves the launch and capture edges, i.e., the launch capture interval or **LCI**, closer together

Tech Area #1: High Resolution Path Delay Measurement Techniques

The process is halted as soon as a condition is met or violated

The condition is usually related to whether the *Capture FF successfully captures* the functional value produced by vector V_2

An estimate of the path delay is computed as $1/frequency_{final}$ where $frequency_{final}$ is the **stop point** frequency

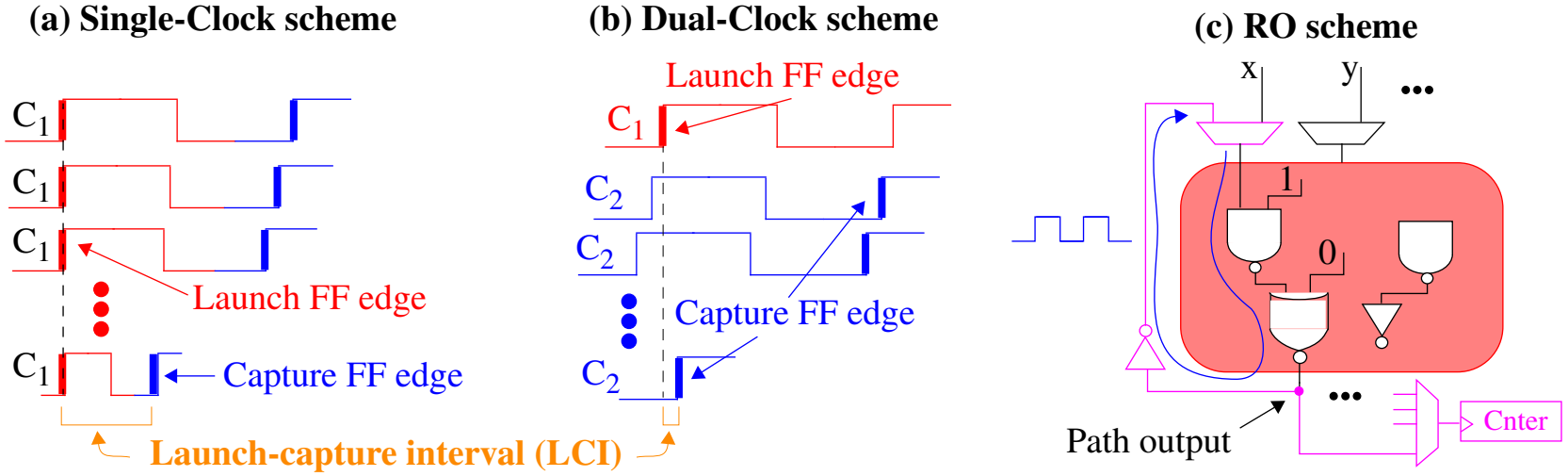
Although this scheme requires the fewest resources, i.e., **only one clock tree** is included on the chip, it *lower bounds* the length of the path that can be measured

For example, short paths would require a very high frequency clock, which creates *undesirable secondary effects*, e.g., power supply noise

Single-Clock schemes which use an externally-generated (ATE) clock constrain the minimum path length even further

Tech Area #1: High Resolution Path Delay Measurement Techniques

The second, called *Dual-Clock* scheme (or ***clock strobing***), also requires repeated application of the 2-vector sequence



On each iteration, the *phase* of the capture clock C_2 is decremented by a small Δt relative to C_1

The additional overhead introduced by the second clock tree is offset by the benefit of being able to **time a path of any length**

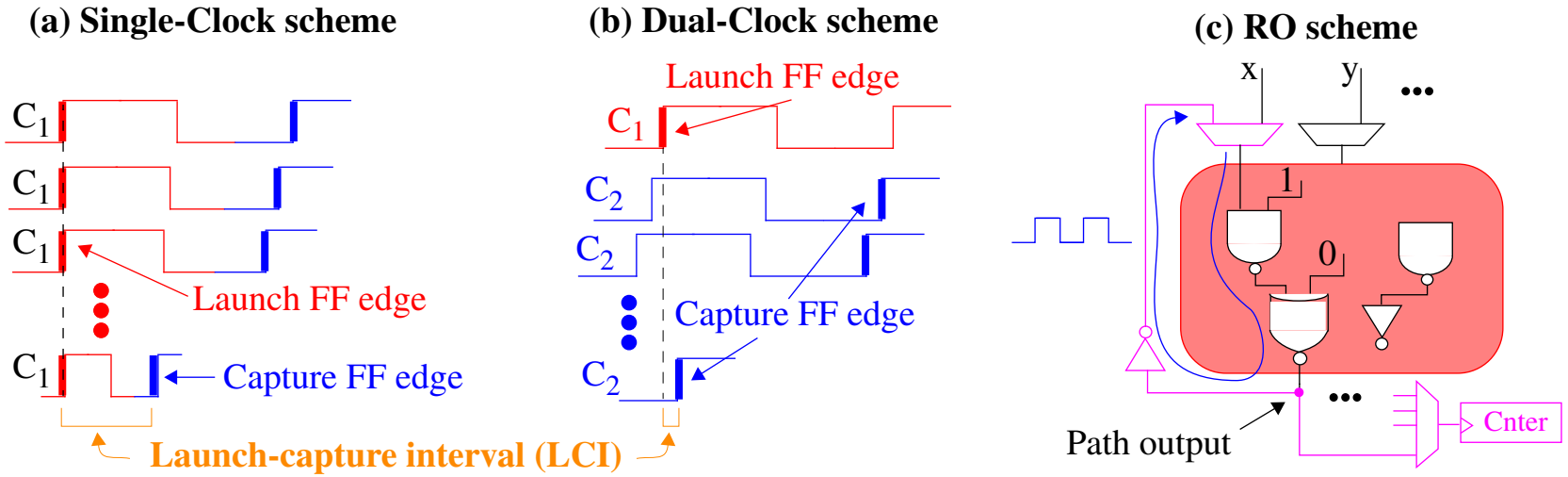
This is possible because the two clock networks are independent and modern DCMs are able to shift C_2 very precisely

Tech Area #1: High Resolution Path Delay Measurement Techniques

Note that power supply noise issues mentioned above are also mitigated

Only **two clock edges** are required to carry out the test instead of three

The third timing mechanism, referred to as the *RO* scheme



It adds the components shown in magenta to the design

Paths in the circuit are timed by creating a **ring oscillator** (RO) configuration where the output of a path is connected back to the input of the path using a MUX

Tech Area #1: High Resolution Path Delay Measurement Techniques

A timing measurement is performed by enabling the MUX connection and then allowing the path to 'ring' for a specific time interval

A counter (*Cnter*) is used to record the number of oscillations

This is accomplished by tieing the output signal from the path to the *clock input of the counter*

The actual path delay is obtained by dividing the time interval by the counter value

No launch-capture event is required in this scheme

Therefore the *clock noise* associated with high frequency clocks in the *Single-Clock* scheme are eliminated

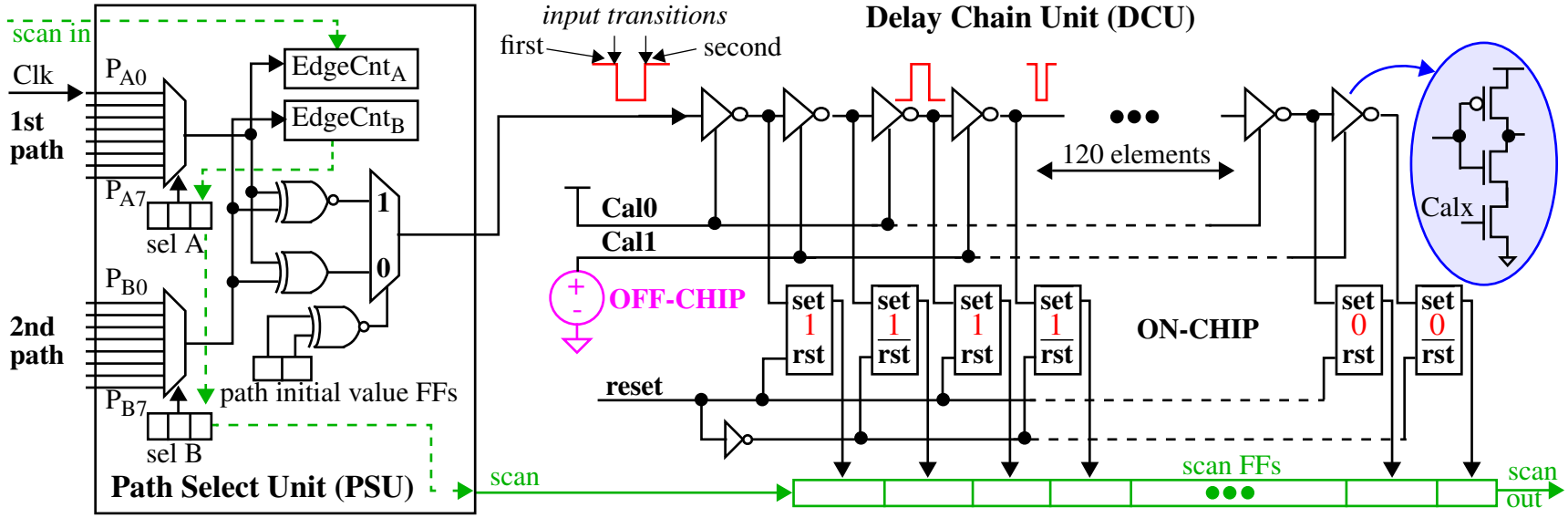
The **main drawback** is related to the limited number of paths that can be timed in this fashion

For example, paths that have **hazards** produce artifacts in the count values

As discussed, hazards are very common in combinational logic circuits

Tech Area #1: High Resolution Path Delay Measurement Techniques

A fourth alternative, called a **time-to-digital converter** (TDC)



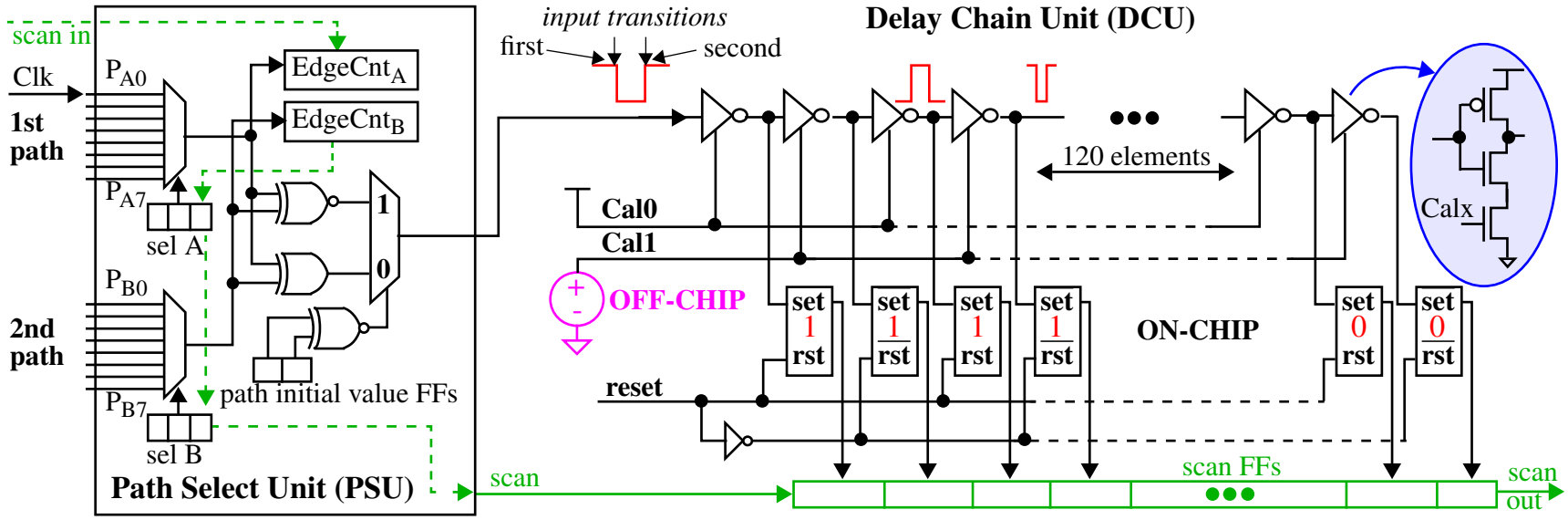
Similar to the *RO* scheme, it eliminates clock strobing, and therefore, is able to obtain path delay measurements that better represent *mission mode* path delays

The TDC is an example of a **flash converter**

A class of converters that digitize path delays very quickly

The *Path Select Unit* shown on the left is responsible for selecting a pair of paths, one of which can be the clock signal

Tech Area #1: High Resolution Path Delay Measurement Techniques



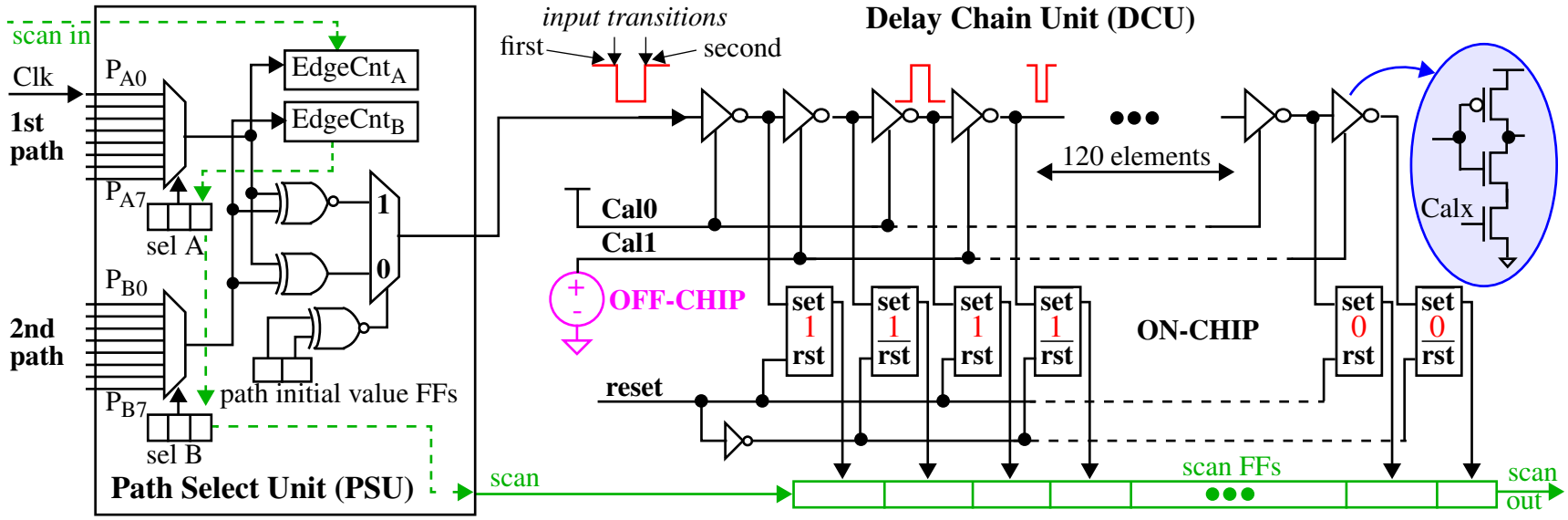
The *Delay Chain Unit* is responsible for creating a digital representation of the relative difference between the delays of the two input paths, P_{Ax} and P_{Bx}

The arrival of a rising or falling transition on one path creates the first edge in the delay chain (labeled *first* in the figure)

While a transition on the second path generates the trailing edge (labeled *second*)

The width of the initial pulse represents the *delay difference* between the two signals

Tech Area #1: High Resolution Path Delay Measurement Techniques



The output of the inverters in the delay chain also each connect to a ‘set-reset’ latch

The presence of a negative pulse (for odd inverters) or positive pulse (for even inverters) changes the latch value from 0-to-1

A digital ***thermometer code*** (TC), i.e., a sequences of 1’s followed by zero or more 0’s is produced in the sequence of latches after a test completes

Calibration can be used to convert the TC to a delay value

Tech Area #1: High Resolution Path Delay Measurement Techniques

A fifth scheme, called REBEL also uses a **delay chain** to obtain timing information

REBEL is a *light-weight embedded test structure* that combines:

- The delay chain component of the TDC (without the pulse shrinking characteristic)
- The clock strobing technique

A significant benefit of REBEL over the TDC is **complete resilience to hazards**

In fact, REBEL is able to provide timing information regarding *each of the edges* associated with hazards in a single launch-capture test

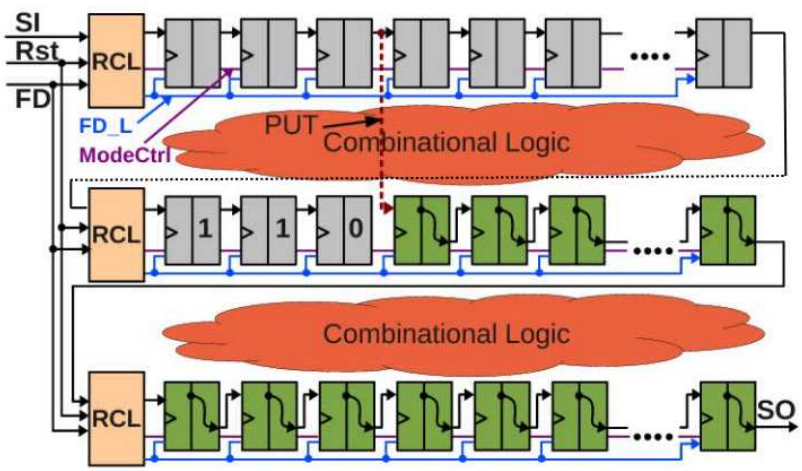
Although process variations add uncertainty and diminish their usefulness, the ability to instantly have knowledge of their presence adds robustness

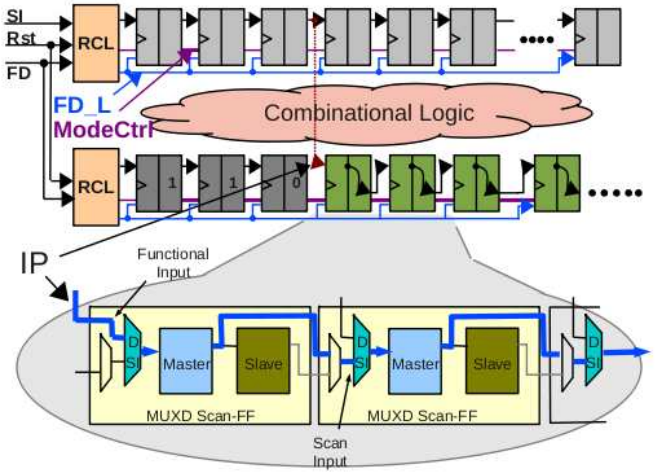
And helps reduce the likelihood of false negative HT detection decisions

REBEL ***leverages the scan chain architecture*** that is already in place to create a delay chain

Tech Area #1: High Resolution Path Delay Measurement Techniques

REBEL creates a delay chain from the existing FFs by creating a *tap-point* between the master-slave components, allowing all the **master latches** to be chained together





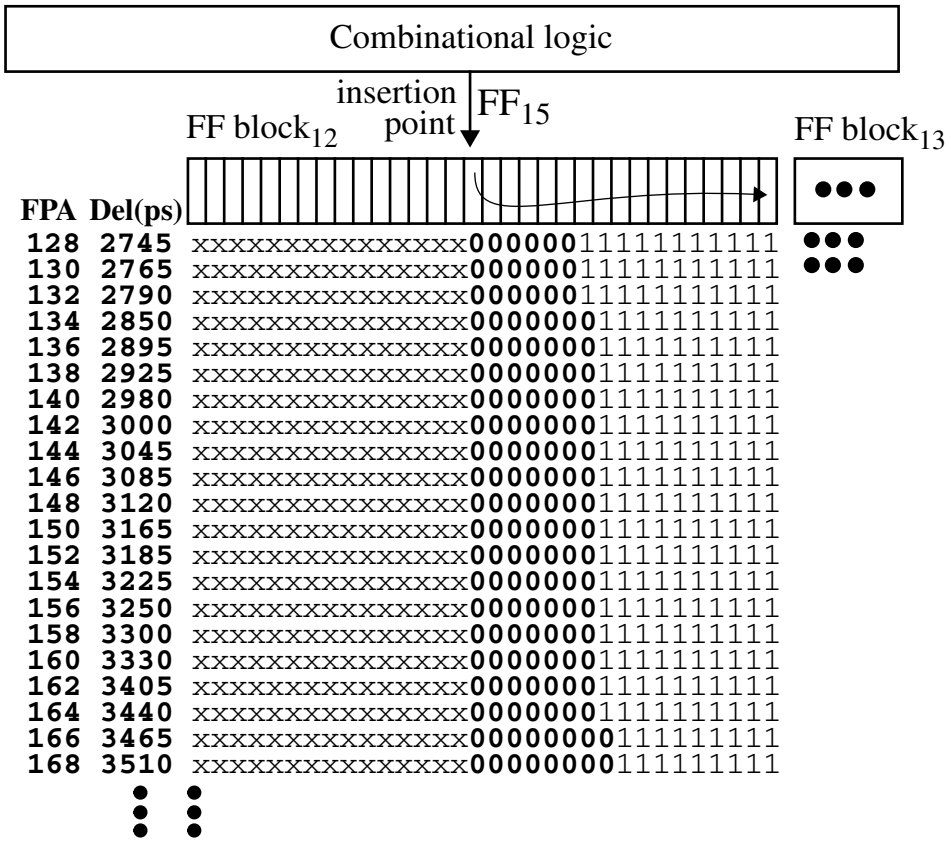
Signal propagate through the combinational logic

One output is designated as the ***insertion point***, which is the signal that is allowed to propagate along the delay chain

Configuration information is 'scanned into' the existing FFs and additional scan chain logic is enabled to create the delay chain during the launch-capture test

Tech Area #1: High Resolution Path Delay Measurement Techniques

The **digital snapshot** result of a launch-capture test



Tech. Area #2: Dealing with Process Variations

A significant benefit of techniques designed to detect HT in fabricated chips is the availability of a ***golden model***

Which is not available for Soft IP Trojans

The golden model assumes all design data prior to mask and chip fabrication steps, e.g., HDL, schematic, GDS-II, is considered trusted

A golden model, and simulation data derived from it, provides a trusted reference to which hardware data can be compared

Path delay methods attempt to **identify anomalies** in the hardware data that cannot be explained by the golden model

Distinguishing between changes in delay introduced by a HT and those introduced by process variation effects is a significant challenge

Failing to do so leads to **false positive** and **false negative** HT detection decisions

Tech. Area #2: Three Approaches for Dealing with Process Variations

- **GoldenSim-based** and **GoldenChip-based**

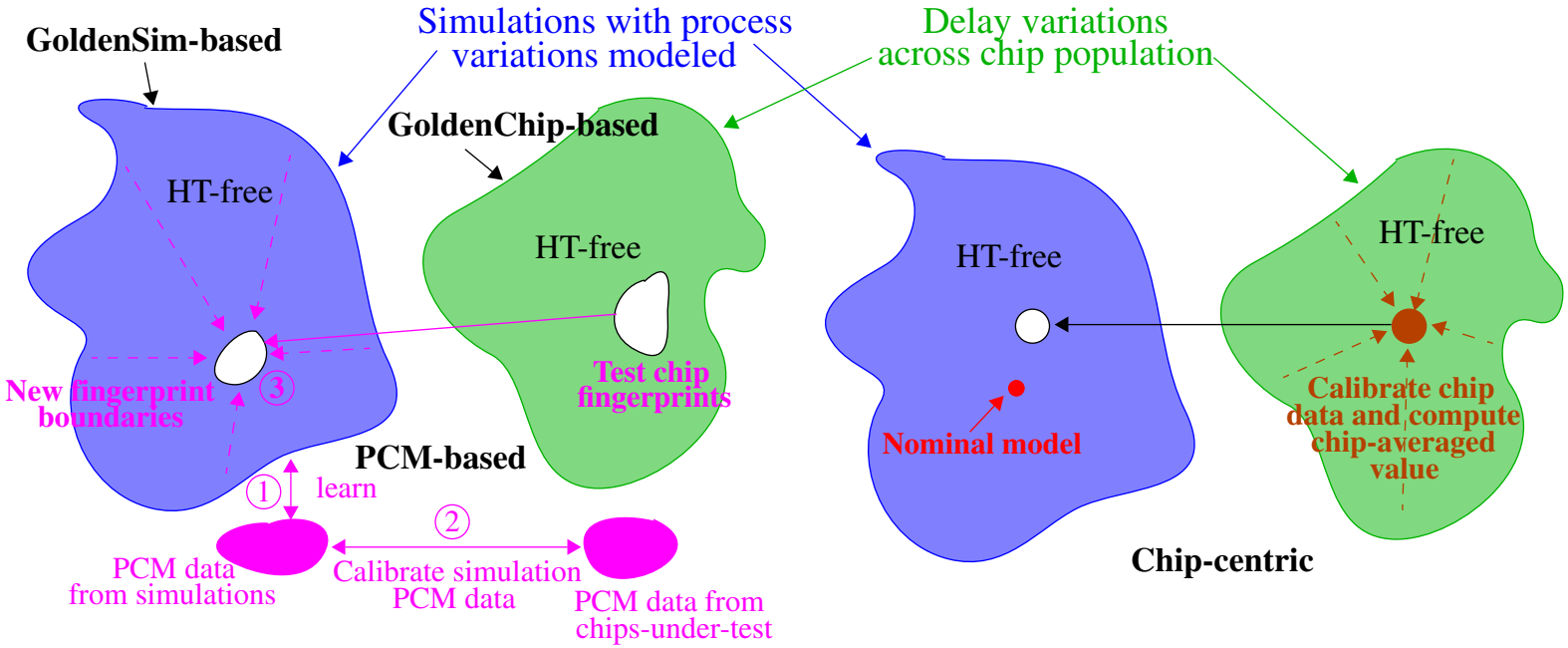
  Creates simulation models or uses HT-free chips, to bound the HT-free space

- **PCM-based**

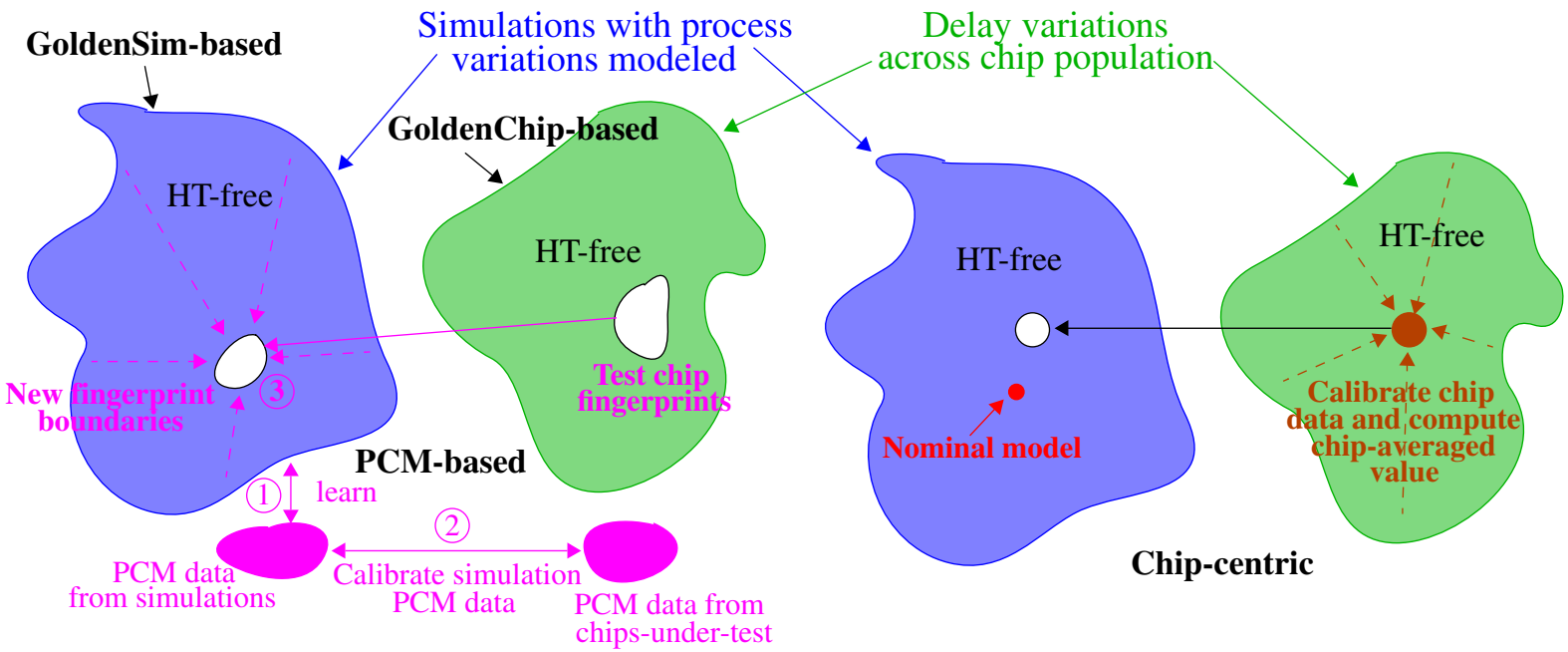
  Uses data from *process control monitors* (PCM) to 'tune' the boundaries of HT-free space derived from golden models using chip-measured test structure data

- **Chip-Centric**

  Creates a *nominal* simulation model and *calibrates and averages* path delays to the nominal model (or data from HT-free chips)



Tech. Area #2: Three Approaches for Dealing with Process Variations



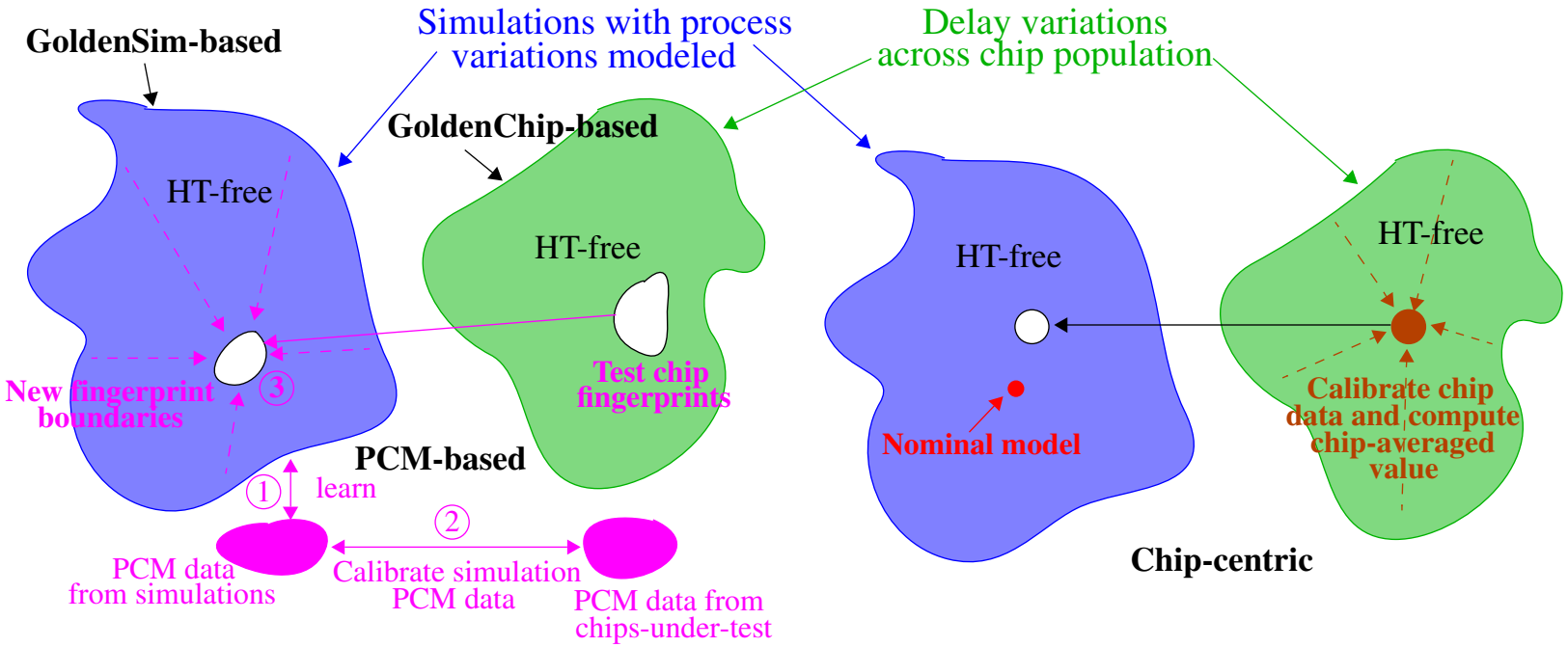
All approaches create a *bounded HT-free space* that represents normal variations in path delays introduced by process variations and/or measurement noise

Data collected from the test chips is compared with this *bounded HT-free space*

Data points that fall outside the boundaries are called **outliers**

Chips that produce outlier data points are *considered HT candidates*

Tech. Area #2: Dealing with Process Variations



The 2-D shapes labeled *Simulations with process variations modeled* and *Delay variations across chip population* can in fact be multi-dimensional

Here, each dimension representing one path delay or one of **multiple features** extracted from the set of path delays using statistical techniques, e.g., PCA

GoldenChip-based and **GoldenSim-based** techniques typically train a classifier using HT-free data from chips or simulations, resp.

Tech. Area #2: Dealing with Process Variations

Both techniques can be expensive in terms of reverse-engineering effort, model development and simulation time

- **GoldenChip-based** methods measure delays from HT-free chips, which are then destructively validated to be HT-free using techniques discussed earlier
- **GoldenSim-based** methods typically use data from Spice-level simulations of a resistor-capacitor-transistor (RC-transistor) model of the *golden* design

For **GoldenChip-based**, delayering technologies utilized for GoldenChip-based methods can take weeks or months

For **GoldenSim-based**, CAD tools effort is non-trivial

Mentor Graphics Calibre must first be used to create the RC-transistor models of the layout using complex process models obtained from the foundry

The modeling files can be very large, e.g., 100's of MB, even for relatively small designs on order of 20,000 gates

Transient simulation times can easily extend to weeks and months

Tech. Area #2: Dealing with Process Variations

Of even greater concern for **GoldenSim-based** techniques is the level of mismatch that can exist between the simulation results and the hardware

Foundry models in advanced technologies have become very complex, providing the user with a variety of statistical evaluation methodologies

For example, *Fixed corners* and *Monte Carlo*

Fixed corner models are provided to enable the user to predict worst-case and best-case performance of the chip by modeling the range of global process shifts

Unfortunately, this typically *expands* the HT-free space beyond what is required to represent the behavior of the chips-under-test

The expansion leads to a **decrease in the sensitivity** of HT methods and increases the level of mismatch between simulation and hardware data

Moreover, foundry models typically provide limited capabilities for modeling **within-die variation effects**, making it difficult to predict delay uncertainties

Tech. Area #2: Dealing with Process Variations

These modeling and simulation challenges are compounded by

- **Measurement noise** that occurs during chip testing
- **Non-zero jitter and drift tolerances** introduced by the tester during the generation and delivery of high frequency clocks

Taken together, these issues work to increase in the possibility of false positive and false negative HT detection decisions

Tech. Area #3: Test Vector Generation Strategies

Earlier, we discussed several challenges regarding test vector generation, including LOS/LOC limitations and circuit hazards

A last issue deals with an important distinction that exists between fault models used in manufacturing test and those required for detecting HT

The manufacturing test community developed several fault models, e.g., **transition delay faults** and **path delay faults**, to handle a wide variety of defect mechanisms

For example, the transition delay fault (**TDF**) model assumes defects occur on individual *nodes* in the circuit, and cause slow-to-rise and slow-to-fall behavior

The path delay fault (**PDF**) model, on the other hand, makes no such assumptions,

It accounts for defects which may in fact be distributed across one or more logic gates and wires that define the paths

Therefore, the PDF model provides more complete information about the integrity of the tested chip

Tech. Area #3: Test Vector Generation Strategies

Unfortunately, obtaining 100% PDF coverage requires all (or a large fractions) of the paths in a chip to be tested

For even moderately sized circuits, the costs associated with the generation and application of a complete PDF test set **is prohibitive**

The number of paths can be exponentially related to the number of inputs

Therefore, most chip companies generate and apply **TDF** vectors instead because the number of such tests **is linear** to the number of circuit nodes in the design

Fortunately, for the security and trust community, the **TDF model is a better match** to the types of malicious modifications an adversary is likely to make to the layout

Tech. Area #3: Test Vector Generation Strategies

There are **two** important points to consider with regard to test generation for HT detection

Although far fewer tests are required under the TDF model to obtain high levels of HT coverage, there are typically **many choices** for the *path* that tests each node

A variety of techniques are proposed by authors of published work including:

- Random vectors
- An ***incremental-coverage*** strategy
- Traditional TDF vectors

Others leverage the TDF model and direct ATPG to target the **shortest paths** through the node

The assumption here is that the additional delay added by the HT has a **larger fractional impact** on the path delay

The traditional TDF model for defects, on the other hand, typically target the **longest paths**

Tech. Area #3: Test Vector Generation Strategies

The *length of the path* relates to the second important point regarding test generation

Automatic test equipment is outfitted for manufacturing test, which is focused on testing the longest paths

For test cost reasons, it is common that **only one clock frequency** is used to apply TDF tests to the chip

This is true b/c the primary goal of manufacturing test is to ensure that the delays of all tested paths are less than the **upper bound**

The *most sensitive tests* for defects therefore are those that test **the longest paths**

This is true because the longest paths minimize the **slack**, i.e., the difference between the clock period and the delay of the tested path

Tech. Area #3: Test Vector Generation Strategies

Many believe that these manufacturing test constraints for defects **are not sufficient** for providing high levels of HT coverage

This is reflected in the proposed use of *clock sweeping*, *clock strobing* and other on-chip embedded test structures for obtaining precise delays

In other words, the slacks inherent in tests for defects provide *too many opportunities* for adversaries to 'hide' the additional delay of the HT in the slack

Therefore, a *paradigm shift* is required regarding the manner in which delay testing is carried out on the test floor

Clock sweeping and **clock strobing** are expensive in terms of test time

And HT methods which use these clocking strategies need to account for:

- The *higher levels of clock noise* associated with high frequency clocks
- Invalidations introduced by circuit hazards

It remains to be seen how the economic tradeoffs of delay-based HT detection schemes will play out