Imperial College London

Timing and fault instrumentation with shadow registers

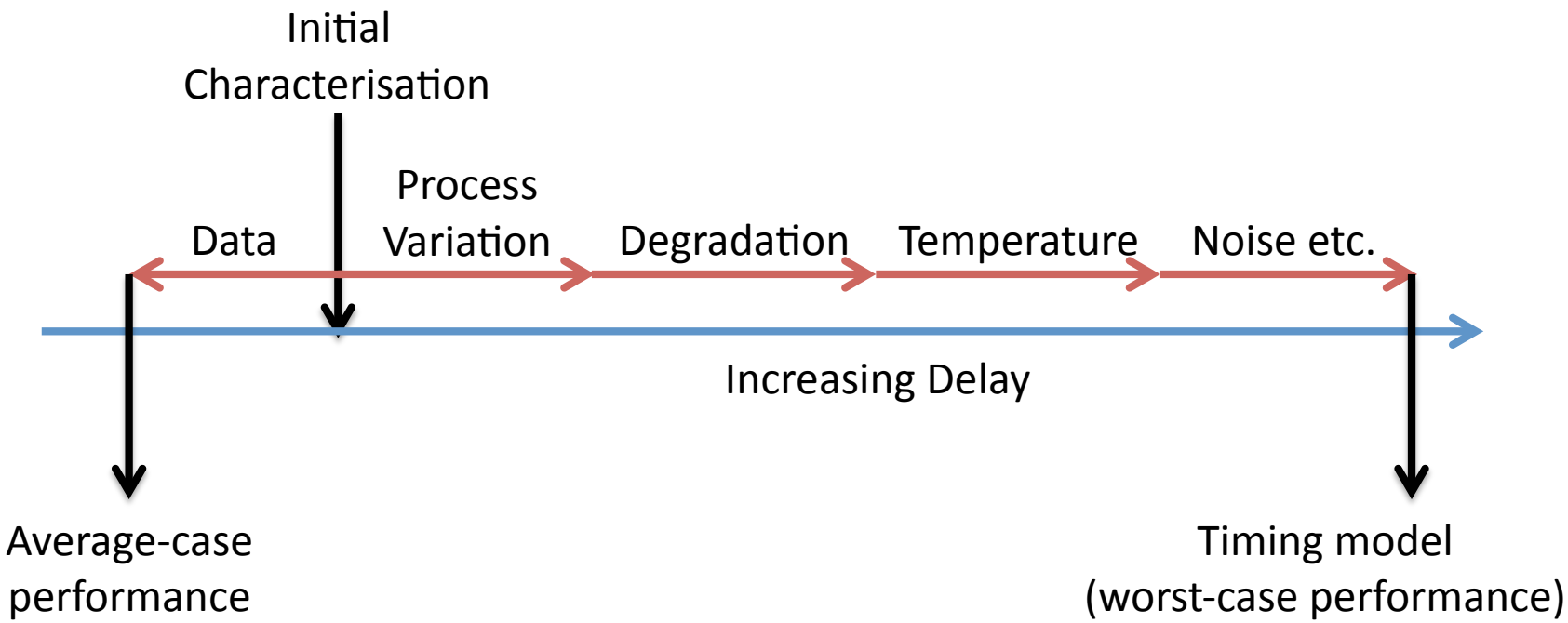
Edward Stott, Josh Levine, Peter Y.K. Cheung

Why circuit instrumentation?

- Timing models are conservative
- Uncertainty means every device is operated at worst-case performance
- But true performance can vary:
  - between devices
  - spatially across a single device
  - over time
- We measure performance characteristics at run-time to adapt to uncertainty
- We have developed an automated, low overhead flow for instrumenting FPGA designs

Imperial College London

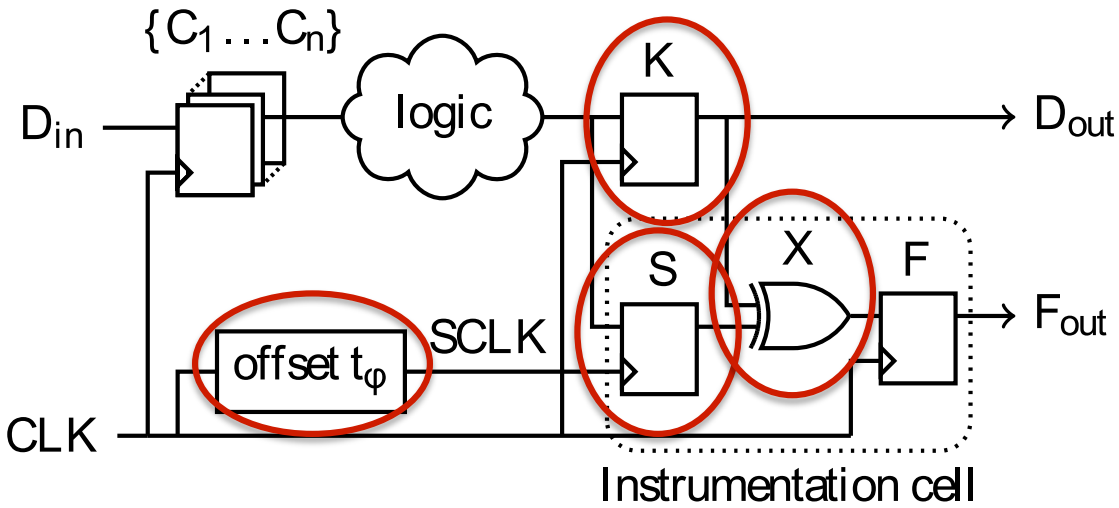
Types of operating margin



Imperial College London

Instrumentation cell

- Attached to certain registers in the application circuit
- Based around a shadow register
- Driven by variable phase clock

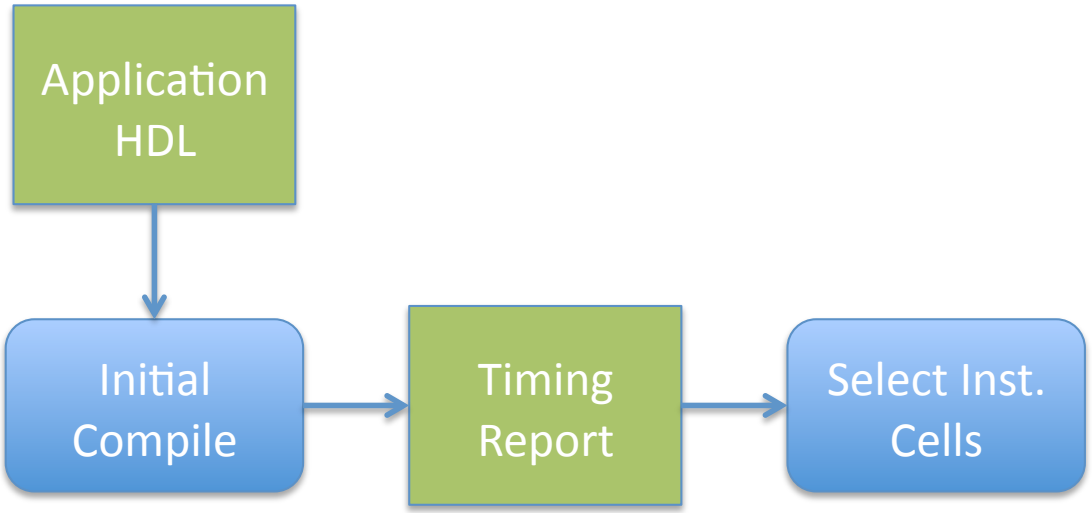


Instrumentation modes

- Timing slack measurement
  - Sweep shadow clock phase offset
  - Look for maximum phase lead with no discrepancies
  - No faults (or metastability) introduced into application circuit
- Timing fault detection
  - Like Razor
  - Fixed shadow clock phase lag
  - Shadow clock always samples correct data
  - Discrepancies indicate timing faults in application circuit

Imperial College London

Design Flow – Register Selection

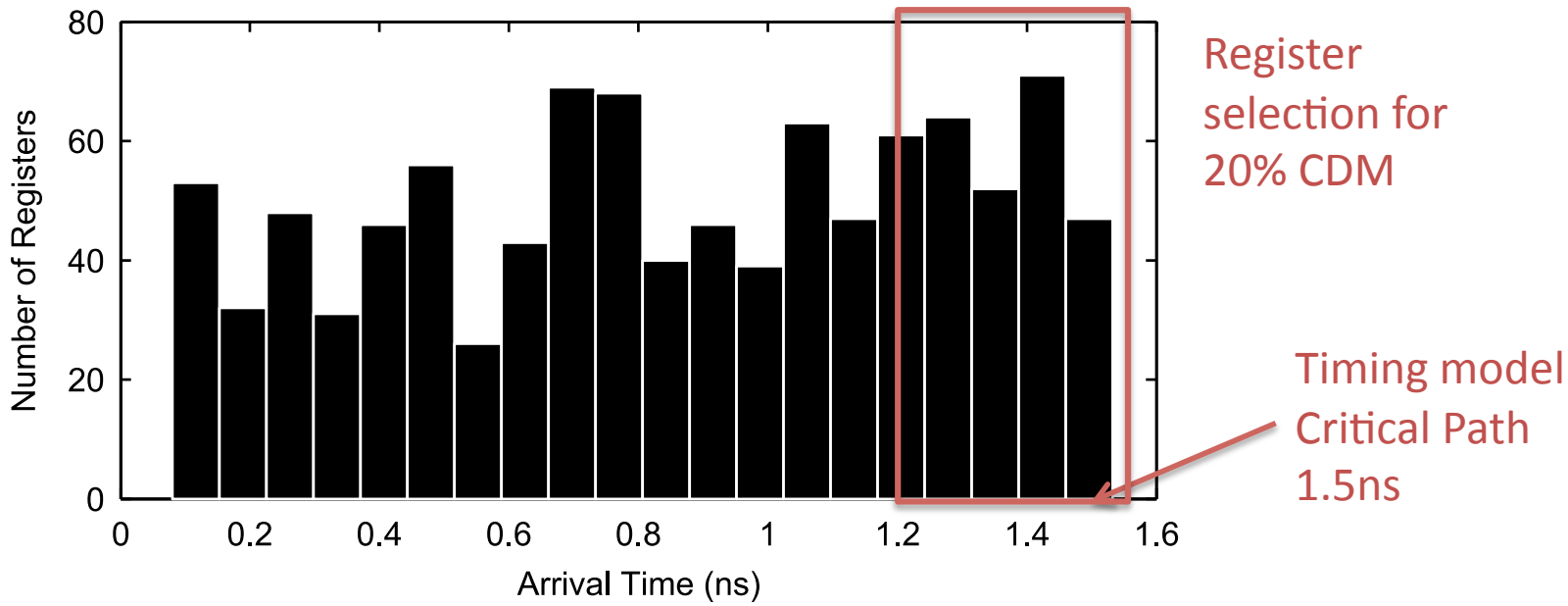


Imperial College London

Instrumentation cell selection

- Typical usage scenario: monitor critical registers in a design
- Define a critical register as

$$t_{arr} > (1 - C)t_{crit}, \text{ C = } \textit{Critical Delay Margin}$$



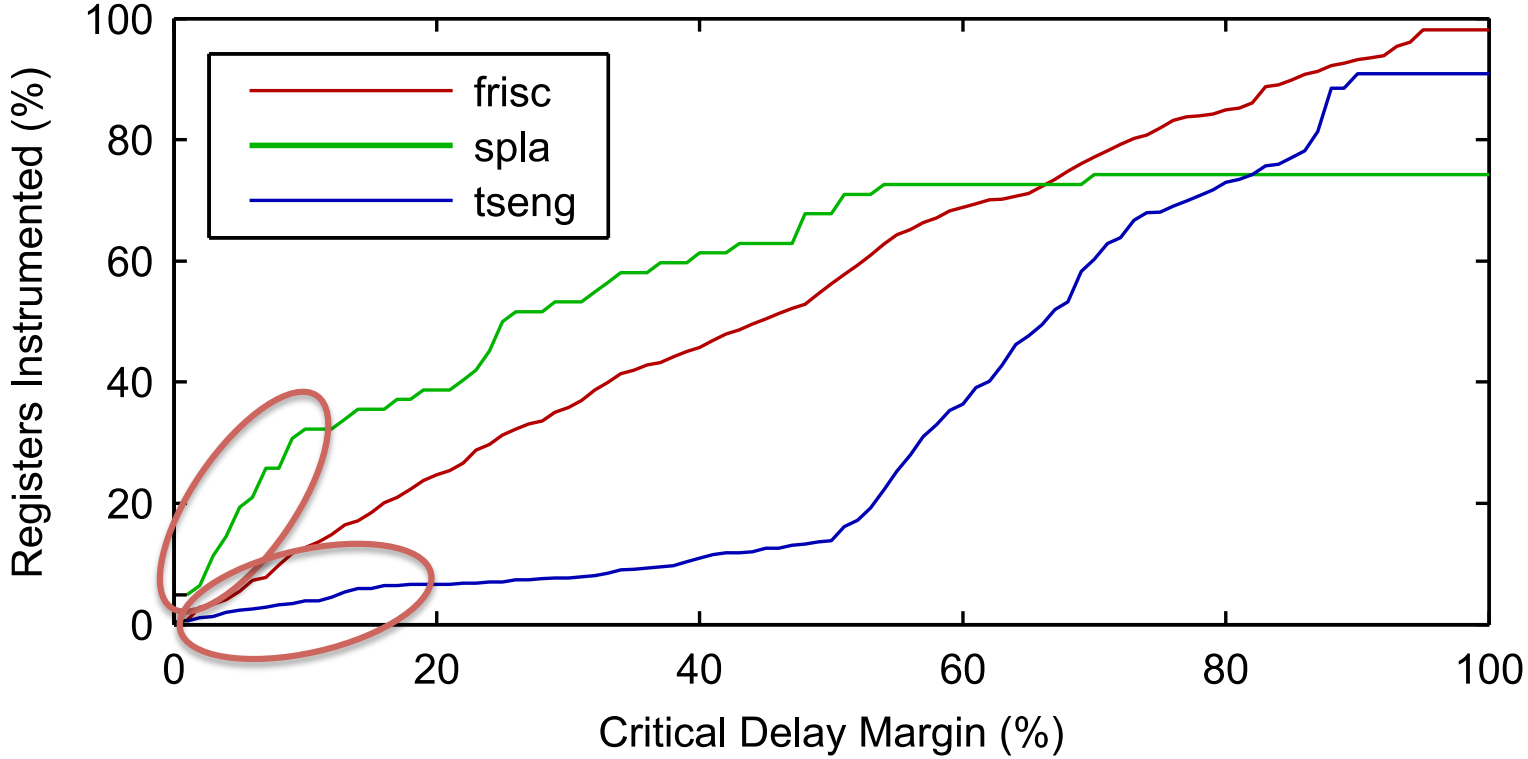
How Much Critical Delay Margin?

- Must capture true critical path to measure timing slack
- Cover sources of variability
  - Process variation
  - Temperature and voltage differentials
  - Noise
  - Uneven degradation
- For timing fault detection – must monitor any potential failure
  - Desired overclocking factor

Imperial College London

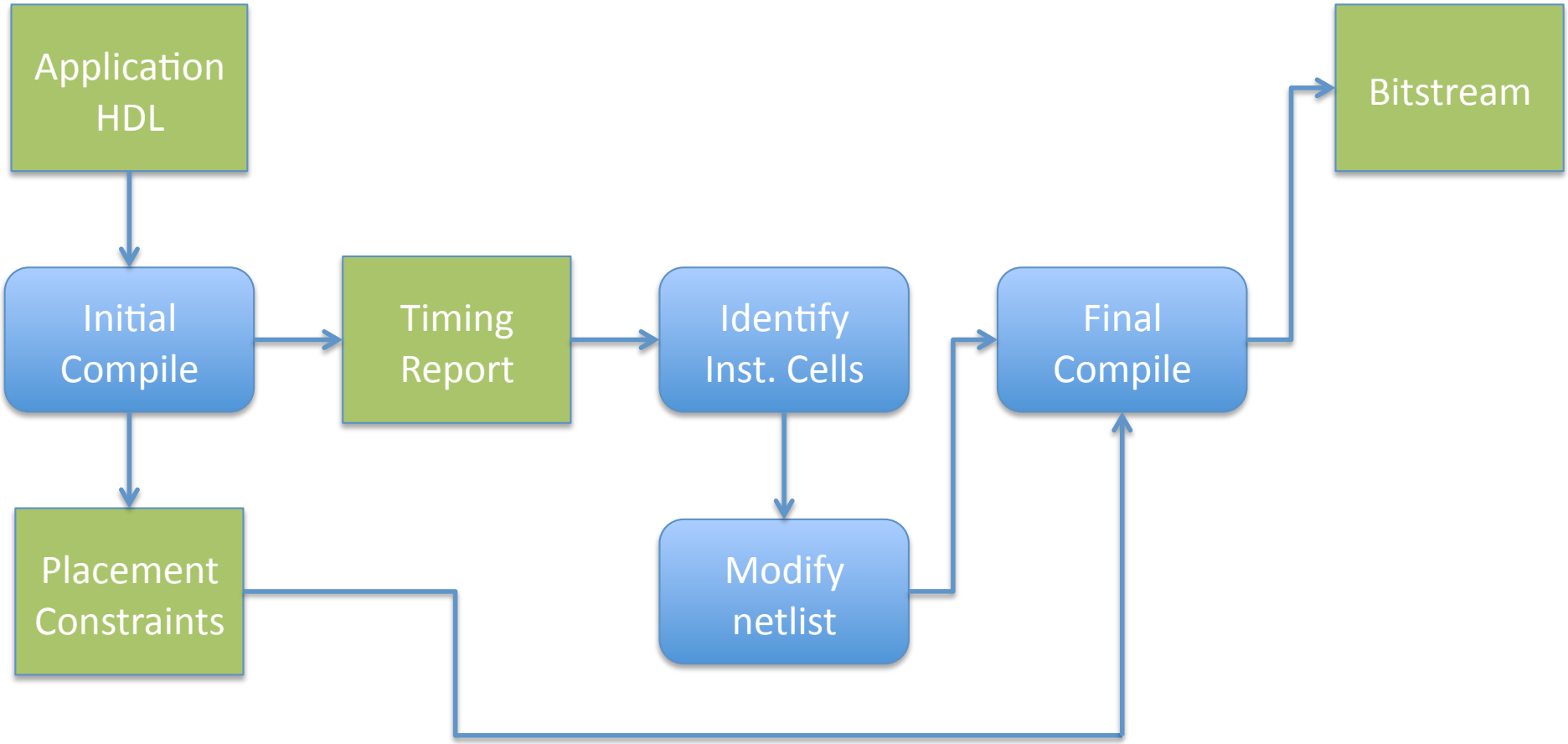
Instrumentation Overhead

- Overhead depends on timing distribution



Imperial College London

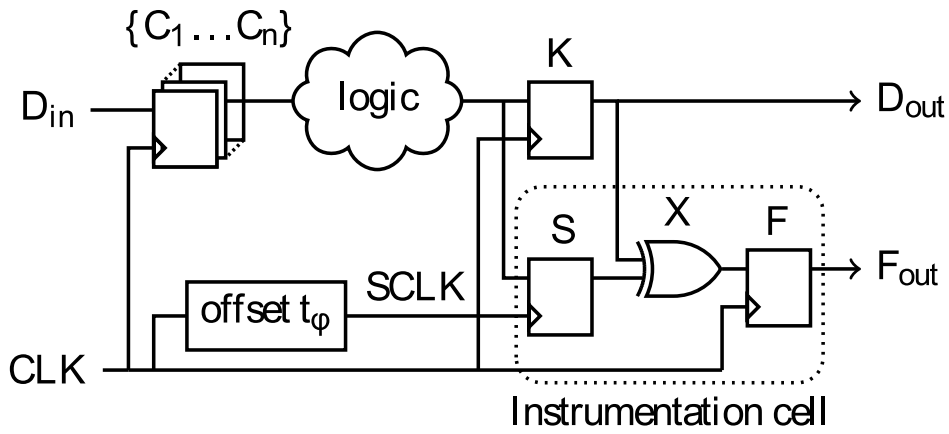
Design Flow – Design Modification

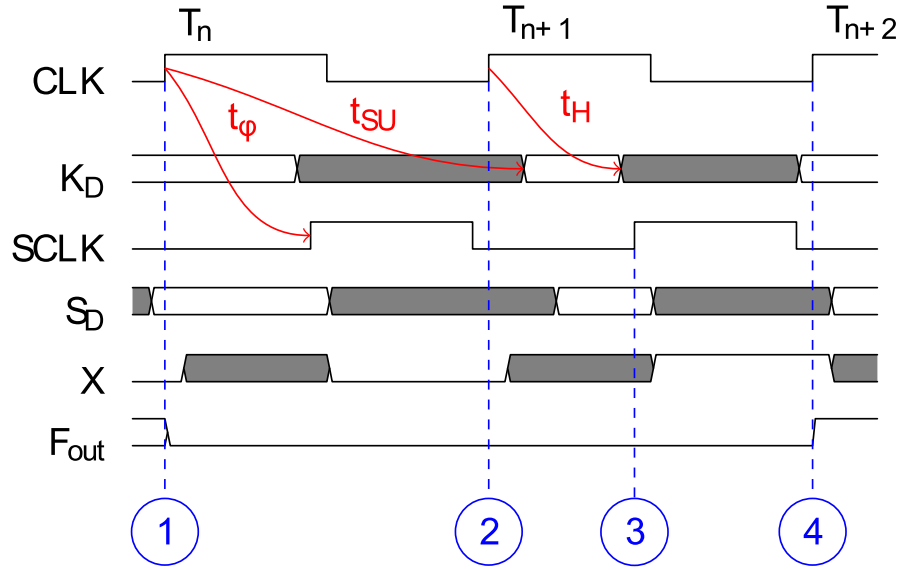


Imperial College London

Timing for fault detection

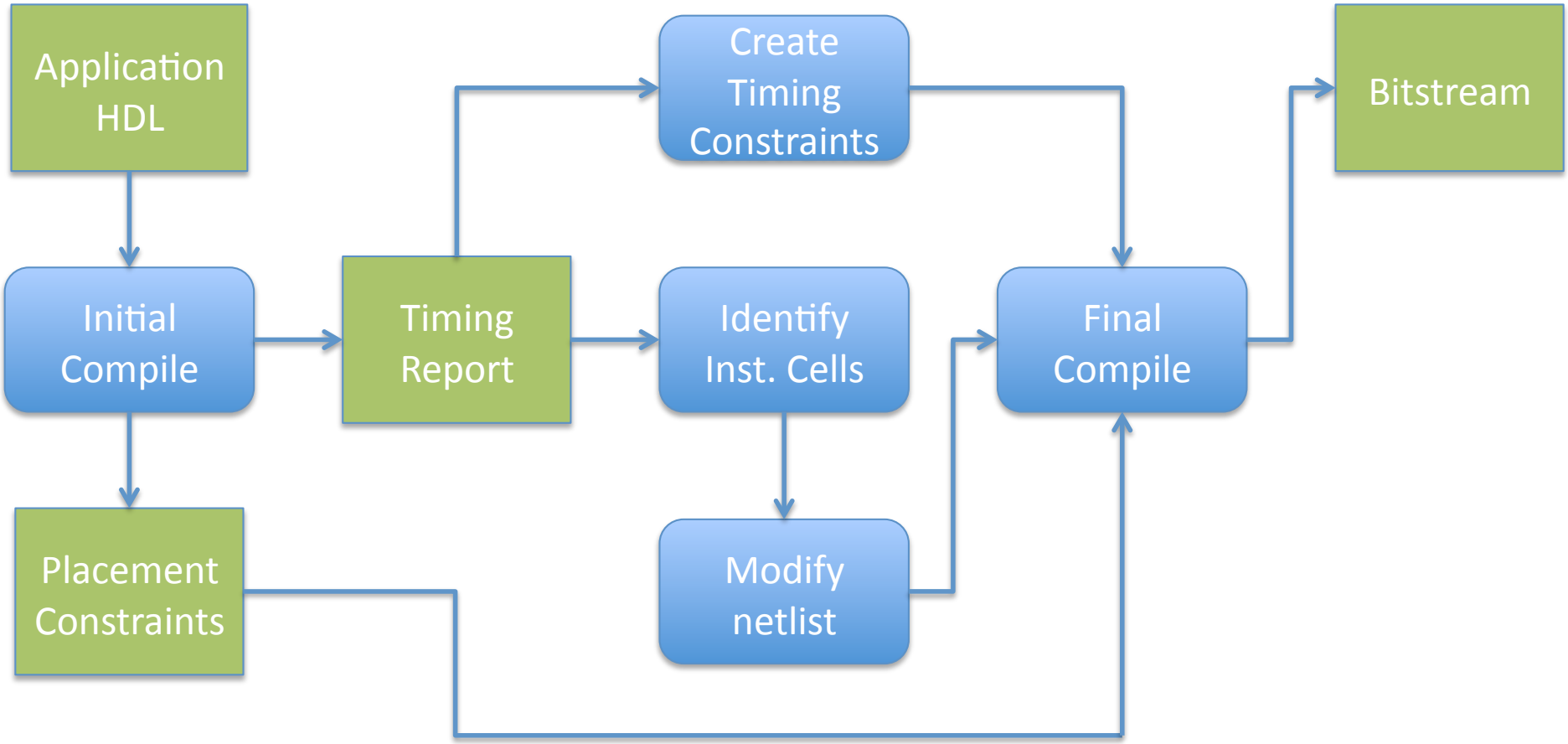
- Timing error detection requires additional constraints
- Shadow register must not sample fast edges from next clock cycle





Imperial College London

Design Flow – Timing Constraints

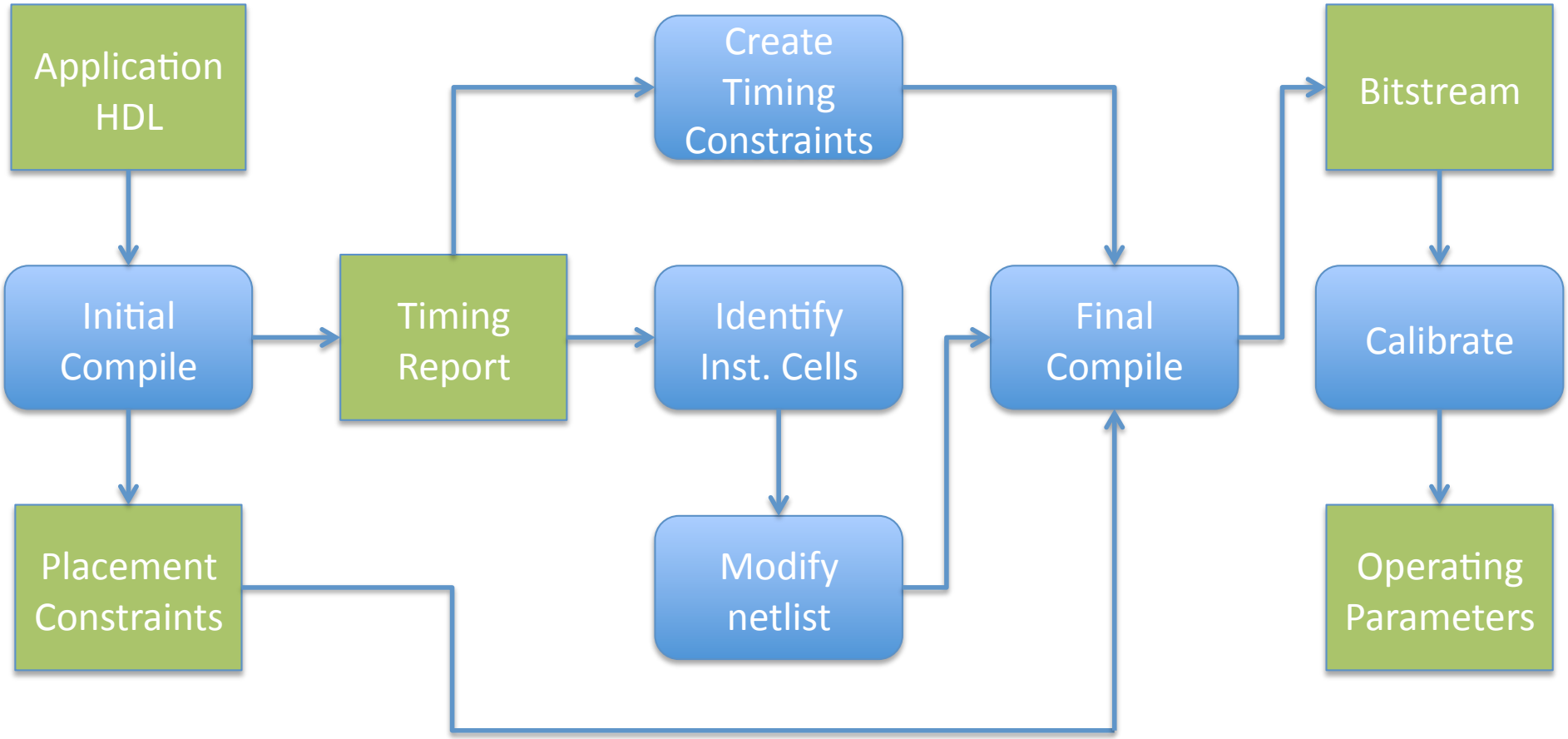


Calibration

- Timing models are not accurate enough to determine operating parameters
  - Approx. 2× difference between ‘fast’ and ‘slow’ models in FPGA
- Measure key parameters in silicon:
  - Shadow path delay offset (correction factor for slack measurement)
  - Shadow register hold slack (limits range of fault detection)
- Offline test
- Uses no additional hardware

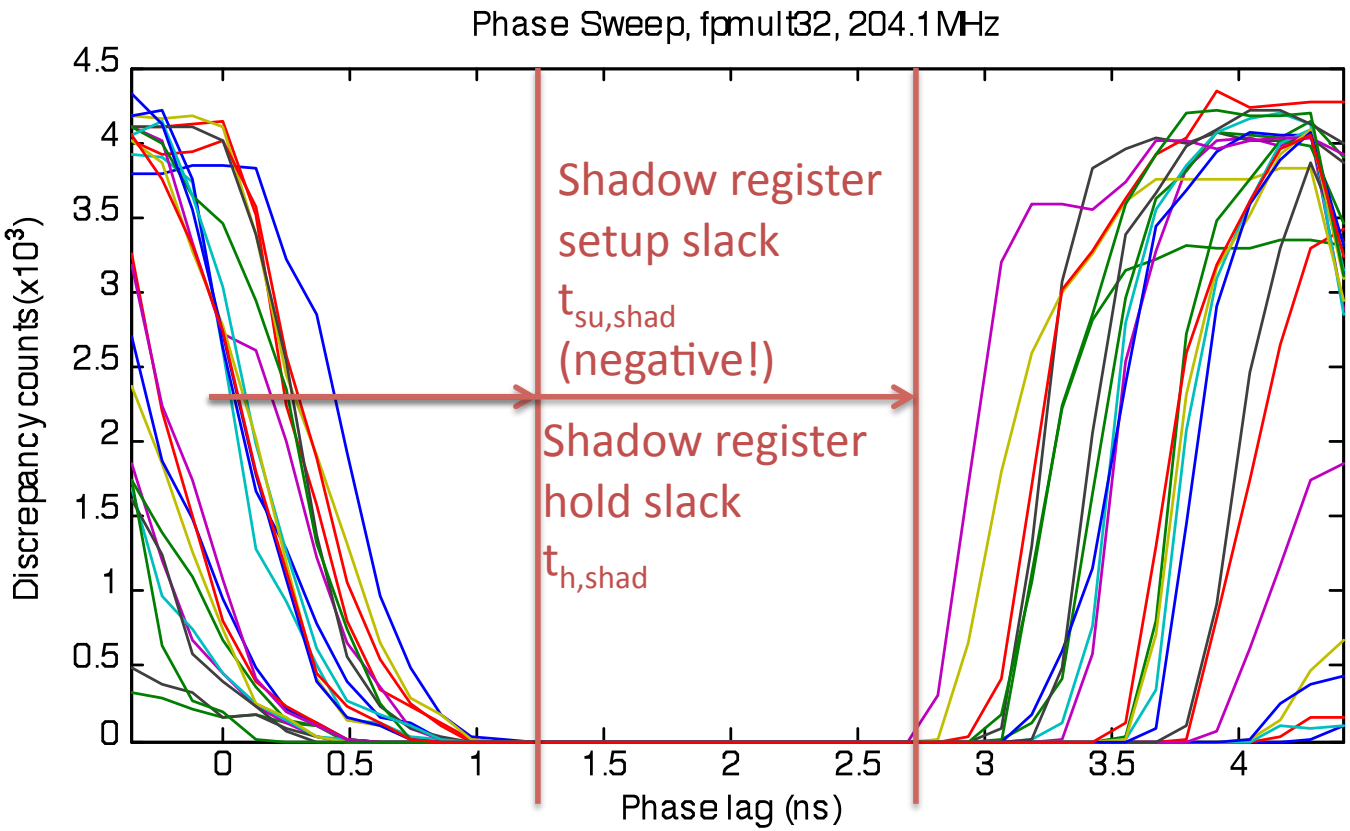
Imperial College London

Design Flow - Calibration



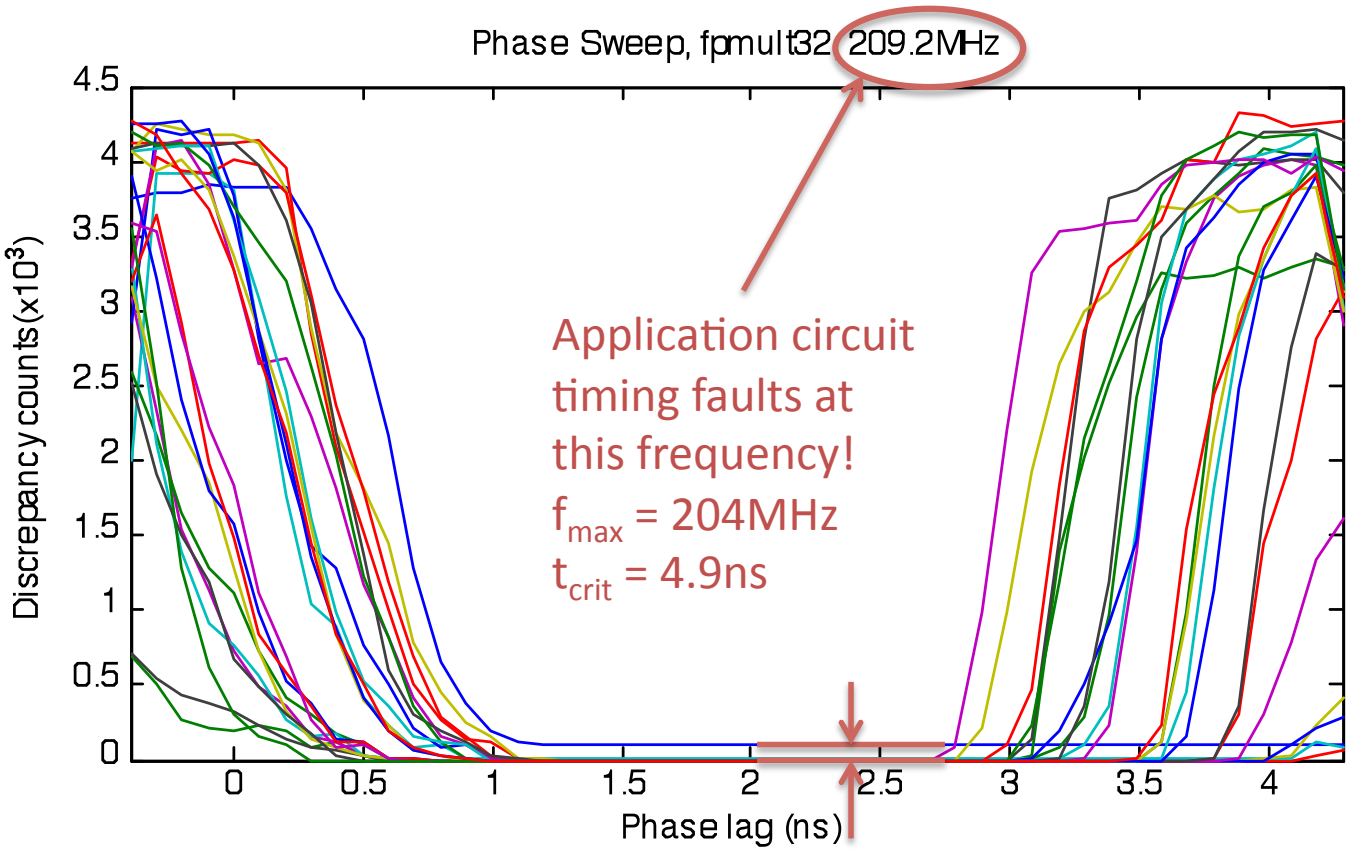
Imperial College London

Calibration



Imperial College London

Calibration



Imperial College London

Operating parameters

- Slack measurement offset = $-t_{su\_shad}$ (when measured at f_{max})
  - Calculated for each instrumentation cell

- Fault detection sample phase lag $t_{\phi} < t_{h,shad}$
  - Determines minimum clock period for fault detection:
  - $t_{clk} = t_{crit} - t_{\varphi} + t_{su,shad}$

Imperial College London

Overhead

Name	ICs	Logic Elements		f_{sta}	
		Δ	%	MHz	%
fpexp32	19	38	+5.0%	76.1	-6.0%
fpexp64	39	78	+2.6%	84.0	-0.1%
fplog32	44	88	+5.1%	95.6	-2.7%
fpmult32	24	48	+14%	135	-4.9%
dct1d	20	40	+7.7%	169	+1.8%
filter	13	26	+4.3%	104	+0.0%

IC = Instrumentation cell

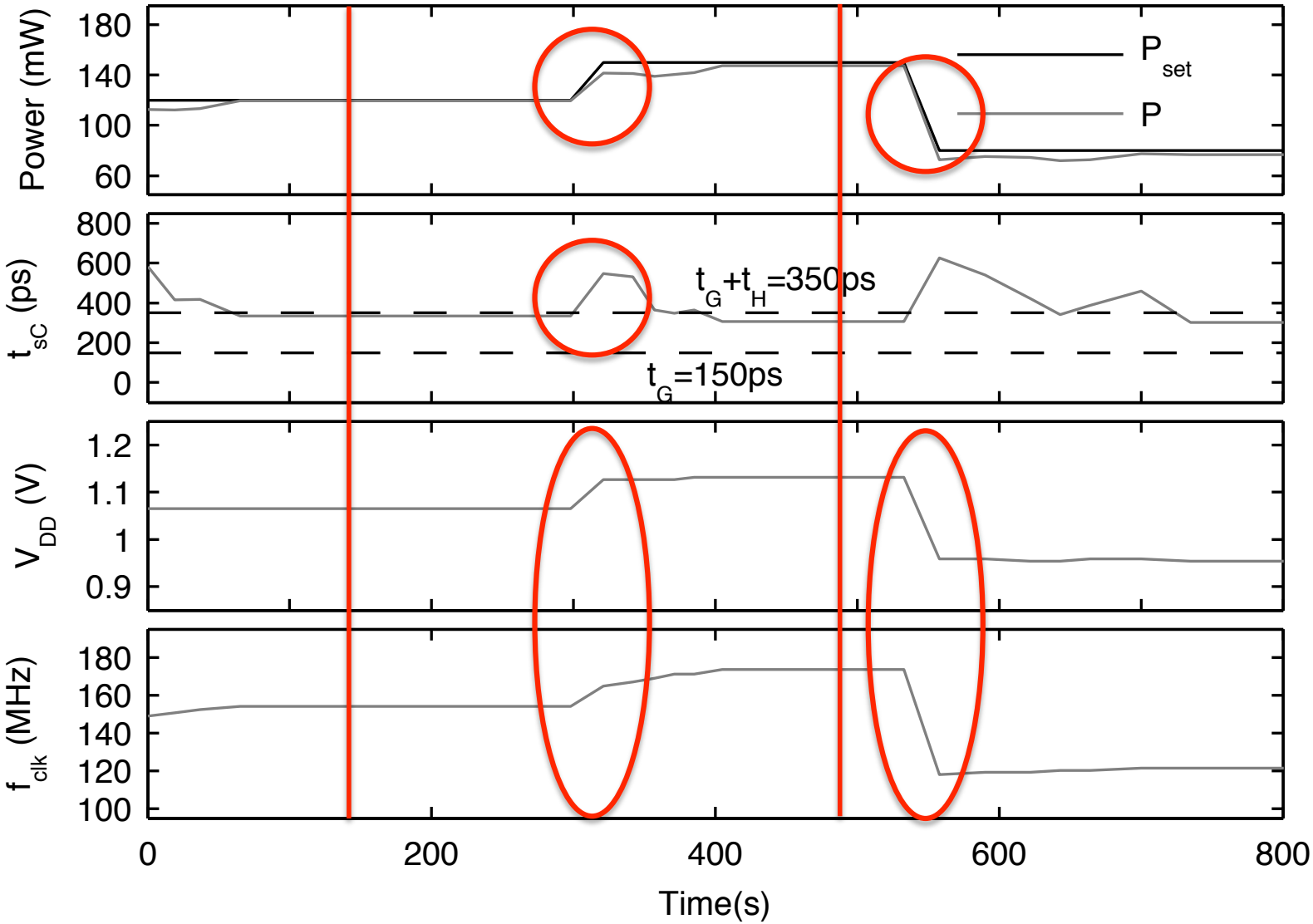
Critical Delay Margin = 10%

Imperial College London

Application – DVFS

- Closed loop dynamic voltage and frequency scaling
- Measure and adapt to: temperature, variation, degradation
- Still maintain guardband for: noise, data

- Setpoint: power limit
- Objective: maximise throughput
- State: timing slack
- Feedback: supply voltage and clock frequency


Power (mW)
180
140
100
60
P set
P
t sC (ps)
800
600
400
200
0
t G +t H =350ps
t G =150ps
V DD (V)
1.2
1.1
1
0.9
f clk (MHz)
180
160
140
120
100
0
200
400
600
800
Time(s)

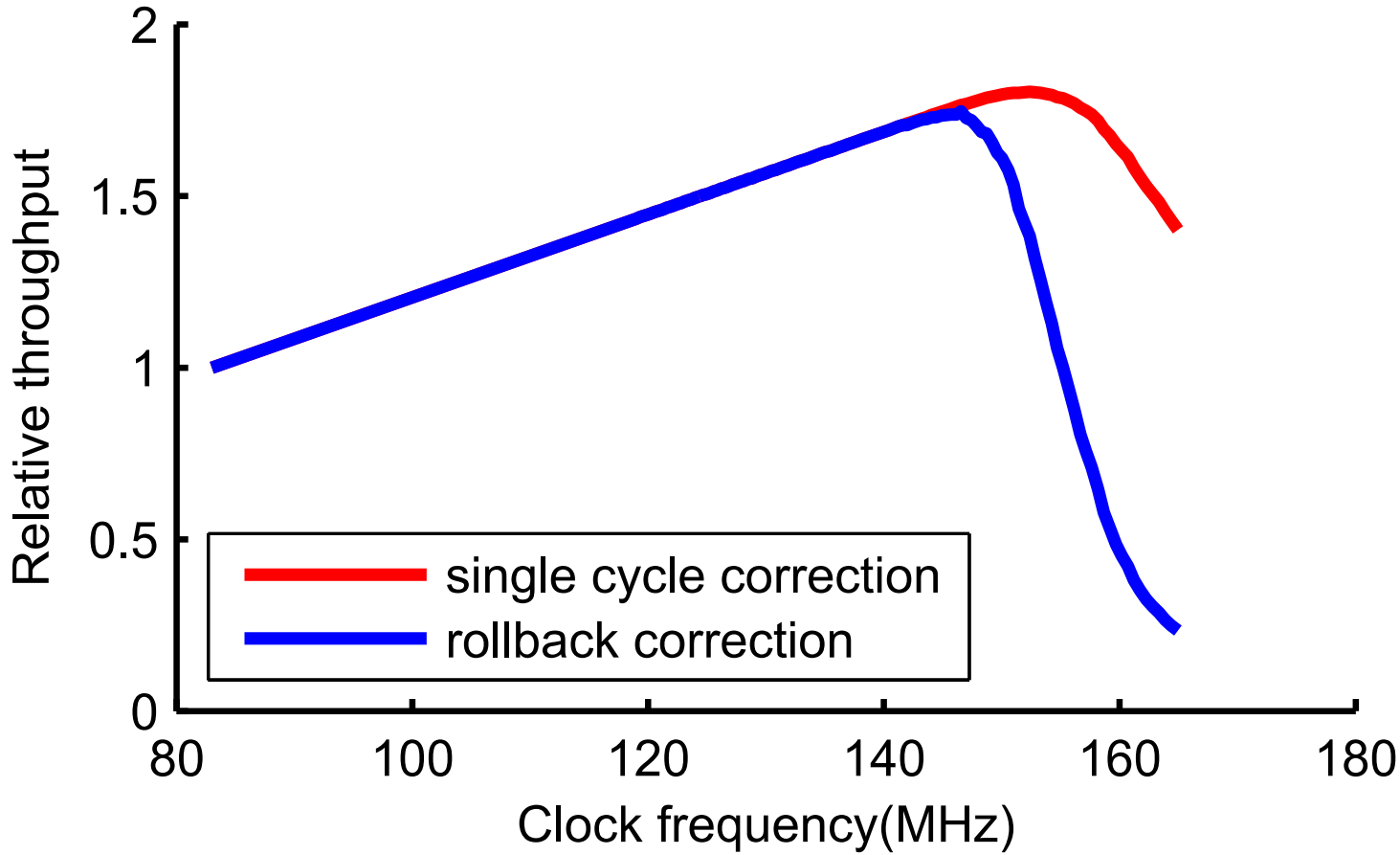
Application – fault correction

- Detect and correct timing faults
  - Correction still to implement!
- Measure and adapt to: temperature, variation, degradation
- Margin for <u>most</u> of: noise, data

- Experiment: Sweep frequency and measure timing fault rates
- Model performance overhead of fault correction
  - Assuming pipeline stall (Razor) or rollback

Imperial College London

Application – fault correction



Conclusion

- Tool flow for automated design instrumentation in FPGAs
  - Register selection
  - Netlist modification
  - Placement and routing lock
  - Timing constraints
  - Calibration
- Measure timing slack or detects timing faults
- Used for operation beyond worst-case
  - Closed loop DVFS with slack measurement
  - Potential use for timing fault correction (work in progress)

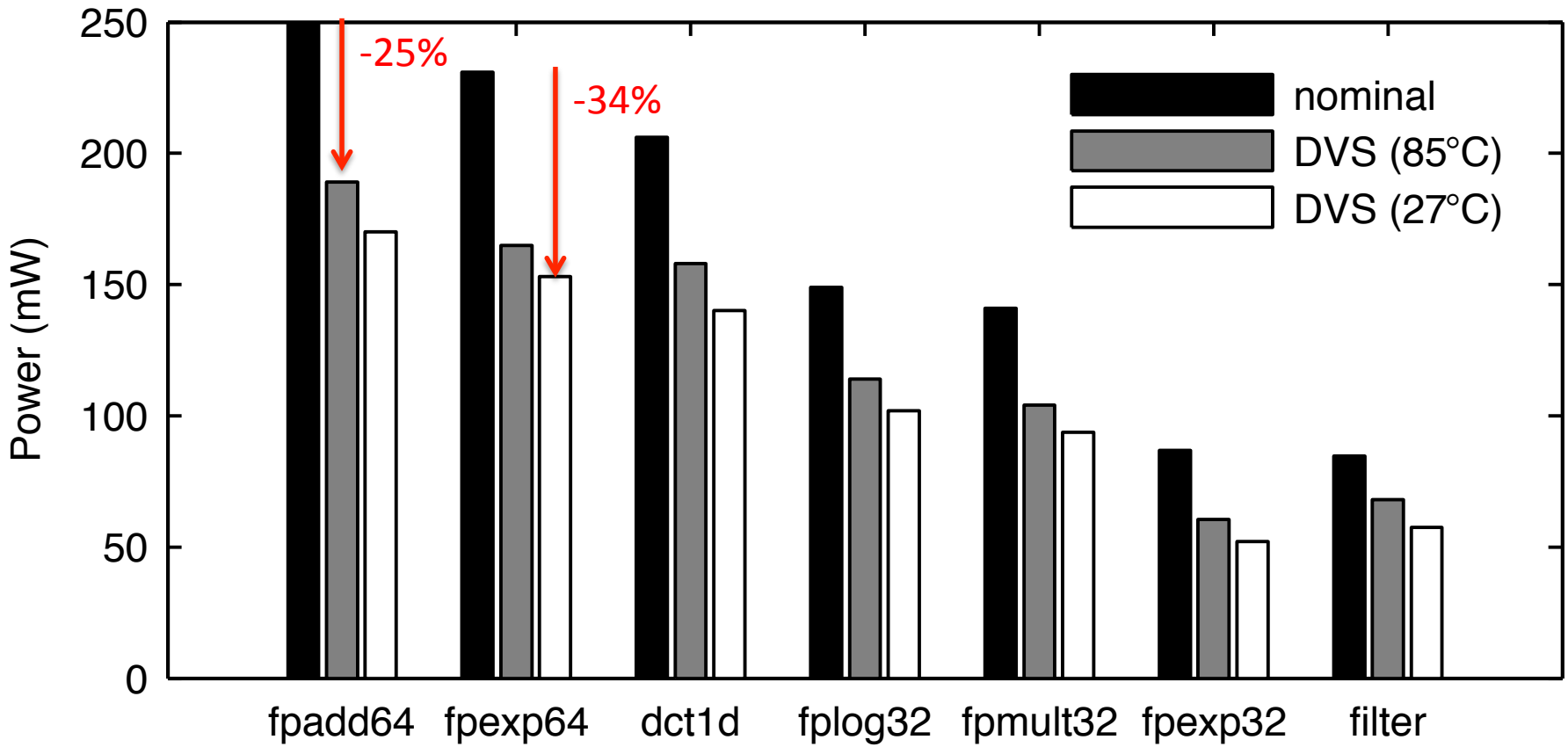
Imperial College London

The End

- Questions?

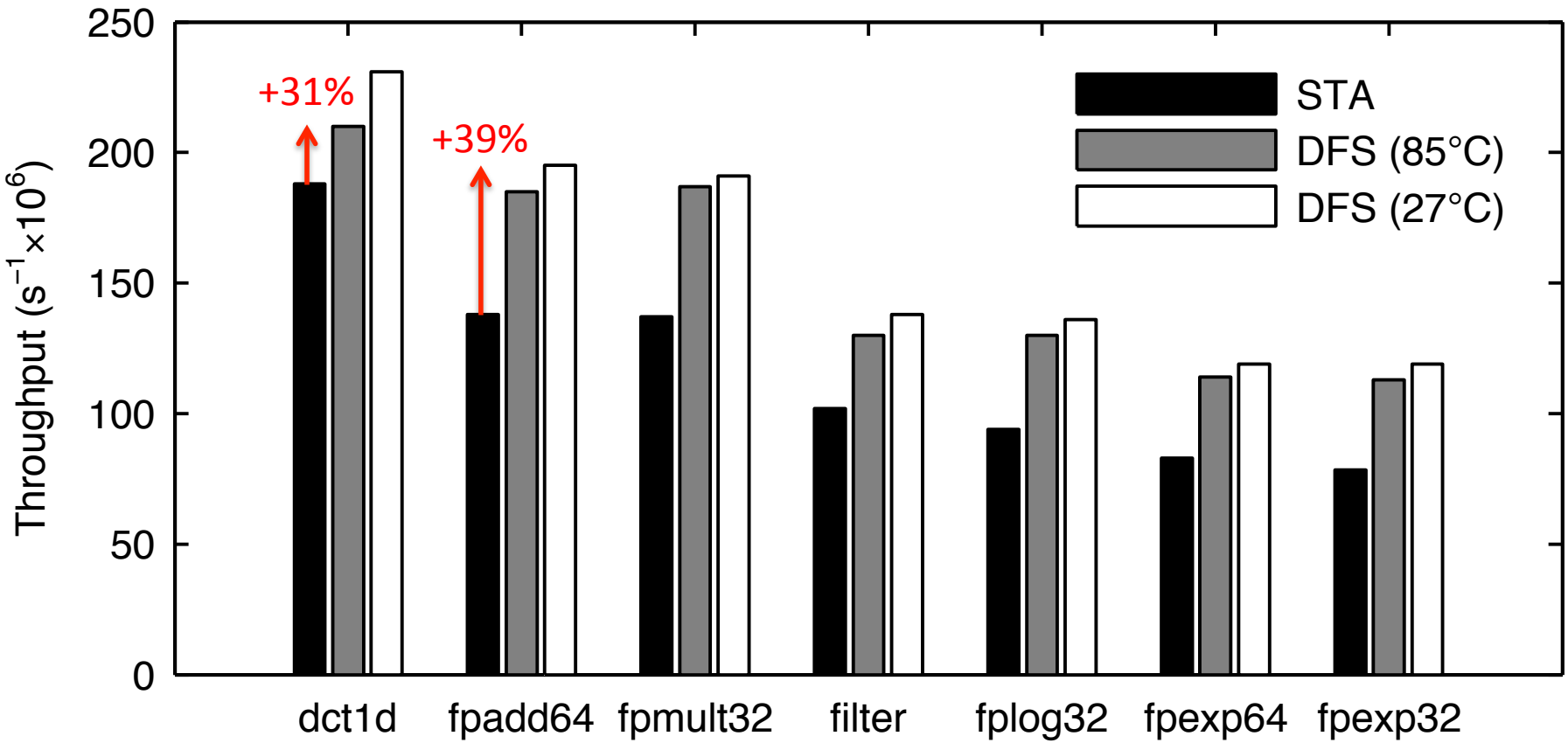
Imperial College London

Dynamic Voltage Scaling Results



Imperial College London

Dynamic Frequency Scaling Results



Imperial College London

Fault correction throughput

Name	f_{sta} (MHz)	stall TP$_{max}$	stall FR $\times 10^{-3}$	rollback TP$_{max}$	rollback FR $\times 10^{-3}$
fpexp32	81.0	1.58×	9.1	1.54×	1.2
fpexp64	84.1	1.80×	18.2	1.75×	0.61
fplog32	98.3	1.45×	21.9	1.40×	0.24
fpmult32	142	1.60×	16.5	1.56×	5.4
dct1d	166	1.29×	2.7	1.29×	2.7
filter	104	1.59×	15.0	N/A	N/A

TP: Throughput, FR: Fault rate

Imperial College London

Fault correction energy

Name	E_{nom} (nJ)	stall E_{min}	stall FR $\times 10^{-3}$	rollback E_{min}	rollback FR $\times 10^{-3}$
fpexp32	1.08	0.60×	4.5	0.63×	0.79
fpexp64	3.12	0.58×	7.4	0.60×	0.85
fplog32	1.40	0.66×	4.3	0.67×	0.21
fpmult32	0.75	0.66×	5.4	0.67×	2.5
dct1d	0.82	0.73×	24	0.75×	1.7
filter	0.77	0.61×	7.6	N/A	N/A