Imperial College London
Department of Electrical and Electronic Engineering

Online Timing Slack Measurement and its Application in Field-Programmable Gate Arrays

Joshua M. Levine

April 2014

Supervised by Professor Peter Y.K. Cheung

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Electrical and Electronic Engineering of Imperial College London
and the Diploma of Imperial College London

Abstract

Reliability, power consumption and timing performance are key concerns for today's integrated circuits. Measurement techniques capable of quantifying the timing characteristics of a circuit, while it is operating, facilitate a range of benefits. Delay variation due to environmental and operational conditions, and degradation can be monitored by tracking changes in timing performance. Using the measurements in a closed-loop to control power supply voltage or clock frequency allows for the reduction of timing safety margins, leading to improvements in power consumption or throughput performance through the exploitation of better-than worst-case operation.

This thesis describes a novel online timing slack measurement method which can directly measure the timing performance of a circuit, accurately and with minimal overhead. Enhancements allow for the improvement of absolute accuracy and resolution. A compilation flow is reported that can automatically instrument arbitrary circuits on FPGAs with the measurement circuitry. On its own this measurement method is able to track the "health" of an integrated circuit, from commissioning through its lifetime, warning of impending failure or instigating pre-emptive degradation mitigation techniques.

The use of the measurement method in a closed-loop dynamic voltage and frequency scaling scheme has been demonstrated, achieving significant improvements in power consumption and throughput performance.

Acknowledgements

I would like to thank my supervisor, Peter Cheung, for taking a chance on me when we first met in 2005, and for providing a research environment with immense intellectual freedom where my creativity could thrive. I am grateful to him and to my second supervisor, George Constantinides, for their insight, guidance and patience. From the inception of this project I have had the privilege of working closely with Ed Stott, my mentor and friend, and I look forward to our continuing collaboration. Thanks also go to the members of the "Reliability Club" and to our research group administrator, Wiesia Hsissen, for bringing order to the chaos.

To my colleagues and friends in the Circuits and Systems Research Group: thanks for the distractions, caffeine breaks, "Thirsty Thursdays" and "Doughts". I'd especially like to thank Adam Powell, Al Smith, David Boland, David Jones, Dom Buchstaller, Eddie Hung, James Davis, James Mardell, Peter Ogden, Rui Duarte, Sam Bayliss, Shane Fleming and Theo Drane; it has been a privilege to work alongside you and I have thoroughly enjoyed the experience. I am also grateful to Chris Chan and Ros Rathouse, for their friendship, advice and support.

Thanks also to my parents, Larry and Tessa, for teaching me the great value of knowledge and always encouraging me in its pursuit, and my brother, Adam, for helping me make this a thesis I am proud of. Special thanks to my girlfriend, Emma Bradley, for getting me through this challenging process, and tolerating me to the end. Final thanks goes to Belka the cat, for keeping me company and monopolising the keyboard!

None of this would have been possible without the funding provided by the EPSRC through grant EP/H013784/1, "Variation-Adaptive Design in FPGAs".

Contents

List of Tables

List of Figures

List of Acronyms

ADC Analogue-to-Digital Converter

ALM Adaptive Logic Module

ASIC Application-Specific Integrated Circuit

BRAM Block Random-Access Memory

CAD Computer-Aided Design

CDM Critical Delay Margin

CMOS Complementary Metal-Oxide Semiconductor

CP Charge Pump

CPU Central Processing Unit

CUT Circuit Under Test

DCT Discrete Cosine Transform

DFS Dynamic Frequency Scaling

DLL Delay-Locked Loop

DSP Digital Signal Processing

DVFS Dynamic Voltage Frequency Scaling

DVS Dynamic Voltage Scaling

EDA Electronic Design Automation

FF Flip-Flop

FIFO First In, First Out

FP Floating Point

FPGA Field-Programmable Gate Array

FRD Failure Rate Detection

HCI Hot Carrier Injection

HDL Hardware Description Language

IC Integrated Circuit

IIR Infinite Impulse Response

ITRS International Technology Roadmap for Semiconductors

JTAG Joint Test Action Group

LAB Logic Array Block

LE Logic Element

LEUM Logic Element Under Monitoring

LF Loop Filter

LFSR Linear Feedback Shift Register

LUT Lookup Table

LZA Leading Zero Anticipator

LZC Leading Zero Count

MCNC Microelectronics Center of North Carolina

MISR Multiple-Input Signature Register

MOS Metal-Oxide Semiconductor

MSB Most Significant Bit

NBTI Negative-Bias Temperature Instability

nMOS N-Type Metal-Oxide Semiconductor

OSM Online Slack Measurement

PBTI Positive-Bias Temperature Instability

PC Personal Computer

PFD Phase-Frequency Detector

PID Proportional-Integral-Derivative

PLL Phase-Locked Loop

PMBus Power Management Bus

pMOS P-Type Metal-Oxide Semiconductor

PSU Power Supply Unit

PWM Pulse Width Modulation

QSF Quartus Settings File

QUIP Quartus University Interface Program

RAM Random-Access Memory

RIPPL Reliability Instrumentation Platform for Programmable Logic

ROM Read-Only Memory

RUM Register Under Monitoring

SMART Self-Monitoring, Analysis, and Reporting Technology

SMI Slack Measurement Insertion

SPI Serial Peripheral Interface

SRAM Static Random-Access Memory

SSTA Statistical Static Timing Analysis

STA Static Timing Analysis

TCL Tool Command Language

TDC Time-to-Digital Converter

TDDB Time Dependent Dielectric Breakdown

TDL Tapped Delay Line

TP Transition Probability

TRC tunable Replica Circuit

USB Universal Serial Bus

VCO Voltage-Controlled Oscillator

VDL Vernier Delay Line

VHDL VHSIC Hardware Description Language

VLSI Very Large-Scale Integration

VQM Verilog Quartus Mapping

List of Nomenclature

$\boldsymbol{V_{DD}}$ Circuit supply voltage

$\boldsymbol{V_{in}}$ Circuit input voltage

$\boldsymbol{V_{out}}$ Circuit output voltage

$\boldsymbol{t_{pd}}$ Propagation delay

$\boldsymbol{C_L}$ Output capacitance

$\boldsymbol{W}$ Transistor gate width

$\boldsymbol{L}$ Transistor gate length

$\boldsymbol{\mu}$ Charge-carrier effect mobility

$\boldsymbol{C_{ox}}$ Oxide capacitance per unit area

$\boldsymbol{V_{th}}$ Threshold voltage

$\boldsymbol{T_{clk}}$ Clock period

$\boldsymbol{t_{cq}}$ Flip-flop clock-to-q delay

$\boldsymbol{t_{su}}$ Setup time

$\boldsymbol{t_{sk}}$ Clock skew

$\boldsymbol{t_s}$ Timing (setup) slack

$\boldsymbol{f_{clk}}$ Operating frequency

$\boldsymbol{f_{max}}$ Maximum frequency circuit operates correctly under given conditions

$\boldsymbol{f_{sta}}$ Maximum frequency as defined by timing model at V_{nom}

$\boldsymbol{f_{cal}}$ Calibration frequency

$\boldsymbol{f_{\mathrm{set}}}$ Requested operating frequency

$\boldsymbol{t_{\phi}}$ Shadow clock phase lead

$\boldsymbol{\Delta t_{\phi}(f)}$ Phase lead step size for frequency f

$\boldsymbol{d_i(t_{\phi})}$ Discrepancy count of RUM i at phase lead t_{ϕ}

$\boldsymbol{t_{\mathrm{sS,i}}}$ Timing (setup) slack at shadow register i

$\boldsymbol{t_{\mathrm{sR,i}}}$ Timing (setup) slack at RUM i

$\boldsymbol{t_{\mathrm{dS,i}}}$ Effective delay at shadow register i

$\boldsymbol{t_{\mathrm{dR,i}}}$ Effective delay at RUM i

$\boldsymbol{t_{\mathrm{sC}}}$ Critical timing (setup) slack

$\boldsymbol{t_{\mathrm{dC}}}$ Critical effective delay

$\boldsymbol{\Delta t_{\mathrm{cal}}}$ Clock period step size for calibration sweep

$\boldsymbol{t_{\mathrm{RS,i}}}$ Shadow register delay offset for RUM i

$\boldsymbol{\Delta t_{\mathrm{dR,slow/fast}}}$ Percentage intra-die delay variation at the slow/fast corner

$\boldsymbol{f_{\mathrm{in}}}$ PLL input frequency

$\boldsymbol{f_{\mathrm{ref}}}$ PLL reference frequency

$\boldsymbol{f_{\mathrm{vco}}}$ PLL voltage-controlled oscillator frequency

$\boldsymbol{f_{\mathrm{out}}}$ PLL output frequency

$\boldsymbol{V_{\mathrm{nom}}}$ Nominal operating voltage

$\boldsymbol{V_{\mathrm{min}}}$ Minimum voltage

$\boldsymbol{I_{\mathrm{leak}}}$ Leakage current

$\boldsymbol{\Delta t_{\mathrm{clk}}}$ Clock period step size for frequency scaling

$\boldsymbol{\Delta V_{DD}}$ Voltage step size for voltage scaling

$\boldsymbol{t_G}$ Timing (setup) slack guardband

$\boldsymbol{t_M}$ OSM latency

$\boldsymbol{t_L}$ Voltage and frequency control latency

$\boldsymbol{t_H}$ Timing (setup) slack hysteresis

$\boldsymbol{P}$ Operating power

$\boldsymbol{P_{set}}$ Requested operating power

$\boldsymbol{P_H}$ Power hysteresis

1 Introduction

Scaling the process technology used to manufacture integrated circuits ha's historically resulted in improvements in power consumption, switching performance and transistor density, as promised by Dennard [20]. This remains the driving force behind the semiconductor industry today. As these technologies have entered the nanometre scale, devices manufactured using them are increasingly experiencing the effects of delay variability. This variability is affecting the devices, not just at the time of manufacturing or commissioning, but during their lifetime, with an increased susceptibility to the effects of environment, operating conditions and degradation [74]. Newly manufactured devices already experience delay variation of as much as ±15% [11], which can deteriorate further by an additional 20% during a 10 year operating life [65].

In order for circuits to function reliably, manufacturers must account for the worst-case delay variation using timing margins. As the variability increases, so to do these margins, eroding much of the improvements achieved by process scaling and potentially making it counter-productive. It may no longer be possible for margins alone to protect against this variability, as doing so would jeopardise the circuit's performance to too great an extent. The coupling between variability and reliability is dominating the benefit of scaling [31] and is a chief concern of the International Technology Roadmap for Semiconductors (ITRS), an organisation representing key industrial leaders worldwide.

The use of sensors to monitor how a circuit is being affected by variability offers part of a solution and may make it possible to improve the viability of continued process scaling. Sensors of this type have yet to be fully realised.

Since this variability typically impacts the circuit's delay, measuring this directly would provide the greatest insight. Timing (setup) slack is the difference between the time that

data is required, and the time it is provided. A positive timing slack indicates that the circuit is operating safely, with a margin by which the delay can increase before causing failure. Measuring timing slack allows for monitoring of the "health" of a circuit and can provide an early warning of deterioration, trigging pre-emptive actions to avoid failure. These sensors form part of a new paradigm for circuit design know as "resilience", whereby systems are able to cope with stress and catastrophe though self-monitoring and adaptation.

This thesis outlines the Online Slack Measurement method (OSM), a technique which uses circuit level sensors to measure the timing slack at critical nodes in the circuit, thereby directly measuring the effects of delay variability. It can do so accurately and with a low overhead, both in terms of area and performance. A Slack Measurement Insertion (SMI) tool flow is described which allows for the measurement circuitry to be automatically added into arbitrary circuits implemented on Field Programmable Gate Arrays (FPGAs) with little-to-no manual intervention.

Dynamic Voltage and Frequency Scaling (DVFS), is an example of a resilient system, whereby the operating parameters of the circuit are controlled in response to measured changes in performance. This thesis demonstrates DVFS using OSM, which is capable of achieving significant improvements in power efficiency or performance in today's devices through the reduction of timing margins. It also has the potential to increase the lifetime of an integrated circuit through self-adaptation.

1.1 Outline

Chapter 2 gives an overview of sources of delay variability and the means by which the effects of this variability can be measured. It points to the need for a method of measuring the timing performance of a circuit directly, while it is operating, in order to establish how it is being affected by variability and from this how "healthy" it is.

Chapter 3 introduces Online Slack Measurement, a technique which is able to directly measure the timing slack in an online circuit, without impacting on its operation. Methods of enhancing this technique to improve accuracy and resolution are described, as is a

technique for identifying which parts of the circuit should be instrumented for monitoring. A variety of circuits are investigated to establish the relationship between delay distribution and amount of the circuit which must be instrumented.

Chapter 4 details how the Online Slack Measurement technique can be mapped to current FPGA architectures, and presents a tool, Slack Measurement Insertion, which can automatically add the measurement circuitry to arbitrary circuits with a minimal overhead, both in terms of the area and timing performance.

Chapter 5 uses Online Slack Measurement in a close loop to quantify the timing performance of the circuit and control its supply voltage and/or clock frequency in order to improve power efficiency, throughput or provide operation beneath a power envelope.

1.2 Published Work

The majority of the work presented in this thesis has been published following peer-review. This section briefly describes the relevant publications.

The principle of Online Slack Measurement was presented at the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays in 2010 [37]. This included calibration and early work on frequency dithering.

In 2012 a paper was published in the IEEE International Symposium on Field-Programmable Custom Computing Machines that described the fully developed Online Slack Measurement technique, calibration, and methodology for the selection of registers for instrumenting [38]. Also included was an analysis of the overheads for monitoring circuits and a provisional study into the use of Online Slack Measurement for Dynamic Voltage and Frequency Scaling.

A paper describing the Slack Measurement Insertion tool flow, and mapping of Online Slack Measurement to the architecture of a current generation FPGA was published at the International Conference on Field Programmable Logic and Applications in 2013 [39].

Also in 2013, Online Timing Slack Measurement formed part of a paper published in the IEEE Design & Test of Computers [56]. This described Online Slack Measurement and its applications in the context of variation and reliability in FPGAs.

Dynamic Voltage Frequency Scaling using Online Slack Measurement automatically added to arbitrary circuits on FPGAs using Slack Measurement Insertion was published at the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays in 2014 [36].

Recently, a paper describing timing error detection using shadow registers in arbitrary circuits, which are instrumented automatically using RIPPL, was presented at the International Conference on Field Programmable Logic and Applications in 2014 [57].

1.3 Statement of Original Contributions

As evidenced by the series of associated publications, this thesis describes a number of original contributions. These are the work of the author except where stated otherwise and include:

- A novel method for measuring timing slack in circuits while they are operating and without disrupting this operation.
- The calibration of this measurement method to improve absolute accuracy.
- An analysis of the accuracy of the measurement method.
- A technique for improving the measurement resolution of the online timing slack measurement method.
- A method for the selection of important registers to be monitored by the measurement method.
- Mappings of shadow registers, to be used for the measurement technique or other applications, to a modern FPGA architecture.
- A tool flow for compiling the relevant hardware necessary for the timing measurement method into arbitrary circuits while minimising timing overheads.
- The use of online timing slack measurement for dynamic voltage and frequency scaling.

- A adaptation of the calibration technique to ensure conservative measurement when used for dynamic voltage and frequency scaling.
- Guardbanding techniques to ensure optimised and save circuit functionality under dynamic voltage and frequency scaling.
- Dynamic voltage and frequency scaling controllers for maximising throughput or efficiency, or providing operation under power constraints, either static or dynamic.

2 Background

2.1 Introduction

Despite manufacturers best efforts, no two integrated circuits are alike. They are affected by delay variation at the time of manufacturing and during the course of their operating life. This chapter reviews the literature relating to the sources of this variation, and methods by which it can be quantified.

2.2 Delay Variability and Timing Failure

The propagation delay of transistors can be affected by a number of parameters. In synchronous circuits, variation in the propagation delay of the transistors which make up the circuit can result in timing failure, whereby the circuit ceases to function correctly. This delay variability may be investigated by analysing a simple CMOS inverter circuit, as shown in Figure 2.1.

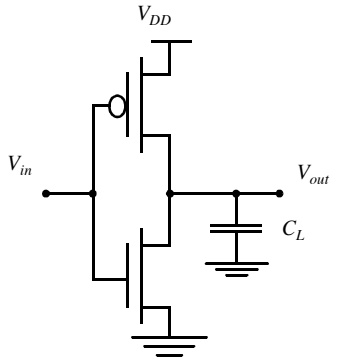


Figure 2.1: CMOS inverter circuit.

This inverter is powered by voltage V_{DD} with an input signal V_{in} and output V_{out} driving output capacitance C_{L}. The propagation delay (t_{pd}) of this gate defines how quickly it

responds to a change at its input with a change at its output, the delay experienced by a signal when passing through the gate. Since the gate may exhibit a different response to rising or falling transitions these are defined separately as $t_{\mathrm{pd,LH}}$ for low to high (rising) and $t_{\mathrm{pd,HL}}$ for high to low (falling), where $t_{\mathrm{pd,LH}}$ is the time taken for the output to rise from 10% to 90% of V_{DD} when there is an appropriate change of input, and $t_{\mathrm{pd,HL}}$ the inverse, falling from 90% to 10%. The overall propagation delay is the average of these as in Equation 2.1.

$$t_{\mathrm{pd}} = \frac{t_{\mathrm{pd,LH}} + t_{\mathrm{pd,HL}}}{2} \tag{2.1}$$

When considering delay variability, the relative change in propagation delay from a nominal baseline, due to changes in transistor parameters are of particular interest. It can be shown that the rising propagation delay is approximately equal to to the parameters in Equation 2.2 and falling Equation 2.3 [29], where W is the gate width, L the gate length, μ the charge-carrier effective mobility, C_{ox} the gate oxide capacitance per unit area and V_{th} the threshold voltage.

$$t_{\mathrm{pd,LH}} \approx \frac{C_{\mathrm{L}} V_{\mathrm{DD}}}{\frac{W_{\mathrm{pMOS}}}{L_{\mathrm{pMOS}}} \mu_{\mathrm{pMOS}} C_{\mathrm{ox}} (V_{\mathrm{DD}} + V_{\mathrm{th,pMOS}})^2} \tag{2.2}$$

$$t_{\mathrm{pd,HL}} \approx \frac{C_{\mathrm{L}} V_{\mathrm{DD}}}{\frac{W_{\mathrm{nMOS}}}{L_{\mathrm{nMOS}}} \mu_{\mathrm{nMOS}} C_{\mathrm{ox}} (V_{\mathrm{DD}} - V_{\mathrm{th,nMOS}})^2} \tag{2.3}$$

From this he effect of variability on delay can be seen. A increase in L or decrease in W or C_{ox} results in an increased delay. Increased V_{th} or μ also contributes to increased delay. μ typically decreases with increasing temperature (except in cases of temperature-inversion), again resulting in increased delay and decreasing V_{DD} has the same effect.

In a synchronous system, this increase in delay can result in timing failure. Timing failure occurs when the delayed data does not arrive at the register sufficiently before the clock edge to meet this register's setup requirement (Equation 2.4) where T_{clk} is the clock period, t_{cq} the clock-to-Q delay and t_{su} the register's setup requirement. t_{sk} is the clock skew, between the source and sink register and has various sources including: systematic (which exists under nominal conditions), random (due to variability in manufacturing

processes), drift (from slow time dependent environmental changes) and jitter (due to high frequency environmental variation).

$$T_{\text{clk}} - t_{\text{sk}} \geq t_{\text{cq}} + t_{\text{pd}} + t_{\text{su}} \tag{2.4}$$

If data arrival at the flip-flop meets this setup requirement (t_{su}), the above inequality is satisfied and the circuit never experiences timing failure. The setup requirement is related to the probability of the flip-flop meeting its t_{cq} specification. If the setup requirement is not met, there is a non-zero probability of timing failure, which increase the more t_{su} is violated. This may result in the flip-flip latching either a previous or indeterminate value, or becomes metastable, requiring an unpredictable amount of time before settling on an output value. The flip-flip becoming metastable can result in timing failure occurring in subsequent flip-flops, which can also become metastable, resulting in complete failure of the circuit.

The difference between the time that data is required ($T_{\text{clk}} - t_{\text{sk}}$) and the time that it is provided ($t_{\text{cq}} + t_{\text{pd}} + t_{\text{su}}$) is the setup slack, hereon referred to as timing slack t_{s}. A positive timing slack implies that the combined delay to the register can be increased up to t_{s} before timing failure occurs. A negative slack implies that the delay is too great for the chosen clock period, and incorrect values are being latched.

Measuring the combined effect delay, setup time and skew ($t_{\text{cq}}+t_{\text{pd}}+t_{\text{su}}+t_{\text{sk}}$), henceforth referred to as effective delay, or timing slack in a newly manufactured integrated circuit allows us to establish the effect of variability on the timing performance of the circuit. Monitoring this through the life of the device measures the impact of environmental and temporal variability. As the combined delay nears the clock period, or the slack tends towards zero, timing failure becomes imminent. Thus, monitoring of delay or slack in an integrated circuit is an excellent metric for the "health" of this circuit.

Timing margins are additional slack added to ensure that timing failure does not occur despite delay variability. Manufacturers use these timing margins to achieve sufficient parametric yield (the number of circuits that meet their specification). These margins impose worst-case functionality on devices, even in better-than worst-case conditions, resulting in increased power consumption and decreased performance.

2.3 Sources of Variability in Integrated Circuits

The behaviour of integrated circuits is primarily affected by three sources of variability: physical, environmental and temporal [74]. The effect of these sources of variation is expected to increase with process scaling [42, 10, 53].

2.3.1 Physical Variability

Process variation is the parametric variation of components in integrated circuits, due to variability in fabrication. It results in variation in parameters including gate oxide thickness, dopant concentration and device geometry and can have an effect on threshold voltage [66]. Process variation can be correlated at the wafer, between dies (inter-die), within a die (intra-die) or uncorrelated (stochastic). In current process technologies, physical variability is responsible for as much as a $\pm 15\%$ variation in circuit delay at the time of manufacturing [11].

2.3.2 Environmental Variability

Environmental variability includes variations in environmental and operational factors, both within and external to the integrated circuit. External factors include fluctuations in supply voltage and package temperature. Internal effects are due to coupling within the circuit and result in delay uncertainty. They are typically high frequency, manifesting as noise, and include: thermal coupling (self-heating [9]), voltage coupling (voltage droop [47]), clock jitter [28] and inductive/capacitive coupling (crosstalk [43]).

2.3.3 Temporal Variability

Temporal variability refers to degradation, which can occur gradually over time, or suddenly, resulting in catastrophic failure. It is manifested as a change in circuit parameters or functionality and is due to a number of physical effects [54].

Negative Bias Temperature Instability (NBTI) has been shown to be the dominant effect in current process technologies [51, 14]. It is caused by trapped charges or defects in the interface region, from negative gate-to-source voltages and as such primarily affects pMOS

transistors. It results in a gradually increased threshold voltage and reduced channel mobility. Degradation due to NBTI occurs all the time the transistor is turned on, not just during switching. Positive Bias Temperature Instability (PBTI) is the equivalent mechanism in nMOS transistors but currently has negligible effect [35].

Hot Carrier Injection (HCI) is the result of defects being accumulated in the interface between the channel and gate and causes a gradual increase in threshold voltage and reduction in mobility. HCI is dependant on the drain current and predominantly occurs during switching [27].

Time Dependent Dielectric Breakdown (TDDB) results from a breakdown in the gate oxide [72]. In mild cases this can lead to an increase in leakage current, in more sever cases failure in the transistors ability to switch. Like NBTI this is TDDB is driven by gate potential and occurs whenever the transistor is on.

Electromigration is caused by the movement in metal ions in conductors, eventually leading to the creation of open and short circuits as these ions erode or build up [16].

The same degradation mechanisms are not responsible for both gradual deterioration and catastrophic failure. Mechanisms that cause variation in threshold voltage (V_{th}), such as NBTI, PBTI and HCI, result in a gradual deterioration in switching performance. Catastrophic failure is caused by TDDB and electromigration, although the physical mechanisms by which this occurs differ.

Experimentation has show NBTI to be the primary factor resulting in degradation in current FPGA technologies [59] and that, while degradation is difficult to model, it is repeatable, being highly dependent on data, supply voltage, temperature and circuit structure [58]. NBTI has been shown to cause shifts in threshold voltage (V_{th}) of up to 50 mV over an operating lifespan of 10 years in 65 nm technologies. This translates to more than 20% deterioration in circuit operating speed [65].

2.3.4 Evaluation of Delay Variability

Process scaling results in an increase in all sources of delay variability as transistor parameters such as channel length and threshold voltage spread. It is becoming a key factor in the development of mainstream digital circuits. As Dennard's Constant Field Scaling

came to an end with 90nm process technologies, the return to Constant Voltage Scaling has resulted in greater degradation due to increased electric fields. Timing margins must already accommodate delay variations of up to 30% for physical variability and an addition 20% for temporal variability. As the timing margins grow further, the benefits of scaling will no longer be realised; there is thus a requirement for new techniques, from transistor technology upwards. Delay measurement and monitoring is one area in which the development of new methods could prove valuable in addressing this.

2.4 Digital Delay Measurement

Measurement of the delay of digital circuits is an active area of research. These measurements are used for a variety of purposes including characterisation of variability and measurement of circuit performance. The primary methods of measuring delay are discussed below.

2.4.1 Delay Inference

These dedicated delay measurement circuits are used to characterise an integrated-circuit die. From these measurements the behaviour of an application circuit on the same die can be inferred.

Ring Oscillator

The Ring Oscillator remains the industry standard for characterising the effect of intra-die process variation and chipwide temperature and voltage variation in integrated-circuits [52]. The ring oscillator consists of an odd number of inverters, connected into a ring. The output of the last inverter is the inverse of the input, and feeding this back to the first inverter results in the ring oscillating freely at a frequency determined by the delay of the inverter elements.

The frequency can be generalised as in Equation 2.5 where f is the oscillation frequency, n the number of inverters and t_{pd} the propagation delay of a single inverter. The frequency of the output can be measured using a simple counter. Typically a large number of inverters

are used in order to keep the oscillation frequency well within a measurable range.

$$f = \frac{1}{T} = \frac{1}{n(2t_{\mathrm{pd}})} \tag{2.5}$$

Through carefully controlled experiments ring oscillators are capable of independently measuring delay, static and dynamic power and temperature [73]. However, the ring oscillator is not without significant weakness. The free running nature makes it susceptible to self-heating due to dynamic power dissipation and it is not possible to distinguish the difference in propagation delay between rising and falling transitions, only the average, which may be significantly faster than the worst-case delay.

Time to Digital Converter

Time to Digital Converters (TDC) provide a method for precisely measuring the delay of the components they are made up from and providing this in a digital representation. They can also be used to measure the propagation delay of a signal through a Circuit Under Test (CUT). They are typically constructed from registers and delay elements (usually buffers) and, like ring oscillators, can be used to measure the effect of intra-die process, temperature and voltage variation, and degradation when measuring a CUT.

Tapped Delay Lines (TDL) [48] consist of a chain of delay elements, with registers D-inputs tapped between them. A measurement is started by pulsing the input to the delay chain high and stopped by pulsing the clock input of the registers high. The stored bit pattern corresponds to the distance that the start pulse has propagated through the delay chain when the measurement is stopped. The delay of the buffer chain can be measured by pulsing the start and stop at a known time interval, and the delay of a CUT by generating a start pulse when the signal enters, and a stop pulse when the signal has propagated through the CUT.

The resolution of the TDL is dependant on the delay of the individual elements and the total delay which can be measured, the delay of the combined chain.

A Vernier Delay Line (VDL)[23], as shown in Figure 2.2, is an enhancement of the TDL which improves measurement resolution. The clock input of the registers taps into an additional chain of delay elements, which have a different delay (t_1) to those of the

D-inputs (t_2). Here, the VDL is started by pulsing the clock input chain and stopped by pulsing the D-input chain. As the start and stop pulses propagate through their respective chains the time difference between the pulses is decreased in each Verner stage. This improves the time resolution to $t_1 - t_2$ when $t1 > t2$.

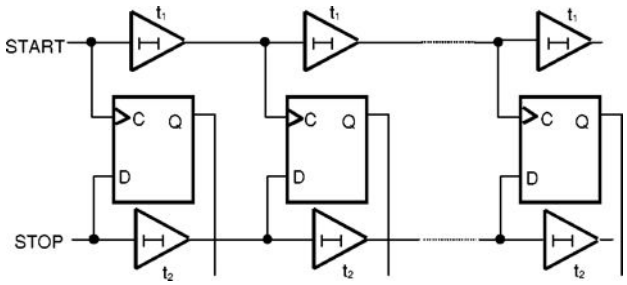


Figure 2.2: Vernier Delay Line from [23].

Where TDC are used to measure a circuit's propagation delay they must be calibrated to remove the effect of variability on the TDC itself. This can be conducted by repeatedly measuring a known time interval. Alternatively, where possible, a Delay Locked Loop (DLL) can be used to bias the delay elements in a closed-loop, actively compensating for any variation in the delay of the TDC itself.

TDC circuits are able to measure their own delay, or the delay of a CUT, accurately and with a high resolution. The primary limitation is difficulty in the generating the of the start and stop signals which must have identical skew so as to cancel out.

Tunable Replica Circuit

Tunable replica circuits (TRC) allow for degradation in the application to be inferred by exploiting its repeatability. Replica circuits are either designed to be particularly susceptible to the different sources of degradation, or replicate the functionality of critical paths from the application circuit. The replica circuits are excited with either stress data that is worst-case for degradation, or vectors from the application circuit, and their performance monitored with TDC [22] or timing error detection circuits like those described in Section 2.4.3 [61, 60].

The worst-case data for degradation depends on the dominant source of degradation. Currently, this is NBTI, which affects pMOS transistors subjected to negative gate-to-

source voltages (V_{gs}). In the case of a CMOS inverter manufactured in current process technologies, the worst-case data for degradation would be a zero input.

The measurement methods discussed above (i.e. ring oscillators and TDCs) are unable to determine degradation as they do not share the structure of the application circuit, or data driving it. TRCs, combined with a measurement technology, have the capability to infer degradation, in addition to being able to quantify intra-die process variation and chipwide temperature and voltage variation with a reasonable cost. The accurate this degradation inference is uncertain and has not been well explored. Measuring the application circuit directly, using the methods discussed below, overcomes the requirement for this inference.

2.4.2 Frequency Sweep Based

These techniques share in common the introduction of timing failure in the circuit to measure delay. As such, they are not suitable for applications where the circuit needs to continue operating correctly while measurement is conducted.

Signature Analysis

Timing measurement using signature analysis exploits the fact that a functioning circuit, configured in the same state and receiving the same set of input data, will produce the same output [32]. The output of the CUT is analysed, either by storing the vectors in a memory, or generating a signature. A signature generator such as a Multiple-Input Signature Register [24] can be used, either internal or external to the device. Using a signature generator simplifies comparison and can improve the time it takes to conduct a measurement as the finite memory would have to be copied during measurement and compared for each output value.

The CUT is clocked at a frequency which is known to be safe, and excited with a set of input test vectors, which can be pre-prepared, or generated procedurally at test time. A Linear-Feedback Shift-Register (LFSR) [7] is suitable as it is deterministic and will produce the same input after re-initialising with the same seed. The CUT is run for a fixed number of clocks cycles, and the output recorded. It is reset, and the clock period decreased.

This process is repeated until such a time as the output differs from that of the previous

clock period. Timing failure has occurred and thus the overall delay of the circuit is between the current and previous clock period.

Signature analysis is suitable for measuring the delay of most CUTs. In complex CUTs care must be taking in selecting the input vectors and measurement duration in order to provide an assurance that the critical path of the circuit has been measured.

Failure Rate Detection

In failure rate detection (FRD) [67], a CUT is sandwiched between two registers which are driven by a variable frequency clock generator as shown in Figure 2.3. An input stimlus generator excites the CUT and the clock frequency is gradually increased, from that at which the circuit is know to work to an upper bound. Timing failure in the CUT is detected by comparing to an at-speed reference generator and the number of errors for each frequency are accumulated by an error counter into an error histogram.

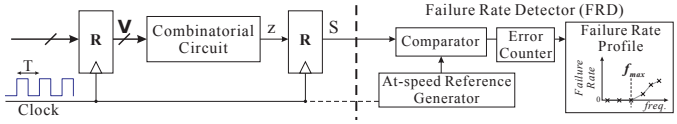


Figure 2.3: Block diagram showing the details of the FRD measurement circuit from [67].

The frequency at which the circuit fails timing, and therefore the delay of the CUT can be found from this histogram. Deriving this from a histogram rather than the point at which the first failure occurs provides greater accuracy and for the measurement and normalisation of clock jitter.

FRD provides a method for measuring real circuits, producing meaningful and realistic characterisation. Unlike Signature Analysis it does not require that the input vectors are identical for each frequency, and the hardware overhead of the reference generator and comparator circuitry is lower than that of an MISR. When used for process characterisation, this allows for more CUTs and associated measurement circuitry to be placed on a single device and finer granularity characterisation to be conducted. However, FRD is limited to testing only one output register at a time and cannot directly support multipath circuits.

Transition Probability

Transition probability (TP) [70, 69] is a low overhead means of measuring delay in combinatorial circuits. It is based on the observation that a circuit exercised with vectors of a fixed transition probability will output vectors of a constant transition probability until a timing error occurs, which will result in a change of output transition probability.

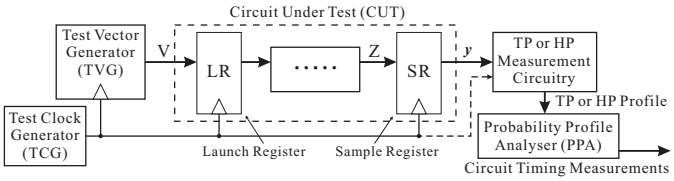


Figure 2.4: A circuit diagram illustrating the principle of TP measurement from [69].

A schematic of TP is shown in Figure 2.4. A CUT is connected to a test vector generator and launch register on its input, and a sample register, and measurement circuitry on its output. This measurement circuitry consists primarily of a transition counter and accumulator.

As in the other frequency sweep measurement methods, the circuit is stimulated with input vectors and the clock frequency gradually stepped up. The output transition probability for each frequency is recorded in the accumulator from which the delay of the circuit can be established, which is aided by the presence of clock jitter and an asymmetry between rising and falling delays. Tuning the transition probability of the input vectors improve the accuracy of measurement and can be conducted automatically [68].

Transition probability offers high accuracy characterisation of real circuits, which can be treated as a black-box. It has a lower overhead than the other frequency sweep methods and does not require that the circuit is exercised with a fixed set of input vectors, just that these vectors have the same transition probability.

2.4.3 Shadow Registers

The techniques described in this section use additional registers, which "shadow" chosen registers in the application circuit. These shadow registers share some of their inputs with the main register that they are shadowing and, depending on the relationship of these inputs, can inform of impending timing failure or the occurrence of timing errors. While

not strictly timing measurement, they provide information relating to timing failure of the circuits which they are monitoring.

Failure Prediction

Failure Prediction [2], provides a warning when a specified guardband has been violated, indicating impending timing failure. Also known as "canary circuits" [34], named after the Miner's Canary which would be taken into a coal mine as an early warning of the presence of toxic gasses, these shadow registers are configured to be more timing critical than the register they shadow, so that timing failure occurs sooner when the circuit's delay increases.

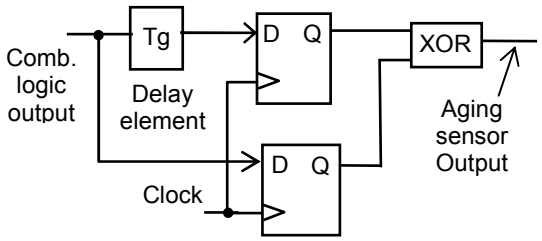


Figure 2.5: A Failure Prediction sensor from [2].

Here, the shadow register shares both the D-input and clock of the main register, but with additional delay inserted into the path of the D-input (see Figure 2.5). This additional delay causes the shadow register to be more timing critical than the main, and fail timing before it.

When there is sufficient slack for both to registers operate correctly, they both latch the same value which is indicated by comparison. When the circuit delay is increased sufficiently, the shadow register will latch the previous value and comparison indicates a guardband violation. The size of the guardband is configured as the amount of delay inserted to the shadow register's D-input.

In some circumstances it is not possible to reliably introduce this additional delay. As an alternative, the clock to the shadow register can be advanced, such that it latches sooner, achieving the same effect [6, 8]. This may be less susceptible to variation, thus achieving more accurate detection. Additional registers can be added with gradually increasing guardbands to provide more timing information [33].

Care must be taken when placing and connecting the shadow registers, so that they are near to the main register, or so that additional path to the shadow register is accounted for.

Failure prediction allows for the monitoring of registers in a circuit at a relatively low cost. It informs when the delay of this circuit exceeds a configured safe value, violating a guardband. It monitors the actual paths of interest, not a different circuit somewhere else on the die, and so can detect directly, guardband violations due to process variation, changes in operating conditions and degradation.

Razor

Razor [25] is a flip-flop capable of detecting and responding to the introduction of timing errors. It is intended for use in processors and the accompanying circuitry exploits features commonly available in these.

The Razor flip-flop is shown in Figure 2.6, it consists of a main pipeline register which is rising edge triggered. This register is augmented with a shadow register which samples on the clock's falling edge. This gives the shadow register additional time (corresponding to half the clock period) to capture the correct state of the data. The Razor flip-flop flags an error when the two registers latch different data, implying that a timing violation has occurred in the main register.

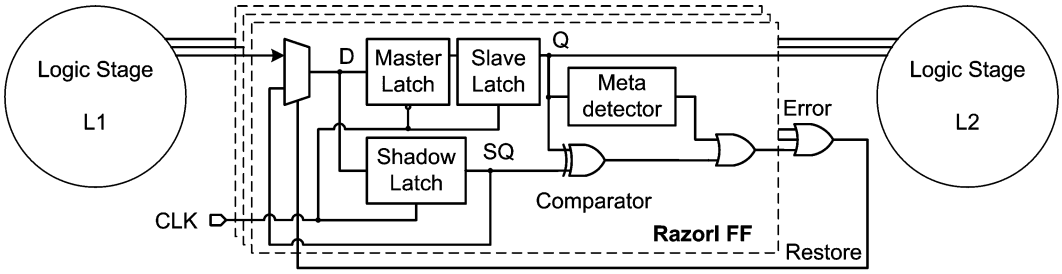


Figure 2.6: A pipeline stage augmented with Razor from [25].

The incorrect value in the main register must now be corrected before it propagates to the next pipeline stage. The error signal is ORed with the signals from all other Razor flip-flops and used to stall, or insert a bubble into the pipeline. The correct value must now be reinserted into the main register from the shadow in the same clock cycle.

Since timing errors may occur in the main register, there is a possibility of this becoming metastable. This makes it impossible to determine by comparison if the latched value is correct. In an attempt to mitigate this risk, a metastability detector is used in the comparator circuit, detecting and triggering correction.

There is a risk that the shadow register, driven by a delayed clock, may latch a value from a short path of the subsequent clock cycle, which would present as a timing failure in the current clock cycle, resulting in a false-positive measurement. This is overcome by applying minimum path length (hold) constraint to the circuit which results in the introduction of buffers to short paths.

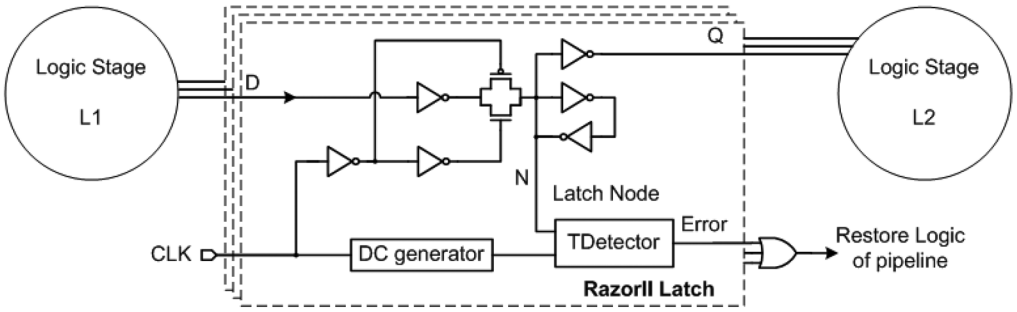


Figure 2.7: The RazorII timing error detecting flip-flop from [18].

Difficulties in the implementation of the metastability detection and restore circuitry (which is liable to becoming timing critical) led to the development of the RazorII flip-flop as shown in Figure 2.7. This performs only error detection, passing responsibility for correction onto the processor architecture. The RazorII flip-flop uses a level sensitive transparent latch instead of main flip-flop. The latch becomes opaque when the clock is low, so never closes when data edges can occur, mitigating the risk of metastability. Timing faults are flagged by a late transition detection triggered by the rising clock edge. The extent of the transition detection window is dictated by the clock duty cycle and hold constraints must still be applied to the circuit in order to avoid false positive error detection due to short paths.

The ability to detect timing failure provided by Razor is the pinnacle of timing measurement technology. Timing faults introduced by all forms of delay variability, including single cycle nose effects can be detected. Combining Razor with closed-loop dynamic voltage and

frequency scaling gives almost all of the benefit of asynchronous techniques, without the prohibitive changes in design methodology. It is a highly appealing concept, but does come with some implementation issues related to design requirements and the need to correct timing errors that have been detected:

- The application of hold constraints will increase the area and power consumption of the design.
- Error recovery is either difficult to design or takes many clock cycles.
- Issues with metastability in the main paths of the circuit, or in the case of RazorII, complicated clocking.
- Requires modification to manufacturers highly optimised processor pipeline structures.

Razor has yet to be demonstrated on arbitrary (non-processor) circuits. It is expected that the overhead of implementing timing error recovery in circuits without existing replay/recovery would be prohibitive. The introduction and detection of timing errors makes Razor non-deterministic and unsuitable for tightly coupled systems and those with hard real-time constraints.

2.4.4 Evaluation of Measurement Methods

The advantages and disadvantages of the aforementioned measurement methods may be compared in the context of the following applications:

Process Technology Evaluation The evaluation of the delay variability characteristics of new process technologies requires simple circuit which are manufactured on test chips. This is historically the mainstay of ring oscillators but their weaknesses (self-heating and averaged rise/fall time) make TDCs an appealing alternative. While slightly more complex, TDCs such as Tapped Delay Lines used to measure the propagation delay of a reference pulse are not thwarted by these weaknesses.

Circuit Timing Measurement It is often desirable to establish the timing performance of a manufactured circuit under various operating conditions (e.g. in order to measure

timing at the design corners). Doing so requires a non-invasive method, which does not require altering of the circuit and can be used with black-box. Both Signature Analysis and TP can be used although the latter achieves this with a lower overhead and reduced measurement time as the circuit does not need to be exercised with a fixed set of input vectors.

Circuit timing measurement may also be useful in other applications, where it can be used to provide highly accurate calibration measurements. Here all techniques are applicable as smaller elements of a large circuit are being measured, with FRD having the lowest overhead, particularly where the at-speed reference generator circuitry is already implemented.

"Health" Monitoring This is becoming increasingly important as the magnitude of temporal variation increases. It is used to provide an indication of imminent timing failure as setup slack is consumed. In principle, any of the online measurement methods which can establish the effect of degradation can be used, however it is difficult to account for mismatches between the intra-die, temperature and voltage variability and degradation between the TRC and application circuit. As such direct measurement is preferable, leaving failure prediction as the desirable option.

Failure prediction only provides a pass/fail indication of circuit health. A technique that is able to perform similar direct, online measurements, but continuous with high accuracy would be of benefit.

Dynamic Voltage and Frequency Scaling This is an adaptive technique whereby the operating parameters of the circuit and controlled in a closed-loop using circuit performance measurements. Any of the online measurement methods described previously can be used for DVFS, however the more sources of delay variability that can be captured by the measurement, the greater the timing margin that can be recovered.

Razor is at the pinnacle, by detecting timing errors that have occurred it is able to remove all timing margins. The circuit is tuned so that it functions "on the razor's edge", with variations such as noise or excitation of a rarely triggered critical path sufficient to cause a timing error. In some applications (e.g. computer vision) this

error need not be corrected and is simply fed back for control, however, in most cases the application is intolerant to error and it must be corrected. The controller balances overhead of correcting the error to achieve optimum throughput.

The cost of implementing Razor, in terms of area, performance and design effort is substantial, particularly in arbitrary circuits. Taking this into account, similar improvements could be achieved using a direct, online timing measurement method with a lower overhead.

Further discussion of DVFS can be found in Section 5.2.

Table 2.1: A comparison of the ability of the measurement methods to quantify sources of delay variation.

	Intra-Die Variation	Inter-Die Variation	Stochastic Variation	Degradation	Temperature	Voltage	Noise
Ring Oscillator	✓				✓	✓	
TDC	✓				✓	✓	
TRC	✓			✓	✓	✓	
Signature Analysis	✓	✓	✓	✓	✓	✓	
FRD	✓	✓	✓	✓	✓	✓	
TP	✓	✓	✓	✓	✓	✓	
Failure Prediction	✓	✓	✓	✓	✓	✓	
Razor	✓	✓	✓	✓	✓	✓	✓

Table 2.2: A comparison of the attributes of the measurement methods.

	Accuracy	Resolution	Area Overhead	Performance Overhead	Arbitrary Circuits	Direct	Online
Ring Oscillator	Low	Medium	Low	Low	-	✗	✓
TDC	Medium	Medium	Low	Low	-	✗	✓
TRC	-	-	Medium	Low	-	✗	✓
Signature Analysis	High	High	Low	Low	✓	✓	✗
FRD	High	High	Medium	Medium	✓	✓	✗
TP	High	High	Low	Low	✓	✓	✗
Failure Prediction	High	Pass/Fail	Medium	Medium	✓	✓	✓
Razor	High	Pass/Fail	High	High	✗	✓	✓

Table 2.1 gives the different sources of delay variability that can be quantified by the various measurement methods discussed. Table 2.2 provides a comparison of the attributes of the measurement methods. For a particular application, the optimum measurement method may be ascertained from these tables, considering the specific requirements.

2.5 Conclusion

This chapter has explored the wealth of research relating to variability, its resultant impact on circuit delay, and the means by which this delay can be quantified.

While historically there has been much focus on faults which result in non-functional circuits, in current process technologies concern is directed at variability's impact on timing performance, particularly the deterioration of this during the course of an integrated circuits lifetime due to degradation. This will typically lead to an abrupt catastrophic failure as critical paths in the circuit are no longer able to meet timing. Ongoing measurement of delay or slack, monitoring the "health" of the circuit, can be used to avoid this occurrence, instigating device replacement or degradation mitigation techniques.

This points to a need for a methods which can directly measure the effect of delay variation on an application circuit itself, accounting for the effects of process, environmental and temporal variation, without the need to infer how the behaviour of a circuit somewhere else on the die relates to the application circuit. It must do this without affecting the normal operation of the circuit and be able to track gradual changes in the delay of the circuit, not just provide an indication of timing failure that has occurred, or is impending. It should be general purpose, applicable to arbitrary circuits, not just processors. This can be achieved by combining the strengths of frequency sweep and shadow register based timing measurement to measure the delay of an arbitrary circuit, accurately and with high resolution, while it is operating online.

A technique capable of performing this measurement would be a valuable contribution to research, both as a method in its own right and as a facilitator to other technologies and is the primary objective of this thesis.

3 Online Slack Measurement

3.1 Introduction

Timing slack is an excellent measure for the health of a circuit. A large amount of slack indicates that the circuit is "healthy", that it would take a significant deterioration to result in timing errors. A small or negative quantity of slack is a concern, indicating the circuit is close to, or beyond the point of failure. The ability to measure timing slack in a circuit while it is operating opens many possibilities. Potential applications include on-silicon timing debug, timing degradation monitoring and dynamic voltage or frequency scaling.

In this chapter, a new method to precisely measure timing slack at selected registers, while the circuit operates normally, is presented. It allows a circuit to be continually monitored for timing performance variation over time; reductions in the amount of slack indicate that the circuit is degrading and changes due to temperature, voltage and other fluctuations can be tracked. The measurement method is enhanced with: 1) a calibration technique to improve absolute accuracy, 2) a dithering method to increase measurement resolution and, 3) a method of selecting registers from the circuit to monitor. The measurement technique is demonstrated on two simple benchmark circuits under a range of operating conditions.

3.2 Principle of Operation

Figure 3.1 illustrates a typical synchronous sequential circuit, which could be the complete circuit or an element of a larger circuit. A set of source registers (Regs 1) feed a block of combinatorial logic, the output of which is captured in sink registers (Regs 2). It is this type of circuit that the Online Slack Measurement (OSM) technique developed here

is applicable.

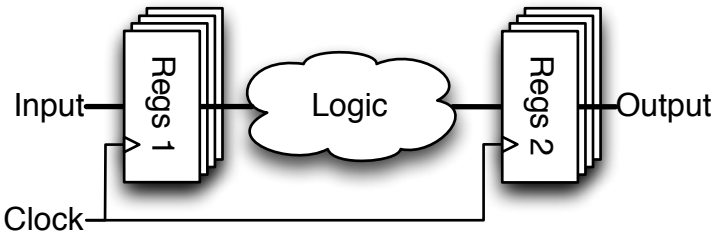


Figure 3.1: A typical synchronous circuit consisting of a block of combinatorial logic surrounded by registers. Bold lines indicate buses.

Circuits are instrumented for OSM through the addition of a sensor to a specific sink register which is known as a Register Under Monitoring (RUM). The sensor is connected to both the D-input and Q-output of the RUM as in Figure 3.2 but uses a different clock signal.

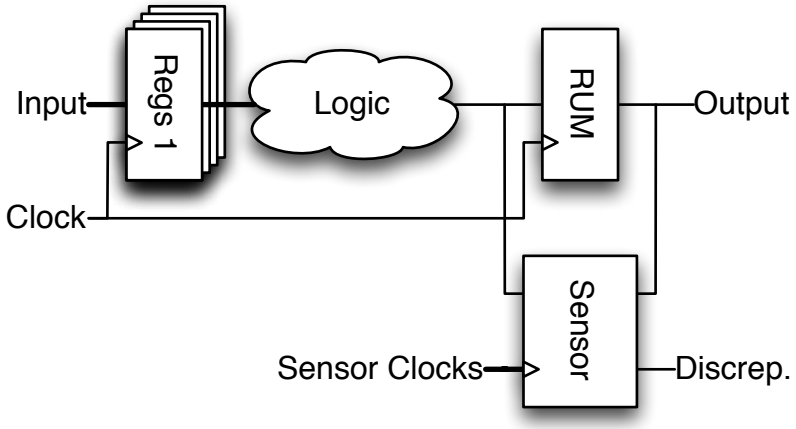


Figure 3.2: Instrumenting a synchronous circuit with the addition of an OSM sensor.

The OSM sensor circuit is shown in Figure 3.3 and consists of two additional registers and an XOR gate. The first register (Reg S) shadows the RUM, sharing its input. This shadow register is driven by a shadow clock with the same frequency as the main circuit's clock, but crucially, a programmable phase lead t_ϕ. It is this shadow register and its associated clock that the basis for the OSM method. The XOR gate is used to compare the outputs of the RUM and shadow register producing a discrepancy signal that is latched by the discrepancy register (Reg D). The shadow register (Reg S) and associated circuitry should be added in such a way as to minimise their effect on the performance of the application

circuit as a whole.

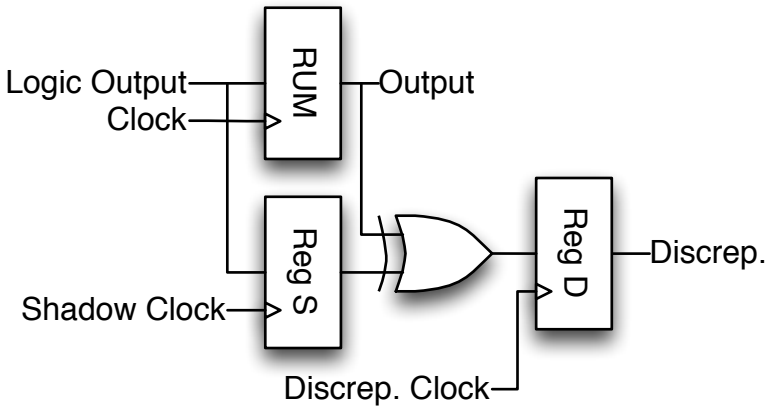


Figure 3.3: Details of the shadow register and associated circuitry for online timing slack measurement. The source register and logic has been removed for clarity.

During a measurement, the main and shadow clocks begin in phase ($t_{\phi 0} = 0$). The shadow clock phase lead, t_ϕ, is gradually increased in small steps such that the shadow clock leads the main clock signal. The shadow register is triggered earlier, gradually eroding slack. At the point at which all slack at the shadow register is eroded, the setup time for this register is violated, an incorrect value is stored. Comparing the values stored in the shadow register and RUM indicates a discrepancy. The exact amount of slack lies somewhere between the last step at which there are no discrepancies, and the next step with one or more discrepancies, thus the t_ϕ step at which this discrepancy occurs forms the upper bound of the slack at the shadow register, the lower bound is the step before the discrepancy. The exact slack lies between these two bounds with a uniform probability. For the purposes of general-purpose measurement the measured slack is assumed to be the midpoint of the bounds.

The measurement can be conducted online since timing failure is only induced in the shadow register. The RUM, its clock and the circuitry which it drives remain unaffected. This differs from other methods, such as Razor, whereby timing failure is introduced at the RUM, resulting in the incorrect value being stored and propagated throughout the rest of the circuit, requiring them to be conducted offline or use some method of timing fault recovery.

Figure 3.4 is a timing diagram showing the two cases: where a discrepancy does and

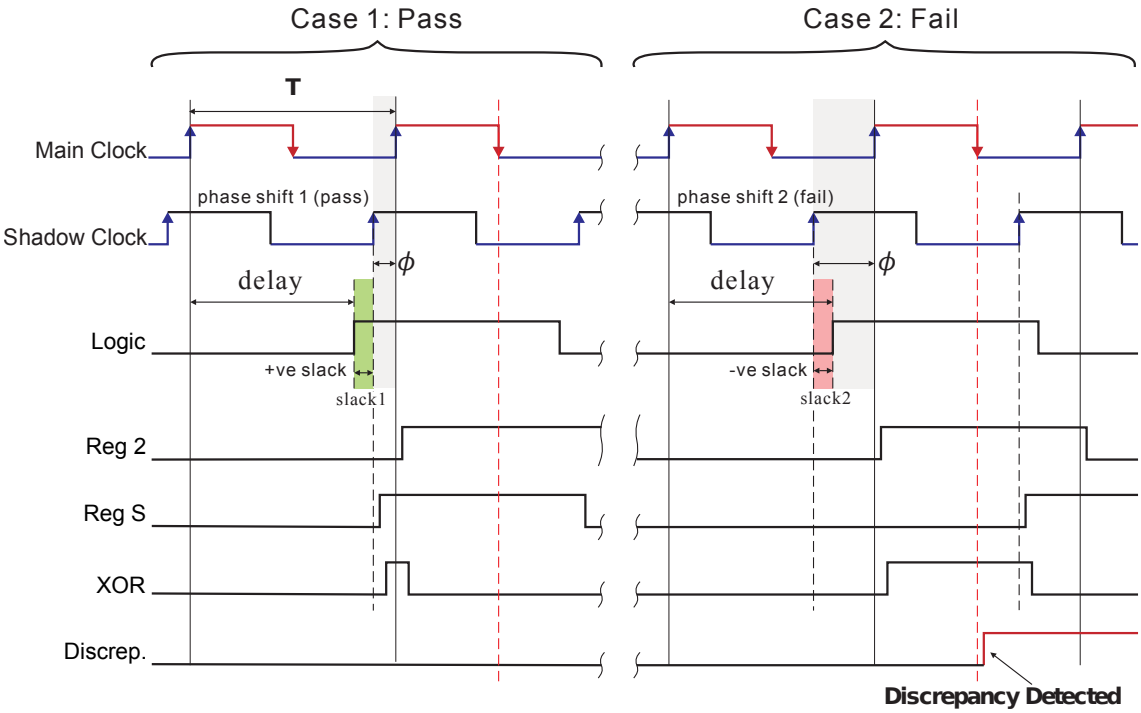


Figure 3.4: Timing diagrams where shadow clock lead (t_ϕ) does and does not result in a discrepancy.

does not occur. In the first case the shadow clock leads the main clock slightly, such that its rising edge occurs just before that of the main clock. There is still sufficient slack in the path to the shadow register such that the setup time for it is met. A glitch is visible at the XOR output since the two registers latch at different times and so briefly contain differing values, but this is not sampled by the discrepancy register. In the second case, t_ϕ leads further, such that the setup requirement for the shadow register is not met, and timing failure occurs. The discrepancy between the two registers is found by comparison and latched by the discrepancy register.

Accumulating the number of discrepancies during each phase step, over a $1/f$ sweep of t_ϕ, yields measurements as shown in Figure 3.6. These discrepancy profiles are cumulative histograms of data delay to the shadow register, where the discrepancy count for each step of phase lead corresponds to the number of times that data violated the setup time of the shadow register for that phase lead, or the number paths with a slack less than or equal to the phase lead. For this example the circuit being monitored is a chain of 32 buffers (`buffer`), exercised at its input with a toggle register. The output register is instrumented

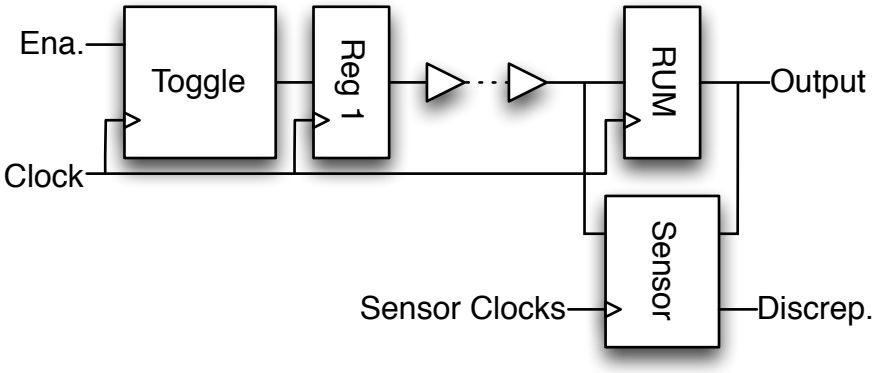


Figure 3.5: OSM instrumented buffer chain circuit.

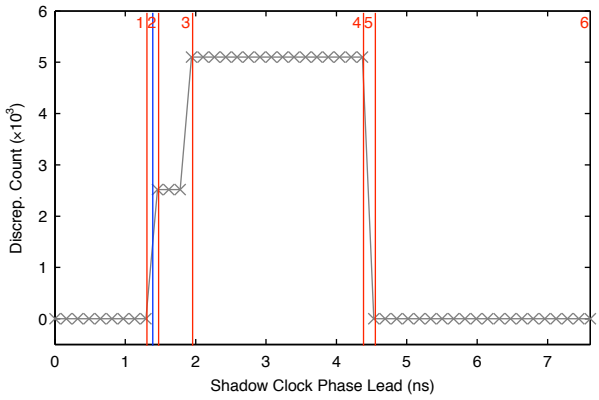


Figure 3.6: An example discrepancy profile for `buffer` running at f_{sta} (128.52 MHz), with regional annotations. The blue line indicates the measured slack, midway between the last zero and first non-zero discrepancy count ($d(t_\phi)$).

with OSM as in Figure 3.5 and the circuit is driven by a 128.73 MHz clock. The different regions of the measurement are demarcated.

In region one of Figure 3.6 the shadow clock is in phase, or slightly leading the main clock. The RUM and shadow register contain the same value so no discrepancy is produced. As t_ϕ is increased further, region two is entered. This region is bounded by the phase lead achieving the last zero discrepancy count, and the first non-zero count. As described previously, the slack at the shadow register corresponds to a phase lead within this region, which for general purpose measurement is taken as the halfway point between the two steps, as shown by the blue line. From here onwards, the setup requirement is not met for the shadow register so it will sometimes latch a different value to the RUM. The size and form of region three is determined by clock jitter and circuit complexity, in this case the step half way through the region being due to the differing rising and falling delays of the circuit. Increasing t_ϕ further still leads to the fourth region, where the RUM and shadow register are consistently sampling data from different clock cycles. Here, a discrepancy will be recorded whenever the input to the RUM and shadow register changes.

Measuring circuit timing by slack erosion (with frequency or period sweep techniques) allows us to establish the combined effect of the source register's clock-to-Q delay, the circuit's propagation delay, the sink register's setup time and any clock skew between the source and sink registers, which is called the effective delay. This can be computed as in Equation 3.1, where $t_{\mathrm{dR,i}}$ is the effective delay at register i, $1/f_{\mathrm{clk}}$ the clock period and $t_{\mathrm{sR,i}}$ the slack at register i. In this example, slack is measured to be 1.38 ns and since the clock period is 7.77 ns, the effective delay is (7.77 ns - 1.38 ns =) 6.39 ns.

$$t_{\mathrm{dR,i}} = 1/f_{\mathrm{clk}} - t_{\mathrm{sR,i}} \tag{3.1}$$

3.2.1 Blind Spot

It is not possible to measure over the entire range of the phase sweep, there is a small blind spot that can obscure measurements at a particular phase lead. This blind spot is due to timing failure in the path between the shadow and discrepancy registers and represents the point at which the advancement of t_ϕ causes the shadow clock to wrap around and lag

the main clock. In Figure 3.6 the blind spot is region five.

The location of this blind spot is primarily dependent on the choice of clock driving the discrepancy register, but is also affected by the delay of the path between it and the preceding shadow register. Table 3.1 shows the blind spot locations for the different discrepancy register clock options.

Table 3.1: The blind spot locations for different discrepancy register clocks.

Discrep. Register Clock	Blind Spot Location
main clock	Start
shadow clock	End
$\overline{\text{main clock}}$	Centre-Right
$\overline{\text{shadow clock}}$	Centre-Left

The location of the blind spot is an important consideration because if region two falls within it, timing measurement is impossible. Additionally, the blind spot occurring in region one complicates the interpretation of measurement results — it is difficult to differentiate between a non-zero discrepancy count due to the blind spot or a genuine discrepancy due to slack being eroded.

Since RUMs typically have near-critical delay, slack is eroded early through a phase sweep. Generally, $\overline{\text{main clock}}$ is used as the discrepancy clock. This allows t_ϕ to exceed $1/2f_{\text{clk}}$ before before reaching the blind spot, with some capacity maintained for calibration, as discussed in Section 3.2.3, on the right hand side of the phase sweep.

A non-zero discrepancy count with no shadow lead ($t_\phi = 0$) implies that there is no or negative slack at the shadow register. Region two will lie on the far side of the blind spot, and the slack bound is between the occurrence of the last zero discrepancy count and the subsequent non-zero. Since the phase has wrapped around, here the slack is calculated as the midpoint between the phase step bounds less the clock period.

3.2.2 Discrepancy Storage

In order to perform measurements, it is necessary to store information on discrepancies for each of the steps of t_ϕ. Discrepancy profiles as shown earlier can be produced by connecting the discrepancy output from the sensor to an asynchronous counter (Figure 3.7). At each phase step the value of the counter is read, stored and the counter reset. Each shadow

register in the design can have a dedicated counter or counters can be shared.

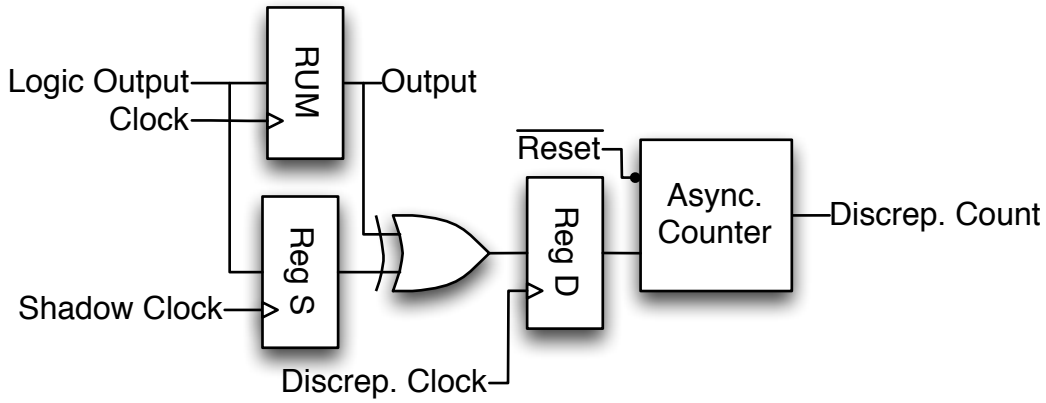


Figure 3.7: The discrepancy output can be connected to an asynchronous counter to produce an error profile.

Ultimately, the presence of only a single discrepancy is required in order to determine the available slack. Rather than using a counter, a discrepancy latch can be used. This latches high after the occurrence of one discrepancy and is reset for each phase step. Implementing a discrepancy latch requires feedback and reset logic as detailed in Figure 3.8, a very low overhead for each OSM sensor.

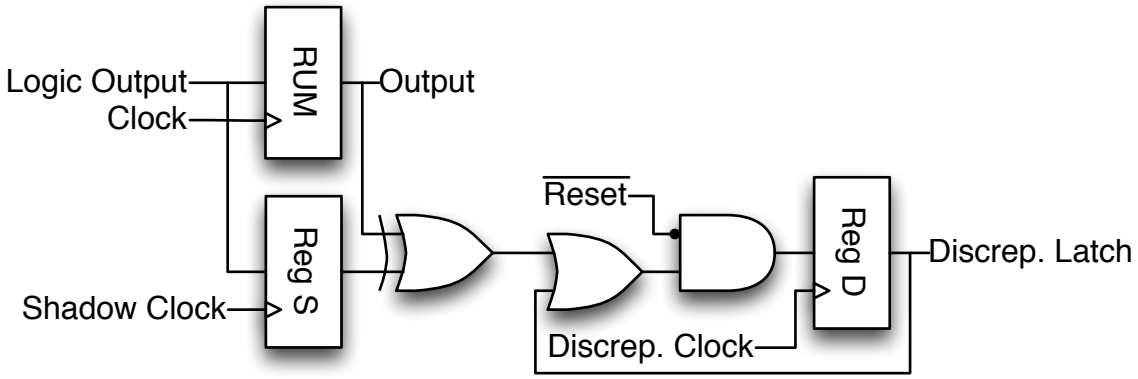
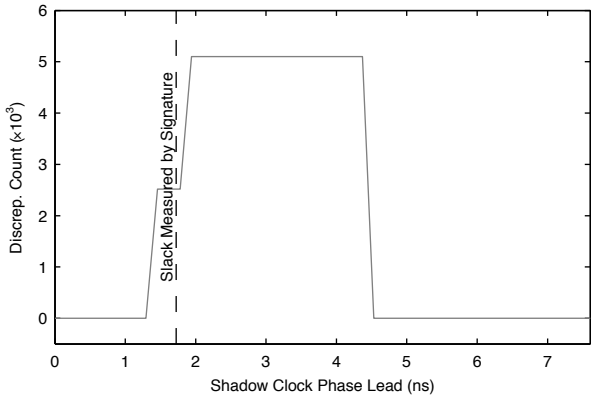


Figure 3.8: Using a discrepancy latch rather than counter, which latches when a single discrepancy has occurred.

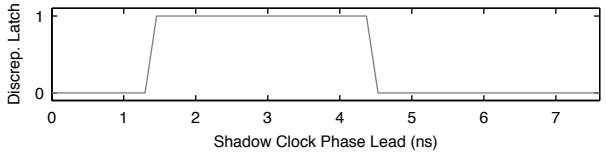
Figure 3.9 shows both discrepancy latch and count measurements of the `buffer` circuit.

3.2.3 Calibration

The OSM technique described thus far assumes that the effective delay at the RUM (t_{dR}) is identical to that at shadow register (t_{dS}) and that the two clocks have the same skew.



(a) Discrepancy Profile



(b) Discrepancy latch plot

Figure 3.9: Discrepancy profile and discrepancy latch plot for `buffer` operating at f_{sta} (128.52 MHz). Also shown is overall circuit slack measured by Signature Analysis using an MISR.

In practice this is not the case, measurements are made at the shadow register and from these the slack at the RUM is inferred. In order to avoid impacting performance of the application circuit, it is preferable that priority is given to the placement of the RUM during compilation. Often this necessitates that the shadow register and associated circuitry be situated away from the RUM, exacerbating the difference in delay.

If in an application the absolute accuracy of the measurement is not critical, e.g. tracking degradation, then the relative measurements achieved without calibration may be sufficient. If a worst-case measurement is needed, constraints can be applied to push the shadow register away from the RUM and the measured slack can be taken as the lower bound of the range rather than the midpoint, making the online slack measurement conservative. Some of the delay disparity can be accounted for using Static Timing Analysis (STA) to estimate the difference in delay between the RUM and shadow register, but measurements with the highest accuracy necessitate that the difference between the effective delay between the RUM and shadow register be accounted for through a calibration process.

The delay offset between the RUMs and shadow registers can be measured using a

one-time offline calibration process as defined in Algorithm 1. The calibration algorithm determines the shadow path offsets t_{RS} for each RUM i, by comparing timing slack measurement of the shadow path with offline frequency sweep measurements at the RUM. A timing slack measurement is made at a safe operating frequency f_{cal}, generally the STA $f_{\max}$ (f_{sta}). This measurement establishes the effective delay at the shadow register t_{dS}.

Algorithm 1 Shadow register delay offset calibration using shadow register-supported frequency sweep.

for all $i \in \{\text{RUMs}\}$ **do**
  set PLL frequency f_{cal}
  measure $t_{\mathrm{sS,i}}$
  $t_{\mathrm{dS,i}} \leftarrow 1/f_{\mathrm{cal}} - t_{\mathrm{sS,i}}$
  measure $f_{\max}$
  $f \leftarrow 0.95 f_{\max}$
  done $\leftarrow$ FALSE
  repeat
    set PLL frequency f
    measure $d_i(t_\phi)$ for $0 \leq t_\phi < 1/f$
    if $0 \in d_i(t_\phi)$ **then**
      $t_{\mathrm{dR,i}} \leftarrow 1/f - \Delta t_{\mathrm{cal}}/2$
    else
      done $\leftarrow$ TRUE
    end if
    $f \leftarrow \frac{1}{1/f - \Delta t_{\mathrm{cal}}}$
  until done
  $t_{\mathrm{RS,i}} \leftarrow t_{\mathrm{dR,i}} - t_{\mathrm{dS,i}}$
end for

The offline frequency sweep is conducted with shadow-register support and must traverse the maximum operating frequencies of all of the RUMs in the circuit. It works in reverse to normal OSM, with timing failure being caused in the RUM and detected by comparison to the shadow register. The frequency is stepped up from f_{cal} in increments corresponding to a period step of Δt_{cal} and shadow register discrepancy counts recorded for each frequency at all phase leads (t_ϕ) between 0 and $f1/f_{\mathrm{clk}}$. Any frequency at which the RUM is operating correctly will produce a $d_i(t_\phi) = 0$ for one or more steps of t_ϕ. The effective delay at the RUM t_{dR} is set to the reciprocal of the midpoint between the frequency at which $d_i(t_\phi) > 0$ for all t_ϕ and the preceding frequency step.

This offline shadow register-supported frequency sweep is similar to the error detection technique of Razor [25] and [67], except that t_ϕ is swept, instead of being static phase offset. This simplifies the implementation, as t_ϕ can scan to any point after all the edges from one clock cycle have arrived, but before the earliest edges from the next cycle. Without this

facility, complex timing constraints are needed to ensure that the shadow register samples clean data. The shadow path calibration offset, t_{RS}, is calculated from the difference of t_{dR} and t_{dS}. The slack at the RUM is found by subtracting the shadow register delay offset from the measured setup slack at the shadow register, $t_{\mathrm{sR}} = t_{\mathrm{sS}} - t_{\mathrm{RS}}$.

The calibration offset is subject to variation due to factors including self-heating, temperature and degradation. In current process technologies the effects of these are small, and a one-time calibration conducted at the time of commissioning is sufficient for accurate measurement over the circuit's lifetime. As the impacts of variability increase, this will no longer be adequate. Since each shadow register is typically located near its corresponding RUM and experiences the same data stimuli, it is expected that deviations in the shadow and RUM delays will be correlated. As such, calibration offset variation is proportional to the variation in slack measured at the shadow register, so such change can be used to compensate for the deviation in calibration offset. Alternatively, the circuit can be taken offline occasionally to re-calibrate.

3.2.4 Measurement Accuracy

The main limit to the accuracy of OSM is dictated by the resolution of the clock phase steps, $\Delta t_{\phi}(f)$. Other factors that affect it include: the accuracy of phase steps, the amount of clock jitter and noise. When calibration is used, the combined resolutions of the various measurements involved must be considered. An analysis of the resolution intervals for calibration is given in Table 3.2. These resolution intervals are dependent on the resolution or step size of the slack measurement at calibration frequency, delay measurement for the RUM and slack measurement during operation.

The step sizes achievable depend on the method by which the main and shadow clocks are generated. If the phase step size is a function of clock frequency, overall resolution can be improved by measuring the setup slack at the shadow register (t_{sS}) for calibration at a frequency that provides maximum phase resolution and correct circuit operation.

It should be noted that OSM can overestimate the amount of slack available by up to half a phase step. Using calibration can exacerbate this, potentially overestimating slack by up to half of one calibration frequency step, calibration phase step and measurement

Table 3.2: Resolution intervals for slack measurement quantities

Qty.	Derivation	Interval
Calibration		
$t_{\mathrm{sS,i}}$	Inferred from $d_i(t_\phi)$	$[-\Delta t_\phi(f_{\mathrm{cal}})/2, +\Delta t_\phi(f_{\mathrm{cal}})/2]$
$t_{\mathrm{dS,i}}$	$1/f_{\mathrm{cal}} - t_{\mathrm{sS,i}}$	$[-\Delta t_\phi(f_{\mathrm{cal}})/2, +\Delta t_\phi(f_{\mathrm{cal}})/2]$
$t_{\mathrm{dR,i}}$	Calibration frequency sweep	$[-\Delta t_{\mathrm{cal}}/2, +\Delta t_{\mathrm{cal}}/2]$
$t_{\mathrm{RS,i}}$	$t_{\mathrm{dR,i}} - t_{\mathrm{dS,i}}$	$[-(\Delta t_{\mathrm{cal}} + \Delta t_\phi(f_{\mathrm{cal}}))/2, +(\Delta t_{\mathrm{cal}} + \Delta t_\phi(f_{\mathrm{cal}}))/2]$
Measurement		
$t_{\mathrm{sS,i}}$	Inferred from $d_i(t_\phi)$	$[-\Delta t_\phi(f_{\mathrm{clk}})/2, +\Delta t_\phi(f_{\mathrm{clk}})/2]$
$t_{\mathrm{sR,i}}$	$t_{\mathrm{sS,i}} - t_{\mathrm{RS,i}}$	$[-(\Delta t_\phi(f_{\mathrm{cal}}) + \Delta t_\phi(f_{\mathrm{clk}}) + \Delta t_{\mathrm{cal}})/2, +(\Delta t_\phi(f_{\mathrm{cal}}) + \Delta t_\phi(f_{\mathrm{clk}}) + \Delta t_{\mathrm{cal}})/2]$

phase step ($(\Delta t_\phi(f_{\mathrm{cal}}) + \Delta t_\phi(f_{\mathrm{clk}}) + \Delta t_{\mathrm{cal}})/2$). This is not a necessarily problem if the technique is only used for measuring slack to monitor the health of a circuit, however, if the slack measurement is to control the voltage and/or frequency of the circuit, such overestimation could cause timing failure. See Section 5.3.1 for strategies used to avoid this underestimation.

3.2.5 Dithering

The resolution achievable by OSM is dependent on the size of phase steps $\Delta t_\phi(f)$ and in turn the shadow clock generator. Typically these steps are coarse when compared to the frequency steps used for offline measurement. One strategy for improving measurement resolution is to use a clock delay circuit to offset the shadow clock phase by half a phase step ($t_\phi/2$). Multiplexing in this clock delay circuit allows discrepancies to be sampled at phases half way between the normal measurements, effectively doubling the measurement resolution. Alternatively, a buffer can be multiplexed into the shadow path, increasing the delay and achieving the same result. In both cases, the multiplexing can be substituted for additional shadow registers with extra data or clock delay, simplifying implementation.

Where the above enhancements are not possible, frequency dithering allows the resolution of OSM to be improved by altering the clock frequency of the circuit by a small amount during operation. The increase in resolution that can be achieved using frequency dithering is limited to the granularity with which frequency can be varied and comes at the expense of a longer measurement time, but does not require any additional registers,

or data/clock delay to be added within the circuit.

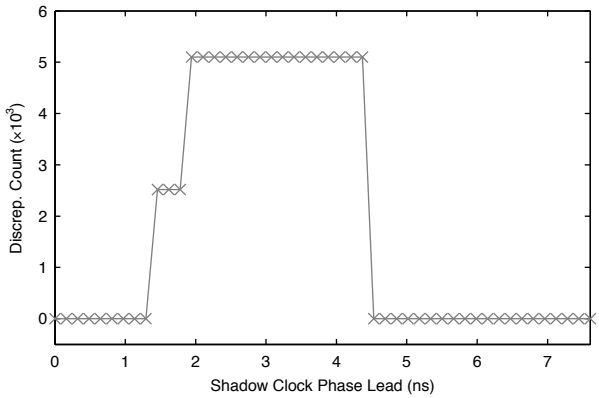
Assuming that Δt_ϕ is constant over all frequencies, the measurement resolution can be doubled by measuring the circuit at two frequencies. OSM is conducted at the operating frequency and upon completion, the frequency of the main and shadow clock is altered by an amount corresponding $\Delta t_\phi(f)/2$ in period. This offsets the OSM phase steps, measuring between each of the existing measurements.

Frequency dithering can be used for slack measurements of the shadow register both for calibration and for online measurement. A frequency sweep is required for calibration, which yields discrepancy counts at a range of frequencies. Dithering the frequency of the corresponding slack measurement at the shadow register drastically increases the accuracy of the calibration offset. When used for online measurement, dithering is only applicable to circuits which can tolerate small changes in operating frequency. The changes required can be less than 1% and may be within specification for the circuit or interface. Systems particularly suited to frequency dithering are loosely coupled, providing more freedom in varying the clock frequency.

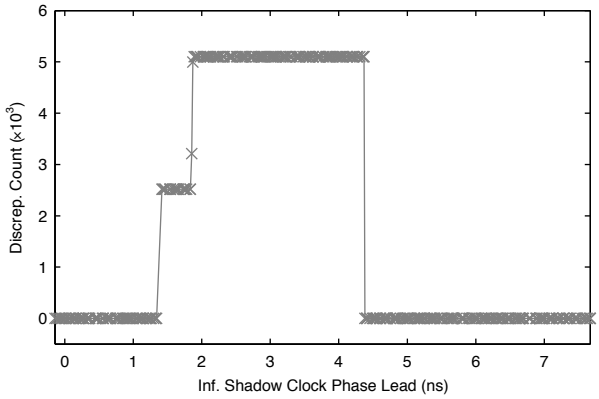
Since the slack measurements during dithering are for different frequencies, the slack measurement bounds are aggregated as inferred delays, t_{dS} calculated as $1/f_{\mathrm{clk}} - t_{\mathrm{sS}}$. The true slack lies between the inner bounds of this set and, for general-purpose usage, the midpoint between the upper and lower inner bound is used.

Depending on the clock generation mechanism, the phase step size, $\Delta t_\phi(f)$, may differ for each frequency, meaning that the additional measurements may not lie exactly in between samples at the nominal frequency. In this instance, the resolution achieved by dithering is non-constant and the selection of suitable frequencies to dither becomes complicated. The selection of optimal frequencies and clock generator parameters for frequency dithering is a topic for future work.

Figure 3.10 shows normal and dithered measurements for the `buffer` circuit. The discrepancy counts sampled for each phase lead are labelled with a cross marker; the increase in resolution provided by dithering is clearly visible.



(a) Normal



(b) Dithered

Figure 3.10: Normal and dithered uncalibrated discrepancy profile for `buffer` clocked at f_{sta} of 128.52 MHz.

3.2.6 RUM Selection

Registers with the least slack (or greatest effective delay) should be instrumented, as small changes in the delay of paths preceding these critical and near-critical registers will have the most detrimental impact on the functionality of the circuit as a whole. The registers can be identified from static timing analysis (STA) of the circuit.

A metric known as Critical Delay Margin (CDM) is used to choose these registers using information from STA. They are selected if they fulfil the condition in Equation 3.2, where $t_{\mathrm{dR,i,STA}}$ is the effective delay at the candidate sink register, i, computed from the timing model, $t_{\mathrm{aC,STA}}$ the maximum effective delay in the circuit (the critical path) and CDM the Critical Delay Margin from 0% to 100%. An example of RUM selection is given in Section 3.3.5.

$$t_{\mathrm{dR,i,STA}} \geq (1 - \mathrm{CDM}) t_{\mathrm{aC,STA}} \tag{3.2}$$

The degree of coverage required is a parameter to be chosen by the circuit designer; the decision is dependent on the variation in delay between the timing model and the circuit, both at commissioning and during the lifetime of the device. At commissioning the circuit is affected by process variation. The impact of inter-die variation is negligible, since all of the device is affected the relative ordering of critical paths identified from STA remains the same. Intra-die process variation results in different parts of the device having different performance, altering which registers is critical and near-critical and introducing inaccuracy between real and STA orderings.

The effect of degradation is more complex, if it is assumed that all paths experience the same relative changes in delay then only the slowest path at commissioning would need be instrumented, since it would always remain the critical register. Given the variation observed in degradation rates [59] this approach would not be valid. Making the assumption that the worst-case relative change in delay can affect any path independently, a coverage can be chosen to ensure that any path which could become critical when subjected to this delay increase is monitored. This is pessimistic since it makes the assumption that when subjected to degradation the fastest path monitored could slow down to become critical,

while all the slower paths monitored remain sub-critical.

A knowledge of the range of relative intra-die variability for a given device can provide insight into a suitable CDM. The relative intra-die variability dictates how the ordering of critical registers will change between timing analysis, commissioning and the circuit's lifetime. Knowing that intra-die variability can result in a given path being 2.5% faster, or 7.5% slower than STA estimates, and that the critical path has an effective delay of 10 ns allows us to compute its minimum effective delay under variation, $(10\text{ns} * (1 - 0.025) =)$ 9.75 ns. All registers in the circuit with an effective delay meeting or exceeding this when subjected to variation must be instrumented. These registers have a STA estimated delay greater than or equal to $(9.75\text{ns}/(1 + 0.075) =)$ 9.07 ns. This is achieved by a CDM of $(1 - (9.07\text{ns}/10\text{ns}) =)$ 9.3%. This is generalised in Equation 3.3, where $\Delta t_{\text{dR,fast}}$ is the percentage intra-die variability at the fast corner, and $\Delta t_{\text{dR,slow}}$ the slow corner.

$$CDM = 1 - \frac{1 - \Delta t_{\text{dR,fast}}}{1 + \Delta t_{\text{dR,slow}}} \tag{3.3}$$

Currently STA does not expose the different granularities of variability. Using the slow and fast timing information generally provided by STA does allow for the computation of a CDM, but as this includes inter-die variability the coverage is excessive, typically approximately 50%.

3.3 Measurement Experiment

In order to establish the effectiveness of the OSM method, measurement experiments were conducted on a Field-Programmable Gate Array (FPGA). Using an FPGA allows for experiments to be performed on real hardware, without the time and expense of producing test chips.

The experiments are conducted on a thermally-controlled FPGA with an adjustable core voltage and power measurement. The FPGA used is an Altera Cyclone IV EP4CE22F17C6 with 22,000 Logic Elements (LEs), a 65 nm device produced on a low-power process. This FPGA is mounted on a DE0-nano development board manufactured by Terasic.

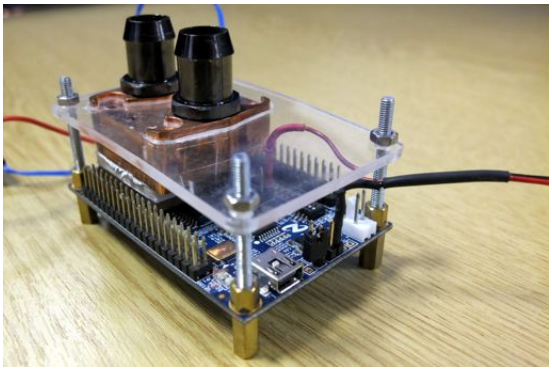
3.3.1 Temperature Control

In order to perform controlled experiments, the temperature of the FPGA must be regulated. A system has been developed that allows the temperature of the FPGA package to be controlled with an accuracy of <0.1 °C and a high slew rate without affecting the temperature of the surrounding components. A thermoelectric heat pump is mounted against the top of the FPGA package. The package of the Cyclone IV EP4CE22F17C6 is smaller than the thermoelectric device, so a stepped copper heat-spreader block is placed between them, this avoids some tall components surrounding the FPGA. The temperature sensor, a PT100 resistance thermometer, is clamped in a recess in the heat-spreader and is used to measure the FPGA's temperature. The heat dissipated by the thermoelectric device is removed using a water cooling system. A water-block is situated on the opposite side of the Peltier to the FPGA and coolant is pumped to a radiator where heat dissipated into the environment. The entire assembly is clamped onto the FPGA using a laser-cut acrylic plate, machined to index with the complex upper face of the water-block and fastened to standoffs at the corners of the development board.

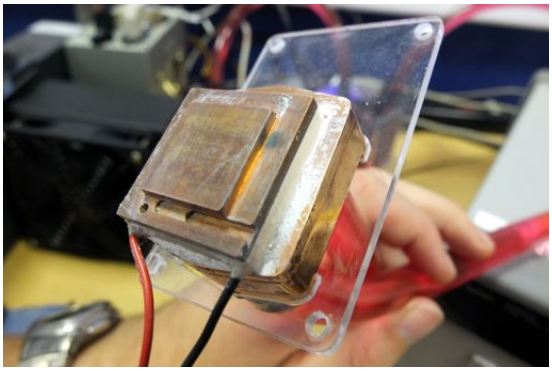
The temperature sensor's resistance is measured using a 22-bit ADC with automatic internal offset and gain calibration. This interfaces with a Atmel AVR microcontroller, on an Ardunio development board, over an SPI bus. The heat pump is driven by a H-bridge with a high frequency PWM signal. A PID controller is used to control the heat pump to a temperature setpoint. The microcontroller connects to a test computer, using a USB interface, which can play back pre-defined temperature schedules or maintain the FPGA at a fixed temperature.

The temperature control system is able to cool the FPGA significantly below the lower temperature specification (0°C) and above the higher (85°C), allowing for experiments spanning these corners.

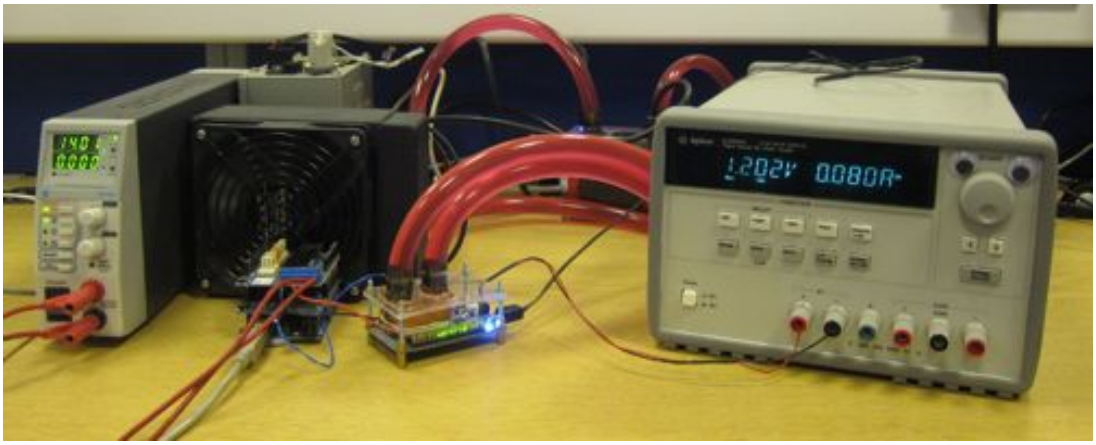
Unlike high-end FPGAs, the Cyclone family lack die temperature sensors and as such it is not possible to measure the thermal resistance between the FPGA die and heat-spreader. The resulting lag in temperature is managed in static temperature experiments by allowing the temperature of the package to equalise before conducting the experiment. During experiments where temperature is varied, the slew is slow, replicating changes in



(a) Temperature controlled FPGA assembly.



(b) Heat-spreader sandwich.



(c) Entire experiment showing from left: temperature controller PSU; temperature controller and radiator; FPGA; FPGA core PSU.

Figure 3.11: Temperature and voltage controlled FPGA experimental hardware.

ambient temperature, and so the effect of this lag is expected to be minimal. Should it be desirable to conduct high slew experiments, this thermal resistance could be accounted for using datasheet values, or the FPGA substituted for a family which includes a die temperature sensor, such as the Stratix family from Altera.

3.3.2 Voltage Control and Power Measurement

Powering the reconfigurable core of the FPGA from an external supply allows the voltage to be easily varied and power consumption measured. A precision programmable power supply is used, which is interfaced with the test computer and configured as required during an experiment. The FPGA development board has been modified to isolate the on-board core power supply, which is supplied by the programmable power supply instead.

Experimentation has shown that this FPGA will operated down to 0.9 V from the 1.2 V nominal. Below this level failure of the power-on reset, configuration and PLL circuitry prevents operation.

3.3.3 Clock Generation

The Phase-Locked Loop (PLL) clock generators common to current generation FPGAs is able to generate the necessary main and variable phase shadow clocks. Both the frequency and phase relationship of the clocks can be configured during device run-time and only a single PLL is required as both clocks have the same frequency. The size of the phase steps, $\Delta(t_\phi)$, is a function of the frequency configuration of the PLL and in the Cyclone IV, varies between 96 ps and 208 ps. A full discussion on the use of PLLs for OSM is in Section 4.2.1.

3.3.4 Benchmark Circuits

Two simple benchmark circuits were used for the experiments: the buffer chain (as described earlier) and a 64-bit unsigned ripple-carry adder (`intadd64`). The one sink register in `buffer` was instrumented and `intadd64` was instrumented to a CDM of 10%, as described in the following section. Input vectors for `intadd64` were generated by a Linear Feedback Shift Register (LFSR). Each sensor was connected to a discrepancy counter,

allowing full delay information to be collected.

3.3.5 Benchmark Instrumentation

The benchmark circuit, `intadd64`, was compiled using an Altera Quartus II without any monitoring hardware, and static timing analysis conducted in order to determine the criticality of sink registers. The timing report for the most timing critical 32 sink registers is shown on the left of Table 3.3. It gives the slack for each of these registers ($t_{\mathrm{sR,i,STA}}$) as well as source, clock skew and data delay. The right hand side of the table shows the effective delay ($t_{\mathrm{dR,i,STA}}$) for each of the sink registers, which has been computed as $1/f_{\mathrm{clk}} - t_{\mathrm{sR,i,STA}}$, with f_{clk} of 200 MHz. The critical effective delay (t_{dC}) is therefore (5.000ns $-$ $-$0.244ns $=$) 5.244 ns.

The inclusive CDM for each of these sink registers can now be found. This is the minimum CDM for which the sink register will be instrumented for OSM. The inclusive CDM is computed as $1 - t_{\mathrm{aC}}/t_aR, i, STA$ for each sink register. In order to instrument a circuit to a given CDM, sink registers with an inclusive CDM of less than or equal to this must be instrumented. In this case a CDM of 10% is specified, so all registers from intadd[63] down to intadd[54] are selected to be instrumented.

The OSM sensors can now be added to the selected sink registers in the circuit netlist and the circuit recompiled, constraining the placement and routing of the original circuit. Compiling the circuit incrementally in this manner allows priority to be given to the benchmark/application circuit. In the first stage, the compiler efforts are directed to optimising the circuit, thereafter, sink registers are selected and OSM sensors and associated circuitry added, minimising the impact on the performance of the application circuit.

Table 3.4 gives STA timing information for the instrumented sink registers (RUMs) and shadow registers. The first column shows the effective delay estimates for the RUMs before the addition of OSM. The next two columns give the effective delay for the instrumented RUMS and shadow registers. Instrumenting the RUMs using the described approach results in a negligible increase in RUM delay, as shown in the fifth column. The last column shows the difference in effective delay between the instrumented RUMs and shadow registers. Giving priority to the application circuit means that the shadow register effective delay

Table 3.3: STA timing report for `intadd64`, with an f_{clk} specification of 200 MHz. The right hand side shows the computed effective delay and inclusive CDM, the minimum CDM that the sink register will be instrumented.

Slack (ns)	Sink Reg.	Source Reg.	Clock Skew (ns)	Data Delay (ns)	Eff. Delay (ns)	Inc. CDM
-0.244	intadd[63]	b[0]	-0.084	5.155	5.244	0.0%
-0.230	intadd[62]	a[1]	-0.084	5.141	5.230	0.0%
-0.128	intadd[61]	b[0]	-0.084	5.039	5.128	2.0%
-0.114	intadd[60]	a[1]	-0.084	5.025	5.114	2.2%
-0.012	intadd[59]	b[0]	-0.084	4.923	5.012	4.2%
0.002	intadd[58]	a[1]	-0.084	4.909	4.998	4.4%
0.104	intadd[57]	b[0]	-0.084	4.807	4.896	6.4%
0.118	intadd[56]	a[1]	-0.084	4.793	4.882	6.7%
0.220	intadd[55]	b[0]	-0.084	4.691	4.780	8.6%
0.234	intadd[54]	a[1]	-0.084	4.677	4.766	8.9%
0.336	intadd[53]	b[0]	-0.084	4.575	4.664	10.8%
0.350	intadd[52]	a[1]	-0.084	4.561	4.650	11.1%
0.452	intadd[51]	b[0]	-0.084	4.459	4.548	13.0%
0.466	intadd[50]	a[1]	-0.084	4.445	4.534	13.3%
0.568	intadd[49]	b[0]	-0.084	4.343	4.432	15.3%
0.582	intadd[48]	a[1]	-0.084	4.329	4.418	15.5%
0.707	intadd[47]	b[0]	-0.061	4.227	4.293	17.9%
0.721	intadd[46]	a[1]	-0.061	4.213	4.279	18.2%
0.823	intadd[45]	b[0]	-0.061	4.111	4.177	20.1%
0.837	intadd[44]	a[1]	-0.061	4.097	4.163	20.4%
0.939	intadd[43]	b[0]	-0.061	3.995	4.061	22.4%
0.953	intadd[42]	a[1]	-0.061	3.981	4.047	22.6%
1.055	intadd[41]	b[0]	-0.061	3.879	3.945	24.6%
1.069	intadd[40]	a[1]	-0.061	3.865	3.931	24.8%
1.171	intadd[39]	b[0]	-0.061	3.763	3.829	26.8%
1.185	intadd[38]	a[1]	-0.061	3.749	3.815	27.1%
1.287	intadd[37]	b[0]	-0.061	3.647	3.713	29.0%
1.301	intadd[36]	a[1]	-0.061	3.633	3.699	29.3%
1.403	intadd[35]	b[0]	-0.061	3.531	3.597	31.2%
1.417	intadd[34]	a[1]	-0.061	3.517	3.583	31.5%
1.519	intadd[33]	b[0]	-0.061	3.415	3.481	33.4%
1.533	intadd[32]	a[1]	-0.061	3.401	3.467	33.7%
1.635	intadd[31]	b[0]	-0.061	3.299	3.365	35.7%

Table 3.4: STA estimates for the RUMs and shadow registers of `intadd64` before and after instrumenting.

RUM	Orig.	Instrumented		Delay Diff.	
	RUM Delay (ns)	RUM Delay (ns)	Shadow Delay (ns)	RUM (ns)	Shadow (ns)
intadd[63]	5.244	5.243	5.942	0.001	0.699
intadd[62]	5.230	5.230	6.117	0.000	0.887
intadd[61]	5.128	5.127	6.028	0.001	0.901
intadd[60]	5.114	5.114	6.179	0.000	1.065
intadd[59]	5.012	5.011	5.894	0.001	0.883
intadd[58]	4.998	4.998	5.672	0.000	0.674
intadd[57]	4.896	4.895	5.784	0.001	0.889
intadd[56]	4.882	4.882	5.810	0.000	0.928
intadd[55]	4.780	4.779	5.479	0.001	0.700
intadd[54]	4.766	4.766	5.467	0.000	0.701

is typically greater than that of the RUM. In circuits larger than the `intadd64`, larger shadow delay offsets would be expected as the shadow register would be placed further from the application circuit due to reduced spare resources.

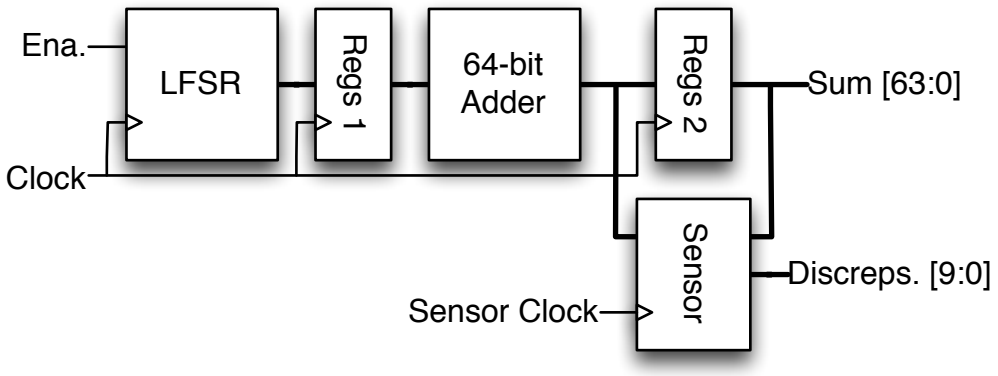


Figure 3.12: A 64-bit unsigned adder instrumented with OSM.

The instrumented `intadd64` benchmark circuit is shown in Figure 3.12, with the inputs to the adder driven by a 128-bit LFSR, and the OSM sensors connected to the 10 most significant output registers.

3.3.6 Offline Measurement

In order to evaluate the accuracy of the OSM method, it must be compared to an existing, proven technique. Offline frequency sweep based delay measurement is an accurate and proven method and thus suitable as a golden standard for comparison. The shadow register-supported measurement method described previously for calibration allows highly accurate golden measurements of the RUMs to be made without any additional circuitry.

Table 3.5 gives the STA estimates and offline frequency sweep measurements for the RUMs of `intadd64`. The STA estimates for delay for the RUMs are always greater than the measured effective delay because of the pessimism inherent in the timing model, which must guarantee that the circuit functions in the worst-case corner, despite the circuit being operated in nominal conditions during the experiment.

Intuitively, the delay of each ripple-carry adder output is expected to increase monotonically with its significance, but this is not the case in these measurements. Three of the outputs have a measured delay that is less than their preceding bit by between 0.01 ns and

Table 3.5: STA estimates and offline frequency sweep measurements for RUMs in `intadd64`.

RUM	RUM Delay		Delay
	STA (ns)	Measured (ns)	Diff. (ns)
intadd[63]	5.243	4.01	1.23
intadd[62]	5.230	4.04	1.19
intadd[61]	5.127	3.93	1.20
intadd[60]	5.114	3.93	1.18
intadd[59]	5.011	3.83	1.18
intadd[58]	4.998	3.85	1.15
intadd[57]	4.895	3.76	1.14
intadd[56]	4.882	3.77	1.11
intadd[55]	4.779	3.69	1.09
intadd[54]	4.766	3.68	1.09

0.03 ns, or 0.3% to 0.7% relative to the overall delay of the circuit. While the ripple-carry adder contributes the majority of the delay in `intadd64`, the source and sink registers and associated routing can effect changes in delay of this magnitude. The changes in the ordering of criticality are likely the result of variations of the low-level implementation of the circuit, or intra-die process variation. Similar variation in ordering would be expected if the circuit were assigned to different regions in the same device, or a different FPGA.

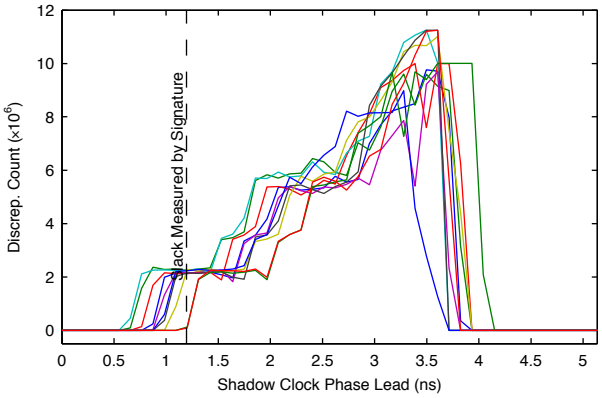
3.3.7 Online Slack Measurement

Measurements were conducted with the instrumented benchmark circuits running at f_{sta} and the FPGA powered at its nominal voltage (V_{nom}) of 1.2 V. The package was controlled to an ambient temperature of 27 °C.

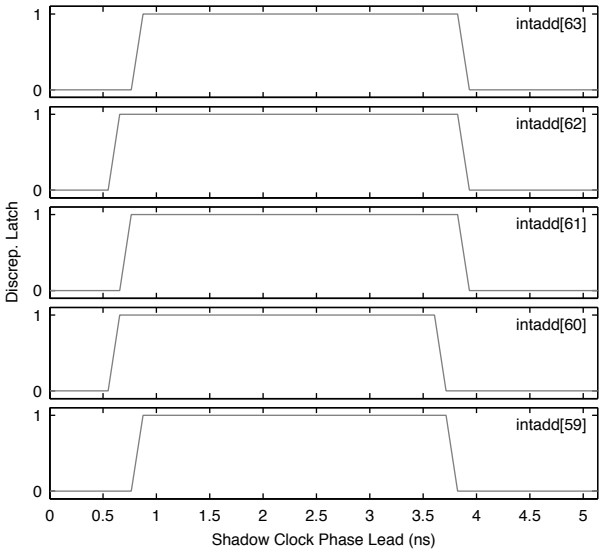
Plots of the measurements are shown in Figures 3.9 and 3.13 for the `buffer` and `intadd64` respectively. The discrepancy profile for the adder circuit (Figure 3.13a) shows considerable complexity as compared to the buffer chain (Figure 3.9b). As the significance of the ripple-carry adder's output increases, it depends on more signal paths than less significant outputs (including the carry output of the previous stage). It would be expected that more significant outputs would have more complex discrepancy profiles, but this is not particularly apparent in Figure 3.13a.

Given that the discrepancy profile is a cumulative histogram of data delay, it would be presumed that the number of discrepancies would rise monotonically with increasing

shadow clock phase lead (until the blind spot) as in Figure 3.9b. Figure 3.13a does not exhibit this behaviour. As the shadow clock phase lead is increased, timing failure occurs with greater frequency. In order for this timing failure to be detected as a discrepancy, the RUM and shadow registers must latch different values, but this does not always occur. The shadow register may sample a glitch, or experience metastability that resolves to the same value as in the RUM. These false negatives result in the non-monotonic behaviour observed, but do not affect measurement, as only one discrepancy is required. Certain shadow clock phase leads may result in more false negatives due to the number and behaviour of paths failing timing.



(a) Discrepancy profile



(b) Discrepancy latch plot

Figure 3.13: Uncalibrated discrepancy profile and discrepancy latch plot for the OSM instrumented `intadd64` circuit running at f_{sta} (190.69 MHz), with slack measured by Signature Analysis.

Table 3.6 explores the accuracy of the OSM method. Measurements were made at each shadow register and compared to the high accuracy offline measurements of the RUM as described in Section 3.3.6. In order to compare slack measurements to the RUM delays, these are converted to inferred slack at the frequency at which the circuit was clocked during OSM, f_{sta}, 190.69 MHz. The phase step size ($\Delta t_{\phi}(f_{\text{clk}})$) at this operating frequency was 109 ps.

There is large difference evident between the slack inferred at the RUM and measured using OSM at the shadow register, with a mean error of 0.55 ns. This disparity is due to the difference in path length to the two registers, as indicated by the STA estimates in Table 3.4. The "Crit." (critical) row gives the measurements that are worst-case for timing, as it is these that are timing critical and limit the overall performance of the circuit. These online slack measurements may be sufficient for tracking changes in delay (i.e. for degradation monitoring), but lack absolute-accuracy.

Table 3.6: Online slack and shadow register-supported offline frequency sweep measurement for `intadd64`.

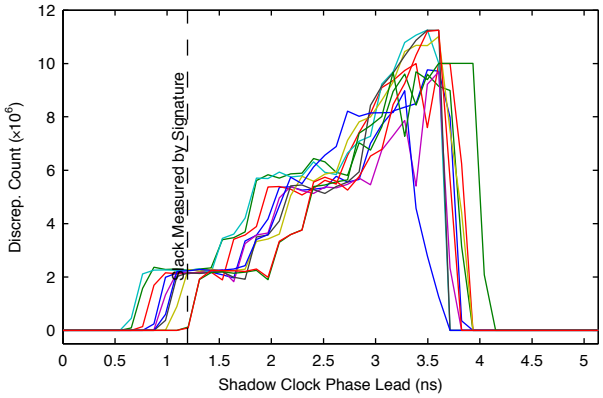
RUM	RUM Delay (ns)	Inf. Slack (ns)	OSM Slack (ns)	Error (ns)	Error (%)
intadd[63]	4.01	1.23	0.8	0.43	35.0%
intadd[62]	4.04	1.20	0.6	0.60	50.0%
intadd[61]	3.93	1.31	0.7	0.61	46.6%
intadd[60]	3.93	1.31	0.6	0.71	54.2%
intadd[59]	3.83	1.41	0.8	0.61	43.3%
intadd[58]	3.85	1.39	0.9	0.49	35.3%
intadd[57]	3.76	1.48	0.9	0.58	39.2%
intadd[56]	3.77	1.47	0.9	0.57	38.8%
intadd[55]	3.69	1.55	1.1	0.45	29.0%
intadd[54]	3.68	1.56	1.1	0.46	29.5%
Crit.	4.04	1.20	0.6	0.60	50.0%
Mean				0.55	40.1%
Max				0.71	54.2%

Calibration

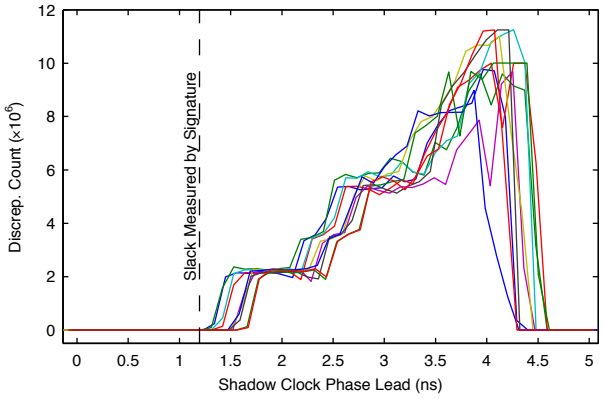
The measurements were calibrated using the presented technique. An offline shadow register-supported frequency sweep was used to find the delay to the RUMs and the delay at the shadow register was inferred by OSM from measurements at 245.56 MHz. This

frequency is lower than the minimum operating frequency for RUMs, $f_{\max}$, (so that the circuit is operating correctly) and provided the lowest phase step size, and thus slack measurement resolution, a $\Delta t_{\phi}(f_{\text{cal}})$ of 102 ps. The frequency sweep was conducted with a period step (Δt_{cal}) of 10 ps, resulting in a combined measurement resolution of ± 111 ps.

A discrepancy profile plot for uncalibrated and calibrated measurements of `intadd64` is shown in Figure 3.14. When compared to Figure 3.14a, the discrepancy profile of Figure 3.14b is translated to the right, since the amount of slack available at the RUM is greater than that at the shadow register and calibration has corrected for the difference in path length.



(a) Uncalibrated



(b) Calibrated

Figure 3.14: Uncalibrated and calibrated discrepancy profiles for `intadd64` operating at f_{sta} (190.69 MHz), with slack measured by Signature Analysis.

The calibrated slack measurements for the circuit are in Table 3.7. Calibration has reduced the mean absolute error from 0.55 ns to 0.07 ns. The maximum error when

comparing calibrated online slack measurement to the high accuracy offline measurements is 0.13 ns, slightly greater than the measurement resolution, the difference being the result of measurement noise or inaccuracy in the offline frequency sweep used for calibration.

Table 3.7: Calibrated online slack and shadow register-supported offline frequency sweep measurement for `intadd64`.

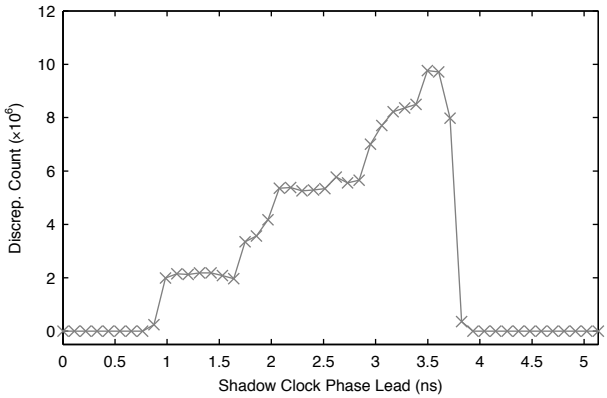
RUM	RUM Delay (ns)	Inf. Slack (ns)	OSM Slack (ns)	Cal. Offset (ns)	Cal. Slack (ns)	Error (ns)	Error (%)
intadd[63]	4.01	1.23	0.8	-0.5	1.3	0.07	5.7%
intadd[62]	4.04	1.20	0.6	-0.7	1.3	0.10	8.3%
intadd[61]	3.93	1.31	0.7	-0.7	1.4	0.09	6.9%
intadd[60]	3.93	1.31	0.6	-0.8	1.4	0.09	6.9%
intadd[59]	3.83	1.41	0.8	-0.6	1.4	0.01	0.7%
intadd[58]	3.85	1.39	0.9	-0.5	1.4	0.01	0.7%
intadd[57]	3.76	1.48	0.9	-0.7	1.6	0.12	8.1%
intadd[56]	3.77	1.47	0.9	-0.7	1.6	0.13	8.8%
intadd[55]	3.69	1.55	1.1	-0.4	1.5	0.05	3.2%
intadd[54]	3.68	1.56	1.1	-0.4	1.5	0.06	3.8%
Crit.	4.04	1.20	0.6		1.3	0.10	8.3%
Mean						0.07	5.3%
Max						0.13	8.8%

Dithering

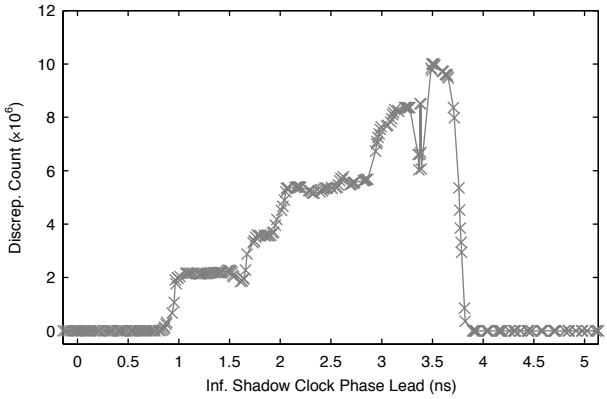
Dithering was used to improve the resolution of both calibration and measurement. For calibration, measurements were aggregated for all frequencies below the minimum $f_{\max}$ for RUMs. These measurements must be taken to establish the RUM delays, so there is no overhead to this dithering. During OSM the circuit was dithered between the nominal operating frequency of 190.69 MHz (f_{sta}) and 186.42 MHz in frequency steps corresponding to a 24 ps change in period, yielding measurements at six different frequencies.

Figure 3.15 shows normal and dithered uncalibrated discrepancy profiles for the `intadd64` circuit. In order to combine measurements an inferred slack is found for each phase lead at each frequency. The increase in measurement resolution is obvious with the discrepancy profile showing a greater number of samples and revealing significantly more detail than OSM without dithering.

Measurement results for dithering of calibration are displayed in Table 3.8. Since the phase step size is not constant for these measurements, the resolution is variable. Here,



(a) Normal



(b) Dithered

Figure 3.15: Normal and dithered uncalibrated discrepancy profile for the intadd[63] RUM in `intadd64`, clocked at f_{sta} of 190.69 MHz.

an average resolution 5 ps is achieved, but with a worst-case of 65 ps, still a significant improvement over the 101 ps resolution before dithering. Dithering improves the resolution of calibration, reducing the maximum absolute slack measurement error to 0.13 ns.

Table 3.8: Measurements showing calibration offset resolution improvement through frequency dithering.

RUM	RUM Delay (ns)	Inf. Slack (ns)	OSM Slack (ns)	Cal. Offset (ns)	Cal. Slack (ns)	Error (ns)	Error (%)
intadd[63]	4.01	1.23	0.8	-0.50	1.3	0.07	5.7%
intadd[62]	4.04	1.20	0.6	-0.66	1.3	0.10	8.3%
intadd[61]	3.93	1.31	0.7	-0.66	1.4	0.09	6.9%
intadd[60]	3.93	1.31	0.6	-0.80	1.4	0.09	6.9%
intadd[59]	3.83	1.41	0.8	-0.64	1.4	0.01	0.7%
intadd[58]	3.85	1.39	0.9	-0.48	1.4	0.01	0.7%
intadd[57]	3.76	1.48	0.9	-0.63	1.5	0.02	1.4%
intadd[56]	3.77	1.47	0.9	-0.65	1.6	0.13	8.8%
intadd[55]	3.69	1.55	1.1	-0.47	1.6	0.05	3.2%
intadd[54]	3.68	1.56	1.1	-0.48	1.6	0.04	2.6%
Crit.	4.04	1.20	0.6		1.3	0.10	8.3%
Mean						0.06	4.5%
Max						0.13	8.8%

Dithering both calibration and measurement, where this is possible, achieves maximum measurement accuracy. Fully dithered measurements for `intadd64` are in Table 3.9. Dithering between the six operating frequencies reduces the 109 ps nominal resolution to 18 ps on average and a worst-case of 107 ps. The maximum error is reduced to 0.07 ns, almost half of that achieved without dithering and could be further reduced with careful selection of dithering frequencies.

3.3.8 Slack Variation

Experiments were conducted in order to show that online timing slack measurements can track changes in slack due to external factors. Temperature, voltage and operating frequency all have an affect on the slack available, either through variation of circuit delay or clock period. Unless varied, the FPGA was powered at nominal voltage (1.2 V) and clocked at the f_{sta} of 190.69 MHz. All measurements are uncalibrated and not dithered.

Altering the operating frequency has a direct impact on slack. Figure 3.16 shows uncalibrated slack measurements for the 10 RUMs of `intadd64` operating at a range of

Table 3.9: Measurements showing resolution improvement with both calibration offset and OSM dithering.

RUM	RUM Delay (ns)	Inf. Slack (ns)	OSM Slack (ns)	Cal. Offset (ns)	Cal. Slack (ns)	Error (ns)	Error (%)
intadd[63]	4.01	1.23	0.79	-0.50	1.29	0.06	4.9%
intadd[62]	4.04	1.20	0.58	-0.66	1.24	0.04	3.3%
intadd[61]	3.93	1.31	0.67	-0.66	1.33	0.02	1.5%
intadd[60]	3.93	1.31	0.56	-0.80	1.36	0.05	3.8%
intadd[59]	3.83	1.41	0.83	-0.64	1.47	0.06	4.3%
intadd[58]	3.85	1.39	0.96	-0.48	1.44	0.05	3.6%
intadd[57]	3.76	1.48	0.92	-0.63	1.55	0.07	4.7%
intadd[56]	3.77	1.47	0.88	-0.65	1.53	0.06	4.1%
intadd[55]	3.69	1.55	1.13	-0.47	1.60	0.05	3.2%
intadd[54]	3.68	1.56	1.13	-0.48	1.61	0.05	3.2%
Crit.	4.04	1.20	0.6		1.24	0.04	3.3%
Mean						0.05	3.7%
Max						0.07	4.9%

frequencies (show as clock periods). Decreasing the period decreases slack as expected given their equivalence, demonstrated by the linear relationship with gradient of approximately one.

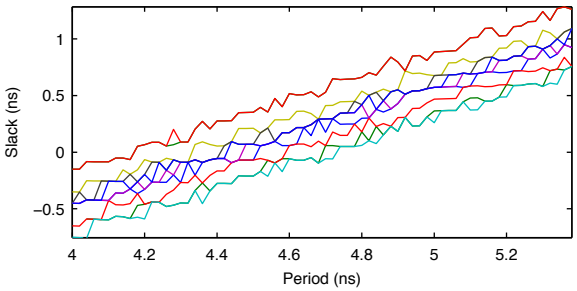


Figure 3.16: Uncalibrated slack measurement response to clock period variation in `intadd64`.

The relationship between slack and temperature is explored in Figure 3.17. The temperature of the FPGA's package was varied across the device corners, from 0 °C to 85 °C. As expected, slack decreases as the circuit slows down with increasing temperature. The total change is fairly modest with the critical path slowing by around 370 ps, corresponding to 4.4 ps/°C. At this scale the quantisation caused by the measurement resolution is quite apparent.

Voltage has a dramatic influence on timing slack, as demonstrated in Figure 3.18, which shows uncalibrated slack measurements. Once the voltage has dropped below 1.15 V all

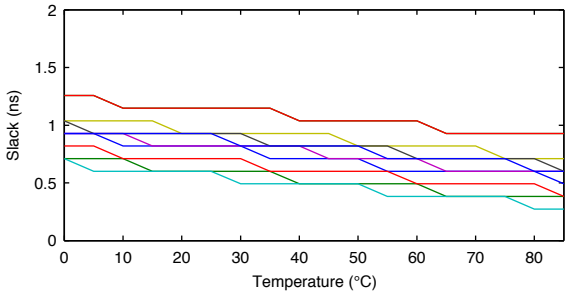


Figure 3.17: Uncalibrated slack measurement response to temperature variation in `intadd64`.

of the slack at the most critical RUMs has been eroded. Its large effect has an impact on calibration; the difference in path length to the main and shadow register increases as voltage is reduced, with the longer path becoming proportionally slower than the short.

Figure 3.19 shows the behaviour of the calibration offsets for the 10 RUMs in `intadd64` as the voltage is varied between 0.9 V and the nominal operating voltage 1.2 V. The changes in voltage due to power supply ripple or noise are too small to be resolved. If techniques such as Dynamic Voltage Scaling are to be employed using OSM the effect of voltage on calibration offsets must be accounted for.

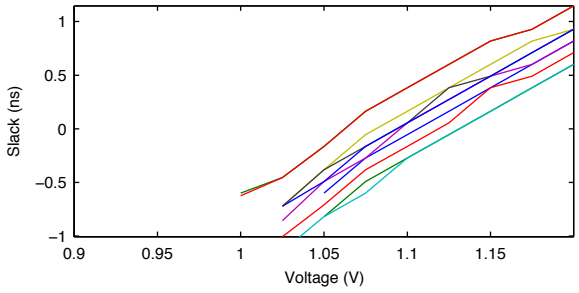


Figure 3.18: Uncalibrated slack measurement response to voltage variation in `intadd64`.

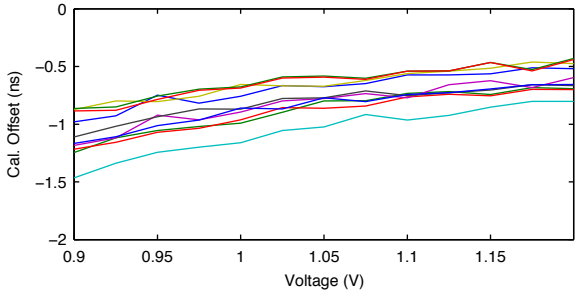


Figure 3.19: Calibration offset variation due to voltage in `intadd64`.

3.4 Overhead

The overhead of implementing OSM can be separated into a fixed cost for the measurement aggregation and control circuitry and a variable cost for the shadow registers and associated circuitry, which depends upon the number of registers monitored. The choice and number of these registers in turn depends on the CDM coverage required and the delay distribution of the circuit. In order to explore the trade-off between hardware overhead and circuit coverage, an experiment has been conducted profiling two sets of benchmark circuits.

The circuits, some of which are combinatorial, were wrapped with registers and compiled in Altera Quartus II for the Cyclone IV architecture, using default options. Timing analysis was performed and registers selected for instrumenting at various CDMs. Instrumenting a register requires the addition of two registers (shadow and discrepancy) and an XOR gate.

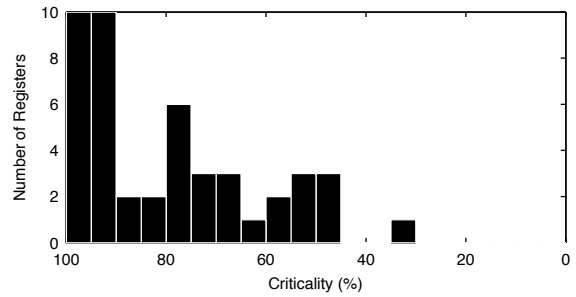
3.4.1 "Toronto 20"

The first set of benchmarks profiled is "Toronto 20" (T20) [49], the twenty large MCNC benchmark circuits [71], commonly used for compiler research in FPGAs. These circuits range in size consisting of from around 1000 to 8000 lookup tables (LUTs) and 17 to 1600 registers. While they are small compared to state of the art circuits currently implemented on FPGAs, they provide a useful range of sizes and, crucially, delay distributions.

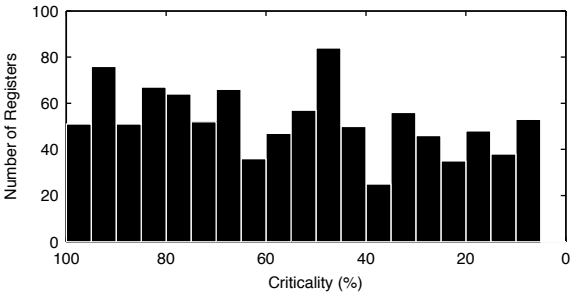
The circuits are supplied in a NET netlist format and were converted to VQM netlist for compilation in Quartus using the NETtoVQM utility supplied in Altera's Quartus University Interface Program (QUIP) [3].

Figure 3.20 shows histograms of delay to the register (as a percentage of the critical delay in the circuit) for three of the T20 benchmarks, illustrating three possible contrasting distribution profiles. `spla` demonstrates a critical-biased distribution, which requires that many registers in the design need to be instrumented to provide a low CDM coverage. `frisc` exhibits a relatively flat delay distribution, meaning that the number of registers which needs to be instrumented grows fairly linearly with CDM and `tesng`'s distribution is biased to the non-critical registers, so there is a low cost of instrumenting the circuit to a moderate CDM.

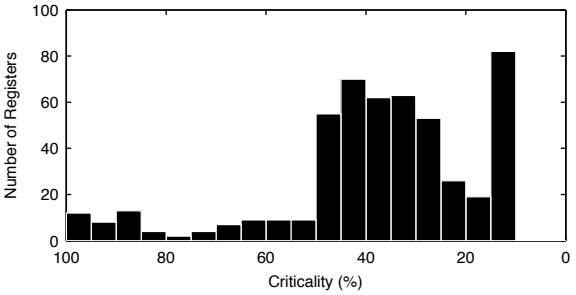
The total number of registers required for a given CDM coverage for the three T20



(a) `spla`



(b) `frisc`



(c) `tseng`

Figure 3.20: Register delay distributions (as a percentage of the critical delay) for three T20 benchmarks.

benchmarks is illustrated in Figure 3.21. This cumulative distribution shows how the cost for each of these different delay distributions grows with CDM, with `spla` rising quickly and levelling off thereafter, `tseng` doing the opposite, and `frisc` rising steady across the entire range of CDMs.

The number of registers instrumented is never 100%, as without information on circuit interfaces, source registers at the very input to the circuit are not instrumented.

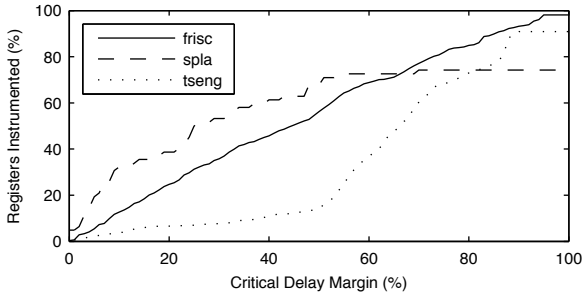


Figure 3.21: Cumulative histogram showing the number of registers that need to be monitored in order to achieve a given CDM, as a percentage of the total number of registers in the circuit.

A detailed breakdown of the T20 benchmark set is given in Table 3.10. The number of registers which require instrumenting to achieve a given CDM are presented. The maximum number of registers that are required to instrument this set of benchmarks to 10% CDM is 137, or 10% of the registers in the circuit. Circuits which require a high percentage of registers to be instrumented typically have a very low number of registers, e.g. `apex4`, which contains only 27 registers and requires that 13 of these (48%) are instrumenting for 10% CDM, representing only 0.84% of the total circuit area. Circuits in which the number of registers instrumented correspond to a larger percentage of the combined area, such as `elliptic`, are generally register rich. The cost of instrumenting circuits for OSM therefore depends on the delay distribution of the circuit and the proportion of logic and registers that it contains.

3.4.2 "Functional"

The T20 benchmarks transpire to not be suitable for the experiments conducted later in this thesis because they are largely synthetic circuits with unknown functionality, some of which are entirely combinatorial. They have been included as they are historically the

Table 3.10: The number of registers that require instrumenting for various Critical Delay Margins in the T20 benchmark set. For the minimum, mean and maximum, the percentages (in brackets) are treated independently.

Benchmark	Base Resources	5% CDM		10% CDM		15% CDM		20% CDM	
	Regs.	RUMs							
alu4	22	3	(14%)	4	(18%)	6	(27%)	6	(27%)
apex2	41	2	(4.9%)	2	(4.9%)	2	(4.9%)	2	(4.9%)
apex4	27	5	(19%)	13	(48%)	17	(63%)	17	(63%)
bigkey	649	7	(1.1%)	54	(8.3%)	139	(21%)	217	(33%)
clma	162	2	(1.2%)	3	(1.9%)	5	(3.1%)	6	(3.7%)
des	501	6	(1.2%)	14	(2.8%)	41	(8.2%)	61	(12%)
diffeq	479	6	(1.3%)	14	(2.9%)	29	(6.1%)	44	(9.2%)
dsip	649	9	(1.4%)	66	(10%)	150	(23%)	210	(32%)
elliptic	1366	48	(3.5%)	137	(10%)	204	(15%)	258	(19%)
ex1010	20	9	(45%)	9	(45%)	9	(45%)	9	(45%)
ex5p	71	12	(17%)	31	(44%)	32	(45%)	32	(45%)
frisc	1021	51	(5.0%)	127	(12%)	178	(17%)	245	(24%)
misex3	28	6	(21%)	13	(46%)	13	(46%)	13	(46%)
pdc	56	6	(11%)	16	(29%)	35	(63%)	37	(66%)
s298	17	4	(24%)	4	(24%)	4	(24%)	5	(29%)
s38417	1597	48	(3.0%)	109	(6.8%)	187	(12%)	247	(15%)
s38584.1	1581	4	(0.25%)	16	(1.0%)	36	(2.3%)	58	(3.7%)
seq	76	3	(3.9%)	13	(17%)	21	(28%)	24	(32%)
spla	62	10	(16%)	20	(32%)	22	(35%)	24	(39%)
tseng	558	12	(2.2%)	20	(3.6%)	33	(5.9%)	37	(6.6%)
Min	17	2	(0.25%)	2	(1.0%)	2	(2.3%)	2	(3.7%)
Mean	449	13	(4.7%)	34	(11%)	58	(17%)	78	(20%)
Max	1597	51	(45%)	137	(48%)	204	(63%)	258	(66%)

standard circuit benchmarks for EDA tool development, and demonstrate effectively the relationship between circuit delay distribution and instrumentation overhead.

A functional set of benchmarks was assembled from a variety of sources. The floating point arithmetic functions are generated by FloPoCo [19], `IIR` is a 16-bit IIR filter with 12 taps generated by Spiral [26] and `DCT` is 8-bit 8-point discrete cosine transform used for reliability experiments.

The same analysis was conducted on this new set of benchmarks and the results are outlined in Table 3.11. These circuits are well pipelined with a more realistic density of registers. The percentage of registers requiring instrumenting is low; even at 20% CDM a maximum of only 69 registers require instrumenting.

Table 3.11: The number of registers that require instrumenting for various Critical Delay Margins in the functional benchmark set. For the minimum, mean and maximum, the percentages (in brackets) are treated independently.

Benchmark	Base Resources	5% CDM		10% CDM		15% CDM		20% CDM	
	Regs.	RUMs							
DCT	168	11	(6.5%)	19	(11%)	22	(13%)	27	(16%)
IIR	314	9	(2.9%)	19	(6.1%)	28	(8.9%)	31	(9.9%)
fpadd64	908	8	(0.88%)	15	(1.7%)	32	(3.5%)	60	(6.6%)
fpexp32	388	11	(2.8%)	19	(4.9%)	19	(4.9%)	19	(4.9%)
fpexp64	1717	10	(0.58%)	34	(2.0%)	48	(2.8%)	66	(3.8%)
fplog32	1023	28	(2.7%)	45	(4.4%)	56	(5.5%)	69	(6.7%)
fpmult32	185	13	(7.0%)	19	(10%)	35	(19%)	53	(29%)
Min	168	8	(0.58%)	15	(1.7%)	19	(2.8%)	19	(3.8%)
Mean	609	13	(2.4%)	24	(4.7%)	34	(6.7%)	46	(8.7%)
Max	1717	28	(7.0%)	45	(11%)	56	(19%)	69	(29%)

3.5 Other Factors for Consideration

The previous sections have discussed the theory and method of OSM. Additional factors affecting the implementation and usage of OSM are discussed below.

3.5.1 Performing Measurements

Various configurations can be used when applying OSM to an application circuit, the choice largely depends on the nature of the delay variability being monitored. Discrepancy counters can be used for each RUM, or shared between groups of RUMs, alternatively

discrepancy latches can be used with a much lower overhead. Discrepancy latches provide all of the information necessary to conduct measurements as described thus far, however, the additional information collected using discrepancy counters may be of use when tackling the problem of critical path excitation (Section 3.5.3 as discussed in Section 3.6.1.

Measurement cycles can be conducted repeatedly, or scheduled at particular intervals, depending on the rate of delay change being measured. Relatively slow changes in delay due to degradation could be monitored with measurements on a daily or weekly basis, whereas variation due to temperature may require more frequent measurement cycles.

Rather than stepping through the entire range of phases, OSM can be configured to check a variety of phase leads, providing calibrated failure prediction, alerting if there has been a guardband violation. Work is currently in progress on using OSM for Razor-style timing error detection, by operating a fixed lagging, rather than leading clock.

3.5.2 Measurement Latency

The time taken (in clock cycles) to conduct a complete slack measurement is the product of the length of the measurement period, and the number of phase steps to complete a phase sweep. This can be reduced by checking a variety of phase leads, or only sweeping as far as required to perform a measurement, until the first phase resulting in a discrepancy for each shadow register.

3.5.3 Path Excitation

Offline timing measurement methods can use input test vectors specifically designed to ensure that the critical paths in the circuit are excited and therefore measured. Unlike these methods, OSM must make do with the data propagating through the circuit during normal operation. In some cases long measurement periods (thousands or millions of clock cycles, corresponding to milliseconds to seconds of measurement latency) are sufficient to provide a reasonable assurance of critical path excitation, however, there are cases where this will not be sufficient, such circuits with critical paths that are only excited when irregular and specific data is input to the circuit, or those with complex state.

In order to explore this problem further, the excitation of the ripple-carry adder used

as an example throughout this chapter (`intadd64` can be modelled analytically. The ripple-carry adder is a relatively simple combinatorial circuit with a critical path that is only excited under one, uncommon, input condition. The critical path of the ripple-carry adder passes from the least significant bit inputs to the most significant bit, or carry output along the carry chain. In order for this to be excited, a carry must be generated by the least significant bit inputs and this propagated through the adder without being annihilated. A carry is generated when both input bits to a full-adder stage are one (**0b1** + **0b1**). The carry is propagated when only one of the two input bits are one (**0b1** + **0b0** or **0b0** + **0b1**). Assuming that each bit of the input is uniformly and independently generated, the probability of generating a carry at the least significant bit is $p_{\text{gen}} = 1/4$ and the probability of propagating this carry is $p_{\text{prop}} = 1/4 \times 1/4 = 1/2$. Assuming that it is initialised to zero, the probability of critical path excitation for an n-bit ripple carry adder is therefore $p_{\text{excite}} = 1/2^{n-1} \times 1/4 = 1/2^{n+1}$; the expected number of clock cycles before critical path excitation is simply $E_{\text{excite}} = 1/p_{\text{excite}} = 2^{n+1}$ (the number to guarantee excitation is greater in accordance with Binomial probability theory).

Conducting this analysis for an 8-bit ripple-carry adder gives $E_{\text{excite}} = 2^{8+1} = 512$ clock cycles. With a nominal clock frequency of 150 MHz this corresponds to a measurement period of 3.41μs for each phase lead, which is quite manageable. The measurement period increases exponentially with the number of input bits, 0.87 ms for 16-bits, 57.3 s for 32-bits and 7800 years for 64-bits! Clearly, measuring until there is an expectation of critical path excitation is not feasible for all circuits. The ripple-carry adder is somewhat of a pathological case, with a single critical path that passes through all stages of the circuit and is dependent on all inputs, nevertheless, this demonstrates the need for strategies capable of addressing the problem of path excitation, some of which are discussed in Section 3.6.1.

3.5.4 Metastability

The shadow registers used in OSM sample data at varying times, thus it is probably that the data signal will change value at the same time as the shadow register latches. This can result in metastability at the output of the shadow register, with a chance that it propagates onwards. Unlike some other measurement techniques (i.e. Razor), in OSM

the RUM and its output remain unaltered, so these will be unaffected by metastability. The discrepancy register sampling the comparison signal aids in reducing the likelihood of the metastability propagating and the measurement being taken over many clock cycles reduces any potential impact.

3.6 Future Work

3.6.1 Path Excitation

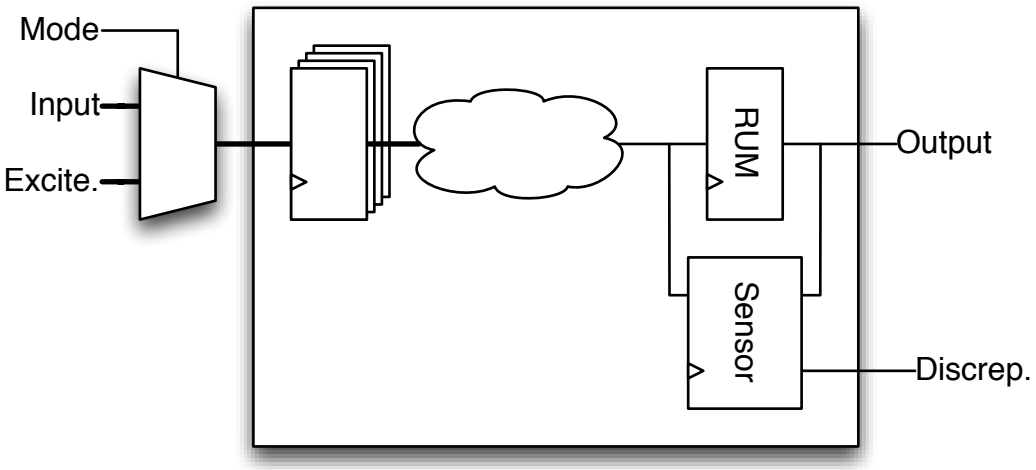
The analysis in Section 3.5.3 demonstrates that extending the measurement period is not an adequate solution to the path excitation problem for many circuits. Some preliminary concepts that may help address this issue are hence discussed.

Excitation Vector Injection

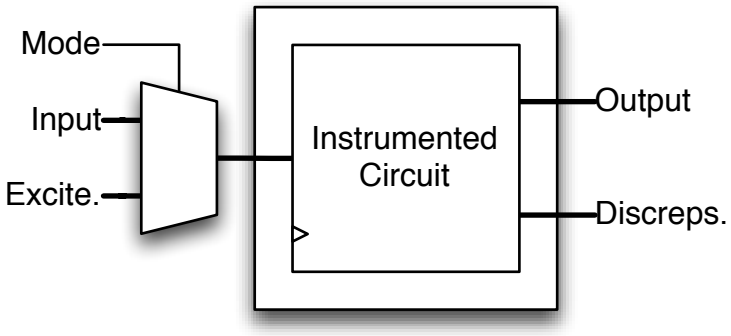
A solution may lie in the profiling of the instrumented critical paths by offline excitation vector injection. Figure 3.22 shows the two extremes. In the low level case, a multiplexer is inserted at the source register for every RUM in the circuit. Doing so comes at a significant cost, both in terms of the additional area for the multiplexer and routing, and performance. Instead, a multiplexer can be added at the input to the circuit, as in Figure 3.22b, the cost of doing so is much lower, but the selection of vectors that excite the critical paths throughout the circuit is a significantly more difficult problem, and the number of cycles to do so is likely to be larger.

In practice, some combination of the low and higher level excitation injection mechanisms would be used. As the RUM level multiplexers are moved towards the inputs of the circuit, the area and performance cost is traded for more difficulty in computing excitation vectors and a longer time with the circuit offline profiling the critical paths. With the critical paths profiled, an offset margin can be used to correct for the difference between the measured slack during operation, and the worst-case slack with offline profiling. This is effectively inferring the critical slack at a RUM using excitations in less than critical paths.

The problem with using this approach alone is that there is no knowledge of critical path excitation during operation, and the offset margin is always subtracted from the measured



(a) RUM level



(b) Circuit level

Figure 3.22: RUM and circuit level excitation vector injection mechanisms.

slack; when the critical path is excited, the margin will still be removed, making the measurement conservative. The ability to measure the critical path offline using excitation vector injection is also of use for the subsequently described methods.

Self-TRC

It has been suggested that if the critical path has a low probability of excitation, it's timing can be inferred from well excited RUMs elsewhere in the circuit, as a form of tunable replica circuit. The offset in slack between the critical register and that which is being monitored can be quantified either by scaling STA estimates or through measurement of the unexcitable critical path using excitation vector injection as described above.

Inferring slack from a RUM spatially located near the unexcited critical register would account for some of the variability effects to be accounted for. Data correlations, such as may be achieved by using a less significant adder output register to infer the slack at the most significant bit could even account for some degradation.

Discrepancy Profiles

The additional information collected when using discrepancy counters for OSM may be of use for the problem of path excitation. A reference discrepancy profile can be measured with either an extended measurement period and/or excitation vector injection. Low discrepancy counts at small phase leads (in the left tail of the profile) signify that an event is rare and cannot be relied upon for measurement. This can be used to build an offset margin against more common events, as in excitation vector injection described above. The above requires that discrepancy counters are used only once, so could be shared, with discrepancy latches used for measurement during normal operation.

The distribution at the left tail of the reference discrepancy profile could potentially be used to reconstruct the left tail when critical paths are unexcited during measurement. Figure 3.23 illustrates an example discrepancy profile, with the critical path unexcited in blue, and the tail, which is reconstructed from the distribution established from a reference measurement in green. When the critical path is not excited, the slack is overestimated as 1.40 ns, with reconstruction this is reduced to 1.29 ns. This technique would require dis-

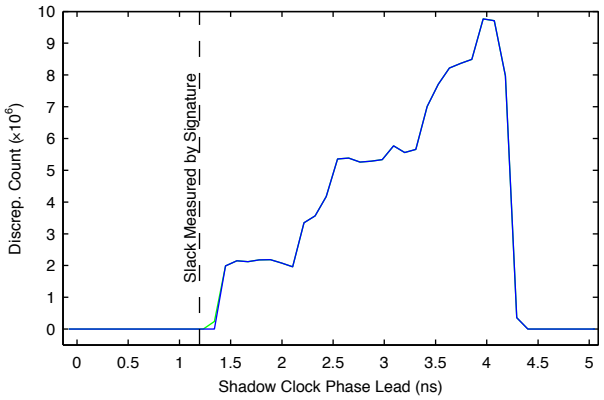


Figure 3.23: An example discrepancy profile showing slack measurement where critical path hasn't been excited in blue, and reconstructed tail in green.

crepancy counters for RUMs requiring tail reconstruction during normal operation. Unlike the margining proposed with excitation vector injection, or earlier herein, tail reconstruction would not be inherently conservative, and if the critical path happened to be excited in a given measurement period this would be accounted for.

3.7 Conclusion

In this chapter, a novel online timing slack measurement method is presented. This method is able to accurately measure the slack at chosen registers without affecting the functionality of the circuit. Registers are instrumented through the addition of shadow registers, that share the register's input but use a different clock. This shadow clock has a variable phase, allowing for the slack in the path to the shadow register to be gradually eroded to the point of timing failure, from which the slack to the shadow register can be inferred.

A calibration method is presented to allow the slack at the monitored register to be precisely inferred and dithering the clocks allows the resolution to be improved.

A technique for selecting registers to instrument in the circuit shown, this is used to analyse a variety of benchmark circuits in order to explore the relationship between circuit delay distributions and the number of registers requiring instrumenting in order to achieve a given coverage.

Finally, the ability of OSM to measure changes in slack due to varying frequency, temperature and voltage is confirmed experimentally.

Continually measuring slack using OSM allows for the "health" of a circuit to be monitored throughout its life, tracking variations in environmental conditions and degradation, making it possible to trigger pre-emptive actions to avoid timing failure.

4 OSM Sensor Insertion

4.1 Introduction

The previous chapter demonstrated that timing slack can be measured in an online circuit through the addition of shadow registers, driven with a clock of the same frequency, but a varying phase lead to the application circuit's clock. The measurement method was demonstrated, and shown to be accurate on several small benchmark circuits, using an FPGA as the test platform. In conducting these experiments it became apparent that the OSM technique was particularly well suited to measurement on FPGAs as the rich and flexible clock resources available on even the low-end FPGAs provides the functionality required to perform this monitoring with almost no cost. The overhead is slightly higher in Application-Specific Integrated Circuits (ASICs), where the infrastructure required to generate the varying phase clock and routing it to the sensors does not already exist.

The Altera Cyclone IV family of FPGAs have been chosen as a research platform for this work. These are low-cost, low-power devices with a traditional island-style architecture. Their simplicity lends them to exploratory research in measurement and the associated automation tools.

One of the great strengths of OSM is its ability to be calibrated. Instead of having to place the sensor adjacent to the register it is instrumenting, generally requiring resources to be reserved throughout the FPGA, the sensor can be added to the application circuit where resources are available. This allows us to lock the application circuits placement and routing, maintaining its timing. After the insertion, any difference in path length between the RUM and shadow register can be accounted for. Doing so maintains the timing behaviour of the application, avoiding a chicken-and-egg problem that can be the case if resources are not reserved, where adding the shadow registers impacts on the timing

of the circuit, altering which registers need to be instrumented to achieve a given coverage.

This chapter describes the configuration and use of the FPGA clock resources for OSM and the mapping of the shadow registers to an FPGA architecture. A CAD tool flow is presented that allows for arbitrary circuits to be automatically instrumented for OSM with minimal intervention. An investigation into the overheads, both in terms of increased area and delay, for instrumenting these circuits is presented.

4.2 Mapping of OSM to FPGAs

Performing OSM requires the addition of shadow registers to chosen registers in the application circuit. These shadow registers need to be driven by a clock of the same frequency as that driving the application's registers, but with a varying phase relationship. This section discusses the generation of this clock, and the mapping of shadow registers in the context of the Altera Cyclone IV architecture. This architecture contains features that are found across all FPGAs and as such, although the mapping of OSM has been conducted to one specific architecture, the general principles are transferable.

4.2.1 Clock Generation

The current generation of commercial FPGAs generate internal clock signals using analogue Phase-Locked Loops (PLLs), which output clock signals that have a phase related to that of an input signal. The Cyclone IV family of FPGAs contain PLLs with a block structure shown in Figure 4.1. An FPGA in this device range has between two and eight of these, each with five clock outputs. The EP4CE22F17C6 device used for experiments in this thesis contains four PLLs, one at each corner of the die.

These PLLs consist of a prescale counter (N), phase-frequency detector (PFD), charge pump (CP), loop filter (LF), voltage-controlled oscillator (VCO), feedback counter (M) and postscale output counter (C). N divides the input clock to produce a reference clock signal. The phase and frequency of this reference signal are compared to the feedback signal by the PFD, which produces an error signal. This error signal is fed to the CP and low-pass LF, together which generate a voltage which drives the VCO. The VCO signal is

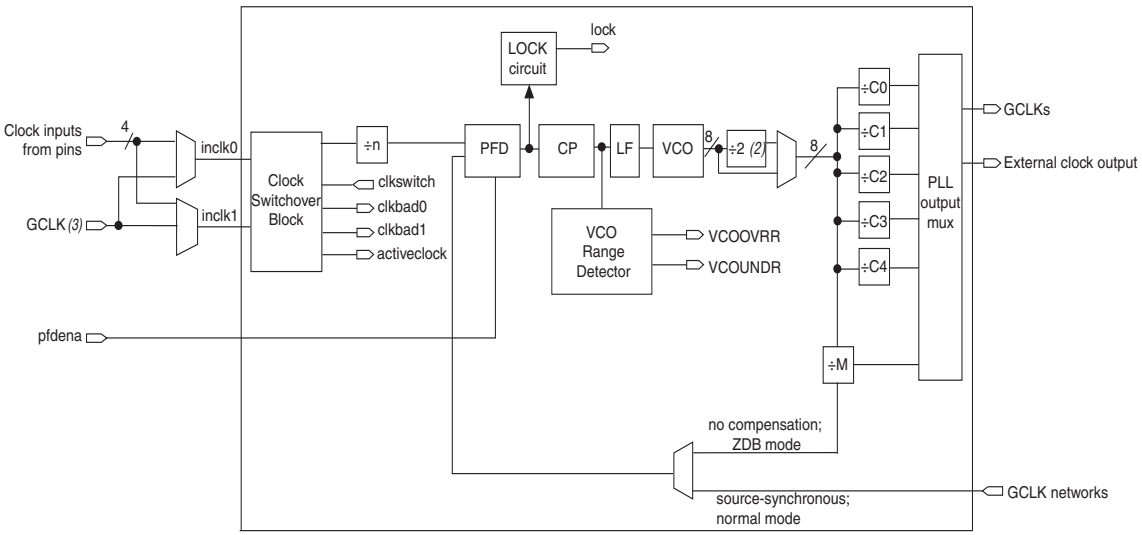


Figure 4.1: A block digram of the Cyclone IV PLL from [4].

fed back through the M counter to the PFD, resulting in a negative feedback control loop. A drift in output phase or frequency results in an increase in the error signal which drives the VCO in the other direction in order to reduce the error. The VCO is said to be locked when the reference signal matches the feedback signal and thus there is no error.

The reference signal is produced by the prescale counter dividing the input signal frequency (f_{in}) by N (Equation 4.1). The feedback counter multiplies the frequency of the reference signal (f_{ref}) by M, producing the f_{vco} frequency (Equation 4.2. The frequency of each of the outputs (f_{out}) is the f_{vco} divided by the C counter (Equation 4.3).

$$f_{\text{ref}} = \frac{f_{\text{in}}}{N} \tag{4.1}$$

$$f_{\text{vco}} = f_{\text{ref}} \times M = \frac{f_{\text{in}} \times M}{N} \tag{4.2}$$

$$f_{\text{out}} = \frac{f_{\text{in}} \times M}{N \times C} = \frac{f_{\text{vco}}}{C} \tag{4.3}$$

The VCO has eight phase taps which provides the ability for fine resolution phase-shifting, producing a phase offset that is independent of process, voltage and temperature variation. The size of a phase offset for each step is dictated by the frequency of the VCO

and the number of phase taps, and is described by Equation 4.4. The phase of a particular PLL output can be varied independently of the other outputs, meaning that OSM requires only one additional output of the PLL that is generating the application circuit's clock to generate the phase varying shadow clock.

$$\Delta t_{\phi}(f_{\text{out}}) = \frac{T_{\text{vco}}}{8} = \frac{1}{8 \times f_{\text{vco}}} = \frac{N}{8 \times f_{\text{in}} \times M} \tag{4.4}$$

The VCO in the Cyclone IV PLL is specified to operate between 600 MHz and 1.3 GHz, as a result $\Delta t_{\phi}(f)$ is bounded between 96.16 ps and 208.33 ps. This dictates the OSM measurement resolution on the Cyclone IV architecture. A given PLL output frequency can be synthesised with a variety of different M, N and C counter configurations. In order to achieve best measurement resolution for a given output frequency, the space of these parameters has been explored exhaustively to produce a table of optimal PLL configurations for each output frequency. Figure 4.2 shows the phase step size for all output frequencies between 50 MHz and 1.3 GHz.

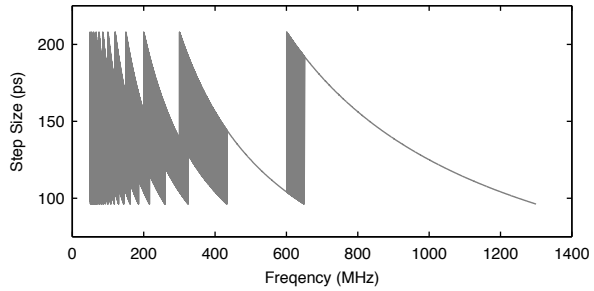


Figure 4.2: Plot showing the step sized achieved for output frequencies with OSM optimised PLL configuration.

Since the Cyclone III architecture, these PLLs can be reconfigured during runtime (using the `altpll_reconfig` megafunction), providing the ability to perform offline frequency sweep measurements by varying the frequency, and OSM by stepping the phase while the FPGA is running and without reconfiguring the entire bitstream.

4.2.2 Shadow Register Mapping

Unlike in an ASIC, where access can be made available to fork the inputs and outputs of any register in a design to an OSM sensor, in an FPGA access is restricted to the signals

that the architecture provides. This section discusses the various strategies for shadowing registers in Logic Elements, memory and DSPs in the Cyclone architecture.

Logic Elements

Figure 4.3 shows a block digram of a Logic Element (LE) in the Altera Cyclone IV architecture. These LEs contain a 4-input SRAM based Lookup Table (LUT), with associated carry chain logic, and a register, with synchronous load and clear, asynchronous clear, clock and clock enable. These LEs are grouped into Logic Array Blocks (LABs), which contain 16 LEs, their carry chain, control signals, and local interconnect.

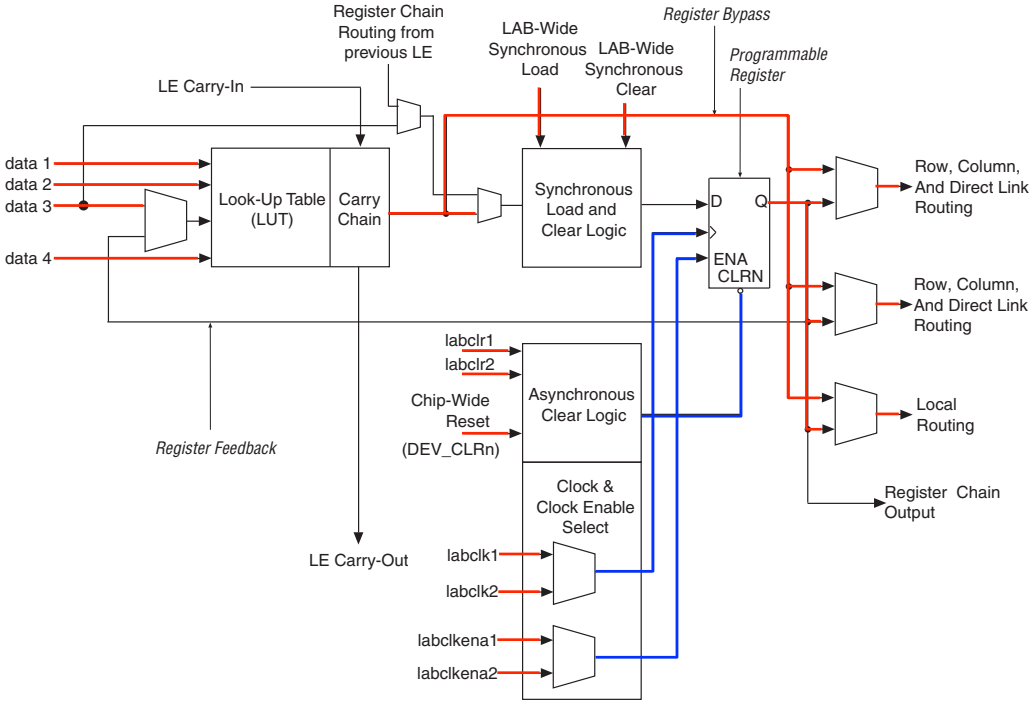


Figure 4.3: The Cyclone IV Logic Element, with externally accessible signals in red and signals accessible from inside the LAB in blue, from [4].

The externally accessible wires in the Cyclone IV LE are indicated as red lines in Figure 4.3, blue lines indicate signals accessible from within the LAB. In order to instrument a given register for OSM, all of its input signals (except the clock) need to be connected to a shadow register (with its own clock signal) and the output of the sink and shadow registers compared. Not all of these signals are directly accessible, particularly the D-input to the register. This is constructed from either the combinatorial output of the LUT

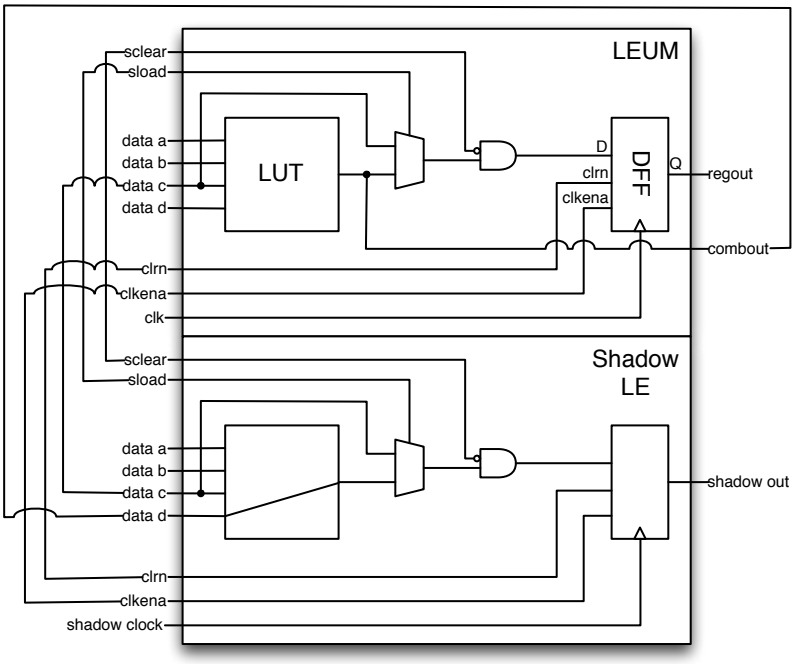
and the synchronous data, load and clear inputs or the output of the previous register routed through the register chain. The D-input must be reconstructed and connected to the shadow register along with the other clear and clock enable signals. In practice, the register chain routing is generally unused and in the event that it is, the signal can be forked at its source, which leaves us with the remaining signals.

The naïve solution to implementing a shadow register is to replicate the LE containing the sink register (LEUM) to be instrumented, copying its LUT configuration and connecting to all of its inputs, however this is not practical. Routing depopulation means that not all of the input signals connected LEUM can be directly connected to another LE within the LAB, without some of these signals first leaving and re-entering the LAB, resulting in increased routing congestion and an increase in delay variation, potentially impacting on the accuracy of measurement.

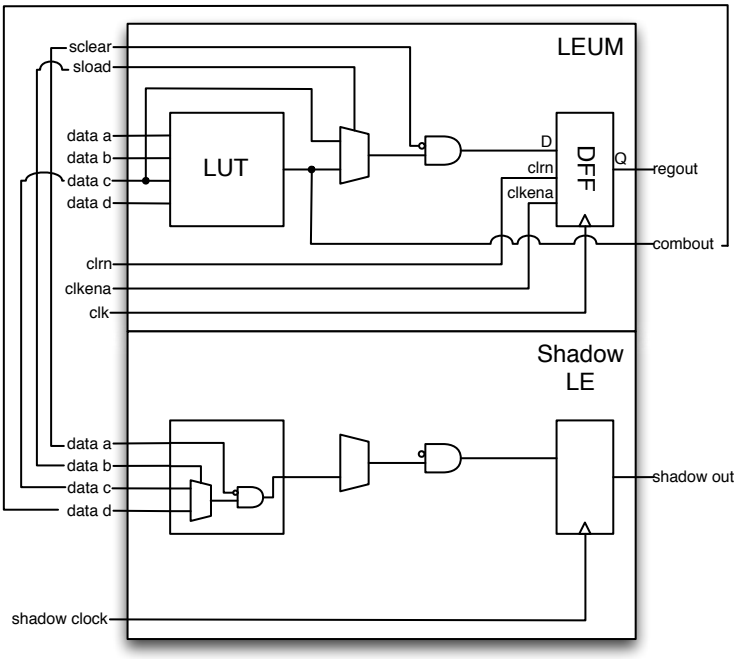
Four strategies have been developed to shadowing a sink register in a Cyclone IV LE, shown in Figure 4.4 and described below:

Mirroring: The functionality of the asynchronous clear and clock enable signals can not be efficiently replicated, so if these are used, mirroring is the most appropriate approach to shadowing a sink register. The shadow register exactly duplicates the configuration of the sink register, with the LUT in the shadow LE feeding the combinatorial output of the LEUMs LUT to the shadow register. This configuration most closely replicates the LEUM and thus the timing behaviour of the two registers will be well machined, and the OSM sensor can be most accurately calibrated.

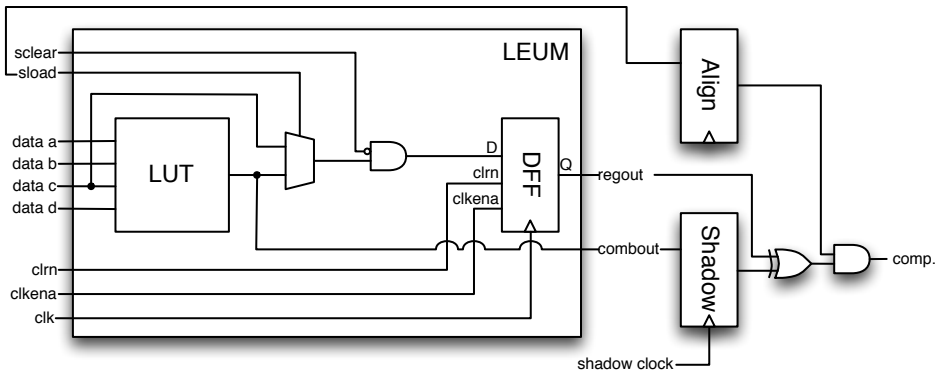
LAB control signals are used, so if the shadow LE can be packed within the LEUM's LAB, these signals do not need to be regenerated. The LAB is limited to two clock signals, and the shadow register requires one so packing can only occur if there is one existing clock. The discrepancy register often requires a different clock signal to the RUM and shadow register, so it is not possible to pack this into the LEUM's LAB. If it is not possible to pack the shadow LE into the LEUM's LAB, use of the LAB the shadow LE is placed in may be restricted as the control signals must be the same as the LEUM.



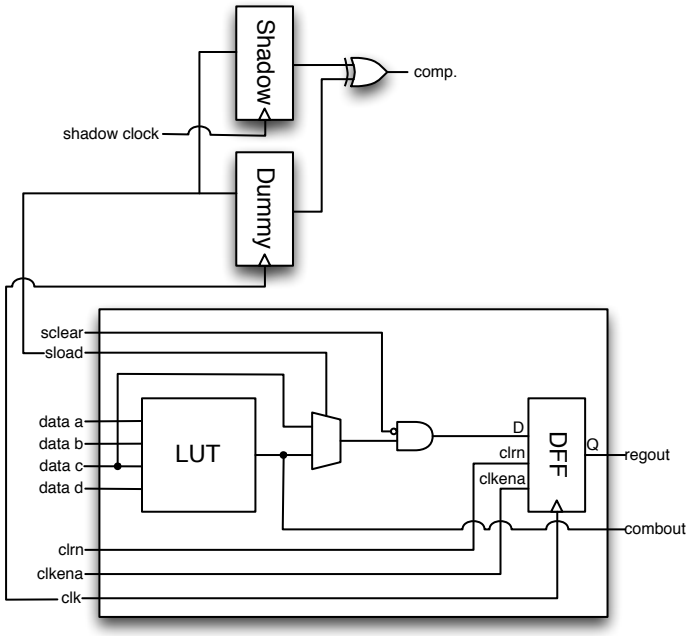
(a) Mirroring



(b) Emulation



(c) Enable



(d) Dummy

Figure 4.4: The various shadow register mapping strategies for the Cyclone IV LE. Where appropriate the shadow register has been shown outside of a LE, and comparator and/or discrepancy register negated for the sake of clarity.

Mirroring is the general-purpose LE mapping as it facilitates shadowing the full register functionality as provided by the architecture.

Emulation: Where asynchronous clear and clock enable are not used, the synchronous load and clear functionality can be emulated using the shadow LE's LUT. This frees the shadow register from the LAB control signals allowing it to be placed in a LAB where these signals are already being used. This may reduce the total LAB utilisation, allowing more registers to be instrumented in designs with high resource utilisation.

Since the delay of the LUT isn't the same as the synchronous load and reset circuitry, using this approach can introduce some measurement error which cannot be calibrated for.

Enable: If the near-critical paths arrive exclusively through the LUT feeding the sink register, the OSM sensor need only monitor this signal. The synchronous load signal is used to enable the OSM sensor only when the sink register has been fed by the LUT output. The converse can be applied if the synchronous data signal is critical.

This mapping requires an additional register to align the synchronous load signal and LUT as a OSM sensor enable.

Dummy: A dummy register allows for the monitoring of a signal which may not be otherwise accessible. For example, if it was desirable to monitor the synchronous load signal, without inferring its timing by monitoring the D-input to the sink register, this could be done by adding a dummy register driven directly by synchronous load. The output of this dummy register is then used for comparison against the shadow register, instead of the application sink register. It is not possible to calibrate for the difference in delay to the sink register and dummy register, so this must be placed as near to the sink register as possible. Dummy registers require that just one signal is forked off the measurement cell, reducing the routing overhead, and may also offer a solution in situations where the sink register is not accessible for shadowing, such as in embedded blocks.

Embedded Blocks

Embedded blocks on the Cyclone IV come in the form of 18x18 fixed-point multipliers and 9 kb memories. These embedded blocks present some problems with signal observability, as one side of the register (either input or output) are not exposed. The OSM mapping strategy for these are as follows:

Embedded Multipliers: The Cyclone IV embedded multiplier is shown in Figure 4.5. The blocks have optional registers at their input and output. Since one side of these registers is not visible to the FPGA fabric, they cannot be monitored with OSM sensors. Instead, the registers must be pulled out of the multiplier and implemented in LEs. Doing so may adversely affect the timing performance of the application circuit.

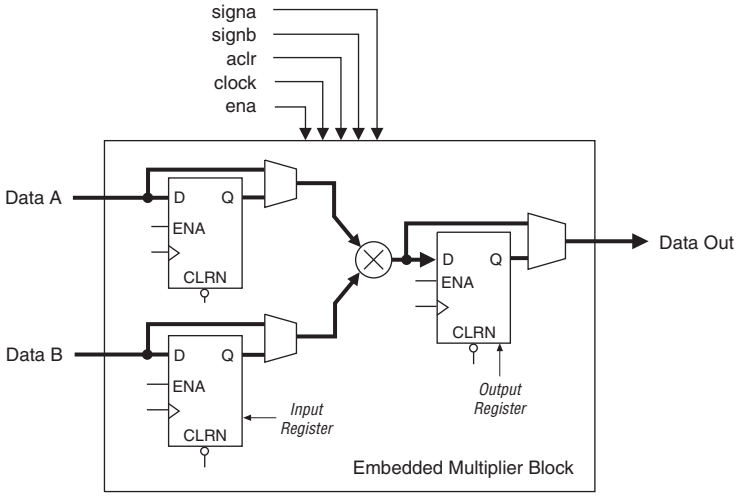


Figure 4.5: The Cyclone IV embedded multiplier from [4].

Memory Blocks: The Cyclone IV memory blocks can be configured as single and dual-port RAM, shift-register, ROM and as a FIFO. The registers in memory blocks cannot be bypassed, so it is not possible to instrument them in the standard way. In some cases the functionality can be implemented in soft-logic instead, for example a barrel shifter often performs faster implemented in soft-logic rather than memory, albeit with a higher resource utilisation.

In practice, embedded blocks are significantly faster than the soft-logic in an FPGA and are unlikely to be near-critical in well designed circuits. Thus they will not need to be monitored, except when a high CDM coverage is specified. In the EP4CE22F17C6 the embedded multipliers are specified to run at 286 MHz and memory blocks at 315 MHz; a typical circuit implement on this device will have an f_{sta} of 100-200 MHz.

4.3 Online Slack Measurement Insertion Tool

The Slack Measurement Insertion (SMI) tool aims to add OSM to arbitrary circuits with minimal intervention and impact on circuit timing performance. SMI (including the register selection algorithm set out in Section 3.2.6) was developed by the author and is integrated into Reliability Instrumentation Platform for Programmable Logic (RIPPL), a framework created co-operatively with Dr Edward Stott and Dr Nachiket Kapre.

4.3.1 RIPPL

RIPPL is a platform for conducting reliability experiments, currently on the Cyclone family of FPGAs. RIPPL interfaces with the external temperature control and power supply and measurement hardware (from Section 3.3) and the FPGAs PLL, facilitating measurement experiments. Altera's "Virtual JTAG" interface provides communication between TCL running on a host computer and hardware on the FPGA, even in the least expensive development boards (such as the Terasic DE0-nano), which have no other interface infrastructure. RIPPL provides hardware modules for input vector generation (either procedural using Linear-Feedback Shift Registers or through loading into input memory), output vector collection, fast counters (to be used as discrepancy counters or in offline Transition Probability measurement), and signature generators.

4.3.2 Tool Flow

SMI is an end-to-end automated compile flow that plugs into the existing vendor's tools, and can readily be used by a designer without knowing the details of the OSM technique. The input of the tool is a conventional HDL design entry (in Verilog or VHDL). It is

integrated into Altera Quartus II compiler, but the principles are equally applicable to other vendors compilers using the interfaces they provide. Quartus II v13.0 is the current release at the time of writing and has been used for SMI, but it should be compatible with versions as old as v5.0.

Currently the framework is configured to cater to running experiments, rather than instrumenting circuits that will be used in the field. The primary difference is that rather than the application interfacing with an external system, the input and output vectors are generated and stored within the device. This can be modified trivially to allow for real operating circuits to be instrumented with OSM.

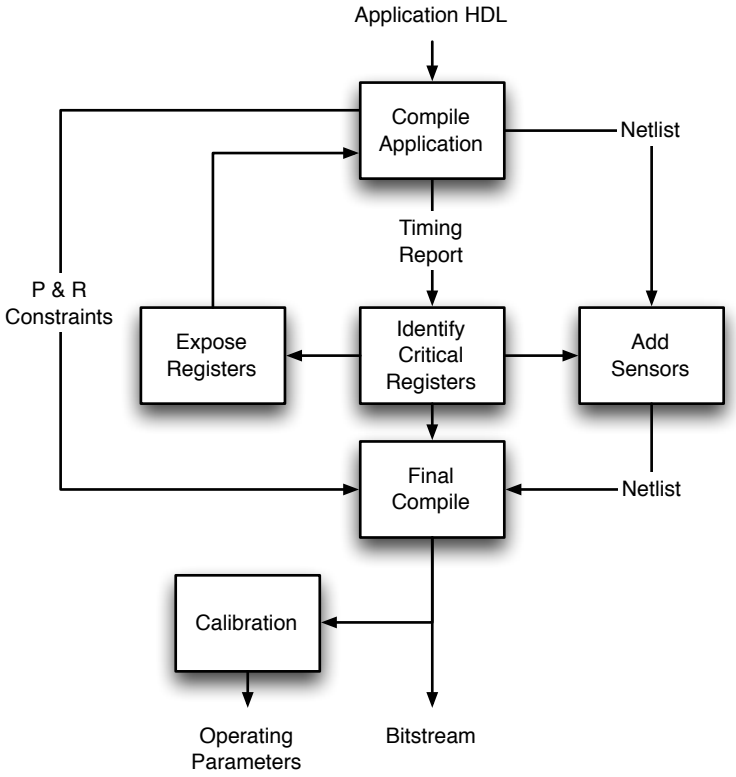


Figure 4.6: The SMI compile flow. Details regarding the identification of critical registers may be found in Section 3.2.6 and calibration in Section 3.2.3.

The SMI compile flow is illustrated in Figure 4.6. The flow is based on principles described in Altera's Quartus University Interface Program (QUIP) [3]. It was chosen to perform the necessary circuit modification in this way, as Altera recommends the use of an intermediate netlist for large-scale changes and it is better documented than alternative techniques. The steps to the compile are detailed below:

Design Entry: The application circuit is instantiated into a wrapper (`op_wrap`) containing the input vector generator/memory and output memory. For non-experimental purposes the design entry input would be the application circuit and the interface to the external system. Figure 4.7 illustrates the wrapped application circuit.

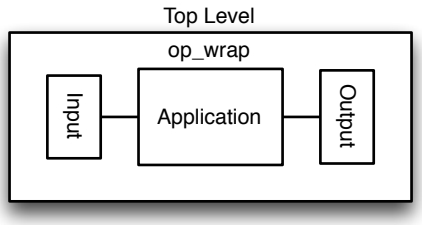


Figure 4.7: Application circuit instantiated within wrapper.

Compile Application: The wrapped application circuit is compiled (synthesised, mapped, placed and routed) in Quartus II, as the top level of the design. A flattened netlist, timing report, and routing and placement constraints are extracted for the application.

Identify Critical Registers: Near-critical registers (those with the least setup slack) that meet a specified CDM (RUMs) are identified from the STA timing report. If these registers are not accessible in the netlist, the circuit must be modified. If the registers are inaccessible because they have been packed or inferred into an embedded block, they are automatically extracted. Otherwise, manual intervention is required.

Add Sensors: The RUMs are identified in the netlist and OSM sensors are added.

Final Compile: The instrumented netlist is instantiated within a top level containing interface and control logic, clock generators and signature generators used to confirm the functionality of the application circuit and vector generation/collection. The output is a bitstream for the instrumented application circuit.

Calibration: The bitstream is automatically programmed onto the FPGA and calibration conducted, producing the shadow path offsets (t_{RS}) necessary for accurate OSM.

4.3.3 Flow Details

The previous section discussed the compile flow from a higher level perspective, this looks at some of the details. Despite this being the recommended method for performing large-scale post-compile modifications to a circuit, a significant effort was required in order to integrate SMI into the Quartus compile. This is likely to be because of the hierarchical and memory modifications required to give full compilation priority to the application circuit.

The application circuit is manually instantiated into a wrapper entity where it sits between an input vector generator/memory and output memory. The experiments conducted for this thesis use random inputs generated procedurally using the LFSR option. RIPPL uses Altera's "SignalTap" for in-system memory reading and editing, which is used by the host computer to load input vectors and read output traces. This is disabled during the first compile, as the associated circuitry can congest the area surrounding the memory, limiting placement and routing opportunities for the application circuit. The memory is kept (prevented from being synthesised away) using dummy address generators.

The first compile of the wrapped application (`op_wrap`) is conducted using a settings file (QSF), `op_wrap_isolated.qsf`, which specifies `op_wrap` as the top level entity. The netlist format used is a Verilog Quartus Mapping (VQM), which is a restricted form of the Verilog language standard. It supports only wires (no registers) and atomic primitives (LUTs, registers, I/O, memory, multipliers etc.). The generation of VQMs is no longer supported by default in Quartus II, so the assignment shown in Listings 4.1 is required in the QSF to enable this.

Listing 4.1: This assignment is required to enable VQM generation in current versions of Quartus II.

```
set_global_assignment -name INI_VARS "qatm_force_vqm=on"
```

The compile, netlist and placement and routing extraction is conducted as shown in Listing 4.2. This outputs a routing constraints file `routing.rcf`, placement constraints are written into the QSF file as location assignments and a flattened VQM netlist is written to `op_wrap_isolated.vqm`.

Listing 4.2: Compile and export netlist, placement and routing.

```
# Compile the wrapped application circuit
quartus_sh --flow compile op_wrap_isolated.qsf

# Back annotate placement
quartus_cdb op_wrap_isolated.qsf --back_annotate=lc

# Back annotate routing
quartus_cdb op_wrap_isolated.qsf --back_annotate=routing

# Extract netlist
quartus_cdb op_wrap_isolated.qsf --vqm=op_wrap_isolated.vqm
```

The timing report is generated by running the timing script shown in Listings 4.3 in `quartus_sta`. This reports the setup timing from all source nodes, to all sink nodes in the design, showing just the worst-case setup timing for each sink register. This is the information that is required to establish which registers to instrument according to the specified CDM coverage.

Listing 4.3: Generate timing report used to establish register criticality.

```
report_timing\
        -from_clock main_clock\
        -to_clock main_clock\
        -from [get_keepers *]\
        -to [get_keepers *]\
        -setup -npaths 10000 -nworst 1 -detail summary\
        -file setup_timing_isolated.txt
```

Sink registers are selected for instrumenting based on their reported slack and checked to see if are inaccessible, such as those contained within `altsyncram`, `lpm_mult` or `mac_mult` primitives. If registers have automatically been packed into embedded blocks, or embedded blocks inferred, the registers can be removed with the assignments like those shown in Listings 4.4 or manually, and compile process begun from the start.

Listing 4.4: Assignments to disable register packing or shift-register recognition in specified entities.

```
# Disable register packing
set_instance_assignment\
        -name AUTO_PACKED_REGISTERS_STRATIXII\
        -to *\
        -entity <entity name> off

# Disable automatic shift-register recognition
set_instance_assignment\
        -name AUTO_SHIFT_REGISTER_RECOGNITION\
        -to *\
        -entity <entity name> off
```

With the sink registers to be instrumented identified and accessible, the VQM netlist can be parsed to find the inputs to these registers and OSM sensors inserted into the netlist. Listings 4.5 is a "mirroring" OSM sensor constructed from atomic primitives in VQM syntax. All of the signals driving the RUM are connected to the shadow register, any not connected are set to a default value. The discrepancy signals are aggregated into a bus and exposed as an output in the module declaration.

Listing 4.5: A "mirroring" OSM sensor to be added to the netlist.

```
// Shadow Register
wire shadow_<index>
dffeas shadowreg_<index> (
.d(<RUM_d>),
.clrn(<RUM_clrn>),
.prn(<RUM_prn>),
.ena(<RUM_ena>),
.asdata(<RUM_asdata>),
.aload(<RUM_aload>),
.sclr(<RUM_sclr>),
.clk(shadow_clk),
.q(shadow_<index>));
defparam shadowreg_<index>.is_wysiwyg = "true";

// Compare output of RUM and shadow registers
wire compare_<index>;
```

```
cycloneive_lcell_comb compare_<index> (
        .datac(<RUM_q>),
        .datad(shadow_<index>),
        .combout(compare_<index>));
defparam compare_<index>.sum_lutc_input = "datac";
defparam compare_<index>.lut_mask = "0FF0";
defparam compare_<index>.is_wysiwyg = "true";

// Discrepancy Register
wire discrep_<index>;
dffeas discrepreg_<index> (
        .d(compare_<index>),
        .clk(!clk),
        .q(discrep_<index>));
defparam discrepreg_<index>.is_wysiwyg = "true";
```

In some cases, the RUM D-input is driven by a "feeder", where the adjacent LUT serves only to route a signal through to the register. When the instrumented netlist is compiled, Quartus removes and replaces non-functional primitives such as feeders. Forking the output of this feeder to a shadow register stops the feeder from being removed and results in an additional feeder being inserted into the path to the RUM, causing the near-critical path to gain additional delay. In order to overcome this, the signal must be forked to the shadow register before the feeder, as in Listings 4.6. Feeder LUTs can be identified as having a single input, usually on the datad port (which is fastest) and a LUT mask which buffers this input to the combinatorial output ("FF00"). Generally the instance and intermediate signal name contain the keyword "∼feeder".

Listing 4.6: Instrumenting a RUM with feeder register.

```
// Feeder LUT
cycloneive_lcell_comb \d_input~feeder_I  (
        .dataad(feeder_input),
        .combout(\d_input~feeder ));
defparam \d_input~feeder_I .sum_lutc_input = "datac";
defparam \d_input~feeder_I .lut_mask = "FF00";
```

```
// RUM
dffeas fed_register (
        .d(\d_input~feeder )
        .clk(clk),
        .q(fed_register_out));
defparam fed_register.is_wysiwyg = "true";

// Shadow Register -
// connect to feeder_input not the RUM D-input (\d_input~feeder )
wire shadow;
dffeas shadow_reg (
        .d(feeder_input),
        .clk(shadow_clk),
        .q(shadow));
defparam shadow.is_wysiwyg = "true";
```

In addition to adding shadow registers, various signals are brought out of the module so that they can be attached to signal generators, used to confirm functionality. If input vector memories are used these and the output memories are converted to have "SignalTap" enabled and the dummy address generators removed.

Quartus makes extensive use of escaped identifiers in the names of signals and instances. This allows for names which encode the hierarchy from which they originate. In some cases, when the netlist has been modified, Quartus re-infers the hierarchy from these names, changing signal or instances names. The placement and routing constraints are applied to these names, so if they are changed, the new names will not match and the constraints cannot be applied. This is overcome by obfuscating the identifier names in the netlist and associated constraint files, replacing the "|" character with "__". An example of the identifier name obfuscation is shown in Listings 4.7.

Listing 4.7: An example of identifier obfuscation for the `fpmult32` VQM netlist.

```
// Before identifier obfuscation
wire \FPMultiplier_8_23_400_Wrapper:test_inst|o_R_d1[29] ;
wire \FPMultiplier_8_23_400_Wrapper:test_inst|FPMultiplier_8_23_400:test|
    IntAdder_33_400:RoundingAdder|X_d1[30] ;
```

```
dffeas \FPMultiplier_8_23_400_Wrapper:test_inst|o_R_d1[30]~I (
                .clk(\clk~inputclkctrl ),
                .d(\FPMultiplier_8_23_400_Wrapper:test_inst|o_R_d1[30]~91 ),
                .q(\FPMultiplier_8_23_400_Wrapper:test_inst|o_R_d1[30] ));
defparam \FPMultiplier_8_23_400_Wrapper:test_inst|o_R_d1[30]~I .power_up = "low
    ";
defparam \FPMultiplier_8_23_400_Wrapper:test_inst|o_R_d1[30]~I .is_wysiwyg = "
    true";

// After identifier obfuscation
wire \FPMultiplier_8_23_400_Wrapper:test_inst__o_R_d1[29] ;
wire \FPMultiplier_8_23_400_Wrapper:test_inst__FPMultiplier_8_23_400:
    test__IntAdder_33_400:RoundingAdder__X_d1[30] ;

dffeas \FPMultiplier_8_23_400_Wrapper:test_inst__o_R_d1[30]~I (
                .clk(\clk~inputclkctrl ),
                .d(\FPMultiplier_8_23_400_Wrapper:test_inst__o_R_d1[30]~91 ),
                .q(\FPMultiplier_8_23_400_Wrapper:test_inst__o_R_d1[30] ));
defparam \FPMultiplier_8_23_400_Wrapper:test_inst__o_R_d1[30]~I .power_up = "
    low";
defparam \FPMultiplier_8_23_400_Wrapper:test_inst__o_R_d1[30]~I .is_wysiwyg = "
    true";
```

During the first compile the application wrapper was the top level design entity. When the instrumented netlist is compiled it is instantiated within the RIPPL framework. The placement and routing constraints have absolute identifiers which will not reference correctly during the compilation of the instrumented application and framework. The hierarchy must be modified to account for this, with the application circuit now being instantiated from within `BBTester:TST0|_wrap:OP_inst`. The placement constraints accept a "$*$" wildcard and the routing requires the absolute path. An example is given in Listings 4.8.

Listing 4.8: An example of hierarchy correction of constraints in `fpmult32`.

```
# Routing constraint before hierarchy correction
signal_name = FPMultiplier_8_23_400_Wrapper:test_inst|FPMultiplier_8_23_400:
    test|IntMultiplier_24_400:SignificandMultiplication|IntAdder_49_400:
    Adder_final4_0|Add0~4 {
```

```
        LE_BUFFER:X40Y14S0I12;
        R4:X37Y14S0I25;
        LOCAL_INTERCONNECT:X39Y14S0I34;
        dest = ( FPMultiplier_8_23_400_Wrapper:test_inst|FPMultiplier_8_23_400:
            test|round~6, DATAC ), route_port = DATAA;
}

# Routing constraint after hierarchy correction
signal_name = BBTester:TST0|OP_wrap:OP_inst|FPMultiplier_8_23_400_Wrapper:
    test_inst__FPMultiplier_8_23_400:test__IntMultiplier_24_400:
    SignificandMultiplication__IntAdder_49_400:Adder_final4_0__Add0~4 {
        LE_BUFFER:X40Y14S0I12;
        R4:X37Y14S0I25;
        LOCAL_INTERCONNECT:X39Y14S0I34;
        dest = ( BBTester:TST0|OP_wrap:OP_inst|FPMultiplier_8_23_400_Wrapper:
            test_inst__FPMultiplier_8_23_400:test__round~6, DATAC ), route_port
             = DATAA;
}

# Placement constraint before hierarchy correction
set_location_assignment LCCOMB_X39_Y14_N0 -to "FPMultiplier_8_23_400_Wrapper:
    test_inst__FPMultiplier_8_23_400:test__round~6"

# Placement constraint after hierarchy correction
set_location_assignment LCCOMB_X39_Y14_N0 -to "*FPMultiplier_8_23_400_Wrapper:
    test_inst__FPMultiplier_8_23_400:test__round~6"
```

The instrumented circuit and associated constraints are now ready for the final compilation which produces an instrumented bitstream. A QSF (`op_wrap.qsf`) is constructed using `'BBTester'` as the top level and applying the placement and routing constraints. Various optimisations are disabled in order to maintain the application circuit, these include "register duplication" and "beneficial skew optimisation". A complete compile must be executed due to the additional framework circuitry. The completed circuit is shown in Figure 4.8, which shows the wrapped application circuit now surrounded by the RIPPL framework.

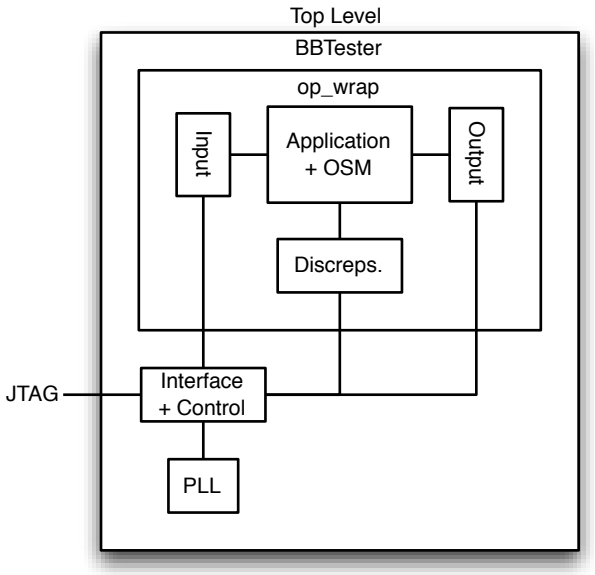


Figure 4.8: Structure of the application circuit, within RIPPL after SMI.

With this compile complete, the bitstream can be automatically programmed to a target FPGA and calibration executed. For the sake of simplicity, the measurements and calibration are currently controlled by the host computer, using RIPPLs TCL interface, but this could easily be migrated to run entirely on the FPGA. The calibration procedure returns a file containing the calibration offsets. Which are used for OSM when the FPGA and application are commissioned.

4.4 Results

The functional set of benchmarks were automatically instrumented for OSM using SMI. The resource utilisation and timing model frequency (f_{sta}) for these benchmarks before instrumentation is given in Table 4.1. The compiled circuits utilise embedded multipliers and memory making them a realistic test for the OSM technique and SMI flow. The "mirroring" mapping has been used as this allows shadowing of all register configurations, making it the most generally applicable.

The following sections discuss the area and performance overheads incurred for instrumented these benchmark circuits for OSM.

Table 4.1: Base resource utilisation for the benchmarks in the functional benchmark set.

Benchmark	LUT	Reg.	Mem. (b)	DSP	f_{sta} (MHz)
DCT	329	168	0	0	158.93
IIR	558	314	0	0	102.67
fpadd64	1524	908	222	0	142.31
fpexp32	648	388	18464	4	80.42
fpexp64	2683	1717	87975	44	84.80
fplog32	1486	1023	29116	18	97.52
fpmult32	226	185	0	8	143.04
Min	226	168	0	0	80.42
Mean	1065	672	19397	11	115.67
Max	2683	1717	87975	44	158.93

4.4.1 Area Overhead

The resource overhead of instrumenting with OSM depends on the circuit's delay distribution and desired CDM coverage as described in Section 3.4. Table 4.2 shows the overhead in terms of an increase in LUTs and registers for 5%, 10%, 15% and 20% CDM coverage. The increase corresponds to two registers (shadow and discrepancy) and one LUT (for comparison) for each RUM instrumented.

The area overhead of instrumentation starts off small with a mean increase of just 1.5% LUTs and 4.9% registers. This increases with the CDM, reaching a mean increase of 6.3% LUTs and 20% registers for 20% CDM. Despite the majority of the circuits using embedded blocks, these only present a problem in one case, `fpadd64` at 20% CDM. Here registers that require instrumenting fall within the Leading-Zero Counter (LZC) and shifter logic which has been inferred into embedded memory. This is overcome by disabling the automatic inference of the LZA/shifter into memory, so that it is implemented using soft-logic resources instead. Doing so increase the base number of LUTs used by 109 (7%) and registers by 218 (24%) but frees the 222 b of M9K embedded memory.

Excluded from these overhead counts are allowances for a measurement controller. The design of the controller will depend on the host system. A basic implementation might consist of:

- An additional output and clock routing from the existing clock generator.
- A measurement period counter to govern the duration of each phase step — 24 LEs.

Table 4.2: Area overheads for instrumenting the functional benchmarks for OSM with coverages between 5% and 20% CDM.

Benchmark	5% CDM				10% CDM				15% CDM				20% CDM			
	ΔLUT		ΔReg.		ΔLUT		ΔReg.		ΔLUT		ΔReg.		ΔLUT		ΔReg.	
DCT	11	(3.3%)	22	(13%)	19	(5.8%)	38	(23%)	22	(6.7%)	44	(26%)	27	(8.2%)	54	(32%)
IIR	9	(1.6%)	18	(5.7%)	19	(3.4%)	38	(12%)	28	(5.0%)	56	(18%)	31	(5.6%)	62	(20%)
fpadd64	8	(0.52%)	16	(1.8%)	15	(0.98%)	30	(3.3%)	32	(2.1%)	64	(7.0%)	169	(11%)	278	(31%)
fpexp32	11	(1.7%)	22	(5.7%)	19	(2.9%)	38	(9.8%)	19	(2.9%)	38	(9.8%)	19	(2.9%)	38	(9.8%)
fpexp64	10	(0.37%)	20	(1.2%)	34	(1.3%)	68	(4.0%)	48	(1.8%)	96	(5.6%)	66	(2.5%)	132	(7.7%)
fplog32	28	(1.9%)	56	(5.5%)	45	(3.0%)	90	(8.8%)	56	(3.8%)	112	(11%)	69	(4.6%)	138	(13%)
fpmult32	13	(5.8%)	26	(14%)	19	(8.4%)	38	(21%)	35	(15%)	70	(38%)	53	(23%)	106	(57%)
Min	8	(0.37%)	16	(1.2%)	15	(0.98%)	30	(3.3%)	19	(1.8%)	38	(5.6%)	19	(2.5%)	38	(7.7%)
Mean	13	(1.5%)	26	(4.9%)	24	(2.9%)	49	(9.4%)	34	(4.2%)	69	(13%)	62	(6.3%)	115	(20%)
Max	28	(5.8%)	56	(14%)	45	(8.4%)	90	(23%)	56	(15%)	112	(38%)	169	(23%)	278	(57%)

- A phase step counter — 8 LEs.
- A state machine — 4 LEs.
- A register to record the minimum slack — 8 LEs.
- Calibration offset storage for each OSM sensor — 4 LEs per sensor.

The majority of this overhead remains constant, regardless of the number of RUMs, and will be amortised in larger application circuits.

4.4.2 Timing Overhead

The additional management circuitry and forking near-critical paths to connect OSM sensors has an impact on the timing performance of the application circuit. Table 4.3 shows the f_{sta} overhead for the addition of OSM to the application circuits. Since the flow is designed to preserve the application's placement and routing, the effect on timing is limited. Forking the signal generally results in a small increase in delay, impacting on the overall model operating frequency, as it is not always possible to perfectly preserve the routing. In some cases the addition of OSM can actually improve timing estimates as in the `DCT` benchmark. Increasing the coverage should not have a significant impact on performance, as this instruments increasingly less critical registers, however the increase in routing and congestion does typically contribute to an overall reduction in f_{sta}.

Removing the LZC/shifter from memory in the `fpadd64` improves the performance of the applications f_{sta} slightly, from 142.31 MHz to 145.37 MHz (2.15%).

The timing impact of the additional fanout and loading introduced by instrumenting a RUM has been quantified experimentally to establish the true impact of this if placement and routing were perfectly maintained. Transition Probability was used to measure the delay to a OSM instrumented register, then the sensor was detached using Quartus II's ChipEditor. Without the sensor, the delay to the register was 3.507 ns, which increased to 3.516 ns when instrumented, a change of only 9.00 ps or 0.25%. The impact is likely to be so small since in an FPGA, the routing, and hence the additional fanout and loading, is already present.

Table 4.3: Timing overheads for instrumenting the functional benchmarks for OSM with coverages between 5% and 20% CDM.

Benchmark	5% CDM		10% CDM		15% CDM		20% CDM	
	Δf_{sta}(MHz)							
DCT	8.74	(5.5%)	7.32	(4.6%)	6.36	(4.0%)	8.71	(5.5%)
IIR	-0.07	(-0.07%)	-0.94	(-0.92%)	-2.01	(-1.96%)	-2.01	(-1.96%)
fpadd64	-3.36	(-2.36%)	-4.00	(-2.81%)	-4.09	(-2.87%)	2.93	(2.1%)
fpexp32	-1.96	(-2.44%)	-1.96	(-2.44%)	-1.96	(-2.44%)	-1.17	(-1.45%)
fpexp64	-0.16	(-0.19%)	-1.81	(-2.13%)	-2.46	(-2.90%)	-0.55	(-0.65%)
fplog32	-0.04	(-0.04%)	-1.04	(-1.07%)	-0.02	(-0.02%)	-0.02	(-0.02%)
fpmult32	-3.73	(-2.61%)	-6.09	(-4.26%)	-9.12	(-6.38%)	-11.27	(-7.88%)
Min	-3.73	(-2.61%)	-6.09	(-4.26%)	-9.12	(-6.38%)	-11.27	(-7.88%)
Mean	-0.08	-	-1.22	-	-1.90	-	-0.48	-
Max	8.74	(5.5%)	7.32	(4.6%)	6.36	(4.0%)	8.71	(5.5%)

4.5 Future Work

4.5.1 Mapping to Adaptive Logic Modules

With the introduction of the Cyclone V family of devices in 2013, all FPGAs from the major vendors (Xilinx and Altera) have moved away from the simple 4-LUT, including those at at the low cost and low power end of the spectrum. The Cyclone V shares architectural features with the Stratix family of devices, namely the Adaptive Logic Module (ALM). Like Xilinx's Slices these logic blocks feature a multitude of operating modes, including a being able to implement a variety of LUT sizes.

A block diagram of the new ALM is shown in Figure 4.9. While the increase in operating modes is likely to complicate the mapping of OSM to the ALM architecture, the abundance of additional registers, which can be driven by a choice of three clocks (for each LAB), offer exciting opportunities for the placement of shadow registers within the ALM containing the RUM. These registers have limited application (e.g. duplication to reduce loading and simplify routing, synchronisation and packing from functions outside the ALM) so are unused in the majority of cases and available to be employed as shadow registers. The extra routing to these shadow registers would lie within the ALM, with more deterministic delay which may reduced the necessity for calibration.

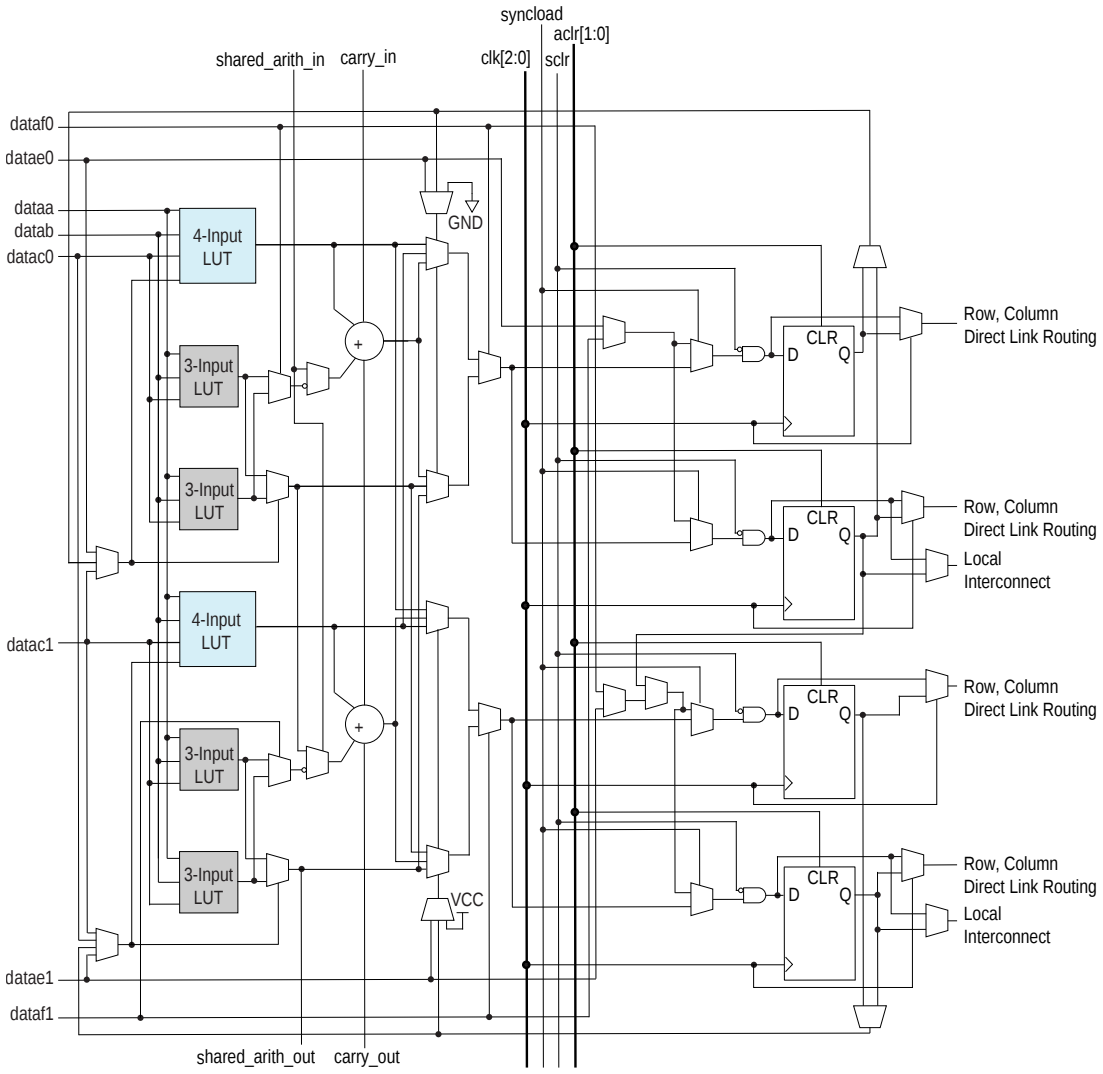


Figure 4.9: Block diagram of the Cyclone V ALM from [5].

4.5.2 Shadowing Embedded Memory

If the memory is being used as a RAM, with the same clock for reading and writing, the block memory's read-during-write behaviour can be exploited to shadow the memory, similarly to [12]. With the read-during-write behaviour configured to "New Data", data loaded into memory is available at the output port on the rising edge of the same clock cycle on which it was written. This allows the RAM to be treated as a register, where the written data could be compared to that in a shadow register with a variable phase clock, as is normal in OSM. In order for the written data to be presented on the read port, the read-enable signal must be asserted during write, additionally, the OSM sensor must only be enabled during writing operations.

4.5.3 Path Excitation

Path excitation (Section 3.5.3) allows vectors to be injected into the application circuit to stimulate known critical paths. Doing so in an ASIC requires the addition of multiplexers at some level of the circuit, which can incur a large performance and area overhead. When OSM is applied to circuits on an FPGA, the FPGAs reconfigurability can be exploited, modifying the circuit to isolate critical paths and provide access for vector injection, similarly to the technique described in [69].

4.5.4 Razor

Razor-like timing error detection, inserted automatically into arbitrary circuits on FPGAs using SMI [57] is currently being investigated. There has been some success detecting timing errors and are developing novel methods to correct these, without the area and performance overheads normally associated with Razor implementations.

4.6 Conclusion

This chapter has introduced SMI, a compile flow that can automatically insert OSM shadow registers into arbitrary circuits, currently on the Altera Cyclone IV family of FPGAs. SMI makes OSM almost as easy to use as scan-tests in VLSI circuits. The results demonstrated

that the area overhead in hardware is manageable and performance cost minimal, both a small price to pay for information provided.

This work opens opportunities for optimising FPGA designs for power and/or performance, mitigating variation and degradation issues and the reliability of circuits.

With mappings for a wide selection of commercial FPGAs, perhaps provided by the device vendors, SMI would be able to provide a turn-key approach to monitoring arbitrary circuits implemented on FPGAs for temporal timing variation.

5 Dynamic Voltage and Frequency Scaling

5.1 Introduction

In exploring OSM's ability to track changes in timing slack due to variation in clock frequency, supply voltage and external conditions such as temperature, it became apparent that OSM would be ideally suited as a feedback measurement for closed-loop dynamic voltage and frequency scaling (DVFS).

As process scaling continues, it is expected that delay variability will increase, both in magnitude of the delay variation and the speed with which degradation occurs. OSM offers a solution to monitoring this in the future, but currently the effect of is limited. DVFS the "killer app" for OSM today.

DVFS promises to alleviate some of the operating margins required to guarantee safe operation of circuits under the effect of variation. This section discusses the use of OSM for DVFS, to directly quantify the timing performance of circuits while they are operating. By feeding this information back through a controller, the voltage and/or clock frequency can be tuned in a closed-loop to optimise power efficiency, throughput or a combination. This technique offers an excellent trade-off between the degree of optimisation and implementation overhead, in terms of area and performance.

As variation increases with process scaling, operational timing margins will have to grow in order to compensate. FPGAs are particularly affected by large margins as their function is not known during device manufacturing and may change during its lifetime. The resources in the device may be used in any configuration and any combination of them may fall on a critical path.

This chapter describes a low overhead, yet accurate, DVFS system for performance and lifetime enhancement, a mechanism for calibrating and guardbanding to ensure safe and

optimised operation and controllers for a variety of operating modes including maximising throughput, power efficiency or a combination of the two and meeting interactive constraints, including power.

5.2 Background

Operational timing margins exist to ensure that a circuit continues to work across a range of conditions that impact timing performance. These parameters are built into Static Timing Analysis (STA) models and are inherently pessimistic, resulting in worst-case operation, even in better than worst-case conditions. The dominant margins in a system are detailed below:

Variation: STA must guarantee operation over all (or nearly all) possible delay variations due to process variation, including inter-die, intra-die and stochastic. The average device performs better than the STA model, but will not be able to exploit this.

Degradation: Temporal variation due to degradation must be included in STA models to ensure correct operation over the whole device lifetime. This is wasteful when a device is new, and cannot accommodate the non-deterministic nature of many degradation mechanisms.

Temperature: Switching performance varies with temperature, typically deteriorating as temperature rises. STA margins must assume the worst-case for this parameter.

Noise: Several stochastic effects influence circuit timing on a cycle-to-cycle basis, these include thermal noise, crosstalk, power supply ripple and clock jitter. To ensure correct operation STA must consider the possibility that all of these effects are compounded.

While not strictly margins, other factors that have an impact on efficiency include:

Load: A system must be able to meet throughput and latency requirements under the worst-case computational workload. If it runs under these assumptions with a varying workload, there will be idle clock cycles and wasted energy.

Data: The data delay experienced at a particular register depends on the transitions that occur on all its input nodes. Usually only the slowest of these is considered, the critical path. However, the critical path may not be exercised frequently and the average delay may be significantly faster. It may even be impossible for some physical paths to be exercised due to dependencies between their inputs.

The margins listed have differing statistical properties and the gains to be made by reducing them depend on the application. For example, a mission critical system with device must meed guaranteed timing specifications, so increasing the average throughput over many varying devices is not of much use — the slowest must still meet the specifications. Conversely, a data centre with may devices that share a workload could make significant power savings and performance gains since it is the average efficiency and throughput of the device that will determine the performance of the cluster as a whole.

5.2.1 Dynamic Voltage and Frequency Scaling

In normal use these margins result in large timing model safety guardbands, which in turn require that the device be operated more conservatively than would be necessary in all but the worst-case operating conditions. Timing models impose lower clock frequencies and higher supply voltages are used to improve worst-case delay.

$$P_{\text{total}} = P_{\text{dynamic}} + P_{\text{static}} \tag{5.1}$$

$$P_{\text{total}} = aC_{\text{L}}V_{\text{DD}}^2 f + I_{\text{leak}}V_{\text{DD}} \tag{5.2}$$

The effect of these increased supply voltages can be seen in Equations 5.1 and 5.2 [52], where a is the circuit activity factor, f the operating frequency and I_{leak} the leakage current. Dynamic power increases linearly with increasing frequency and quadratically with increasing voltage, static power linearly with increasing voltage.

Binning or Speed Grading [17] is a static technique commonly used by integrated circuit manufactures in an attempt to reduce the size of these margins. Individual or batches of dies are characterised for timing performance at a range of voltages using one of the

many delay measurement techniques. From these characterisation each die is allocated to a bin/grade. The results of the characterisation are used to inform the timing model in selecting operating frequencies and supply voltages for each bin/grade. Speed binning allows for a reduction in margins covering inter-die process and voltage variation.

DVFS uses real-time information on operating parameters (e.g. temperature, or circuit delay) to control the device's voltage and/or frequency in an attempt to conserve power (DVS) or increase throughput (DFS). This information is typically gleaned using techniques described in Section 2.4 and implementations using these techniques are detailed as follows:

Lookup Table: STA is used to evaluate multiple timing corners for different operating voltages, and these voltage frequency pairs are stored in a lookup table. The frequency can be varied to accommodate changes in load and the voltage scaled to compensate. Adding a temperature sensor or ring oscillator allows for reduction of the environmental margin. Lookup Table DVFS is the most commonly used commercially, with examples including Intel's x86 SpeedStep [55, 30] and others [63].

TDC: Time to Digital Circuits can be used to measure the performance of the die under inter-die process and environmental variation and scale the voltage to accommodate changes in delay. Since measurements are indirect some of these margins must be still be included to account for discrepancies between the behaviour of the sensor and application circuit. TEAtime [64] and [21] are examples of TDC used for DVS in arbitrary circuits. [15] demonstrates the use of TDC for DVS of arbitrary circuits on FPGAs, [45] DVS and [40] DVFS for a processor on FPGAs, and [41] DVFS in processor.

TRC: TRC achieves as TDC above, but also allow for some of the margins protecting against degradation to be reduced. [60] uses TRCs to perform DFS on a processor pipeline and [22] which uses TRC for DVS on a processor.

Failure Prediction: There are a multitude of examples of failure prediction being used for DVFS, including [13] where DVS scales a ARM processor and [46], which presents a tool for automatically adding failure prediction to circuits on an FPGA and demonstrates DVFS on an processor. Since the application circuit is being measured di-

rectly, all process, environmental and temporal margins can be removed, leaving just the margins required to avoid the introduction of timing errors due to noise. Since failure prediction provides only a pass/fail indication it limits the control mechanisms that can be used, making it difficult to avoid oscillation in voltage and/or frequency. Oscillation is detrimental to performance since there is a cost to changing the voltage (losses in the power supply) and frequency (clock stall while oscillator locks onto new frequency) so it is preferable to do so only when required by a change in circuit performance.

Razor: Both Razor [25] and RazorII [18] are DVS schemes for processors. Detecting timing errors and controlling voltage to meet a constant error rate allows for the eradication of all timing margins, including noise and worst-case data. Whilst it is an attractive prospect it has not yet found commercial application due to the various challenges in implementation. The non-deterministic nature makes it unsuitable for DVS of the tightly-coupled, deterministic applications frequently implemented on FPGAs.

The first reported example of Razor being mapped to FPGAs is [12], where it is used to dynamically scale the voltage of a processor array. The Razor flip-flop is mapped to a block memory where it is assumed that critical paths terminate. This is an interesting application which shows promise, but is far from a generic implementation of Razor that could be used with arbitrary circuits.

5.2.2 Summary

Measuring slack at the critical registers in the circuit directly, overcomes the inference required by indirect measurement methods (temperature sensors, TDC and TRC), allowing more timing margins to be reduced and improving the potential gains to be yielded in terms of power consumption or throughput.

OSM fits between failure prediction and timing failure detection (e.g. Razor) in terms of performance, implementational overhead and difficulty. It provides a full timing measurement, rather than the pass/fail indication of failure prediction, allowing for more complex control mechanisms to be implemented that converge more quickly and are stable under

constant conditions. Razor introduces errors into the circuit, these errors must be corrected, feeding the value from the shadow register back into the main path. Even in a processor, where the necessary stall circuitry already exists, this is difficult and expensive to implement. In Razor there is also a risk of metastability in the main register of the circuit, which could propagate. This is avoided in failure prediction and OSM where the timing of the main paths of the circuit are unaltered.

A comparison of various DVFS methods, including OSM, and the margins which they can reduce is shown in Table 5.1. OSM achieves most of the benefits of Razor and asynchronous circuits, except for the fast and stochastic noise and data dependence. These cannot be measured with OSM as they are a cycle-by-cycle effect which will not be accounted for as discrepancies are accumulated over many clock cycles.

Table 5.1: A summary of DVFS methods, including OSM, and the operating margins that they can reduce

Method	Load	Inter-Die Var.	Intra-Die Var.	Deg.	Temp.	Noise	Data
Lookup Table	✓						
Char.	✓	✓					
TDC	✓	✓			✓		
TRC	✓	✓		✓	✓		
Failure Prediction	✓	✓	✓	✓	✓		
Razor	✓	✓	✓	✓	✓	✓	✓
OSM	✓	✓	✓	✓	✓		

5.2.3 Life Extension through DVS

An interesting application of DVS is to increase the lifespan of a device. This is illustrated in the simulation results of NBTI degradation in Figure 5.1, based upon degradation models from [59], which was conducted by Dr. Edward Stott. The simulation assumes that there must be a certain headroom of supply voltage V_{DD} over threshold voltage V_{th} for the circuit to meet timing requirements. Normally, a static V_{DD} would include a guardband to allow for a decrease in V_{th} (pMOS is affected by NBTI: V_{th} is negative) over the lifespan of the product. Once this guardband is eroded, the device fails.

With DVS this guardband is not necessary. Instead, V_{DD} is gradually increased to follow the change in V_{th} via timing slack measurements. This means that V_{DD} is lower when the

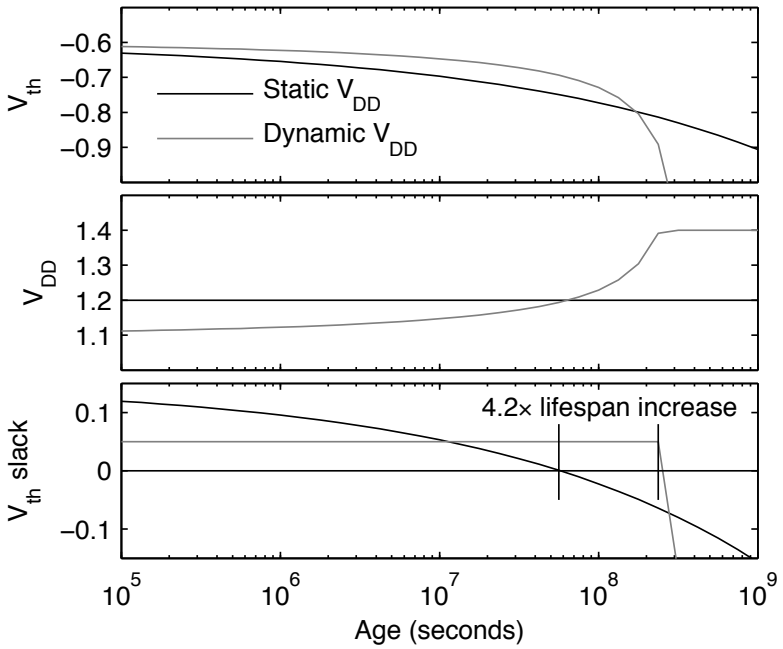


Figure 5.1: DVS can be used to extend lifespan by scaling V_{DD} to match changes in V_{th} due to degradation.

device is new, which reduces the rate of degradation. V_{DD} can also be scaled a certain amount above the normal supply voltage V_{nom}, allowing operation to continue even when the degraded V_{th} would normally mean timing failure. In this simulation a 4.2× increase in lifespan is forecast based on the point where V_{th} headroom is exhausted.

5.3 Dynamic Voltage and Frequency Scaling using OSM

This section describes dynamic voltage and frequency scaling using closed-loop control of timing slack measured with OSM.

5.3.1 Slack Measurement

Using slack measurement for DVFS, rather than general purpose measurement, necessitates some modification that is detailed in the following sections.

Measurement Resolution

The resolution of online slack measurement affects the ability to ensure that circuit remains timing safe. If a slack of 50 ps were reported, with a resolution of ±100 ps, the circuit may

Table 5.2: Resolution intervals for conservative slack measurement

Qty.	Derivation	Interval
Calibration		
$t_{\mathrm{sS,i}}$	From $d_i(t_\phi)$	$[-\Delta t_\phi(f_{\mathrm{cal}}), +0]$
$t_{\mathrm{dS,i}}$	$1/f_{\mathrm{cal}} - t_{\mathrm{sS,i}}$	$[-0, +\Delta t_\phi(f_{\mathrm{cal}})]$
$t_{\mathrm{dR,i}}$	f_{clk} sweep	$[-\Delta t_{\mathrm{cal}}, +0]$
$t_{\mathrm{RS,i}}$	$t_{\mathrm{dR,i}} - t_{\mathrm{dS,i}}$	$[-(\Delta t_{\mathrm{cal}} + \Delta t_\phi(f_{\mathrm{cal}})), +0]$
Measurement		
$t_{\mathrm{sS,i}}$	From $d_i(t_\phi)$	$[-0, +\Delta t_\phi(f_{\mathrm{clk}})]$
$t_{\mathrm{sR,i}}$	$t_{\mathrm{sS,i}} - t_{\mathrm{RS,i}}$	$[-0, +(\Delta t_\phi(f_{\mathrm{clk}}) + \Delta t_{\mathrm{cal}} + \Delta t_\phi(f_{\mathrm{cal}}))]$

not be meeting timing and be safe from errors. The main limitation to the resolution of slack measurement arises from the size of a phase step $\Delta t_\phi(f_{\mathrm{clk}})$. This is determined by the FPGAs implementation of the PLL clock generation hardware, and its configuration to produce an operating frequency. The accuracy of the shadow path offset calibration must also be considered, this is dependent on $\Delta t_\phi(f_{\mathrm{cal}})$.

Measurement and calibration can be altered so that they are conservative, ensuring that the actual slack is always greater than or equal to the measured slack. This is achieved by taking the lower or upper bounds for $t_{\mathrm{sS,i}}$, as appropriate for measurement and calibration, rather than the midpoint. In measurement, the lower slack bound is used, setting $t_{\mathrm{sS,i}}$ to be the lowest t_ϕ for which $d_i(t_\phi) = 0$. As such, $t_{\mathrm{sS,i}}$ is less than or equal to the true slack. The calibration offset, $t_{\mathrm{RS,i}}$, is subtracted to find $t_{\mathrm{sR,i}}$. In order to preserve this accuracy range, $t_{\mathrm{RS,i}}$ should be greater than or equal to the true offset. During calibration $t_{\mathrm{sS,i}}$ is set to the upper slack bound, corresponding to the lowest t_ϕ for which $d_i(t_\phi) > 0$ and the upper bound is used for $t_{\mathrm{dR,i}}$. This results in the possibility that $t_{\mathrm{sR,i}}$ may be up to $\Delta t_\phi(f) + \Delta t_{\mathrm{cal}} + \Delta t_\phi(f_{\mathrm{cal}})$ less than the exact slack, but never greater.

An updated interval analysis for conservative slack measurement as used with DVFS is given in Table 5.2.

Calibration

The calibration offset $t_{\mathrm{RS,i}}$ is affected by variation as previously discussed. While the impact of temperature changes and degradation is small, voltage has been shown to have a significant effect (Figure 3.19). When OSM is used for DVS and DVFS, calibration must be conducted at a range of voltage nodes and $t_{\mathrm{RS,i}}$ interpolated for the current V_{DD}. Since the

relationship between $t_{\mathrm{RS,i}}$ and V_{DD} is concave, a linear interpolation will underestimate $t_{\mathrm{RS,i}}$ and thus overestimate $t_{\mathrm{sR},i}$. For reasons discussed in the previous section, this is unacceptable, so measurements must be conducted at at least three voltage nodes and higher-order interpolation used. The calibration algorithm now has an outer voltage loop as shown in Algorithm 2.

Algorithm 2 Shadow register delay offset calibration for DVFS

```
for V ← V_nom to V_min do
  for all i ∈ {RUMs} do
    f_clk ← f_cal
    measure t_sS,i
    t_dS,i ← 1/f_cal − t_sS,i
    measure f_max
    f_clk ← 0.95 f_max
    done ← FALSE
    repeat
      set PLL frequency f
      measure d_i(t_φ) for 0 ≤ t_φ < 1/f
      if 0 ∈ d_i(t_φ) then
        t_dR,i ← 1/f
      else
        done ← TRUE
      end if
      f ← 1/(1/f − Δt_cal)
    until done
    t_RS,i(V) ← t_dR,i − t_dS,i
  end for
end for
```

5.3.2 Guardbanding

Without any fault tolerance or correction, it is important that the system does not scale voltage or frequency to the point where timing errors are introduced. In Section 5.3.1, OSM for DVFS is analysed, demonstrating that the actual slack during a measurement is greater than or equal to the measured slack. However, slack thresholds for voltage and frequency scaling must still include a guardband to allow for timing fluctuations that vary faster than can be measured. The guardband must be set so that if violated, the condition is detected and corrected (by adjusting voltage or frequency) before the remaining timing slack is eroded and timing failure results.

Two sources of timing fluctuations for guardbanding purposes are considered. Stochastic effects are uncorrelated and include phenomena such as power supply ripple, clock jitter and noise. Drift effects are longer term and include external temperature variation and

degradation.

The guardband calculation is:

$$t_{\mathrm{G}} = \sum_{m} a_m \quad + \quad \sum_{n} b_n (t_{\mathrm{M}} + t_{\mathrm{L}})$$

t_{G} is the guardband, t_{M} is the measurement latency and t_{L} is the control latency. a_m are the stochastic timing fluctuations, expressed as peak-to-peak variation in critical path delay, and b_n are the timing drifts, expressed as a maximum change in critical path delay per unit time. b_n are multiplied by the sum of the measurement and control latencies to obtain the maximum change in delay that could occur during the period the system needs to measure and respond to a guardband violation. All the timing fluctuation terms are summed to calculate the guardband. If any fluctuation parameter is dependent on frequency or voltage then its maximum value is taken over the operating range. During operation, the controller takes action if overall timing slack falls below the guardband slack. Hence, a slack deficit occurs when $t_{\mathrm{sC}} < t_{\mathrm{G}}$.

It is important to allow for data variation. Since timing slack measurement relies on normal operating inputs to stimulate paths, it cannot be guaranteed that the most critical paths are always stimulated. The guardband can account for this in two ways. The measurement latency, t_{M}, can be set to a period during which critical path excitation can be reasonably expected; this could be millions of clock cycles. Alternatively, an extra stochastic fluctuation factor (a_m) can be introduced that represents the maximum difference between the measured and true slack of a critical path. As indicated in Section 3.5.3, the critical paths in certain circuits may be excited only incredibly rarely; the potential techniques discussed in Section 3.6.1 may off solutions to this.

The guardband required could be generated by device vendors, based on the timing margins used for unscaled operation. Providing separate a_m and b_n parameters would allow the designer to compute the combined guardband according to the measurement and control latency used. Alternatively, characterisation of the application circuit across its corners can help inform the magnitude of a_m and b_n, and hence the size of guardband required.

The performance cost incurred by this guardband is explored experimentally in Section 5.5.3.

5.3.3 Hysteresis

The guardband defines when the circuit is running with too little slack, the lower threshold in the control regime. An upper threshold is required above which there is sufficient slack to safely reduce voltage or increase clock frequency. A hysteresis band is needed so that an adjustment does not immediately cause a guardband violation, thus resulting in constant oscillation between slack deficit and surplus. For voltage scaling the hysteresis band is defined for as:

$$t_{\mathrm{H}} = \max(\Delta t_{\mathrm{clk}}, \Delta V_{\mathrm{DD}} \frac{\mathrm{d}t_{\mathrm{sC}}}{\mathrm{d}V})$$

Where t_{H} is the hysteresis, Δt_{clk} is the period increment for frequency scaling, ΔV_{DD} is the voltage increment for voltage scaling and $\frac{\mathrm{d}t_{\mathrm{sC}}}{\mathrm{d}V}$ is the maximum sensitivity of delay to voltage. In operation, there is a slack surplus when $t_{\mathrm{sC}} > t_{\mathrm{G}} + t_{\mathrm{H}}$

Hysteresis must also take into account the accuracy of slack measurements. If while operating optimally, between the guardband and hysteresis threshold, a measurement were to underestimate the amount of slack, indicating that there was a deficit, the controller would increase the voltage or decease frequency to achieve safe operation. With an insufficiently large hysteresis threshold a surplus state could result, initiating scaling in the opposite direction and oscillation. Since the measurement accuracy is biased so as to not overestimate the amount of slack, the hysteresis threshold becomes:

$$t_{\mathrm{H}} = \max(\Delta t_{\mathrm{clk}}, \Delta V_{\mathrm{DD}} \frac{\mathrm{d}t_{\mathrm{sC}}}{\mathrm{d}V}) + \Delta t_{\phi}(f_{\mathrm{clk}})$$

Hysteresis is also used in power constrained DVFS to avoid oscillation between slack and power surplus states. Power hysteresis, P_{H}, is chosen such that when a system is using "optimal" power ($P = P_{\mathrm{set}}$), a single step of voltage or frequency made to optimise timing slack will not result in a power surplus ($P < P_{\mathrm{set}} - P_{\mathrm{H}}$). P_{H} depends on Δt_{clk} and ΔV_{DD}, plus characterised sensitivities of power to delay and power to voltage:

$$P_{\mathrm{H}} = \max(\Delta t_{\mathrm{clk}} \frac{\mathrm{d}P}{\mathrm{d}t_{\mathrm{clk}}}, \Delta V_{\mathrm{DD}} \frac{\mathrm{d}P}{\mathrm{d}V}) + \Delta t_{\phi}(f_{\mathrm{clk}})$$

More sophisticated control mechanisms may be able to achieve oscillation free operation without the need for a hysteresis guardband, but the simple ad hoc controller described herein as a proof of concept still attains significant improvements.

5.3.4 Control Algorithm

The general form of the DVFS system is illustrated in Figure 5.2. The state of the application circuit, its slack and/or power, is continually measured and compared to a setpoint, with adjustments made as necessary. The controller can operate in one of three modes, depending on whether an external constraint is given for voltage, throughput or power. The modes are listed in Table 5.3 with their optimisation objectives and control actions.

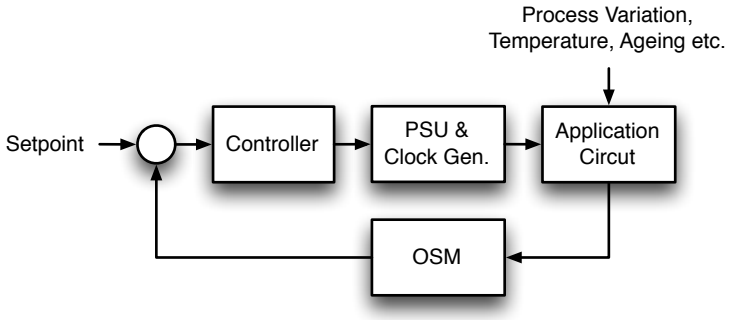


Figure 5.2: Block diagram of DVFS controller

Three forms of controller have been designed for different applications, they are: statically constrained, dynamically constrained and power constrained.

Table 5.3: The system can be controlled in one of three modes, depending on which parameter is constrained

Mode	Constraint	Goal	Scaling
DVS	Throughput	Min. Power	Voltage
DFS	Voltage	Max. Throughput	Frequency
DVFS	Power	Max. Throughput	Frequency and Voltage

Statically Constrained

The most simple controller optimises operation under static constraints. Either voltage or frequency are fixed, typically at the nominal voltage or timing model frequency. If the frequency is fixed, the voltage is reduced until the timing slack reaches a minimum safe level, resulting in a minimum power consumption for the required throughput. With voltage fixed the frequency adjusted achieving a maximum throughput. Power efficiency or throughput are maximised by removing most of the timing margin. Voltage or frequency is adjusted to track any changes in circuit timing from sources such as temperature variation or degradation.

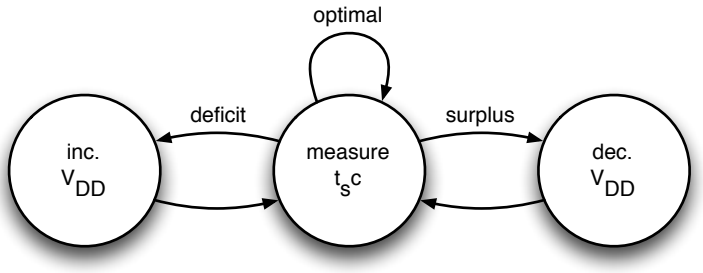


Figure 5.3: State diagram for dynamic voltage scaling with static throughput constraint

The statically constrained controller is a simple stepping algorithm, shown in Figure 5.3 for DVS mode. Timing slack for the application circuit (t_{sC}) is monitored on a continual basis. Voltage is increased (or frequency decreased) on detection of a slack deficit. Surplus slack is removed by reducing the voltage (or increasing frequency). In the case of DVS, hard limits are imposed on the minimum and maximum voltage, to ensure the FPGA remains functional and isn't damaged.

Dynamically Constrained

Systems with a number of voltage-throughput operating points commonly use traditional STA lookup based DVFS. As computational workload varies, voltage and frequency are scaled in tandem to reduce power consumption. In this implementation, voltage or frequency is set according to a system constraint, while the other parameter is scaled freely to optimise timing slack.

Dynamic constraints may change frequently, too quickly for the stepping controller to

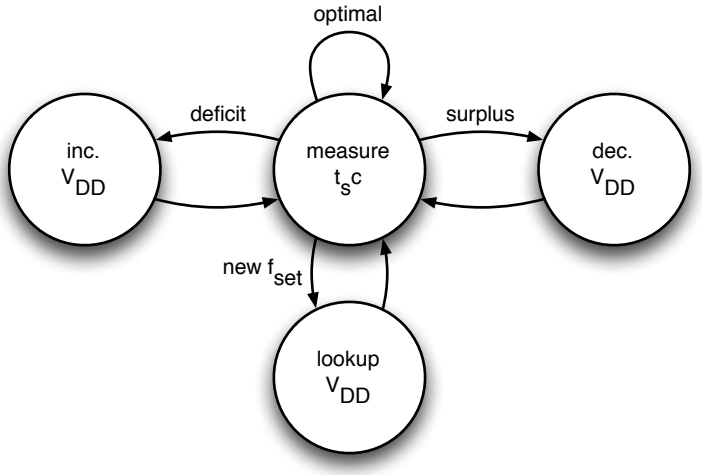


Figure 5.4: State diagram for dynamic voltage scaling with dynamic throughput constraint

track effectively. The ordering in which parameters are set is also important, e.g. if voltage is reduced, clock frequency must first be lowered to a safe level. To address this, the controller uses a table to make coarse changes to voltage and frequency, then making fine adjustments based on slack measurements. The table is generated from offline characterisation over the expected operating corners, with a suitable guardband added to allow for potential inaccuracies, such as degradation. This table could be updated during run-time with data from slack measurements.

The dynamically constrained DVS controller is shown in Figure 5.4. t_{sC} is measured and responded to as with the statically constrained controller. When a frequency constraint (f_{set}) is changed, the lookup table is consulted for the associated near-optimal voltage and the circuit configured in a specific order; frequency first when the frequency decreases and voltage first when the frequency increases.

Power Constrained

DVFS can be used to make a system operate with maximum throughput, within a power envelope. This can be achieved to some extent using a model or table that estimates power from voltage and frequency parameters, but inaccuracy arises due to power's dependence on data and other factors. Table 5.4 details the effect on slack and power with changes to voltage and frequency variables. Power rises and falls proportionally with both voltage and

frequency, but crucially slack responds in opposing directions. This allows a closed-loop controller to freely adjust both frequency and voltage in order to meet a power constraint.

Table 5.4: Control variables and responses for power constrained DVFS

Variable	Action	Slack Resp.	Power Resp.
Voltage	↑	↑	↑
	↓	↓	↓
Frequency	↑	↓	↑
	↓	↑	↓

The power constrained controller is illustrated in Figure 5.5. As with the other dynamically constrained controller modes, continuous measurements of t_{sC} are made, and changes in power constraint are made using a table lookup. The power constraint P_{set} is used to index a voltage and frequency pair which provide a power close to, but less than P_{set} and a guaranteed safe timing slack.

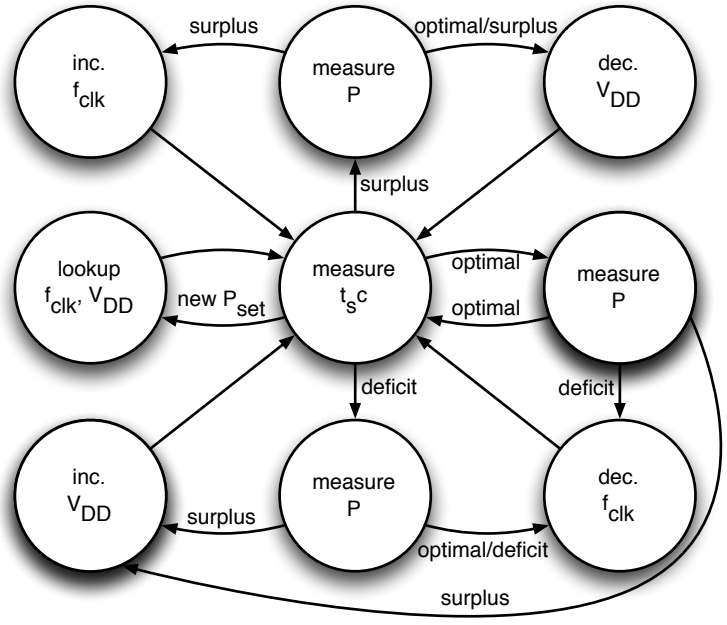


Figure 5.5: State diagram for dynamic voltage and frequency scaling with dynamic power constraint

The controller for power constrained operation is more complex than the other dynamically constrained modes. Power measurements as well as t_{sC} must be considered and with voltage and frequency unconstrained there is an additional output parameter to control. Control adjustments are made to eliminate any surplus or deficit in one input variable

without tending the other towards deficit. For example, if t_{sC} is optimal and P is in deficit, then f_{clk} will be reduced as this will reduce power consumption, without reducing t_{sC}. Conversely, if t_{sC} is optimal and P is in surplus, V_{DD} will be increased. The effect of the two outputs outputs are opposing, with V_{DD} affecting t_{sC} and P in the same direction, and f_{clk} affecting them in opposite directions.

5.4 Experiment

The "functional" set of benchmark circuits was used to investigate the effectiveness of DVFS using OSM. The circuits were instrumented automatically with a CDM of 10% using SMI. Experiments were conducted using RIPPL, which facilitates interfacing with the slack measurement circuitry, controlling clock frequency and core voltage, varying the FPGA's package temperature and provides the ability to accurately measure power consumption.

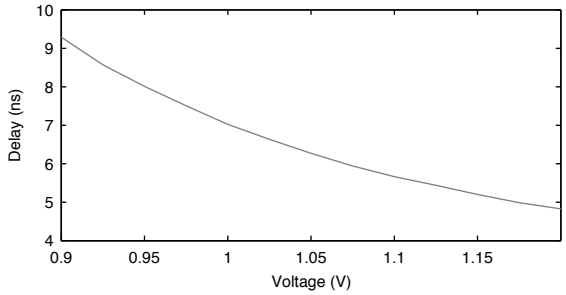
The benchmark circuits were fed with pseudo-random uniform input stimuli from an LFSR, which is free running during the period of the measurements. Each phase was measured over 200 million clock cycles (tuned empirically) to provide a high probability of critical path excitation. At a 150 MHz operating frequency this corresponds to 1.33 s and with a maximum resolution PLL configuration achieving a 70 step phase sweep, the total measurement latency is around 95 s.

The DVFS controller was implemented as a TCL script that integrates with RIPPL, constantly monitoring slack at the instrumented registers in the circuit and reacting, as described in Section 5.3.4, with voltage and/or frequency steps as appropriate. Temperature profiles and dynamic constraints can be applied independently and the response observed.

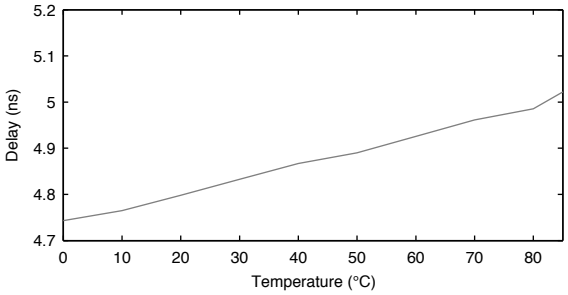
Experiments were conducted as per Section 3.3. The FPGA is able to operate down to 0.9 V from the nominal 1.2 V. Below this, failure of the power-on reset, configuration and PLL circuitry prevents operation. The core voltage has not exceeded nominal during these experiments so to avoid excessive degradation and potential damage.

5.4.1 Characterisation

A series of experiments were conducted, profiling benchmark circuits under a range of conditions in order to establish suitable parameters for the guardband, hysteresis and step size required by the closed-loop DVFS controller. Delay measurements were performed using offline frequency-sweep techniques in order to give the greatest measurement accuracy, and power measurements using the core voltage sourcing PSU.



(a) Voltage

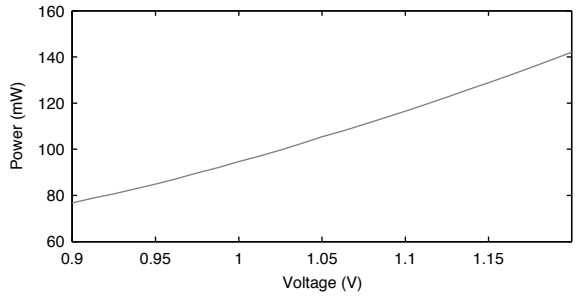


(b) Temperature

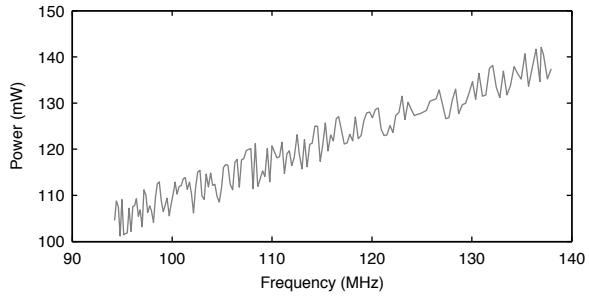
Figure 5.6: The effect of voltage and temperature variation on critical path delay in `fpmult32`.

The impact of voltage variation on delay is explored in Figure 5.6a. The voltage is varied between nominal of 1.2 V and 0.9 V, the lowest voltage at which the FPGA functions reliably, and with the circuit clocked at f_{sta}. Across this range the delay varies by 4.47 ns, with a maximum increase in delay of 92.74%. The package temperature was varied across the device corners and delay measured, the results of which is shown in Figure 5.6b. Temperature has a weaker effect on delay, with an overall increase of 279 ps from 0 °C to 85 °C (5.9%).

Both operating voltage and frequency exhibit a strong effect on power consumption. Voltage and power have a quadratic relationship, with power quickly falling off as voltage



(a) Voltage



(b) Frequency

Figure 5.7: The effect of voltage and frequency variation on power consumption in `fpmult32`.

is reduced (Figure 5.7a). Reducing core voltage from 1.2 V to 0.9 V decreases power by 65.36 mW (45.26%). The response to voltage variation is more linear. Figure 5.7b shows the power consumption as the operating frequency is lowered from the timing model frequency (f_{sta}) of 137.93 MHz to 104.06 MHz, resulting in a power reduction of 33.28 mW (24.09%). The noise superimposed on this measurement is due to the PLL voltage controlled oscillator, which can produce a significant power variation when producing similar output frequencies and contributes to ± 3 mW of the overall power consumption.

5.4.2 Guardband, Step Size and Hysteresis

Since device vendors do not currently provide the necessary timing margin information required to establish reasonable values for these parameters, these had to be derived through characterisation and empirical tuning of parameters during runtime.

Guardband: The guardband takes into account stochastic and drift effects in order to provide an assurance of safe operation under varying parameters. The impact of the stochastic noise effects is difficult to quantify, but clock jitter in Cyclone architecture

devices has been characterised to have a magnitude of the order of 80 ps [69]. Drift effects include degradation, which is negligible over the measurement and control latency timeframes and temperature. The effect of temperature on delay across the device corners is linear, with a variation of approximately 3.12 ps/°C. A guardband of 150 ps has been selected for these experiments, which provides sufficient coverage for the stochastic variation and a temperature fluctuation of over 20 °C during each measurement and control cycle.

Step Size: When the controller makes voltage or frequency steps, it is desirable that they result in a measurable change in slack. The slack measurement resolution in this Cyclone IV implementation varies between 96.15 ps and 208.33 ps depending on PLL parameters. During frequency scaling, a frequency step directly corresponding to this range can be made. Voltage scaling is more complex as voltage changes cause non-linear changes in slack. At the extreme (stepping down from nominal voltage) the slack measurement resolutions corresponds to a voltage step of between 19 mV and 31 mV. Elsewhere in the voltage range this step will produce a smaller response. The experiments conducted use a frequency step equivalent to a change in clock period of 75 ps, and a voltage step size of 5 mV. These result in changes slightly smaller than the measurement resolution, so several steps may occur before a change in slack is detected.

Hysteresis: The slack hysteresis threshold depends on the variation in slack due to frequency or voltage steps, and the measurement resolution. Combined these require a hysteresis of between 192.30 ps and 417.66 ps, depending on PLL configuration and step size decisions. Experimentation has shown that oscillation free operation can be achieved with a smaller threshold, in part due to the fact that a number of near-critical paths with similar delay are being measured, reducing the impact of measurement resolution. In the DVFS experiments a hysteresis threshold of 100 ps has been used for DVS and DFS operation, and 200 ps for power constrained DVFS, which is compounded by the additional parameters.

Power Threshold: Power constrained DVFS requires an addition threshold, defining the

optimal region for power. This depends on the voltage or frequency effect on power and any associated noise. The power response to frequency is linear, with a 96.15 ps step resulting in a change in 5.0 mW, and a 208.33 ps step 10.6 mW. A 1.1 mW hysteresis band is sufficient for the 75 ps step used in the experiments. The impact on voltage varies, with a maximum variation of 5.1 mW for a 19 mV step and 8.1 mW for 31 mV. The 5 mV step used for these scaling experiments corresponds to a maximum variation in power of 1.4 mW. The voltage controlled oscillator induced noise must also be taken into account, requiring that the hysteresis is increased by 6 mW to 7.4 mW.

The theoretical parameters and those established by empirical tuning and used for the experiments which follow are shown in Table 5.5.

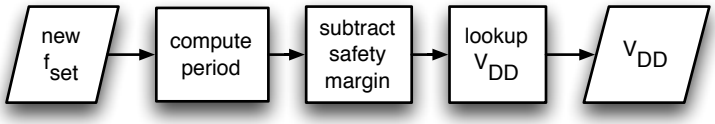
Table 5.5: Theoretical and empirically tuned controller parameters

Parameter	Theoretical	Tuned	
		DVS/DFS	DVFS
Guardband	150 ps	150 ps	150 ps
Step Size	19 mV/96.15 ps	5 mV/75 ps	5 mV/75 ps
Slack Hysteresis	192.30 ps	100 ps	150 ps
Power Hysteresis	7.5 mW	-	7.5 mW

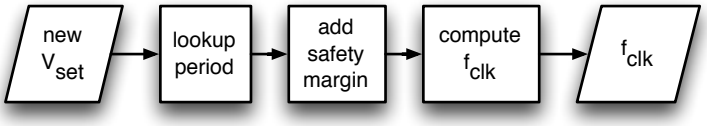
5.4.3 Adaptive Scaling Lookup

The lookup table used to accelerate the adaptive operating modes requires experimental measurements of circuit delay and power consumption over a range of voltage nodes for each circuit. The table consists of voltage, delay and power values, with power measurements made at a frequency corresponding to the reciprocal of the circuit delay. This table can be constructed from data produced during the calibration process, thus requiring no additional measurements. Calibration measures the delay of the RUMs in the circuit ($t_{\mathrm{dR,i}}$) at three voltage nodes, using an offline frequency sweep, taking the maximum of these delays gives the overall circuit delay. Performing a current measurement during the sweep provides all of the necessary data, from which the table can be interpolated.

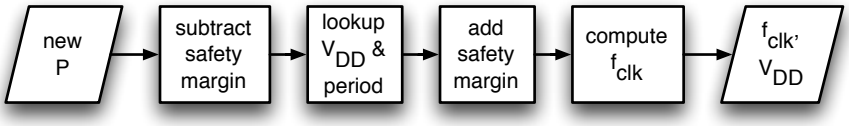
The table does not need to be accurate, since fine tuning is conducted by the controller, but must be conservative, so that the circuit operates safety with a surplus of slack, and



(a) Adaptive DVS



(b) Adaptive DFS



(c) Adaptive DVFS

Figure 5.8: Adaptive lookup for different operating modes

where applicable power, when values from the table are used. A linear interpolation of the required values from the measured parameters is conservative for all of the required parameters. Voltage and delay have a concave increasing relationship: interpolating a delay for a given voltage overestimates the delay and so underestimates the maximum operating frequency, interpolating a voltage for a delay overestimates the voltage, both conservative. Voltage and power (with circuit clocked at $f_{\max}$) is convex increasing: linear interpolation of voltage from power underestimates the voltage, resulting in a power level less than or equal to the specified.

A safety margin must be applied to the values interpolated from the table, to take into account any variation since the production of the table. This is applied to the clock period or power, allowing it to map directly. The process for lookup and margining of the different parameters is shown in Figure 5.8. The DVFS experiments use a period safety margin of 400 ps and a power safety margin of 20 mW.

5.5 Results

The overhead for the benchmark circuits is as per Section 4.4, no additional circuitry is used as the controller is implemented in software on the tethered host machine for ease

of experimentation. The controller was configured with parameters deduced through the characterisation. The following sections detail the results for the various operating modes.

5.5.1 Voltage and Frequency Scaling

Closed-loop DVFS with online timing slack measurement promises to reduce the overheads associated with timing model margins. Figure 5.9 and 5.10 show how throughput and power respectively can be improved using the proposed technique. Figure 5.10 demonstrates DFS, with the core voltage fixed at the nominal (V_{nom}) 1.2 V assumed by the timing model, and clock frequency is scaled to optimise timing slack. Throughputs from the circuit operating at timing model frequency (f_{sta}) are compared to throughputs under DFS at two temperature nodes. For this particular device, there is a substantial increase in throughput for all benchmarks. Delay is slightly lower at 27 °C than 85 °C and the DFS controller is able to adapt to this, running at a higher frequency and thus achieving an increased throughput.

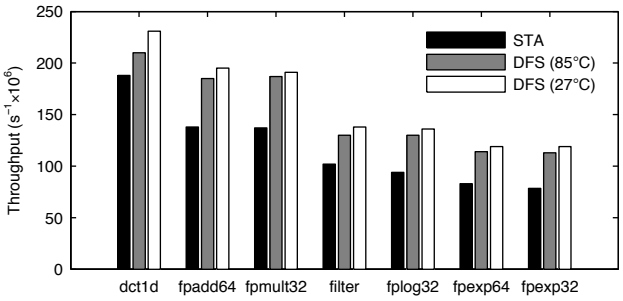


Figure 5.9: Throughput comparison between nominal and DFS. Benchmarks achieve a mean improvement of 38.9% at 27 °C and 30.7% at 85°C

Figure 5.10 demonstrates the corresponding data in DVS mode, with the clock fixed at the f_{sta} of the original uninstrumented circuit and the voltage scaled to optimise timing slack. The chart displayed the power consumption of benchmarks at V_{nom} core supply and dynamically scaled voltage. There are significant reductions in power using DVS. Like throughput, the power scales with temperature. Some correlation would be expected anyway, due to leakage current, but the effect is significantly amplified by voltage scaling.

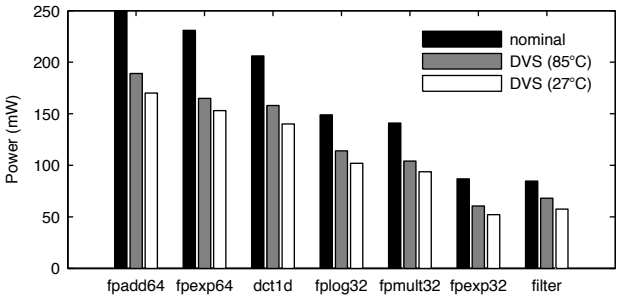


Figure 5.10: Operating power comparison between nominal and DVS. Benchmarks achieves a mean improvement of 33.5% at 27 °C and 24.9% at 85°C

5.5.2 Adaptive Scaling

In this section, the ability of the controllers to adapt to changing external conditions is tested. The static-throughput DVS controller is configured to the timing model frequency f_{sta} and voltage adjusted to optimise timing slack. Figure 5.11 shows the transient response of the system to changing temperature. The FPGA package's temperature is ramped up and down over the full specified operating range according to an external schedule. The temperature's rate of change is configured to 0.1°C/second, faster than would occur due to a change in environmental temperature in normal operating conditions. As temperature increases, timing slack t_{sC} drops below the guardband. The controller responds immediately, increasing the voltage and restoring slack. The reverse occurs when temperature is dropped, slack increases and voltage is reduced. The controller would respond in a similar way to degradation, which manifests as a variation in circuit delay, observable as a change in slack, compensated for by adjusting core voltage.

Systems with a dynamic voltage and throughput constraint are examined next. Figure 5.12 illustrates a system responding to a changing throughput constraint. The DVFS controller consults a table to set the voltage to a conservative level and slack measurements are then used to back off the voltage until optimal timing is achieved, consequently minimal power is used to meet each throughput requirement. Figure 5.13 shows the converse system where voltage is set according to an external schedule. Frequency is set to the safe value from the lookup table increased according to slack measurements to achieve a maximum throughput while achieving the timing slack guardband.

Dynamic power constrained, the most complex DVFS controller is demonstrated in Fig-

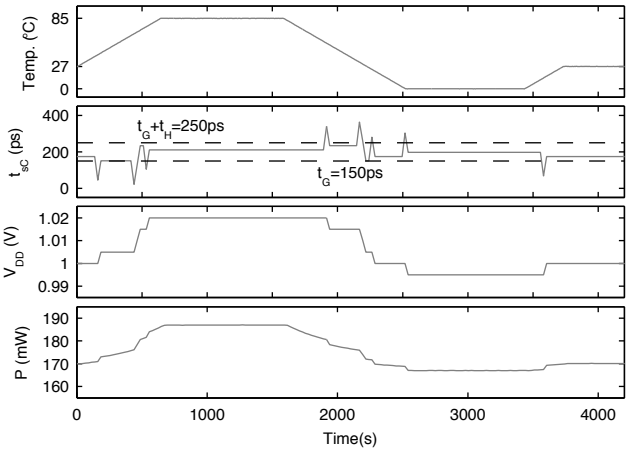


Figure 5.11: Transient response to temperature fluctuation in DVS with `fpmult32` at 27°C

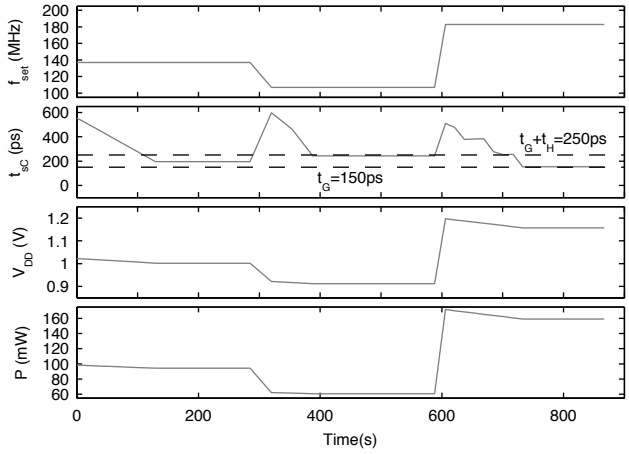


Figure 5.12: Transient response to throughput requirements in DVFS with `fpmult32` at 27°C

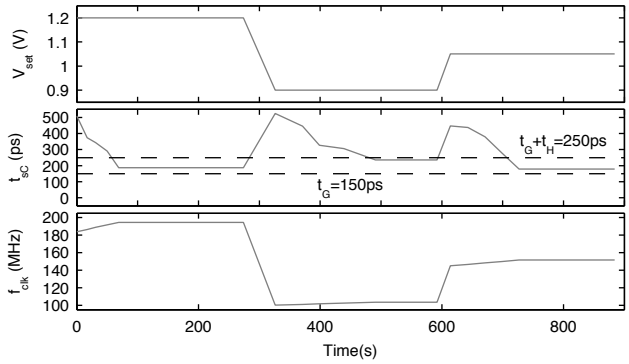


Figure 5.13: Transient response to voltage requirements in DVFS with `fpmult32` at 27°C

ure 5.14. Like the previous modes, an external constraint, power, is set according to an arbitrary schedule. At each power request the table is consulted to find a safe voltage and frequency operating point. The control process in Figure 5.5 is followed with the input parameters adjusted to maximise throughput while respecting the timing slack guardband and power constraint.

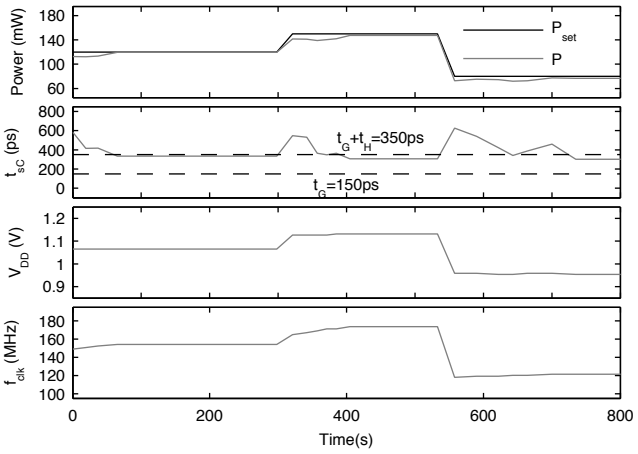


Figure 5.14: Transient response to power requirements in DVFS with `fpmult32`

5.5.3 Guardband Cost

The requirement to impose a guardband in order to avoid the occurrence of timing faults due to rapid changes in delay results in dynamically scaled circuits operating conservatively. While this conservativeness is small as compared to that require for circuits without any DVFS, there is a performance cost nonetheless. In DFS, the effect of the guardband is a reduction in operating frequency as compared to the optimum configuration of the circuit, operating at the cusp of timing failure ($f_{\max}$). The reduction is equivalent to reducing this $f_{\max}$ by an amount equivalent to a guardband sized increase in clock period.

The effect of the guardband on power consumption in DVS circuits is more complex due to the non-linear relationships between the increase in voltage required to achieve the guardband and the increase in power due to elevated voltage. This is explored in Figure 5.15, where the power reductions achieved by DVS of `fpadd64` are given for a variety of guardbands. With a 0 ps guardband (circuit operating just before the point of first failure), the power saving is 49.0%. Introducing a 75 ps guardband results in a saving

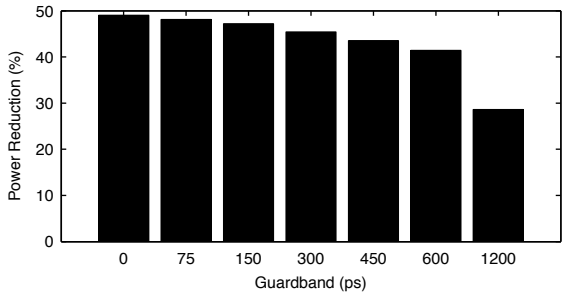


Figure 5.15: Power reduction over nominal circuit operation with DVS of `fpadd64` for a variety of guardbands at 27 °C.

of 48.1%, an increase of 1.7% over operating the circuit with optimum parameters. Further increasing the guardband to 150 ps, as used in these experiments reduces the power saving to 47.2%, 3.4% greater than optimum operation. A 300 ps guardband reduces overall power by 7.1%, but requires 3.5% more power than optimal operation. Both of these trends are quadratic with most of the benefit of optimum operation achieved despite a small guardband, and a large guardband having a significant cost.

5.6 Practical Implementation

Clock generation is already provided by on board PLLs, for the purpose of these experiments the PLL is reconfigured each time a new frequency is required, which stalls the system until the voltage controlled oscillator locks onto the new frequency. It is possible to pre-select the new frequency on a different PLL and use one of the FPGA's clock multiplexers to seamlessly switch between the two clocks in a single cycle.

Only the core supply to the FPGA is varied. This powers soft-logic, DSPs and BRAM, leaving clock management and I/O unaffected. While the experiments presented here have used external power supply equipment, the hardware required to implement DVFS is already available on some of the new high-performance FPGA reference boards containing Altera Stratix and Xilinx Virtex and Zynq devices. These boards use power supplies compatible with the PMBus protocol, which allows the FPGA to vary supply voltage for each of the input rails and measure power over an I^2C bus.

A standard power supply can be modified to allow FPGA control by replacing the voltage feedback resistor with a digital potentiometer. An example of this power supply

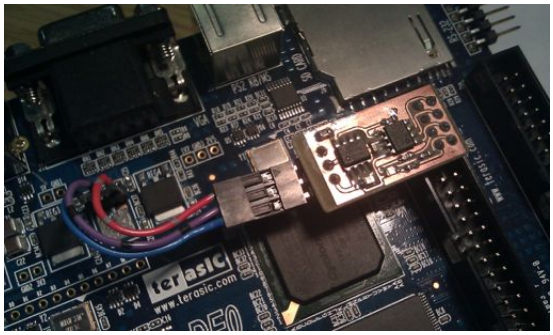


Figure 5.16: Stand-alone power supply with digital potentiometer allowing FPGA self-control of core voltage.

configuration is shown in Figure 5.16, here the power supply on a Terasic DE0 has been replaced with an auxiliary variable supply allowing the FPGA control over its core voltage. Including a current sense resistor and ADC allows the FPGA to measure its own power consumption. The inclusion of these few components to the FPGA board allows all of the DVFS techniques to be conducted at a small additional hardware cost.

The measurement latency can be reduced significantly by measuring at some, rather than all, of the phase steps. The full timing information can be found by measuring at each phase step until all of the shadow registers have detected a discrepancy. Another option is to measure just the phases of interest, limiting the sweep to a range of slacks around operating region. A subsequent implementation of the measurement and controller using this technique has reduced measurement latency to less than 20 s.

Alternatively, measurement can be configured like a two level failure prediction, measuring slack at the guardband and hysteresis thresholds. The different calibration offsets complicate this, requiring that more than just two phase leads are used. This is sufficient for the simple stepping controller described here, but since only a pass/fail indication is provided at the two levels, it does not allow for a more sophisticated control method, which takes into account how far the control variable is from a setpoint.

5.7 Future Work

All of the DVFS control modes demonstrate that DVS using OSM is able to safely maximise throughput or power efficiency under changing external conditions and operating

constraints. Despite using a lookup table to accelerate large steps in operating points, the controller can still take some time to converge on an optimum configuration. A reduction in measurement latency as described above would allow for more steps to be made in the same period of time, accelerating convergence. The primary focus of future work is to utilise a more advanced control technique that would reach optimal operation in fewer cycles.

5.8 Conclusion

In this chapter, dynamic voltage and frequency scaling using online slack measurement in FPGAs has been demonstrated. This method can better reduce operating margins than scaling techniques that rely on static timing analysis or inferring circuit performance using timing inference circuits (e.g. TDC or TRC), yet it requires a far smaller overhead than more intrusive methods like Razor and asynchronous design.

Techniques for guardbanding the slack measurements to ensure safe operation have been shown, and controllers that use timing information feedback to improve circuit throughput or efficiency developed. This DVFS system has a range of applications including power efficiency improvement, lifespan extension and system level dynamic operating constraints.

The method has been demonstrated on a number of benchmark circuits, running an a Altera Cyclone IV FPGA. Statically constrained DVS and DFS show the potential to yield significant improvements in power and throughput respectively, even at worst-case temperature corners. It is expected that even greater improvements would be achieved on devices using a smaller, high-performance process. The controllers can also adapt to changes in voltage or throughput constraints and achieve optimised operation under varying workload conditions. A more advanced system, power-constrained DVFS, provides optimised operation within a defined and variable power envelope by scaling both the voltage and frequency.

DVFS using OSM added by automatically with SMI is an almost turn-key approach to DVFS on FPGAs. It takes in a hardware description of an application circuit and outputs an instrumented and calibrated bitstream, with bindings to a voltage and frequency scaling

framework, with different controller modes offering a solution to most dynamic scaling requirements. It offers a comprehensive solution for dynamic scaling of arbitrary circuits on FPGAs with indicative improvements 25% to 40% as a reduction in power consumption or increased throughput.

6 Conclusions

The ability to determine the "health" of an operating circuit opens up the possibility for significant improvements in a variety of areas including reliability, power consumption and timing performance.

This thesis has described a novel online timing slack measurement technique, capable of accurately measuring the timing slack at selected registers in a circuit, without affecting the circuit's functionality. A calibration method improves absolute accuracy and frequency dithering of the clock facilitates resolution improvement. A method for selecting what registers in the circuit need to be instrumented has been described, and how the delay distribution of different circuits affects the overhead of instrumentation explored.

This measurement is capable of detecting delay variation due to process, environmental and temporal variation, which is confirmed through a series of experiments.

Continually measuring slack allows for the "health" of a circuit to be monitored throughout its life, tracking variations in delay and enabling triggering of pre-emptive action or warning of impending timing failure.

The necessary circuitry to perform the measurements can be added to arbitrary circuits implemented on FPGAs (currently Altera Cyclone family) using the Slack Measurement Insertion compile flow. This makes using the OSM method almost as easy as scan-tests used for VLSI circuits, requiring little-to-no manual intervention.

Combining the measurement method and compile flow in a closed-loop dynamic voltage and frequency scaling system provides an almost turn-key solution to the reduction of timing safety margins, and is capable of yielding improved power consumption or throughput performance. Experiments with a variety of application circuits show significant indicative improvements of between 25% and 40% as a reduction in power consumption or increased

throughput, with the promise of even greater improvement in smaller, higher performance devices, offering a comprehensive solution for dynamic scaling of arbitrary circuits on FPGAs.

OSM fits into the context of existing timing measurement methods, which have been reviewed in Section 2.4. It combines various attributes of these techniques providing direct, online measurements, in arbitrary circuits, which are continuous, and accurate with reasonable resolution and low overhead, both in terms of area and performance. While the existing techniques each have some of these capabilities, none combine them all.

However, further work is required in order to make these novel techniques widely usable and robust. This has been described in each of the three technical chapters (Sections 3.6, 4.5 and 5.7). The primary limitation of the base OSM method is the problem of critical path excitation as discussed extensively in Section 3.5.3. Because measurements are made online, only paths which are excited by the circuit's input data can be measured, this input data may excite critical paths infrequently or never. Possible solutions to this are discussed in Section 3.6.1.

The use of the SMI tool is currently restricted to the soft-logic of one family of FPGAs from a single manufacturer. However, the principle of OSM is generally applicable; for global utilisation in FPGAs it must either be mapped to each FPGA architecture and a tool implemented for the different manufacturer EDA tools, or integrated directly into the fabric of the devices themselves. Adoption to ASICs would require transistor level design of the sensors and tool support. Manufacturers integrating OSM more closely into the low level chip design may provide insight into some of the parameters relating to instrumentation coverage and margins for DVFS control.

Finally, more sophisticated DVFS control algorithms that make full use of the continuous timing measurements provided by OSM would achieve performance improvements over the ad hoc control used.

The work described in this thesis directly addresses current concerns of the semiconductor industry (as highlighted by the ITRS) in providing sensors, which can be used to improve the robustness and performance of a system, despite increasing variability and reliability concerns, and demonstrating these sensors can be easily implemented and used.

Bibliography

[1] A. Agarwal, V. Zolotov, and D. T. Blaauw. Statistical clock skew analysis considering intradie-process variations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(8):1231–1242, 2004.

[2] M. Agarwal, V. Balakrishnan, A. Bhuyan, B. Paul, and S. Mitra. Optimized Circuit Failure Prediction for Aging: Practicality and Promise. In *Proc. IEEE International Test Conference*, pages 1–10. IEEE, Oct. 2008.

[3] Altera. Quartus University Interface Program (QUIP). 2005.

[4] Altera. Cyclone IV Device Handbook, 2013.

[5] Altera. Cyclone V Device Handbook. 2014.

[6] A. Amouri and M. Tahoori. A Low-Cost Sensor for Aging and Late Transitions Detection in Modern FPGAs. In *Proc. International Conference on Field Programmable Logic and Applications*, pages 329–335. IEEE, Sept. 2011.

[7] P. Bardell, W. McAnney, and J. Savir. *Built-In Self-Test for VLSI: Pseudorandom Sequences*. John Wiley & Sons, Somerset, New Jersey, 1987.

[8] V. Bexiga, C. Leong, J. Semiao, I. Teixeira, J. Teixeira, M. Valdes, J. Freijedo, J. Rodriguez-Andina, and F. Vargas. Performance Failure Prediction Using Built-In Delay Sensors in FPGAs. *Proc. International Conference on Field Programmable Logic and Applications*, pages 301–304, Sept. 2011.

[9] S. Bhunia and S. Mukhopadhyay, editors. *Low-Power Variation-Tolerant Design in Nanometer Silicon*. Springer US, Boston, MA, 2011.

[10] D. Boning and S. Nassif. Models of process variations in device and interconnect. *Design of High Performance Microprocessor Circuits*, pages 98–118, 2000.

[11] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter variations and impact on circuits and microarchitecture. *Proceedings 2003. Design Automation Conference (IEEE Cat. No.03CH37451)*, (mV):338–342, 2003.

[12] A. Brant, A. Abdelhadi, D. H. Sim, S. L. Tang, M. X. Yue, and G. G. Lemieux. Safe Overclocking of Tightly Coupled CGRAs and Processor Arrays using Razor. In *Proc. Annual International Symposium on Field-Programmable Custom Computing Machines*, pages 37–44. IEEE, Apr. 2013.

[13] T. Burd, T. Pering, A. Stratakos, and R. Brodersen. A dynamic voltage scaled microprocessor system. *IEEE Journal of Solid-State Circuits*, 35(11):1571–1580, Nov. 2000.

[14] G. Chen, K. Chuah, M. Li, D. Chan, C. Ang, J. Zheng, Y. Jin, and D. Kwong. Dynamic NBTI of PMOS transistors and its impact on device lifetime. In *Proc. IEEE Symposium on International Reliability Physics*, pages 196–202. IEEE, 2003.

[15] C. Chow, L. Tsui, P. Leong, W. Luk, and S. Wilton. Dynamic voltage scaling for commercial FPGAs. In *Proc. IEEE International Conference on Field-Programmable Technology.*, pages 173–180. IEEE, 2005.

[16] P. Clarke, A. Ray, and C. Hogarth. ElectromigrationâĂŤa tutorial introductionâĂă. *International Journal of Electronics.*, 69(3):333–338, Sept. 1990.

[17] B. Cory, R. Kapur, and B. Underwood. Speed binning with path delay test in 150-nm technology. *IEEE Journal of Design & Test of Computers*, 20(5):41–45, Sept. 2003.

[18] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw. RazorII: In Situ Error Detection and Correction for PVT and SER Tolerance. *IEEE Journal of Solid-State Circuits*, 44(1):32–48, Jan. 2009.

[19] F. de Dinechin, C. Klein, and B. Pasca. Generating high-performance custom floating-

point pipelines. In *Proc. International Conference on Field Programmable Logic and Applications*, pages 59–64. IEEE, Aug. 2009.

[20] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. LeBlanc. Design of ion-implanted MOSFET's with very small physical dimensions, 1974.

[21] S. Dhar, D. Maksirnovi, and B. Kranzen. Closed-loop adaptive voltage scaling controller for standard-cell ASICs. *Proc. International Symposium on Low Power Electronics and Design*, pages 103–107, 2002.

[22] A. Drake and R. Senger. A distributed critical-path timing monitor for a 65nm high-performance microprocessor. *Proc. IEEE International Conference on Solid-State Circuits*, pages 398–399, Feb. 2007.

[23] P. Dudek, S. Szczepanski, and J. Hatfield. A high-resolution CMOS time-to-digital converter utilizing a Vernier delay line. *IEEE Journal of Solid-State Circuits*, 35(2):240–247, Feb. 2000.

[24] F. Elguibaly and M. El-Kharashi. Multiple-input signature registers: an improved design. In *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, volume 2, pages 519–522. IEEE, 1997.

[25] D. Ernst, S. Das, S. Pant, R. Rao, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: a low-power pipeline based on circuit-level timing speculation. In *Proc. International Symposium on Microarchitecture.*, pages 7–18. IEEE Comput. Soc, 2003.

[26] A. Gacic, M. Puschel, and J. Moura. Fast automatic software implementations of FIR filters. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages II–541–4. IEEE, 2003.

[27] C. Guerin, V. Huard, and A. Bravaix. The Energy-Driven Hot-Carrier Degradation Modes of nMOSFETs. *IEEE Transactions on Device and Materials Reliability*, 7(2):225–235, June 2007.

[28] D. Harris and S. Naffziger. Statistical clock skew modeling with data delay variations. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 9(6):888–898, 2001.

[29] R. T. Howe and C. G. Sodini. *Microelectronics: an integrated approach*. Prentice Hall, Upper Saddle River, NJ USA, 1997.

[30] Intel. Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor, 2004.

[31] ITRS. International Technology Roadmap for Semiconductors, 2011.

[32] K. Katoh, T. Tanabe, H. M. Zahidul, K. Namba, and H. Ito. A Delay Measurement Technique Using Signature Registers. In *Proc. Asian Test Symposium*, pages 157–162. IEEE, 2009.

[33] T. Kehl. Hardware self-tuning and circuit performance monitoring. *Proc. IEEE International Conference on Computer Design*, pages 188–192, 1993.

[34] T. Kunitake, Y., Sato, T., Yasuura, H., and Hayashida. A Selective Replacement Method for Timing-Error-Predicting Flip-Flops. *Journal of Circuits, Systems and Computers*, 21(06):1240013, Oct. 2012.

[35] J. Lee. PBTI-Associated High-Temperature Hot Carrier Degradation of nMOSFETs With Metal-Gate/High-k Dielectrics. *IEEE Electron Device Letters*, 29(4):389–391, Apr. 2008.

[36] J. M. Levine, E. Stott, and P. Y. Cheung. Dynamic voltage & frequency scaling with online slack measurement. In *Proc. ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, FPGA '14, pages 65–74, New York, New York, USA, 2014. ACM Press.

[37] J. M. Levine, E. Stott, G. A. Constantinides, and P. Y. Cheung. Health monitoring of live circuits in FPGAs based on time delay measurement (abstract only). In *Proc. ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, FPGA '11, page 284, New York, New York, USA, 2011. ACM Press.

[38] J. M. Levine, E. Stott, G. A. Constantinides, and P. Y. Cheung. Online Measurement of Timing in Circuits: For Health Monitoring and Dynamic Voltage & Frequency Scaling. In *Proc. IEEE International Symposium on Field-Programmable Custom Computing Machines*, pages 109–116. IEEE, Apr. 2012.

[39] J. M. Levine, E. Stott, G. A. Constantinides, and P. Y. K. Cheung. SMI: Slack Measurement Insertion for online timing monitoring in FPGAs. In *Proc. International Conference on Field Programmable Logic and Applications*, pages 1–4. IEEE, Sept. 2013.

[40] A. Nabina and J. L. Nunez-Yanez. Adaptive Voltage Scaling in a Dynamically Reconfigurable FPGA-Based Platform. *ACM Transactions on Reconfigurable Technology and Systems*, 5(4):1–22, Dec. 2012.

[41] M. Nakai, S. Akui, K. Seno, T. Meguro, T. Seki, T. Kondo, A. Hashiguchi, H. Kawahara, K. Kumano, and M. Shimura. Dynamic voltage and frequency management for a low-power embedded microprocessor. *IEEE Journal of Solid-State Circuits*, 40(1):28–35, Jan. 2005.

[42] S. Nassif. Delay variability: sources, impacts and trends. In *Technical Papers IEEE International Solid-State Circuits Conference.*, pages 368–369. IEEE, 2000.

[43] S. Nazarian, A. Iranli, and M. Pedram. Crosstalk analysis in nanometer technologies. In *Proc. ACM Great Lakes Symposium on VLSI*, page 253, New York, New York, USA, 2006. ACM Press.

[44] K. Nowka, G. Carpenter, E. MacDonald, H. Ngo, B. Brock, K. Ishii, T. Nguyen, and J. Burns. A 32-bit PowerPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling. *IEEE Journal of Solid-State Circuits*, 37(11):1441–1447, Nov. 2002.

[45] J. Nunez-Yanez. Dynamic voltage scaling in a FPGA-based system-on-chip. *... Logic and Applications, ...*, pages 459–462, Aug. 2007.

[46] J. Nunez-Yanez. Energy proportional computing in commercial FPGAs with adaptive voltage scaling. *Proc. FPGAworld Conference*, pages 1–5, 2013.

[47] P. Pant and J. Zelman. Understanding Power Supply Droop during At-Speed Scan Testing. In *2009 27th IEEE VLSI Test Symposium*, pages 227–232. IEEE, May 2009.

[48] T. Rahkonen and J. Kostamovaara. The use of stabilized CMOS delay lines in the digitization of short time intervals. In *Proc. IEEE International Sympoisum on Circuits and Systems*, pages 2252–2255. IEEE, 1991.

[49] V. B. Rose and Jonathan. VPR: A New Packing, Placement and Routing Tool for FPGA Research. *International Workshop on Field-Programmable Logic and Applications*, 1997.

[50] M. Rothberg. Disk drive for receiving setup data in a self monitoring analysis and reporting technology (SMART) command, May 2005.

[51] D. K. Schroder and J. a. Babcock. Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing. *Journal of Applied Physics*, 94(1):1, 2003.

[52] A. K. Singh and S. A. Kumar. *Digital VLSI Design*. PHI Learning Pvt. Ltd., 2011.

[53] J. Srinivasan, S. Adve, P. Bose, and J. Rivers. The impact of technology scaling on lifetime reliability. In *Proc. International Conference on Dependable Systems and Networks*, pages 177–186. IEEE, 2004.

[54] S. Srinivasan, P. Mangalagiri, N. Viiaykrishnan, and K. Sarpatwari. FLAW: FPGA lifetime awareness. *Proc. ACM/IEEE Design Automation Conference*, pages 630–635, 2006.

[55] B. Stackhouse, S. Bhimji, C. Bostak, D. Bradley, B. Cherkauer, J. Desai, E. Francom, M. Gowan, P. Gronowski, D. Krueger, C. Morganti, and S. Troyer. A 65 nm 2-Billion Transistor Quad-Core Itanium Processor. *IEEE Journal of Solid-State Circuits*, 44(1):18–31, Jan. 2009.

[56] E. Stott, Z. Guan, J. M. Levine, J. S. J. Wong, and P. Y. K. Cheung. Variation and Reliability in FPGAs. *IEEE Design & Test*, 30(6):50–59, Dec. 2013.

[57] E. Stott, J. M. Levine, N. Kapre, and P. Y. K. Cheung. Timing Fault Detection in FPGA-based Circuits. In *Proc. IEEE International Symposium Field Programmable Custom Computing Machines*, Apr. 2014.

[58] E. Stott, J. S. Wong, and P. Y. Cheung. Degradation Analysis and Mitigation in FPGAs. In *Proc. International Conference on Field Programmable Logic and Applications*, pages 428–433. IEEE, Aug. 2010.

[59] E. A. Stott, J. S. Wong, P. Sedcole, and P. Y. Cheung. Degradation in FPGAs. In *Proc. ACM/SIGDA International Symposium on Field Programmable Gate arrays*, page 229, New York, New York, USA, 2010. ACM Press.

[60] J. Tschanz, K. Bowman, and S. Lu. On-line detection and correction of errors due to fast, dynamic voltage droop events. *Proceeding of Silicon ...*, pages 1–4, 2010.

[61] J. Tschanz, K. Bowman, S. Walstra, M. Agostinelli, T. Karnik, and V. De. Tunable replica circuits and adaptive voltage-frequency techniques for dynamic voltage, temperature, and aging variation tolerance. In *Proc. Symposium on VLS VLSI Circuits*, pages 112–113. IEEE, 2009.

[62] J. Tschanz, K. Bowman, C. Wilkerson, S.-L. Lu, and T. Karnik. Resilient circuits. In *Proc. International Conference on Computer-Aided Design*, page 71, New York, New York, USA, 2009. ACM Press.

[63] J. Tschanz, N. Kim, and S. Dighe. Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging. *Proc. IEEE International Conference on Solid-State Circuits*, pages 292–604, Feb. 2007.

[64] A. Uht. Uniprocessor performance enhancement through adaptive clock frequency control. *IEEE Transactions on Computers*, 54(2):132–140, Feb. 2005.

[65] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao. The impact of NBTI on the performance of combinational and sequential circuits. In

Proceedings of the 44th annual conference on Design automation - DAC '07, page 364, New York, New York, USA, June 2007. ACM Press.

[66] B. P. Wong, A. Mittal, Y. Cao, and G. Starr. *Nano-CMOS Circuit and Physical Design*. John Wiley & Sons, Inc., Hoboken, NJ, USA, Nov. 2004.

[67] J. Wong, P. Sedcole, and P. Cheung. Self-characterization of combinatorial circuit delays in FPGAs. In *Proc. International Conference on Field-Programmable Technology*, pages 17–23, 2007.

[68] J. S. Wong and P. Y. Cheung. Improved delay measurement method in FPGA based on transition probability. *Proc. ACM/SIGDA Inernational Symposium on Field programmable Gate Arrays*, page 163, 2011.

[69] J. S. J. Wong and P. Y. K. Cheung. Timing Measurement Platform for Arbitrary Black-Box Circuits Based on Transition Probability. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 21(12):2307–2320, Dec. 2013.

[70] J. S. J. Wong, P. Sedcole, and P. Y. K. Cheung. A transition probability based delay measurement method for arbitrary circuits on FPGAs. In *Proc. International Conference on Field-Programmable Technology*, pages 105–112. IEEE, Dec. 2008.

[71] S. Yang. Logic Synthesis and Optimization Benchmarks User Guide. (919):0–44, 1991.

[72] Y.-C. Yeo, Q. Lu, and C. Hu. MOSFET Gate Oxide Reliability: Anode Hole Injection Model and its Applications. *International Journal of High Speed Electronics and Systems*, 11(03):849–886, Sept. 2001.

[73] K. M. Zick and J. P. Hayes. On-line sensing for healthier FPGA systems. *Proc. ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, page 239, 2010.

[74] P. Zuchowski, P. Habitz, J. Hayes, and J. Oppold. Process and environmental variation impacts on ASIC timing. In *Proc. IEEE/ACM International Conference on Computer Aided Design*, pages 336–342. IEEE, 2004.