UC Irvine
UC Irvine Electronic Theses and Dissertations

Title

Low Power Reliable Design using Pulsed Latch Circuits

Permalink
https://escholarship.org/uc/item/5ss2z430

Author
Elsharkasy, Wael Mahmoud

Publication Date
2017

Copyright Information
This work is made available under the terms of a Creative Commons Attribution License, availalbe at https://creativecommons.org/licenses/by/4.0/

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA,
IRVINE

Low Power Reliable Design using Pulsed Latch Circuits

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Wael Mahmoud Elsharkasy

Dissertation Committee:
Professor Fadi J. Kurdahi, Chair
Professor Ahmed M. Eltawil
Professor Rainer Doemer

2017

© 2017 Wael Mahmoud Elsharkasy

DEDICATION

To the memory of my Grandfather,
To my beloved Mom and Dad,
To my beloved Wife,
To my little angels Judy and Malek

TABLE OF CONTENTS

Page

LIST OF FIGURES **v**

LIST OF TABLES **viii**

ACKNOWLEDGMENTS **ix**

CURRICULUM VITAE **x**

ABSTRACT OF THE DISSERTATION **xiii**

1 Introduction **1**
1.1 Motivation 1
1.2 Pulsed Latch as a Sequential Element 3
1.3 Background and Related Work 4
1.4 Thesis Contributions 7
1.5 Organization 9

2 Effect of PVT Variations on Pulsed Latches **10**
2.1 Introduction 10
2.2 Prior Work 13
2.3 Pulsed Latch Circuit Design 14
2.4 Process Variations 15
2.4.1 Sources of Process Variations 15
2.4.2 Effect of Process Variations on Pulsed Latches 16
2.5 Voltage Scaling 17
2.5.1 Benefits of Voltage Scaling 17
2.5.2 Effect of Voltage Scaling on Pulsed Latches 18
2.6 Temperature Variations 20
2.6.1 Effect of Temperature Variations on Circuits Performance 20
2.6.2 Effect of Temperature Variations on Pulsed Latches 22
2.7 Conclusion 23

3 Reconfigurable Pulsed Latches **25**
3.1 Introduction 25
3.2 Prior Work 26

3.3 Proposed Design Approaches . . . 27
3.3.1 First Approach . . . 29
3.3.2 Second Approach . . . 31
3.4 Circuit Design . . . 32
3.4.1 Register Design . . . 32
3.4.2 Control Circuit Design . . . 34
3.5 Simulation Setup . . . 36
3.6 Results . . . 38
3.6.1 Reliability Analysis . . . 38
3.6.2 Power and Area Comparisons . . . 41
3.6.3 Combined Metric for Reliability and Power . . . 42
3.6.4 Sensitivity to Clock Rise Time . . . 43
3.6.5 Discussion . . . 44
3.7 Conclusion . . . 45

4 Register File Design using Pulsed Latches 47
4.1 Introduction . . . 47
4.2 Prior Work . . . 48
4.3 Design of a Single Read - Single Write Register File (1R1W) . . . 50
4.3.1 SRAM Based Implementation . . . 50
4.3.2 Flip-Flop Based Implementation . . . 51
4.3.3 Latch Based Implementation . . . 55
4.3.4 Pulsed Latch Based Implementation . . . 56
4.3.5 Comparison between Different 1R1W Register File Implementations . 59
4.4 Design of Multiport Register File . . . 62
4.4.1 SRAM Based Implementation . . . 62
4.4.2 Standard Cell Based Implementation . . . 65
4.5 Proposed Pulsed Latch Based Register File with Virtual Ports . . . 67
4.5.1 Adding Virtual Read Ports . . . 67
4.5.2 Adding Virtual Write Ports . . . 72
4.6 Results . . . 73
4.6.1 Register Files Circuit Designs . . . 73
4.6.2 Register Files Area . . . 79
4.6.3 Register Files Power Consumption . . . 82
4.7 Conclusion . . . 84

5 Future work 86
5.1 Pulsed Latches at the Circuit Level . . . 86
5.2 Pulsed Latches in Register Files . . . 89

6 Conclusions and Summary 91

Bibliography 94

LIST OF FIGURES

Page

1.1 Simple diagram showing the typical timing overhead of a master-slave flip-flop. 5
1.2 A simple diagram of a basic pulsed latch circuit. 6

2.1 Simple diagram of the Transmission Gate Pulsed Latch (TGPL). 12
2.2 A simple diagram illustrating the effect of different pulse widths on the pulsed latch write operation. 12
2.3 The basic construction of a simple pulser. 14
2.4 Sample PDFs of the latch write time and the pulser pulse width showing the region of write failure. 17
2.5 The effect of voltage scaling on the distributions of the latch WR time and the pulser PW at 125°C. 19
2.6 The probability of failure of a TGPL designed at nominal supply voltage at different supply voltages and temperatures. 20
2.7 The effect of temperature variation on both the latch write time and the pulser pulse width running at nominal supply of 1.05V. The bar around different points represents the 1-sigma variation around the mean. 22
2.8 The effect of temperature variation on both the latch write time and the pulser pulse width running at scaled supply voltage of 0.7V. The bar around different points represents the 1-sigma variation around the mean. 23

3.1 A diagram showing arbitrary PDFs for the pulser and the latch when (a) operating at nominal supply voltage, (b) scaling down the supply voltage without configuring the pulser (or having a fixed pulser), and (c) scaling down the supply voltage and configuring the pulser circuit to generate a wider output pulse 28
3.2 A delay path-based pulser. 29
3.3 The proposed header switches-based pulser design. 30
3.4 The proposed MUX-based pulser design. 31
3.5 The structure of the pulser used for the PL-SW based register. 33
3.6 The structure of the pulser used for the PL-MUX based register. 34
3.7 The structure of the circuit used to generate the CTRL signal for the PL-SW and PL-MUX registers. 35
3.8 The layouts of the proposed pulser circuits: (a) PL-SW, (b) PL-MUX. . . . 37
3.9 The latch worst write time and the typical and worst pulser pulse widths for the two configurations of PL-SW at different supply voltages at 25°C. 39

3.10 Probability distribution function for PL-SW before and after configuration for VDD=0.7V at 125°C. 40
3.11 Probability distribution function for PL-MUX before and after configuration for VDD=0.7V at 125°C. 41
3.12 The average energy per cycle normalized to the energy per cycle of the traditional PL register at 1.05V and 125°C. 41
3.13 Box plot of pulse width of PL-SW at 0.7V and 25°Cover 1000-samples of Monte Carlo simulation. 44

4.1 A typical structure of an 8T SRAM bitcell for a 1R1W register file. 52
4.2 A block diagram for a standard cell based register file. 52
4.3 A block diagram for one of the flip-flop registers in the data array showing the usage of the clock gating cell (CGC) for the enabling of the flip-flops. . 54
4.4 A block diagram for one of the pulsed latch registers in the data array showing the replacement of the clock gating cell by the pulser circuit. 56
4.5 The pulser used in the pulsed latch register file (a) The proposed pulser circuit, (b) Timing diagram for the pulser operation 57
4.6 The modified pulser design with the delay buffers shared among different pulsers (a) The modified pulser circuit, (b) Timing diagram for the modified pulser operation 58
4.7 Area comparisons of the three versions of 32-bits/word 1R1W register files for different number of words 61
4.8 Power consumption of the three versions of 32-bits/word 1R1W register files for different number of words 62
4.9 A block diagram for a multiport register file with m read ports and j write ports 63
4.10 A straight forward expansion of the 8T SRAM bitcell to add more read and write ports 64
4.11 The structure of the pulsed latch based register file with virtual ports. 68
4.12 The structure of the proposed read logic that provides multiple virtual read ports. 69
4.13 The structure of the *Address Mux* that give priority to the read ports in the required order. 71
4.14 The structure of the proposed write logic that provides multiple virtual write ports. 73
4.15 The proposed read logic structure for register files with two virtual read ports. 76
4.16 The proposed read logic structure for register files with four virtual read ports. 78
4.17 The proposed write logic structure for register files with two virtual write ports. 80
4.18 Area comparisons of the five implementations of the 32-bits x 32-words register files with different read and write ports 81
4.19 Power consumption of the five implementations of register files with different number of ports 83
4.20 Power consumption of the pulsed latch based register file with 4R2W virtual ports when running the same testbenches of the lower number of ports. . . . 84

5.1 The proposed pulsed latch self gating approach. 89

LIST OF TABLES

Page

3.1 The probability of write failure for the three different register implementations at different supply voltages and different temperatures. 39

3.2 Normalized yield per unit power for the two register implementations using the two proposed pulsed latch designs. 43

4.1 The area of 1R1W SRAM bitcell, flip-flop and latch for the UMC 28nm process. 60

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Professor Fadi J. Kurdahi, who provided me with the guidance and support in all aspects of my research at UC Irvine. His guidance and support always helped me to overcome the challenges I faced in my research. I learned from him how to face challenges, question thoughts, and express ideas.

I would also like to thank my Co-advisor Prof. Ahmed M. Eltwail. The discussion with him always opened new doors to me and helped me overcome several challenges in my research. He always promoted my work and encouraged me in going further.

I would also like to thank Prof. Rainer Doemer for dedicating some of his time to be in my committee.

I would also like to thank Dr. Amin Khajeh for all his guidance and support. Our discussions in our weekly meetings or even individually helped me a lot in my research.

I would also acknowledge the support from my colleagues and friends, especially Dr. Ahmed Nassar, Dr. Muhammed Khairy, Dr. Amr Hussien and Dr. Elsayed Ahmed.

I would also like to thank all my other colleagues and friends for making all those years at UC Irvine a great and enjoyable experience.

Most importantly, I would also like to thank my parents and my brother who were always supporting me and encouraging me with their best wishes. I am blessed with having a caring and loving family.

Finally, this work would not have been finished without the constant support of my beloved wife, Yomna Hamed. Her moral and physical support was always a driving force throughout my PhD. I would also like to thank my little daughter, Judy, for sacrificing some of her play time with me to allow me to finish my work. I would also like to thank my little angle, Malek. His beautiful smile was always a driving force for me through hard times.

CURRICULUM VITAE

Wael Mahmoud Elsharkasy

EDUCATION

Doctor of Philosophy in Electrical and Computer Engineering **2017**
University of California, Irvine *Irvine, CA*

Thesis title: *Low Power Reliable Design using Pulsed Latch Circuits*

M.Sc. in Electrical Engineering **2011**
Alexandria University *Alexandria, Egypt*

Thesis title: *Hardware Implementation of JPEG2000 MQ Encoder*

B.Sc. in Electrical Engineering **2007**
Alexandria University *Alexandria, Egypt*

INDUSTRY EXPERIENCE

Engineering Intern **June 2016–December 2016**
ClariPhy Communications Inc., Irvine, CA.

Developing power characterization flow for DSP modules.

Engineering Intern **June 2014–June 2016**
Broadcom Corporation, Irvine, CA.

Dynamic and leakage power variation analysis.

Engineering Intern **June 2013–December 2013**
Broadcom Corporation, Irvine, CA.

Support for the integration of the HDMI 2.0 standard.

RESEARCH EXPERIENCE

Graduate Research Assistant **Fall 2012–Winter 2017**
University of California, Irvine *Irvine, California*

Visiting Researcher **March 2012–June 2012**
Egypt-Japan University for Science and Technology (E-JUST) *Alexandria, Egypt*

Research Assistant **September 2007–September 2012**
Alexandria University *Alexandria, Egypt*

TEACHING EXPERIENCE

Teaching Assistant **Fall 2012–Fall 2016**
University of California Irvine *Irvine, CA*

EECS159A Senior Design Project I.	Falls 2014-2016
EECS70B Network Analysis II.	Spring 2016
EECS159B Senior Design Project II.	Winter 2016
EECS113 Processor Hardware/Software Interface.	Spring 2015
EECS170B Electronics II.	Winters 2014-2015
EECS119 VLSI.	Fall 2013

Teaching Assistant **September 2007–September 2012**
Alexandria University *Alexandria, Egypt*

Peer Review Activity

IEEE Transactions on Circuits and Systems I.
IEEE Transactions on Circuits and Systems II.

Professional Activities

- Member, IEEE since 2003.
- IEEE Alexandria University Student Branch organizing Team (2003–2007).
- Student Branch Chairman (Alexandria University), IEEE Alexandria & North Delta Subsection, 2006.
- Student Branch Mentor (Alexandria University), IEEE Alexandria & North Delta Subsection, (2008–2011).
- Student Branch Advisor (Alexandria University), IEEE Alexandria & North Delta Subsection, (2011–2012).
- IEEE Alexandria Student Branch representative in Student Branch Congress (SBC) 2006 in Paris.
- One of the organizing team of Egyptian Engineering Day (EED) conference from 2005 till 2009.
- The leader of Alexandria University Science Club team in Egyptian Universities Youth Week 2007.

Honors

- Electrical Engineering and Computer Science Department Graduate Fellowship, University of California Irvine 2012.

- ◇ Distinction with Honor degree at B.Sc and ranked Third on a class of 334 students, Alexandria University 2007.
- ◇ IEEE Student Ethics Competition Certificate, Alexandria University 2006.
- ◇ Official IEEE Outstanding Support of Student Branch, Alexandria University 2004-2005.

SELECTED PUBLICATIONS

- ◇ Wael M. Elsharkasy, Amin Khajeh, Ahmed M. Eltawil, and Fadi J. Kurdahi,“Pulser Self Gating for Power Reduction of Pulsed Latch Circuits” , To be submitted to *2017 International Conference on Computer Aided Design (ICCAD)*, November 2017.
- ◇ Wael M. Elsharkasy, Hasan Erdem Yantir, Amin Khajeh, Ahmed M. Eltawil, and Fadi J. Kurdahi,“Efficient Implementation of Multiport Register Files using Pulsed Latches”, To be submitted to the *International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES)*, October 2017.
- ◇ Wael M. Elsharkasy, Amin Khajeh, Ahmed M. Eltawil, and Fadi J. Kurdahi,“Reliability Enhancement of Low-Power Sequential Circuits Using Reconfigurable Pulsed Latches”, Submitted to the *IEEE Transactions on Circuits and Systems I (TCAS I)*, under review.
- ◇ Ihsen Alouani, Wael M. Elsharkasy, Ahmed M. Eltawil, Fadi J. Kurdahi, and Smail Niar,“AS8-SRAM: Asymmetric SRAM Architecture For Soft Error Hardening Enhancement”, *IET Circuits, Devices & Systems*, 2016.
- ◇ Ahmed Nassar, Fadi J. Kurdahi, and Wael M. Elsharkasy,“NUVA: architectural support for runtime verification of parametric specifications over multicores”, *Proceedings of the 2015 International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, October 2015.
- ◇ Amr M. A. Hussien, Wael M. Elsharkasy, Ahmed M. Eltawil, Fadi J. Kurdahi, and Amin Khajeh,“Low Overhead Correction Scheme for Unreliable LDPC Buffering”, *1st IEEE Global Conference on Signal and Information Processing (GlobalSIP 2013)*, December 2013.
- ◇ Wael M. El-sharkasy, and Mohamed E. Ragab,“Hardware Modelling of JPEG2000 MQ-Encoder”, *The 4th International Conference on Intelligent and Advanced Systems (ICIAS2012)*, June 2012.
- ◇ Mohamed Elmoghany, Mohamed Diab, Moustafa Kassem, Mustafa Khairallah, Omar El Shahat, and Wael M. Elsharkasy, “FPGA Implementation of High Speed XTS-AES for Data Storage Devices”, *International Conference for Internet Technology and Secured Transactions (ICITST)*, December 2011.

ABSTRACT OF THE DISSERTATION

Low Power Reliable Design using Pulsed Latch Circuits

By

Wael Mahmoud Elsharkasy

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2017

Professor Fadi J. Kurdahi, Chair

System-on-Chip (SoC) faced lots of challenges over the past decade. With nowadays applications centered around Internet-of-Everything (IoE), these challenges are expected to be more critical. Among these challenges are the reduction of power consumption for better energy efficiency, the overcoming of different sources of variations to ensure reliable operation and the reduction of design area to reduce the cost and increase the integration. As a result, chip designers find themselves facing lots of problems, trying to build reliable systems that integrate complex level of functionality, on a minimum die size and with a limited power budgets. Among different circuit components in every chip, memory components are of great concern. They consume the majority of the chip area and power, in addition to affecting the entire chip performance and reliability. These include large memory arrays, caches, register files and different sequential elements in the logic paths. Sequential elements play an important and critical role in modern synchronous CMOS circuits. Indeed, they can represent up to 50% of the standard cells used in a chip. In addition, the power consumption of the clock tree, including these elements can be more than half of the total chip power. In addition, they come in the second place after memory to be affected by different sources of variation. Hence, efficient implementation of these elements is of great importance for the design of energy efficient and reliable integrated circuits. Pulsed latches have been proposed as efficient replacement of flip-flops in the implementation of sequential elements.

They can achieve higher performance when compared to traditional flip-flops, and can be designed to be smaller in area and more power efficient. However, the operation of pulsed latch is more sensitive to process, voltage and temperature (PVT) variations. In this thesis, we are proposing a methodology to study the reliability of pulsed latches and we have used it to evaluate the effect of PVT variations on their behavior. In addition, novel approaches to enhance the reliability of pulsed latches without significant degradation in performance, area or power are presented. Also, since sequential elements can be used to build small size register files, pulsed latch implementation of register files are discussed and compared to other traditional implementations, including SRAM and flip-flops. In addition, since multiport register files are very beneficial for quite few applications, novel implementations of multiport register files are also presented. The proposed implementation is proved to highly reduce the significant overhead in area, power and latency associated with the traditional way of designing multiport register files.

Chapter 1

Introduction

1.1 Motivation

The continuing advancements in the process technology have enabled the integration of more transistors on a single chip. This opened the gate to building large and complex systems on chips (SoCs). Together with the advancements in the communication systems, this allowed the existence of powerful mobile devices over the past decade. Although increasing the computing power was one of the main challenges for a long time, other design challenges appeared over the last decade and became of more importance. With the move toward building a smarter world with application centered around the Internet-of-Everything (IoE)[22], these challenges will continue to be the main ones. These challenges include reducing power consumption, improving design reliability, reducing design area to reduce cost, in addition to enhancing security.

The first and may be foremost challenge is power consumption. With the increase of the computation power and the higher integration in modern SoCs, power consumption is significantly increasing. Reducing power is extremely important for SoCs with limited energy resources, ranging from powerful mobile SoCs to sensor nodes with limited processing power. However, with the emergence of the phenomena of dark silicon [26, 59], where it becomes very difficult to power up the entire chip at the same time, power efficient design becomes also critical to even SoCs that are not running on a battery. In addition, high power consumption can increase the on-chip temperature, which in turn, affects the performance, reliability and aging of different on-chip devices.

The second major challenge is design reliability. The chip designers must ensure that the chip will work as expected under different operating conditions. This includes process, voltage and temperature (PVT) variations. Process variation has been identified as one of the main challenges facing the semiconductor industry for sub-100 nm technology nodes [1]. Although sources of process variations can't be eliminated for such nanoscale dimensions, their effect should be considered at design time and different circuit or system techniques should be implemented to guarantee correct functionality. In addition to process variations, the chips usually operate in different power modes in order to save power. This may include changing the supply voltage, since it is the main factor affecting power consumption. Hence, chip designers should guarantee that different parts of the design is working reliably and meeting their timing budget at different supply voltages. In addition, another factor to consider is the on-chip temperature. Since the temperature will be changing depending on the ambient temperature, in addition to the chip operation, this changes should be considered at design time, since temperature will affect the transistors' performance and power.

The third challenge is reducing the chip area. Driven by the highly competitive industry, reducing the chip cost depends on having the smallest possible die size. In addition, reducing the area of different design blocks will allow the integration of more functionality into the same chip area, hence getting higher computation capability.

As a result of these three challenges, SoC designers are in a hard dilemma, trying to build reliable designs that integrate vast amount of logic and memories to do complex functionality in a minimum die size with the minimum power consumption [25]. Among different circuit components in every chip, memory components are of great concern as they consume the majority of the chip area. In addition, they highly affect the chip performance, power and reliability. This includes large memory arrays, caches, register files and different sequential elements in the logic paths. Sequential elements (including flip-flops and latches) play an important and critical role in modern synchronous CMOS circuits. Indeed, they represent up to 50% of the used standard cells in modern chips. Also, the power consumption of the clock tree, including these elements can reach more than half the total chip power [12, 49]. In addition, sequential elements come in the second place after memory to be affected by different sources of variation [42]. Hence, efficient implementation of these elements is of great importance for the design of reliable and energy efficient integrated circuits.

1.2 Pulsed Latch as a Sequential Element

Pulsed latches have been proposed multiple times as an efficient sequential element implementation in replacement of flip-flops. Pulsed latches are latches driven by short pulses generated from the normal clock signal using a pulse generator circuit called a *pulser*. In comparion to flip-flops, pulsed latches have lower timing overhead, in addition to lower power consumption. As an example, our implementation of a 16-bit register, consisting of one pulser driving 16 latches showed power saving of more than 20% when compared to a similar register designed by flip-flops. Also, since the latch's area is much smaller than flip-flop, a significant area saving can be expected when a single pulser is shared between several latches. For the 16-bit register described previously, this are saving was over 30%. However, the operation of pulsed latch is more sensitive to PVT variations. Hence, comprehensive study of each of these effects on the pulsed latch operation and reliability must be performed. Also, novel approaches to enhance the reliability of pulsed latches without significant degradation in performance, area or power must be implemented.

In addition to using sequential elements in the logic paths, they are also sometimes used to build register files for data storage. While flip-flops and latches were traditionally used for such cases, pulsed latches can provide an attractive alternative. Using pulsed latches can results in a register file with smaller area, lower latency and lower power consumption. However, it still needs to be compared with the wide spread SRAM based register files. In addition to the register files with single read and single write ports, which are common in most designs, multiport register files are very beneficial for quite few applications. However, there are significant overhead in area, power and performance associated with the conventional way of designing multiport register files. With the flexibility associated with pulsed latches, they can provide an attractive and more efficient alternative implementation of multiport register files.

1.3 Background and Related Work

Flip-flops are considered the most popular sequential elements used in conventional ASIC designs. This is mainly because of the simplicity of their timing model, which makes the design and timing verification processes much easier. Master-Slave Flip-Flops (MSFFs) are considered the most common and traditional implementations of flip-flops, due to its stable operation and its simple timing characteristics. However, the fact that the MSFF micro-architecture is usually built using two consecutive latches, it takes an appreciable portion of the clock period, power consumption and area. As shown in Figure 1.1, a typical MSFF has a significant nominal timing overhead (sum of the clock-to-Q delay and the setup time) of 6 FO4 (fanout-of-4) and can reach 10 FO4 when considering clock skew and jitter [17]. In addition, the clock network, including the flip-flops, often consumes one third to one half of the total dynamic power of the chip [49, 56]. In addition to the mentioned overheads

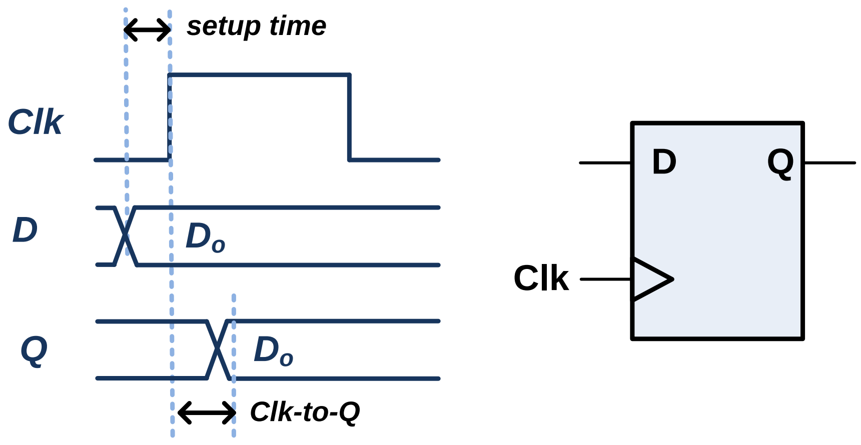


Figure 1.1: Simple diagram showing the typical timing overhead of a master-slave flip-flop.

associated with MSFF, some additional margins, which can reach up to 15% (depending on the sign off methodology), are usually added to the nominal timing margins to ensure correct operation under different process, voltage, and temperature (PVT) variations [4]. This, in turn, increases the already existing high timing and power overheads. For the above reasons, MSFF can be considered as a good choice for low-to-medium performance designs as they provide a good balance between delay, power, and easy design and verification processes for chips working at a relatively low frequency [20].

On the other hand, high performance custom designs tend to use latches due to their lower timing overhead that can reach 2 FO4 in some designs [17]. Although latch based designs are typically robust to clock skew and jitter (due to the latch transparency period), latches have a complicated timing model, which, in turn, complicates the design and the verification processes and increases the risk of hold time violations, especially with PVT variations.

To fill in the missing gap between MSFFs and latches, pulsed latches (sometimes called pulsed flip-flops) have been used in some high-performance designs [64, 11, 19]. Pulsed latches (PLs) are latches driven by short pulses generated from a *pulser* circuit as shown in Figure 1.2. The pulser can be either embedded in the latch (also sometime called pulsed latches with implicit pulser or pulsed flip-flop), or can be separated as a standalone circuit.

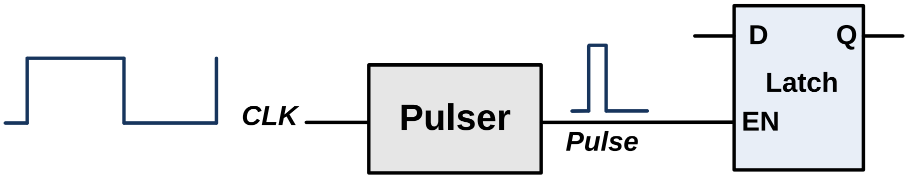


Figure 1.2: A simple diagram of a basic pulsed latch circuit.

If the latter approach is used, a single pulser can be shared by more than one latch. Thus, it has the advantage of area and power consumption savings over the former approach, and it is the focus of our discussion in this thesis. In addition, the pulser usage can eliminate the need for some of the clock buffers used in the clock tree, thus providing an additional amount of power and area savings.

Having only one latch between the input and the output, pulsed latches have lower timing overhead compared to MSFFs. At the same time, since the driving pulse is very short, the transparency period for the latch becomes very narrow, allowing the pulsed latches to have a timing behavior close to that of MSFFs [55], to the extent that they are sometimes classified among flip-flop families [5, 6]. Also, due to the presence of the narrow transparent window of the latch, pulsed latches have an inherent tolerance to clock skew and jitter [49]. Since they have fewer transistors that are triggered by the clock signal, they have the advantage of reducing a significant amount of clocking power [19], and they consume much less leakage power compared to MSFFs due to the smaller area and fewer transistors.

One complication in the pulsed latch design is the choice of the pulser output pulse width. Too short of a pulse width may not be enough for the latches to store the input data correctly, while too long of a pulse width will result in a longer latch transparency window; which, in turn, increase the timing overhead or can violate hold time requirements [50]. This issue becomes more complicated when considering different sources of variations. PVT variations have significant impacts on different circuit components. Since sequential elements, in general, are by nature time sensitive elements, they are among the circuit categories that are highly affected by any PVT variations [42]. Since pulsed latches, in particular, are very time sensitive, good study of the effect of different sources of variations has to be considered. Since some of these variations, such as voltage and temperature, are temporal variations that change over the operating period of chips, careful analysis and design have to be performed to ensure that reliable circuit operation can always be achieved without any significant loss in performance, power or area.

Sequential elements are not only used as standalone elements in logic paths, but it is also used in designing register files. Although Static Random-Access Memories (SRAMs) are commonly used to build register file, standard cell based register files provide a very attractive competitor to SRAMs for small size register files. In addition, they are also highly considered for high performance operations, low power applications and designs working at ultra low supply voltages [44, 44, 12, 14, 15]. Although pulsed latches offered few advantages in area, performance and power over flip-flops, they weren't widely discussed as a better alternative for building register file. Giving the regular structure of register files, in addition to the nature of their operation of accessing an entire register for read and write, pulsed latches with shared pulser represent a perfect fit for such implementations.

1.4 Thesis Contributions

The keys contributions of this thesis can be summarized as follow:

- A comprehensive methodology to study the reliability of pulsed latch circuits is presented. This methodology considers the different behavior of both the pulser and the latches and their effect on the overall operation of the entire circuit. In addition, the proposed methodology is a general one that can be used with other pulsed latch topologies.

- The effects of process, voltage and temperature variations on pulsed latch's behavior are evaluated. We studied the effect of wide range of variations, covering ±6σ of process variation, more than 30% of supply voltage scaling and temperature variation range from -40°Cto 125°C. We show that the worst case scenario for pulsed latches is working at the lowest supply voltage and the lowest temperature.

- Two novel approaches to enhance the reliability of pulsed latches are presented. The two approaches add a reconfiguration ability to pulsed latch circuits. We show that the two approaches allow the usage of supply voltage scaling without any degradation in reliability, measured as the probability of having a write failure. In addition, the added area overhead is very small, 3% or less, while the power overhead is negligible for one of the approaches and around 11% on average for the other approach. The design yield per unit power, which is introduced as a figure of merit to compare the proposed reconfigurable approaches to the traditional one, shows the gained benefits of the proposed approaches at different operating conditions.

- Pulsed latch implementation of register files with single read and single write ports is presented. In comparison with SRAM and flip-flop based register files, it shows a better area and power efficiency when compared to flip-flop based register files. In addition, it extends the register file size range at which the standard cell based register file can be more power efficient than the SRAM based ones.

- Novel implementations of multiport register files using pulsed latches are presented. The proposed implementations allow the execution of multiple read and write operation using a single physical read and a single physical write ports by adding virtual read and write ports. Multiple register file configurations with different number of read and write ports are implemented using different implementation options. For all the different configurations, our proposed implementation shows significant reduction in area and power. In addition, the idea of using a global register file with four read and two write virtual ports that can be configured at run time is discussed. When compared to register files with dedicated number of ports, the power overhead is found to be very small.

1.5 Organization

The rest of the thesis is organized as follow. In Chapter 2, the methodology of evaluating the reliability of pulsed latch is introduced and applied to evaluate the effect of PVT variations on pulsed latches. Chapter 3 presents the two proposed approaches to enhance the reliability of pulsed latches under supply voltage scaling. In Chapter 4, the pulsed latch implementation of register files with one read and one write ports are discussed and compared with other implementation options. In addition, the novel implementations of multiport register file with virtual read and write ports using pulsed latches are presented and compared to other alternative implementations. Chapter 5 highlight other in-progress work, in addition to some future work. Finally, conclusions and summary are drawn in Chapter 6.

Chapter 2

Effect of PVT Variations on Pulsed Latches

In this chapter, we present the effect of different process, voltage, and temperature (PVT) variations on the operation of pulsed latches. First, we start by describing a simple methodology that will be used in our analysis of the effect of PVT variations. After that, we will discuss some details about each of these variation and their effects on circuit behavior in general. Then, we will discuss the effect of each of them on pulsed latch circuits. Also, we will discuss the prior work in the literature that tried to study these effects on pulsed latch circuits.

2.1 Introduction

The operation of pulsed latches (PLs) is based on enabling the latch for a short time using a pulse generated by the pulser circuit. Hence, to study the effects of PVT variations on PL operation, the effects on both the latch write time and the pulser pulse width should be considered. In our methodology, we are proposing to study the effect of process variations on each of the latch and the pulser independently. Then, both effects should be merged together to get a reliability metric for the entire PL circuit. The reliability metric proposed for the pulsed latch circuit is the probability of write failure, which represents the probability of the latch's failure to capture and store a new data within the available transparent timing window provided by the pulser circuit. The same study can be repeated for different voltage and temperature values of interest to get the effects of voltage and temperature variations on the operation of PL circuits.

In this chapter, we are presenting our proposed variability analysis on one of the popular topologies of pulsed latches, which is the Transmission Gate Pulsed Latch (TGPL)[46] showed in Figure 2.1, which is considered one of the most efficient PL architectures. TGPL consists of two separate circuits: a pulser circuit and a transmission gate based latch. The pulser circuit is used to generate a pulse from the clock signal. This pulse will enable the transmission gate after the latch's input inverter and at the same time will break the feedback loop of the two crossed couple inverters (that are used to hold the stored data). Therefore, any new input D will pass to the internal nodes of the latch, appearing at both the latch's output Q and the output of the forward inverter of the crossed coupled structure. When the pulse is ended, the transmission gate is turned off, isolating the latch's internal nodes from the input, in addition to closing the crossed couple inverters' feedback loop to store the latch's current state.

Since the pulse is used to control the write operation of the latch, the choice of the width of this pulse is very critical to ensure correct write operation. As shown in Figure 2.2, if the pulse width is too short, the write operation will completely fail, resulting in unknown data being stored by the latch. Alternatively, if the pulse width is too long, the required data will be stored. However, if the latch's input changes within the same pulse, either this new input will be captured or a corrupted data will be stored by the latch. In both cases, any operation depending on the data stored by this latch would fail.

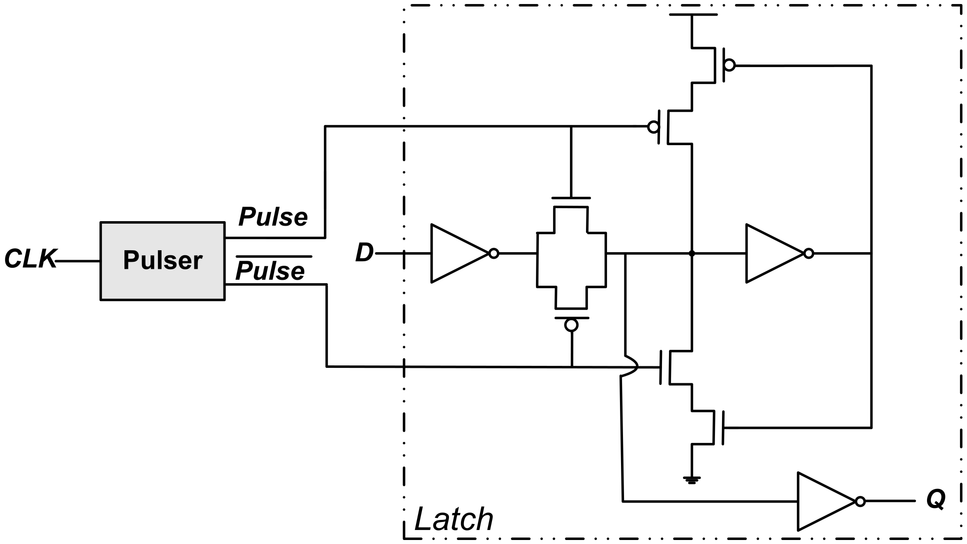


Figure 2.1: Simple diagram of the Transmission Gate Pulsed Latch (TGPL).

Since the pulser circuit is completely separate from the latch, the same pulser can be used to drive multiple latch. Although all our discussion will be focused on the TGPL, the same methodology can be applied to any explicit PL circuit with a standalone pulser that can be shared across few latches.

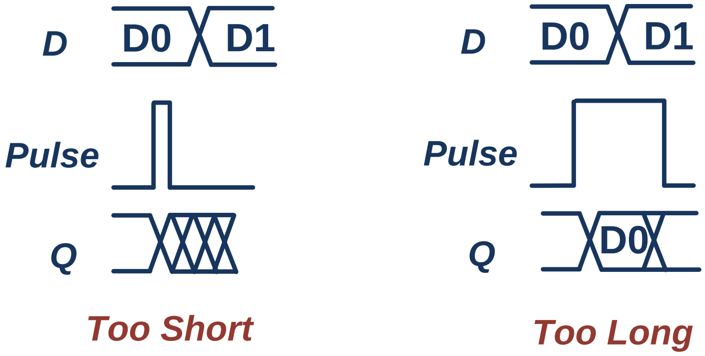


Figure 2.2: A simple diagram illustrating the effect of different pulse widths on the pulsed latch write operation.

2.2 Prior Work

Since sequential elements in general and pulsed latches in particular are very time sensitive elements, careful study of the effect of different PVT variations has to be conducted. However, to the best of our knowledge, no complete study of the effect of wide range of PVT variations on PL was presented in the literature.

In [38], the impact of process variations on several pulsed latches was presented, in addition to two techniques to reduce this impact. However, all the studied PL topologies were the ones with implicit pulser embedded with the latch to form one sequential elements (which is sometime called pulsed flip-flop) and the study didn't cover the process variations impact on the more popular explicit PL architectures, where a standalone pulser is used to drive several latches. In addition, the study focused only on the effect of process variations without considering the effect of voltage and temperature variations as well.

In [8, 9], the effects of PVT variations were studied on different flip-flops and pulsed latches topologies. The study showed that TGPL has the highest performance and resiliency against process variation. In addition, it also showed that TGPL is still highly affected (as much as the other considered topologies) by voltage variations. However, the studied voltage variations was limited to 10% only (which is very limited in comparison to variation ranges applied to nowadays circuits), while the temperature variations studied was only 20°C variation range around 85°C (which covers only certain range of applications). In addition, the study didn't quantify the effect of these variations on the design yield.

In [62], the authors presented a robust pulsed latch design that can work correctly in the presence of process and supply voltage variations. However, only some Monte Carlo simulations were used to test the circuit robustness without any comprehensive study for the effect of the process and voltage variations. In addition, all the simulations were done at room temperature and the effect of temperature was not considered.

In [36], a pulsed latch design with enhanced scan was proposed. The pulse width was chosen large enough to ensure correct operation under different PVT corners, assuming that delay cells can be used to fix any hold time issues. However, the design robustness was verified using only some Monte Carlo simulations and no comprehensive study for the effect of the process and voltage variations was done to determine the needed pulse width.

Therefore, it seems that a complete and comprehensive study of the effect of PVT variations on pulsed latches are still needed. Since pulsed latches (representing the designs' sequential elements) are replicated in large numbers in a chip die and across the entire wafer, a wide range of up to ±6σ of process variation should be considered in the variability analysis [42]. In addition, a wide range of voltage variation should be covered since supply voltage scaling is used extensively in nowadays chips to reduce power. Also, a wide range of temperature variation, that covers a typical industrial range from -40°C up to 125°Cshould be considered to cover most of the modern applications. In addition, the used methodology should be general and the same study can be repeated to any other explicit pulsed latch architectures.

2.3 Pulsed Latch Circuit Design

In our study, we have designed a 16-bit register circuit using the TGPL architecture. The register consists of a single pulser driving sixteen identical latches similar to that shown in Figure 2.1. The pulser used in our register is a traditional delay-chain based pulser shown in Figure 2.3, where the delay unit, usually consisting of an even number of inverters, is responsible for determining the width of the needed output pulse.

The register circuit was designed using the Synopsys 28nm PDK [2, 31]. Initial spice sim-

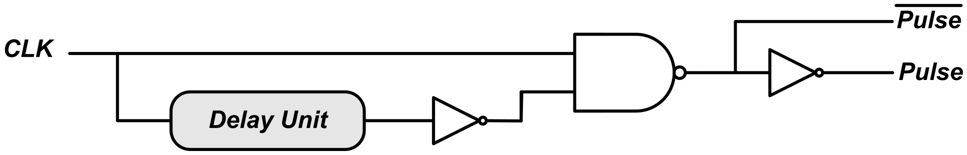


Figure 2.3: The basic construction of a simple pulser.

ulations were carried out to ensure correct operation. Then, the complete layout of the circuit was implemented and the post-layout netlist, including all the parasitic elements, was extracted. All the simulations and analysis shown in this chapter were carried out on the post-layout extracted circuits. A chain of inverters was used as a realistic waveform generator for the register clock signal and data inputs, to make their waveforms closer to the actual implementation. The simulations were done using Hspice and the variability analysis were carried out using Solido Variation Designer [42].

2.4 Process Variations

2.4.1 Sources of Process Variations

Due to the extreme miniaturization of device parameters in current and upcoming technology processes, even a small variation in the manufacturing process may cause parameter variations that can lead to a failed circuit operation [13]. Thus, one of the significant challenges in the design phase is the ability to evaluate the effect of different sources of variations on the functionality of complex circuits and to provide circuit solutions to guarantee correct functionality. Process dependent sources of variability such as effective length variation, oxide thickness variation, Line Edge Roughness (LER), and Random Dopant Fluctuation (RDF) result in variations in the value of the threshold voltages of transistors, which in turn impact the timing and power of digital circuits [40]. The threshold voltage variations due to RDF (which is usually the primary source of threshold voltage variations in planar MOSFETs) are considered as zero-mean Gaussian independent random variables with standard deviation denoted as σ_{Vth} which is given by [40]:

$$\sigma_{Vth} = \sigma_{Vtho}\sqrt{\frac{L_{min}W_{min}}{LW}} \tag{2.1}$$

where σ_{Vtho} is the σ_{Vth} for minimum sized transistor and it is given by:

$$\sigma_{Vtho} = \frac{qT_{ox}}{\epsilon_{ox}}\sqrt{\frac{N_a W_d}{3L_{min}W_{min}}} \tag{2.2}$$

where N_a is the effective channel doping, W_d is the depletion region width, T_{ox} is the oxide thickness, L_{min} and W_{min} are the minimum channel length and width, respectively.

While the scaling down of CMOS technology reduces the nominal supply voltage, the threshold voltages are not scaled by the same factor, leading to a significant reduction of the transistor's available voltage headroom (the difference between the supply voltage and the threshold voltage). Hence, even any small variation in the transistor's threshold voltage can lead to a significant degradation of the circuit behavior or can even cause complete circuit failure.

2.4.2 Effect of Process Variations on Pulsed Latches

The effect of process variations on pulsed latches should be studied carefully. Since pulsed latches are composed of two different components (pulser and latches) of different micro-architecture, the study should include the effect of variations on both the latch write time calculated as the CLK-to-Q delay and the pulser pulse width, the two effect should be combined together to evaluate the variation effects on the entire circuit operation. Due to process variations, both the write time and the pulse width can have a range of values with certain statistics. As shown in Figure 2.4, this is represented by the probability distribution functions (PDFs) of both the write time (Latch WR Time) and the pulse width (Pulser PW). To ensure correct write operation, the pulse width should be larger than the required transparent window for the latch (i.e. time needed to capture the input data and pass it through the internal nodes to the storing cross coupled inverters). The area under the intersection between the two PDFs represents the failure of write operation, since this is the

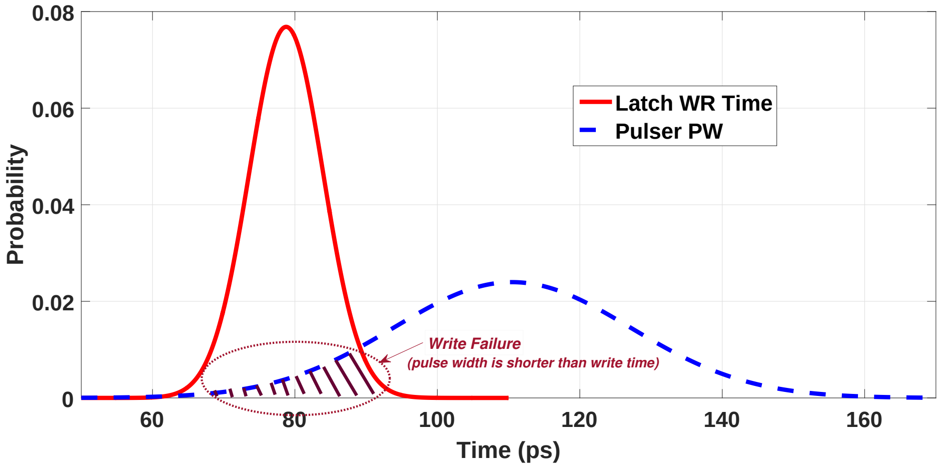


Figure 2.4: Sample PDFs of the latch write time and the pulser pulse width showing the region of write failure.

region where there is a high probability that the pulse width will be smaller than the time needed by the latch to capture the new input data.

Alternatively, knowing the information about the distribution of the latch write time and for easiness of timing analysis and pulser design, a maximum value for the latch write time can be calculated for certain sigma value of the designer choice. In this case, the probability of write failure can be calculated as the probability of having the width of the pulser output smaller than this desired maximum value.

In both cases, depending on the target yield, the designer can determine the minimum acceptable value for circuit failure, and hence, the transistors' parameters can be adjusted to reach the target yield.

2.5 Voltage Scaling

2.5.1 Benefits of Voltage Scaling

Voltage scaling is a popular run-time technique used for reducing the power consumption of different circuits. As shown in Equation 2.3, for a typical power consumption in digital circuits, reducing the supply voltage will significantly reduces dynamic power (with its two components of switching power and internal power) and leakage power [53].

$$P = \alpha f C V_{DD}^2 + I_{leakage} V_{DD} \tag{2.3}$$

where α is the activity factor, f is the operating frequency, and C is the equivalent capacitance.

Meanwhile, the ability to reduce the operating supply voltage is limited by a minimum value determined usually by some timing constrains (critical path delay as an example), in addition to some margins for the PVT variations, and usually adding a margin for aging effects [68]. As the supply voltage is scaled down, the available voltage headroom decreases further and the transistors become more sensitive to any variations. On the other hand, the chip may not be always working with the lowest supply voltage. With different modes of operation of modern chips, it may be required to run at higher supply voltages for certain amounts of time to reach certain target performance [57, 41]. Hence, careful study of the effect of voltage scaling on the design performance and behavior should be considered to ensure reliable operation at different supply voltages.

2.5.2 Effect of Voltage Scaling on Pulsed Latches

The effect of voltage scaling is naturally associated with the increase of timing delays for different circuit components. While this can be handled at design time for several circuit components, the case may not be as easy for pulsed latches. Due to the different micro-architectures of the pulser and the latches, the timing of each of them is independently affected. As shown in Figure 2.5, voltage scaling affects the probability distribution of the pulser and the latch differently. As shown, the means of both distributions are nearly scaled by the same amount when the voltage is scaled down, while the standard deviation are scaled differently. This will result in changing the intersection area between both distributions. Hence, failure probability calculated at one voltage will not be the same when applying voltage scaling. Indeed, PL reliability degrades significantly when scaling down the supply voltage. As shown in Figure 2.6, the probability of write failure for a PL can increase by up to two order of magnitude when the supply voltage is scaled down by around 30%. Even if the PL circuit is designed to operate reliably at an intermediate supply voltage (0.9V as an example), the reliability will still significantly degrade at lower voltages, especially at low operating temperatures.

One possible solution is to design the PL circuit to operate with the needed level of reliability at the lowest possible operating voltage. Since chips usually operate at different supply voltages with different operating modes, when PL circuits are operating at a voltage higher than that minimum value, they will be operating with extra timing margin (the pulse width will be larger than the needed width to achieve the required level of reliability). Hence, this will negatively affect one of the main advantage of PLs which is their low timing overhead, in addition to increasing the risk of hold time violations. The proposed circuit approaches in Chapter 3 will help in forcing PLs to reliably operate with just the needed timing margins at different supply voltages. Hence, this will assure gaining the maximum benefits of

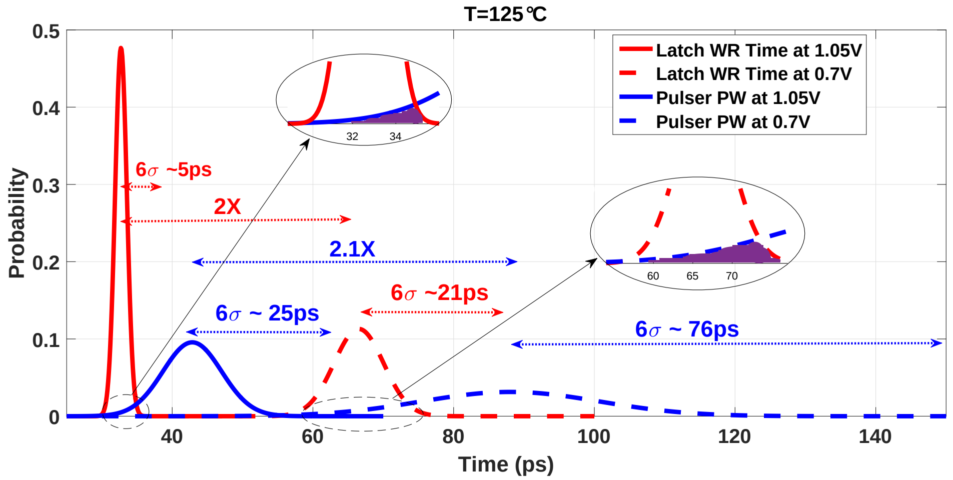


Figure 2.5: The effect of voltage scaling on the distributions of the latch WR time and the pulser PW at 125°C.

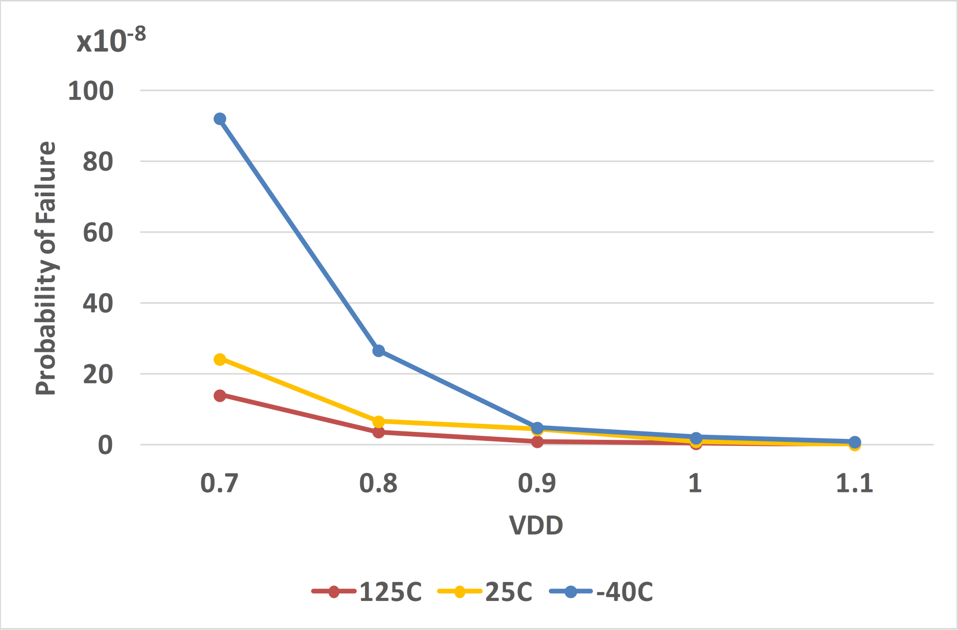


Figure 2.6: The probability of failure of a TGPL designed at nominal supply voltage at different supply voltages and temperatures.

pulsed latches at different operating modes without any unnecessary waste in the design performance.

2.6 Temperature Variations

2.6.1 Effect of Temperature Variations on Circuits Performance

Studying the effect of temperature variation on the designed circuit is very important. Not only does the variation in temperature affects leakage power and performance, but it also affects the probability of having an error during circuit operation, as well as impacting the life span of different chip parts [34]. Factors such as the increase of leakage power with technology process scaling, the nonequivalent down scaling of the supply voltage when compared to geometry scaling, and the increase in the dynamic power associated with the increase in performance required in current designs, all lead to the increase of the operating temperature. In addition, the effect of temperature on the device performance is not always the same. In fact, it can also depend on the value of the supply voltage and the available voltage headroom (i.e. also depends on the threshold voltage). When running at higher supply voltage, the transistors become slower at higher temperatures. However, this relation can be inverted at lower supply voltages, where the transistors can become faster at higher temperature. This effect is called temperature inversion [52]. The relation between the temperature and the delay depends on the variation of the carrier mobility and the threshold voltage with temperature and their effect on the driving current of the transistor I_{on} as shown in Equation 2.4 for the n-channel MOSFET [52]:

$$I_{on} \propto \mu(T)(V_{gs} - V_{th}(T))^{\beta}, \qquad V_{gs} \geq V_{th} \tag{2.4}$$

where T is the temperature, $\mu(T)$ is the carrier mobility and β is the velocity saturation effect factor.

When temperature increases, both $\mu(T)$ and $V_{th}(T)$ decreases. At higher supply voltages, the available voltage headroom is high. Hence, the influence of the decrease of $\mu(T)$ is stronger, deceasing the I_{on}, and consequently making the transistor slower. When decreasing the supply voltage, the available voltage headroom decreases. Hence, any decrease in the threshold voltage ($V_{th}(T)$ in equation 2.4) becomes more effective, making the transistor faster. One important thing to mention is that the significance of the temperature inversion effect can be different from one process node to the other. Even for the same process node, it is still function of the supply voltage and the threshold voltage of the device (i.e. the inversion effect can start at a different supply voltage value for high threshold voltage devices versus low threshold voltage deceives). Therefore, careful study of the effect of temperature variations at different supply voltages is required especially for time-sensitive sequential elements.

2.6.2 Effect of Temperature Variations on Pulsed Latches

The study of temperature effects on pulsed latches is much more critical, since each of the pulser and the latches can have a different response to temperature variations. The study we did shows that both circuits become more sensitive to process variations with the decrease in temperature. However, the pulser is more significantly affected by temperature variations. In addition, the entire PL circuit would have high failure rates when operating at a lower temperature. This can explained as follows. When running at nominal supply voltage, the transistors become faster as the temperature decreases. Since the pulser is more timing sensitive than the latch, the timing margin between the latch write time and the pulser output pulse width will decrease with the decrease in temperature as shown in Figure 2.7. Hence, the probability of write failure is expected to increase with the decrease of temperature. When scaling down the supply voltage beyond certain limit, the relation between the circuit delays and temperature is reversed due to temperature inversion effect. Again, due to the higher sensitivity of the pulser over the latch, the time margin increases as temperature decreases. However, the process variations effect increases significantly with the decrease in temperature. As shown in Figure 2.8, the standard deviation for the latch distribution is

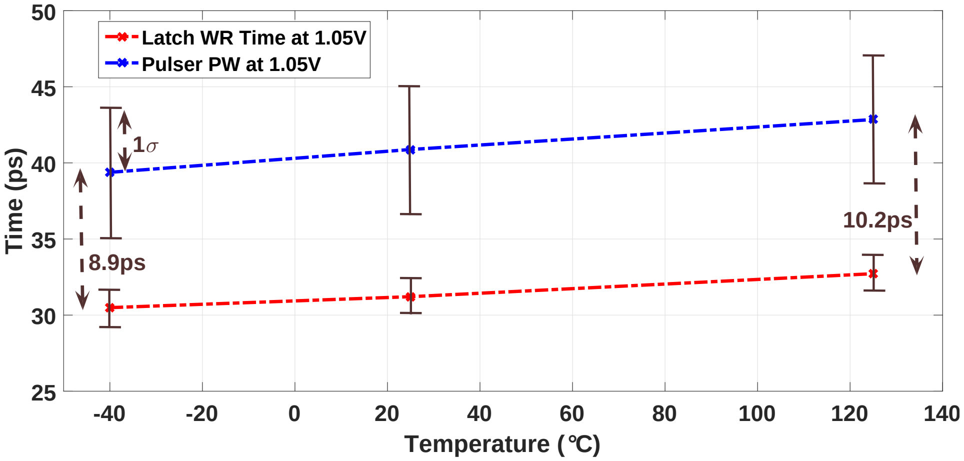


Figure 2.7: The effect of temperature variation on both the latch write time and the pulser pulse width running at nominal supply of 1.05V. The bar around different points represents the 1-sigma variation around the mean.

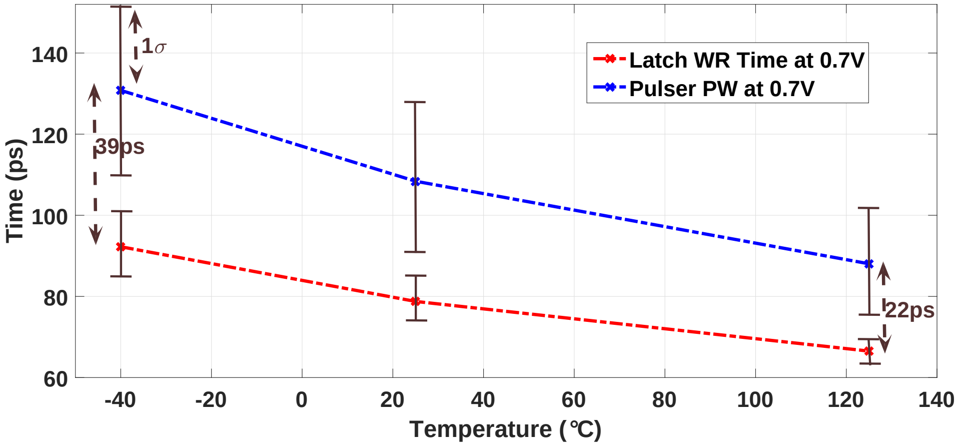


Figure 2.8: The effect of temperature variation on both the latch write time and the pulser pulse width running at scaled supply voltage of 0.7V. The bar around different points represents the 1-sigma variation around the mean.

doubled with the temperature decrease from 125°C to -40°C, while the standard deviation for the pulser distribution increases by more than 60%. This significant increase in the variation of the latch and pulser timing with the decrease in temperature leads to the increase of the probability of write failure.

Therefore, for different supply voltages, even when taking temperature inversion into account, pulsed latches become less reliable at lower temperature. Hence, to ensure correct operation at different temperatures, pulsed latches should be designed at the lowest operating temperature expected for the system. This can be a little different from one process technology to another, but similar results can be expected, at least for planar CMOS devices at closer process nodes.

2.7 Conclusion

In this chapter, we presented an analysis of the effect of process, voltage and temperature variations on the reliability of the traditional TGPL. We first discussed a proposed methodology to study these effects on the behavior of pulsed latch circuits. The methodology includes studying the process variations effect on each of the pulser and the latch independently, and then combine them together to get the probability of failure of the entire pulsed latch circuit. Then, we conducted a general discussion on each of the PVT variations, followed by the study of its effect on pulsed latch circuits.

Chapter 3

Reconfigurable Pulsed Latches

In this chapter, we propose a reconfigurable architecture for pulsed latches (PLs). The proposed architecture will allow pulsed latch to work reliably at different operating conditions without a significant overhead. We start by presenting the prior work in the literature. Then, we propose two different approaches to add the configuration ability to pulsed latches, followed by circuit designs of two registers implementing the proposed approaches. Finally, we present the simulation setup we did to test our proposed circuits and the results obtained for the reliability, power and area.

3.1 Introduction

As described in the Chapter 2, it is not easy to design a pulsed latch circuit that can operate with just the needed timing margins at different supply voltages in the presence of process and temperature variations, while keeping the needed level of reliability. If the circuit is originally designed to work properly with certain level of reliability at the nominal supply voltage, the reliability will significantly degrade when scaling down the supply voltage. If the circuit is originally designed to operate reliably at the lowest supply voltage, then there will be much excess timing margins when it operates at higher supply voltages, as the pulser pulse width will be much wider than needed. Although this will always ensure robust operation at different supply voltages, this excess unneeded timing margins will degrade the performance of pulsed latches at these higher supply voltages. In addition, a wider pulse width will increase the risk of hold time violations. While this can be solved by adding some delay buffers, this will come at the cost of higher power consumption and area.

Since sequential elements such as pulsed latch are prolifically used within the die, any degradation in their performance or reliability can significantly affect the performance of the entire chip or can even cause a large yield loss. In addition, designing a chip that can perfectly operate at just one voltage corner is not an acceptable solution nowadays. Hence, to be able to efficiently reach the needed reliability level while applying supply voltage scaling without much overhead in timing, power or area, the pulser circuit should be reconfigured at run time to generate an output pulse whose width can be controlled based on the operating condition. This will ensure reliable operation against different PVT variations, without unnecessary timing overhead.

3.2 Prior Work

Most of the pulsed latch circuits reported in the literature were designed to work reliably at the worst operating corner. In other words, the pulse width is adjusted to be wide enough to ensure correct write operation at the lowest supply voltage in the presence of process and temperature variations. Although this will guarantee correct operation at higher supply voltages, the generated pulse width will be much wider than needed for correct write operation at these higher voltages. In addition, the designer must consider the risk of possible hold time violations associated with that.

In [64], a pulse generator with a pulse width control option was presented to enhance the reliability against stress caused by Bias Temperature Stability (BTI). In [46], PLs with relatively wide pulse widths were used to allow cycle borrowing and tolerate any clock skew. In order to compensate for any hold time issues, deracer circuits were used to block any incoming fast data before the end of the pulse. In [36], the pulser used for the proposed PL is similar to the traditional pulser studied in this chapter, but it was designed to generate wider pulse for correct operation at different voltages. This paper reported the usage of delay cells to fix hold time issues. Similarly, [11, 19] reported the need of delay buffers to fix hold time violations. These additional delay buffers will add some area and power overhead that can eliminated if the pulsed latch is adjusted to work with just the needed pulse width at different supply voltages. In addition, working with these extra wide pulse width will increase the timing overhead of the pulsed latches when working at higher voltages. Since pulsed latches are usually used in high performance systems, because of their low timing overhead compared to flip-flops, any degradation in their timing characteristics will affect the performance of the entire system.

3.3 Proposed Design Approaches

As mentioned earlier, to be able to reach the needed reliability level at different supply voltages, the pulser circuit should be reconfigured at run time to generate an output pulse whose width can be controlled to ensure robust operation. As shown in Chapter 2, each of the latch write time and the pulser pulse width are represented by probability distributions due to process variations. Shown in Figure 3.1(a) is the PDFs of a pulsed latch circuit designed to operate correctly at nominal supply voltage with high level of reliability. However, when scaling down the supply voltage as shown in Figure 3.1(b), the circuit become less reliable with higher probability of failure (indicated by the increase in the intersection between the

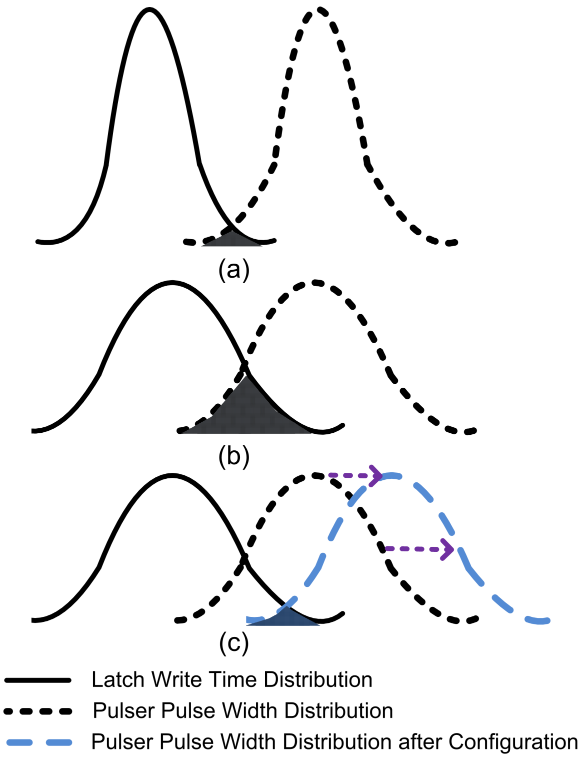

(a)
(b)
(c)
Latch Write Time Distribution
Pulser Pulse Width Distribution
Pulser Pulse Width Distribution after Configuration

Figure 3.1: A diagram showing arbitrary PDFs for the pulser and the latch when (a) operating at nominal supply voltage, (b) scaling down the supply voltage without configuring the pulser (or having a fixed pulser), and (c) scaling down the supply voltage and configuring the pulser circuit to generate a wider output pulse .

two distributions). The required level of reliability can be achieved at the lower supply voltage by increasing the width of the generated pulse. As shown in Figure 3.1(c), this is equivalent to shifting the pulser probability distribution to the right, compensating for the increased variation effects at lower voltages and therefore, decreasing the probability of circuit failure.

In this section, two design approaches are proposed. Both approaches depend on controlling the delay path (the delay unit and its following inverter) of the pulser circuit shown in Figure 3.2 by using an external control signal (CTRL) to generate a controllable pulse width.

The first approach considers splitting the supply rail of the pulser circuit, and applying an additional controllable level of voltage scaling on the delay path when needed. The second approach relies on using multiple delay units in the pulser circuit and choosing a certain delay unit at run-time according to the operating condition. Detailed discussions of these two approaches are presented in the next two subsections.

3.3.1 First Approach

This approach is based on using a virtual supply rail for the delay path of the pulser. This rail is driven from the main supply rail used for the rest of the pulser circuit and the latches. This can be accomplished using header PMOS switches for the delay path of the pulser circuit, similar to the the local power gating topology [18]. This is shown in Figure 3.3, where turning off some of these switches will result in lowering the supply voltage of the delay path. Since this delay path is the main part of the circuit that control the width of the generated pulse, controlling the supply voltage of this path will result in controlling the output pulse width.

Separate control signals can be used for different switches, where at least one of these switches must be always turned ON (i.e. the gate of this PMOS switch should be tied to the ground), giving the maximum output pulse width. The other switches can be turned ON or OFF to achieve the required narrowing of the pulse width. The number of these parallel switches and their sizes will depend on the number and values of the virtual supply voltage levels, which corresponds to the needed pulse widths to achieve the target reliability level at different operating conditions. Since the current of the delay chain represents only 20-30% of the total pulser current, the sizes of these PMOS switches should be reasonable, adding a small area overhead to the pulser circuit.

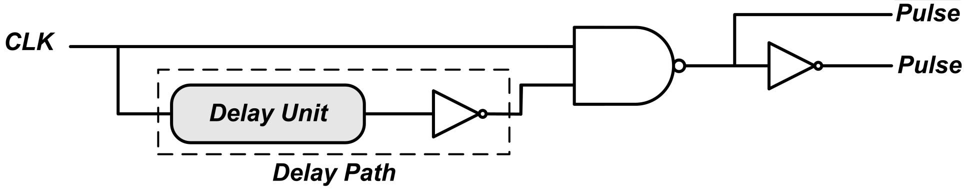


Figure 3.2: A delay path-based pulser.

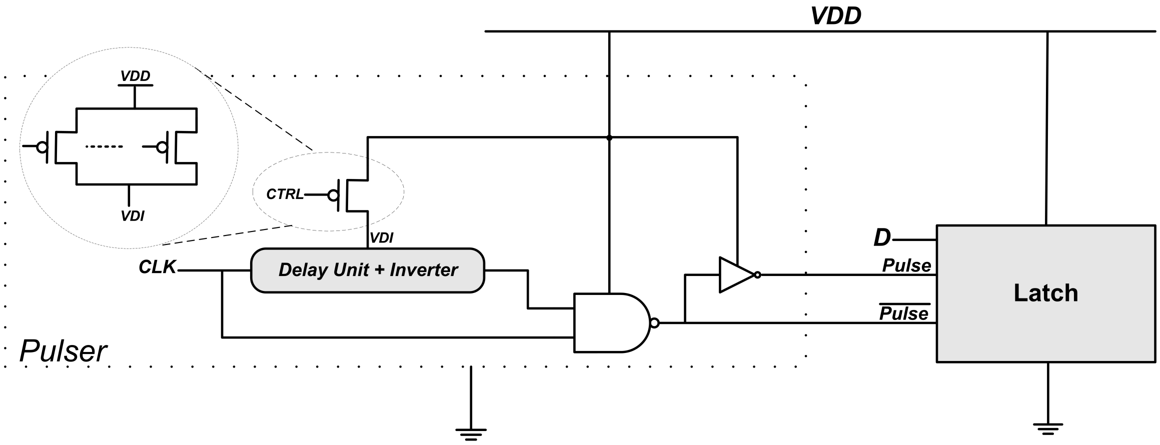


Figure 3.3: The proposed header switches-based pulser design.

During normal operation, when required to operate at the nominal supply voltage, all header switches are ON, driving the whole pulser circuit by nearly the same supply voltage value as the driven latches.

When scaling down the supply voltage, the needed margin for variations in the latch write time increases. By turning OFF some of the pulser header switches, an additional down scaling of the virtual supply of the delay path (*VDI*) is provided; i.e., the delay unit and its following inverter will be running at a slightly lower supply than the rest of the pulser circuit. This additional voltage scaling of *VDI* will result in a small increase in the pulser output pulse width. Since the circuit is already operating with a small voltage headroom at this lower supply voltage, even a very small decrease in *VDI* will be sufficient to produce an adequate increase in the pulse width without having a significant difference between the supply voltages of the delay path and the rest of the pulser circuit. In addition, the remaining pulser circuit (the NAND gate and the output inverter) will act as a voltage level shifter, driving the latches by the same voltage level as their supply voltage.

3.3.2 Second Approach

Since the pulse width depends on the delay unit as shown in Figure 3.2, implementing multiple delay units with different delays can help in generating pulses with different widths. One important design consideration is the ability to choose between these different units post silicon or at run-time. The second proposed pulser design is shown in Figure 3.4. Each delay unit represents a buffer chain that can be implemented in different ways. It can be as simple as a very small delay unit (i.e. just a wire) and up to multiple even number of inverters of different inverter sizes and/or numbers. The output of the multiplexer is used to drive an odd number of inverters, whose final output is connected to the NAND gate.

By selecting a longer delay chain, the latch transparency window can be increased at run time, which is required when scaling down the supply voltage. The shortest delay unit is designed such that, when operating at a nominal supply voltage, the circuit is verified to run with very low probability of failure in the presence of different process and temperature variations. The rest of the delay units are designed depending on the number and values of the supply voltage scaling levels. In addition, the delay of these longer path units must be carefully adjusted such that the generated wide pulse width is just enough to compensate for the expected variation in the latch write time, in order to decrease the possibility of hold-time violations.

Aside from the ability to scale the supply voltage while ensuring correct functionality, the same circuit approach also can be used to utilize additional time borrowing. If, for example,

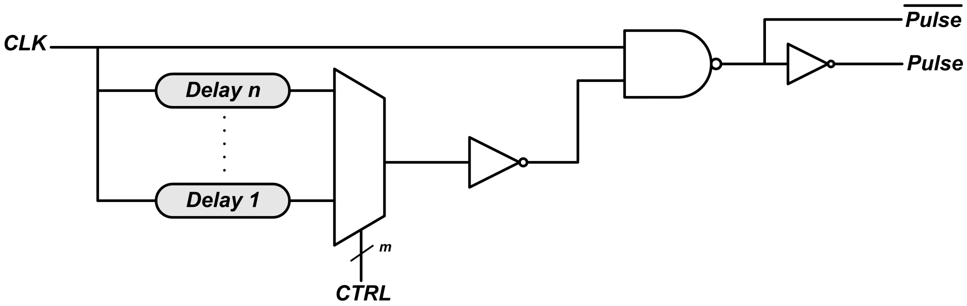


Figure 3.4: The proposed MUX-based pulser design.

it is required to increase the operating frequency of a pulsed latch-based pipeline stage above the normal limit, the pulser of the register at the end of a critical path can be adjusted to generate a wider pulse width. This will increase the allowable operating period for the critical path by borrowing some time from the next non-critical path. Thus, correct operation at higher frequencies can also be obtained.

3.4 Circuit Design

3.4.1 Register Design

To verify the proposed approaches, test circuits of 16-bit register were examined, where three implementation choices for the register were compared. All the three implementations consists of a single pulser driving sixteen identical latches. The first choice is the implementation using the traditional non-configurable pulser shown in Figure 3.2. The pulser was designed at nominal supply voltage to ensure the required reliability level. The second and the third choices are the two proposed pulser implementations, also driving sixteen identical latches. The effect of voltage scaling of one scaling level was applied on all circuits. An extreme value of voltage scaling which is usually around 30% reduction from nominal supply value was used to show the effectiveness of the proposed approaches. The same approaches can be easily extended to any other scaling values.

The traditional PL is designed to have three inverters in its delay path to generate the needed pulse width. The transistor sizing for this pulser (which is also a common part in the two proposed designs) follows rules close to that described in [7]. All the transistors have the minimum length, and only the transistors widths are varied. The first inverter is chosen to be of minimum size to reduce the load on the clock network. Hence, the sizes of the second and the third inverters are adjusted to determine the needed pulse width. With the technology used to implement this design, stacked transistors were used for the second and the third inverters. This is used in order to get the needed delay without the addition of more inverter, which will be larger in area. The NAND gate and its following inverter are sized depending on the load they drive in order to generate reasonable sharp edges for the output pulse.

For the first proposed pulsed latch approach, which is the header switches-based design (PL-SW), the pulser header consists of two switches as shown in Figure 3.5. One switch is always turned ON by tying its gate to ground, while the other one is turned ON or OFF by the input control signal (CTRL). This will add one additional level of voltage scaling to the delay path's virtual supply voltage (*VDI*). The size of the always-on header switch is chosen to ensure the correct operation at the down-scaled supply voltage with the required reliability level, since this switch will determine how much lower will *VDI* be when compared to *VDD*. The size of the other controllable header switch is chosen to reduce the voltage drop across the switches when this switch is turned on, and hence, driving *VDI* to be very close to the main supply voltage *VDD* when running at nominal supply value.

For the second proposed approach, which is the multiplexed delay units-based design (PL-MUX), a pulser with two delay paths was designed as shown in Figure 3.6, having either three or five inverters in the delay path. The sizes of the three main inverters after the

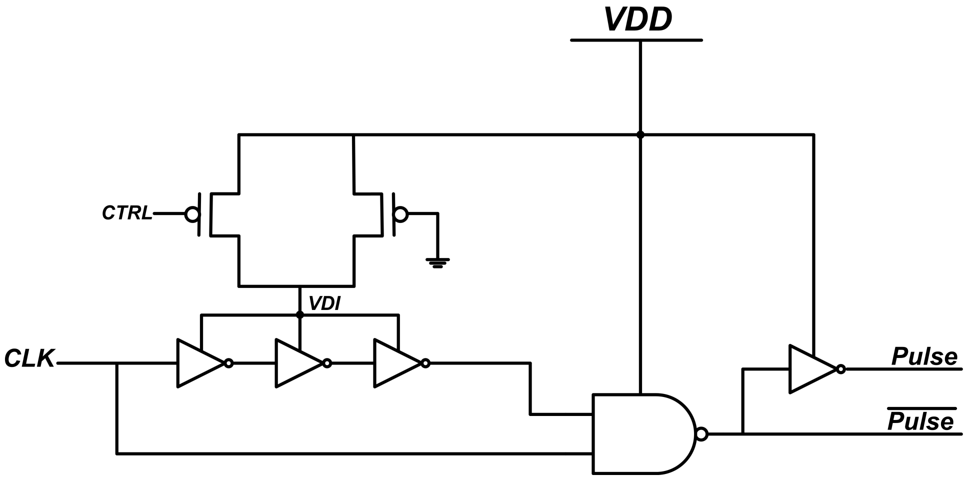


Figure 3.5: The structure of the pulser used for the PL-SW based register.

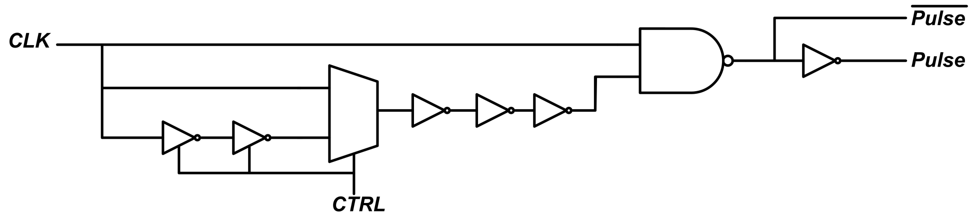


Figure 3.6: The structure of the pulser used for the PL-MUX based register.

output of the multiplexer were chosen to ensure correct operation at nominal *VDD* with the required reliability level. The sizes of the two inverters at the input of the multiplexer were chosen to ensure the same reliable operation when scaling down *VDD*. In addition, to save power, these two inverters can be turned off by the multiplexer control signal when the other delay path is selected. The multiplexer is designed using two transmission gates, one for each multiplexer path. The delay through these transmission gates are considered when sizing the three main inverters at the multiplexer output.

3.4.2 Control Circuit Design

The control signal (CTRL) used to select the required pulse width can be generated by one of two methods. In a system with two supply rails, the CTRL signal can be driven by the same control signal used to switch the power gates used to choose the operating supply rail. For a system with a single supply rail, a circuit can be designed to generate an output voltage that is dependent on the value of the *VDD*.

For the proposed register circuits, the circuit in Figure 3.7 was designed, assuming a system with a single supply rail. The first stage of this circuit is a voltage divider of the supply voltage that generates an output voltage of value a little less than half the *VDD*. The output of this voltage divider is used to drive a pseudo-NMOS inverter circuit. The pull down network (PDN) of this inverter is a strong NMOS, while the pull up network (PUN) is built using a weak PMOS in parallel with a weak NMOS. The presence of both NMOS and PMOS in the PUN is to ensure reliable circuit operation at different process corners, especially

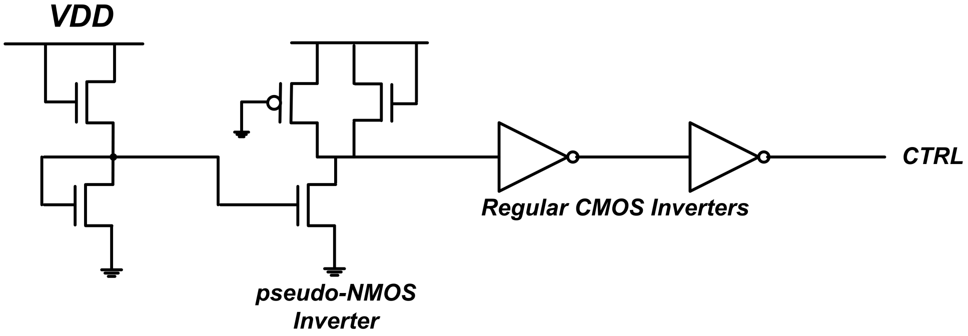


Figure 3.7: The structure of the circuit used to generate the CTRL signal for the PL-SW and PL-MUX registers.

the fast-NMOS slow-PMOS (FS) corner. The output of this pseudo-NMOS inverter passes through few regular CMOS inverters to generate the final CTRL signal.

At nominal supply voltage, when the value of *VDD* is high, the output of the voltage divider will be high enough to strongly turn on the PDN NMOS of the pseudo-NMOS inverter, generating a lower voltage value at its output. Since the PUN of the circuit has weak NMOS and PMOS devices, the generated output of this pseudo-NMOS inverter will be low enough to be interpreted as a logic '0' by the following CMOS inverter.

At scaled-down voltage, when the value of *VDD* is low, the output of the voltage divider will be low enough to make the NMOS of the pseudo-NMOS PDN hardly on. Hence, the always-on PMOS and NMOS of the PUN will generate a high enough voltage to be interpreted as a logic '1' by the following CMOS inverter. The regular CMOS inverters are used to adjust the voltage levels of the CTRL signal and to generate the needed voltage polarity.

Since this circuit can be shared between different resisters in a design block, any area or power overhead associated with it can be negligible. In addition, the same circuit can be replicated to generate additional control signals (for multiple supply voltage scaling levels), where the switching threshold for each circuit can be controlled by adjusting the sizes and the threshold voltages of each transistor.

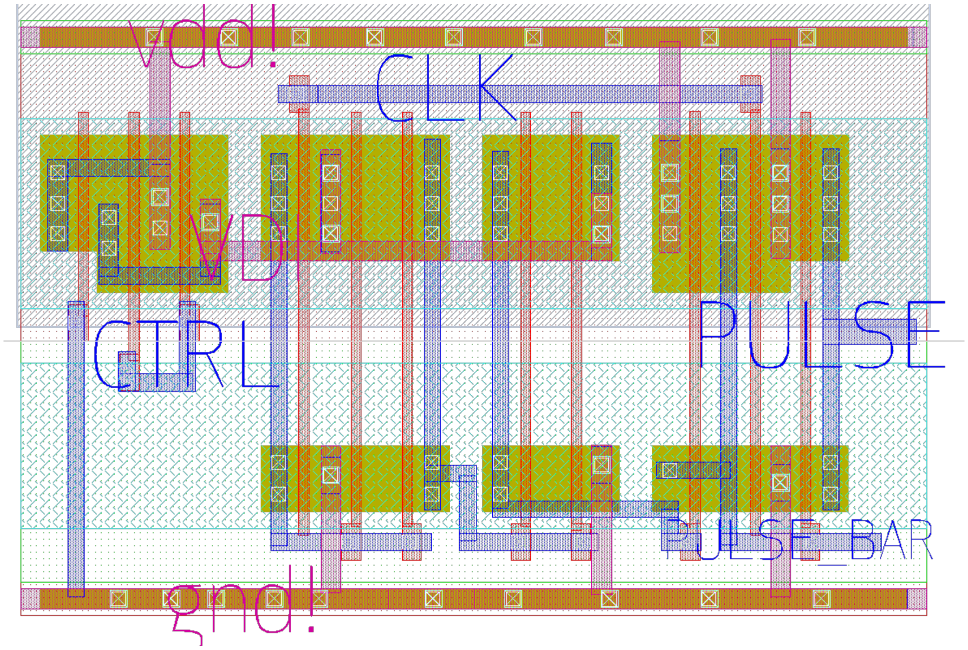
3.5 Simulation Setup

The experiments were carried out using the Synopsys 28nm PDK [2, 31]. All of the implementations were examined at the same frequency of 1 GHz and the rise time of the input clock signal was chosen to be 50ps. The nominal supply voltage used was 1.05V and the same level of supply voltage scaling to 0.7V was applied. The analysis for the effect of temperature variation was conducted to cover a typical industrial range of temperature variation from -40°C up to 125°C. The simulations were done using Hspice and the variability analysis was carried out using Solido Variation Designer [42] and Matlab. All the simulations and analysis were carried out on the post-layout extracted circuits. The power numbers were calculated over a few hundred cycles of common activity levels for the different register implementations. The area calculations were carried out through layouts drawn with Synopsys Custom Designer and verified with Synopsys IC Vaildator using the same 28nm technology. The layouts of proposed PL-SW pulser and PL-MUX pulser are shown in Figure 3.8. For simplicity, the shown layouts are for the pulser circuits only. However, the reported areas were calculated on the complete 16-bit register layout, which include the 16 latches in addition to some buffers.

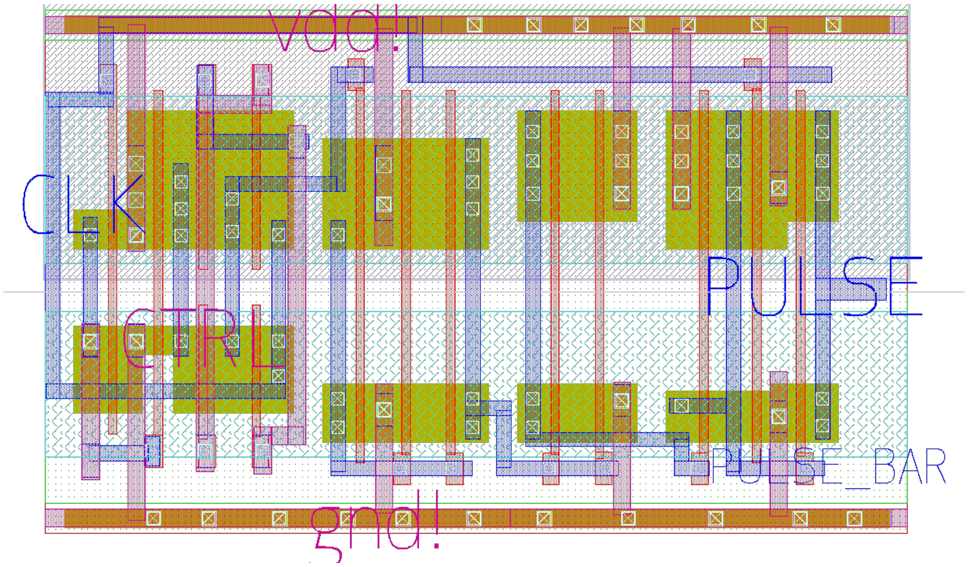
Since sequential elements are replicated in large numbers in a chip die and across the entire wafer, a wide range of up to $\pm 6\sigma$ of process variation was considered in the variability analysis [42], where the High-Sigma Monte Carlo (HSMC) tool of the Solido Variation Designer was used to do the variation analysis.

A high design yield that is higher than 99% was chosen. Hence, a target probability of write failure (P_{WF}) that is less than $1 * 10^{-8}$ was set using [45]:

$$Yield = (1 - P_{WF})^n \tag{3.1}$$



(a)



(b)

Figure 3.8: The layouts of the proposed pulser circuits: (a) PL-SW, (b) PL-MUX.

where n was chosen to be in the order of 10^6 cells.

The probability of write failure P_{WF} can be calculated as the probability of having a pulser pulse width PW smaller than an estimated maximum value for the latch write time (T_{wr-max}). This T_{wr-max} is calculated as the latch write time (T_{wr}) at 6σ of the write time distribution (i.e., $T_{wr-max} = \mu_{wr} + 6\sigma_{wr}$, where μ_{wr} is the mean of the distribution and σ_{wr}

is the standard deviation). Hence, P_{WF} can be calculated as

$$P_{WF} = P(PW < T_{wr-max}) \; * \; P(T_{wr} > T_{wr-max}) \tag{3.2}$$

3.6 Results

3.6.1 Reliability Analysis

Since the traditional PL circuits do not have any configuration ability, they were designed to ensure correct functionality at nominal supply condition. Correct functionality means that the pulsed latch circuit can achieve the target level of reliability (i.e. target probability of failure) when running at nominal supply at the entire range of temperature values in the presence of process variation.

When the supply voltage is scaled down, the traditional PL register becomes more susceptible to write failure. As shown in Table 3.1, the P_{WF} of the traditional PL register is within the required value (less than $1 * 10^{-8}$) at nominal supply voltage. However, P_{WF} increases when the supply voltage is scaled down to 0.7V. Hence, the circuit becomes much less reliable at this lower voltage. The latch maximum write time T_{wr-max} and the pulser pulse widths PWs at different supply voltages are shown in Figure 3.9 for the PL-SW design. The typical values for the PWs are chosen to be the mean of their distributions, while the minimum values for the PW are arbitrary chosen to be 3σ lower than the mean. The short configuration of the PL-SW is nearly the same architecture as the traditional PL. When the supply voltage decreases, the latch T_{wr-max} starts to move away from the typical short PW and closer to the minimum short PW, increasing the probability of write failure as shown in Table 3.1. If the pulser can be configured to generate longer PW at lower supply voltages, the reliable timing relation between the T_{wr-max} and the pulser PW can be regained, where

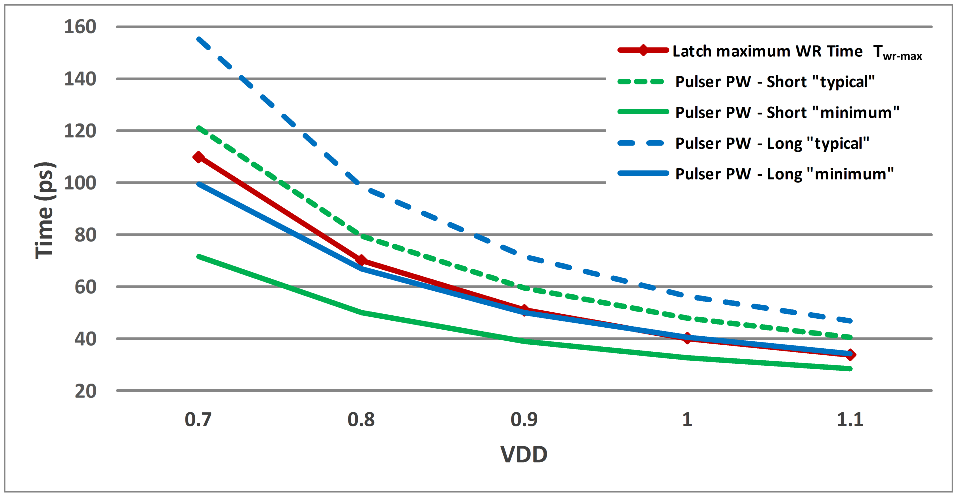


Figure 3.9: The latch worst write time and the typical and worst pulser pulse widths for the two configurations of PL-SW at different supply voltages at 25°C.

the T_{wr-max} becomes closer to the typical PW, lowering the probability of write failure.

For both proposed designs, PL-SW and PL-MUX, the require reliability levels can be achieved at different supply voltages, within the entire temperature range in the presence of process variations without adding any unnecessary timing overhead. With the added reconfiguration ability, both designs were independently characterized to function properly at the two different operating voltages (nominal and 30% down-scaled voltages). This was used to adjust the sizes of the header switches for the PL-SW design, and adjust the design of the delay units for the PL-MUX design.

When running at the nominal supply voltage, all of the header switches are turned on for the PL-SW design, while the multiplexer is switched to the short delay unit for the PL-MUX design. When scaling down the supply voltage, the two proposed approaches depend on

Table 3.1: The probability of write failure for the three different register implementations at different supply voltages and different temperatures.

T	$V_{DD} = 1.05$ V			$V_{DD} = 0.7$ V		
	Traditional PL	PL-SW	PL-MUX	Traditional PL	PL-SW	PL-MUX
125°C	$1.2 * 10^{-9}$	$3.9 * 10^{-10}$	$2.8 * 10^{-10}$	$1.4 * 10^{-7}$	$2.7 * 10^{-9}$	$1.1 * 10^{-12}$
25°C	$3.0 * 10^{-9}$	$8.6 * 10^{-10}$	$5.2 * 10^{-10}$	$2.4 * 10^{-7}$	$2.8 * 10^{-9}$	$1.1 * 10^{-11}$
-40°C	$9.5 * 10^{-9}$	$2.8 * 10^{-9}$	$1.5 * 10^{-9}$	$9.2 * 10^{-7}$	$9.9 * 10^{-9}$	$7.9 * 10^{-9}$

increasing the pulse width. This is accomplished by turning off one of the switches for the PL-SW design or switching to the long delay unit for the PL-MUX design using the CTRL signal generated by the circuit shown in Figure 3.7. As shown in Figure 3.10 for PL-SW and Figure 3.11 for PL-MUX, this results in shifting the probability distribution of the pulser output to the right, compensating for the increased variation effects at lower voltages, and hence, decreasing the probability of circuit failure as shown in Table 3.1. Therefore, the required high level of reliability at different voltages is obtained at the cost of a very small overhead in area and power.

In addition, to ensure a reliable operation of the circuit that generates the CTRL signal, the circuit was tested at different process corners and at the wide temperature range from -40°C up to 125°C. In addition, the correct operation was verified at 5% lower than the 1.05V and at 5% higher than the 0.7V to ensure tolerance to any variations in the supply voltage regulator and the power delivery network.

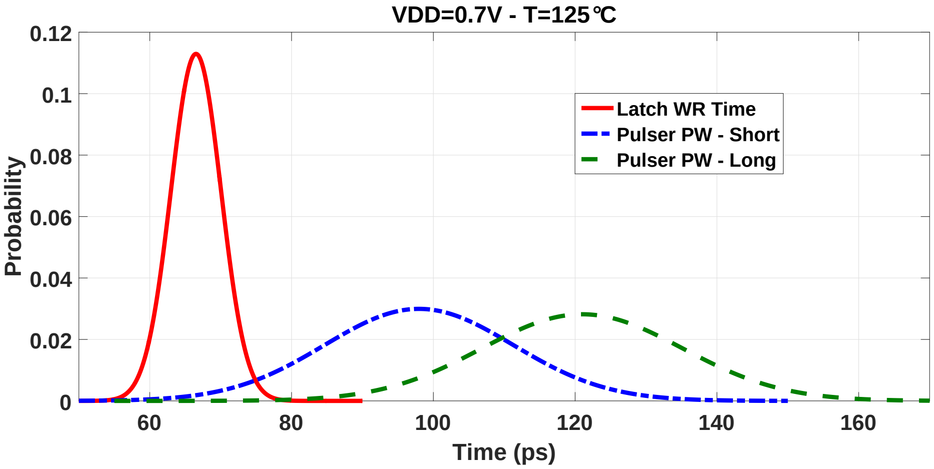


Figure 3.10: Probability distribution function for PL-SW before and after configuration for VDD=0.7V at 125°C.

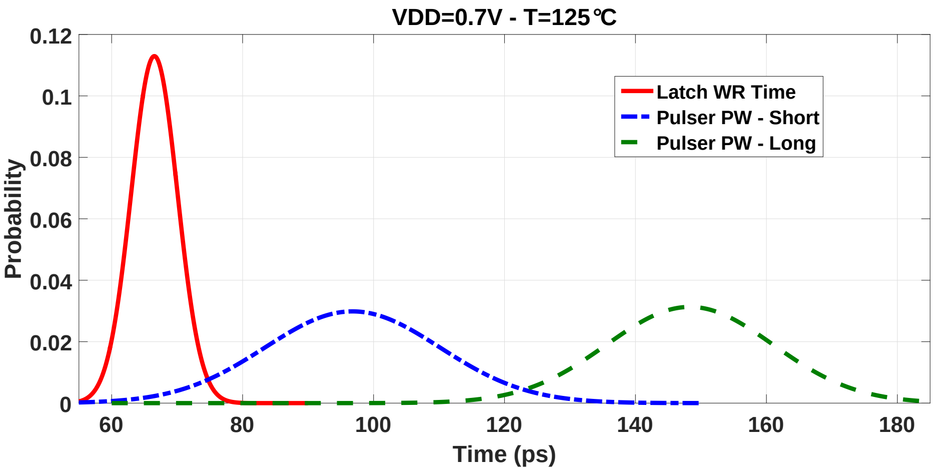


Figure 3.11: Probability distribution function for PL-MUX before and after configuration for VDD=0.7V at 125°C.

3.6.2 Power and Area Comparisons

Since power consumption is an important metric for such circuits, any power overhead associated with the proposed approaches should be minimized. Figure 3.12 shows the average energy per cycle for different register designs normalized to that of the traditional PL register at the 1.05V and 125°C. During normal operation, when running at nominal supply voltage, both the PL-SW register and the PL-MUX are nearly consuming the same amount of energy as the traditional PL register.

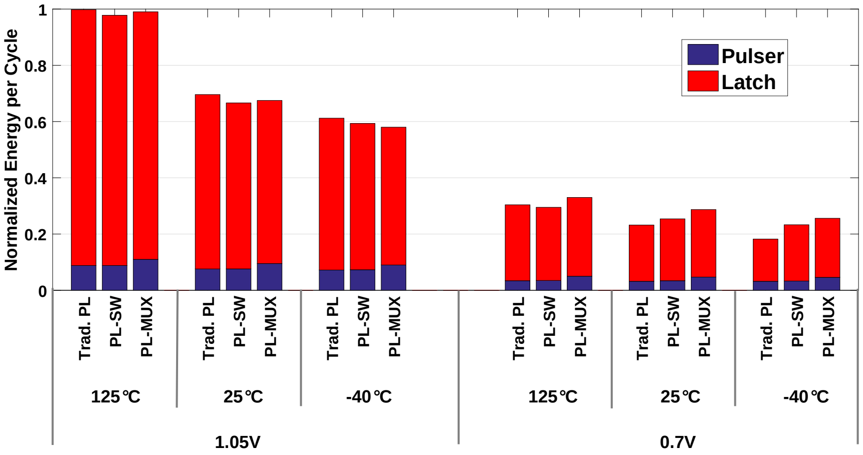


Figure 3.12: The average energy per cycle normalized to the energy per cycle of the traditional PL register at 1.05V and 125°C.

When scaling the supply voltage down to 0.7V, both the PL-SW and PL-MUX registers seem to consumes more power. However, the traditional PL register has higher probability of failure at this lower voltage, which make its energy numbers of no sense unless that working at this low level of reliability is acceptable. Comparing the two proposed approaches, PL-MUX register consumes between 9% to 14% more power than the PL-SW register due to the additional switching delay units.

Regarding the area overhead, each approach has added some transistor to the pulser circuit. However, the area overhead of the added circuits is very small. The overhead in area of the PL-SW register compared to the traditional PL one is only 2.4%, while the area overhead of the PL-MUX register is 3%.

3.6.3 Combined Metric for Reliability and Power

While the area overhead of the proposed approaches can be negligible (3% or less), power overhead is still significant. Hence, there is a need to evaluate the gained benefits from the proposed designs when compared to the added power overhead. A new metric is introduced to evaluate the gained benefits in reliability when compared to the added power consumption. This metric is represented by the design yield, calculated using equation 3.1, per unit power, and it is calculated for the three different register implementations. The results of the design yield per unit power for the two proposed implementations when normalized to that of the traditional PL register at the same voltage and temperature are shown in Table 3.2. If this normalized number is higher that one, this means that the proposed design is more reliable than the traditional design for the same power budget (or the gain in reliability overcomes any increase in power consumption).

When running at nominal supply voltage, the proposed designs are running as reliable as the traditional PL register with negligible power overhead at different temperatures. Hence, their design yield per unit power is nearly the same (or even slightly better) as that of the traditional PL design. When the supply voltage is scaled down, only the two proposed designs can keep the same level of reliability with a very small power overhead at different temperatures. Hence, the two proposed designs show great advantages over the traditional PL register at the entire range of voltages and temperatures.

3.6.4 Sensitivity to Clock Rise Time

The pulser operation depends on the input clock. The clock signal is routed through a clock network by the clock tree synthesis tool (CTS) in order to reduce clock skew between different nodes in the design [29]. While the main goal of CTS is to reduce clock skew, the designer can put some constraints on the maximum transition time for the clock signal. Hence, each internal clock node can have a slightly different transition time. Therefore, the effect of clock transition (specifically the clock rise time since the pulse generation starts at the clock rise) should be studied. Running few simulations with different transition times shows that the two proposed designs are not significantly affected by the clock rise time. This can be shown in Figure 3.13 for PL-SW when running 1000-sample Monte Carlo simulations for different clock rise time. Also, an output pulse was generated successfully with no failure at both 1.05V and 0.7V when running 1000-samples of Monte Carlo simulations while varying the clock rise time up to 150ps. Similar results were also obtained for the PL-MUX design. This makes the two proposed designs very comparable to the design proposed in [23] for the wide range of supply voltage considered in this study.

Table 3.2: Normalized yield per unit power for the two register implementations using the two proposed pulsed latch designs.

T	$V_{DD} = 1.05$ V		$V_{DD} = 0.7$ V	
	PL-SW Register	PL-MUX Register	PL-SW Register	PL-MUX Register
125°C	1.03	1.01	1.18	1.06
25°C	1.04	1.02	1.18	1.04
-40°C	1.04	1.07	1.94	1.79

3.6.5 Discussion

Comparing the results of the two proposed approaches, each has advantages and drawbacks. While the PL-SW design is smaller in area and requires less power, the design of the power switches is more complicated specially if several control level is needed. In addition, when all the switches are ON (i.e. running at nominal supply voltage), there will still be a small voltage drop on the power switches. Hence, the delay unit and its consecutive inverter will still run at a slightly lower voltage than the rest of the pulser circuit, generating a pulse width slightly larger than expected. One possible solution is to increase the sizes of the switches, however, this will increase the pulser area.

On the other side, the PL-MUX is simpler and easier in design. In addition, it can be easily implemented using standard cells. However, its power overhead is much larger due to the increase of dynamic power with the extra inverters. In addition, the area and power will exponentially increase with each additional voltage scaling level, due to the additional delay units and the larger multiplexer.

Hence, for designs with few voltage scaling levels, the PL-MUX design can be preferable over the PL-SW one, as it is easier in design, generates more precise pulse widths, and its overheads (power and area) are reasonable. On the other hand, for designs with large number

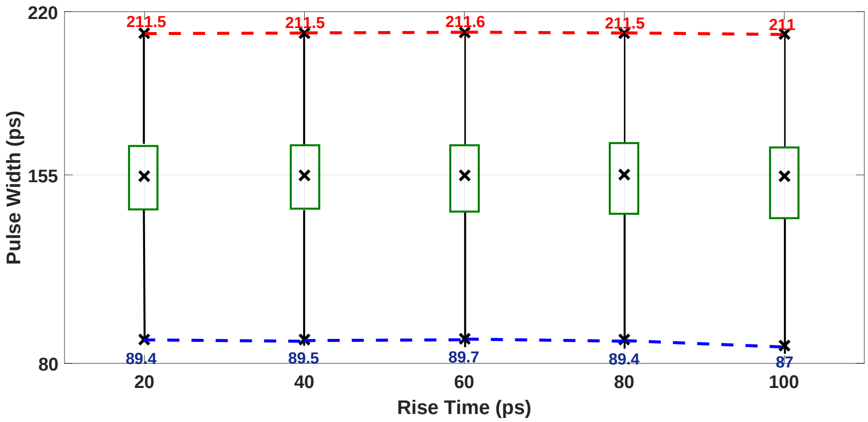


Figure 3.13: Box plot of pulse width of PL-SW at 0.7V and 25°Cover 1000-samples of Monte Carlo simulation.

of voltage scaling levels, the PL-SW design is preferred, as the area and power overheads of the PL-MUX design will be significant.

3.7 Conclusion

In this chapter we presented two proposed approaches for designing a reconfigurable pulsed latch. Both proposed approaches have the advantage of keeping the required level of reliability at different supply voltage conditions with the minimum power and area overhead when compared to the static traditional pulsed latches. Normalized design yield per unit power was introduced as a new metric to evaluate the gained benefits in reliability when compared to the added power consumption.

In addition to the benefits gained in reliability, each of the two approaches is running with just the needed margins at different supply voltage values, hence, minimizing the timing overhead. Without any configuration ability, the pulsed latch should be designed to operate with high reliability at the lowest supply voltage (the pulse width is increased to ensure correct operation with low failure probability). This will result in much wider pulser pulse width when the circuit is operating at higher voltages. Hence, this will add unnecessary extra timing overhead, in addition to increasing the chances of hold time violations. These hold time violations can be solved by adding some delay buffers. However, this will degrade some of the advantages of pulsed latches which are their lower timing overhead and their lower sequential and clock network power. Therefore, reconfigurable pulsed latches are needed. In addition, any overhead associated with adding the configuration ability should be minimized. With the very small area overhead of PL-SW and PL-MUX and the elimination of the need to insert delay buffers, our two proposed approaches are even expected to save area. Since PL-SW doesn't add any significant power overhead, power saving are also expected (PL-MUX can also save power if large number of delay cells must be used).

Although the two proposed approaches was used for the TGPL, both approaches can be easily used with any other pulsed latch topology whose pulser uses a delay chain to control the generated pulse width.

Chapter 4

Register File Design using Pulsed Latches

4.1 Introduction

Register files represent a substantial portion of the energy budget in modern processors [75]. Together with the cache array, they also consume a significant area. Hence, performance, power and reliability of the overall processor operation will be greatly influenced by the register file design. In addition, since register file are accessed frequently, high access power can turn the register file to be the hotspot of any chip. This can, in turn, degrade the performance of the devices and speed up their aging [47].

Static Random-Access Memories (SRAMs) have been known to be used for the implementation of register files in CMOS chips. Since SRAMs are used as memory storage for large number of bits, each single SRAM cell (that is used to store a single bit) is designed in the minimum possible size to ensure correct read and write operation with reasonable noise margins.

Alternatively, standard cells such as flip-flops and latches can be used to store data. However, since these sequential cells are designed in the library for multiple usages inside the main core of the chip, their design criteria and constraints are much different than that of the SRAM cells, which are only used for data storage. That's why the area of a flip-flop or a latch can be 3-5X the area of an SRAM cell. Hence, the SRAM are preferred for implementing medium to large sized register files. However, for a small size register file, the area and power overhead associated with the SRAM peripherals can overcome the area benefit of the single cells, making SRAM less attractive for small size data storage [33]. Hence, for a small size storage, standard cell based register files can be more attractive than SRAM. In addition, for chips that uses ultra low supply voltage or needs to operate at an ultra wide voltage range, standard cell based register files are usually preferred [44, 12, 14, 15].

In this chapter, we will discuss the different implementations of register files in details. We start by studying register files with one read and one write port. Then, we introduce the usage of pulsed latches to build the standard cell based register files instead of the traditional implementations using flip-flops and latches. After that, we discuss the design of multiport register files. We discuss the different methods of implementations, in additions to their drawbacks. Finally, our proposed implementation for multiport register files is presented, followed by simulations and results showing the advantages of our proposed implementation.

4.2 Prior Work

Register file implementation has been discuss extensively in the literature. In [21], a register file architecture composing of multiple banks was introduced to reduce the access time and bypass logic complexity. In [10], a two level register file organization was introduced to reduce the size and number of needed ports for superscalar processors. In [75], different register file circuits were compared. The comparison focused on their energy efficiency as a function of the number of registers and the number of ports.

In [3], a novel asymmetrically ported register file implementation was proposed to reduce power consumption. The reduction in power was obtained because some of the ports can only read and write to the lower significant value of the register instead of accessing the entire register. In [24], the usage of array replication and double pumping was discussed and used to implement register files with four read and two write ports using register files with only two read and one write ports. In [71], the usage of multi-pumping and bank replications of block RAMs for the implementation of register files on FPGA was discussed. A new design using shift register to implement multi-pumping was proposed to save area and power, while keeping high performance.

In [43, 44], standard cell based implementation of register files with single read and write ports was discussed. The usage of either flip-flops or latches to build the register files was discussed and compared with the SRAM implementation. In [12], 16x32-bit register file with two read ports and one write port was implemented using pulsed latches for ultra wide voltage range. In addition, the area and power consumption is compared to a similar register file built using flip-flops. In [73], the usage of STT-RAM to build register files for GPUs was discussed. Two techniques were tested to enhance the performance and power consumption of the implemented register files.

Hence, it seems that a lot of efforts have been done to propose different architecture for an efficient implementation of register files. Some of these efforts were done at the circuit level, while others were done at the architecture level. The aim was to reach an energy efficient implementation of register file that can operate at the needed performance level and occupy the smallest possible area. In addition, novel techniques were examined to add more read and write port without significant degradation in performance and with little area and power overheads.

4.3 Design of a Single Read - Single Write Register File (1R1W)

In this section, we will discuss four different implementation options for a commonly used register file with two ports: one port dedicated for read operation and the other port is dedicated for write operation. Hence, we will call it 1R1W register file. In our discussion, we are assuming a register files of size $W * B$, where W is the number of words (registers) that a register file can store, while B is the number of bits per each word. For simplicity, the register file can be imagined as an array of W rows and B columns[1]. Also, as common to most of the register files, we assume a word access scheme, where the entire register can be accesses for read and write. In addition, a maximum of one clock latency is assumed for read and write operations.

4.3.1 SRAM Based Implementation

SRAMs have been widely used in different applications such as large data storage, caches, and register files. For a regular memory storage, the popular six transistor (6T) SRAM bitcell is used to built the memory array, providing a single port that can handle only one read or one write operation per cycle. Some architectures modified the 6T structure by splitting the word line to two separate lines to enable two read operation or one write operation per cycle [32]. However, simultaneous read and write operations can't be performed unless time multiplexing is done, where write operation occurs on the first half of the clock cycle and the read operation occurs on the second half. However, this needs a very careful timing and puts some constraints on the minimum clock period.

To build a register file with physically separate read and write ports, the memory bitcell

[1] Note: For the remaining of this chapter, the words *register*, *word*, and *row* will be used interchangeably, where all of them mean a single location of the register file with B number of storing elements, where all the B stored bits are accessed at the same time for read or write.

becomes more complicated. A typical modification is by adding two more transistors to form an 8T bitcell as shown in Figure 4.1[67, 37, 39]. The write operation is done similar to the regular 6T SRAM, where the word line selector (*wr*) is turned high by the write row address decoder, while applying the new data to be written and its compliment to the write bitlines (*wr_bit* and $\overline{wr\_bit}$) through the column decoder and the sense amplifiers. The read operation is single-ended, where the read line selector (*rd*) is turned high to connect the read bitline (*rd_bit*) to the selected cell. If the cell is storing '1' (i.e. Q is VDD), then the *rd_bit* is pulled down from its pre-charge value, while if the cell is storing '0', the *rd_bit* keeps its pre-charge value. In addition, the two read transistors isolate the storage node during read operation, which increases the read stability at lower supply voltage [16, 63]. Some other SRAM bitcells adds two more read transistors at the Q_b side to form a differential read port [67]. Hence, the SRAM bitcell becomes a 10T cell which will results in significant area increase.

The remainder of the circuit is close to the regular SRAM, with the exception of having two row decoders to separately decode the read and write address and generate the *rd* and *wr* signals for the bitcells, in addition to two separate column decoders for the read and write data.

4.3.2 Flip-Flop Based Implementation

The structure of the flip-flop based register file (FF-RF) is a little different than the SRAM one. As shown in Figure 4.2, it consists of three separate blocks: the write logic circuit, the flip-flop array, and the read logic circuit.

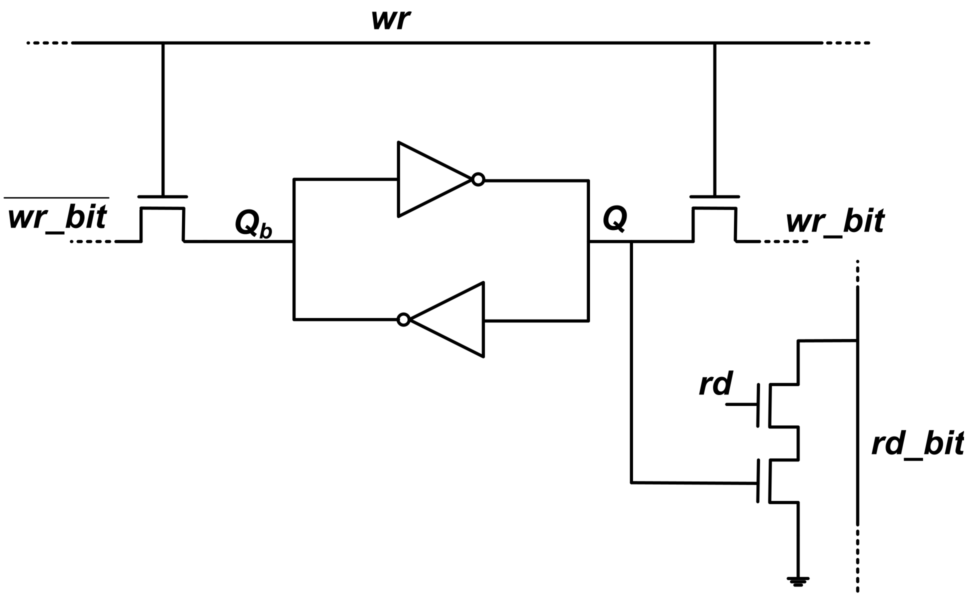


Figure 4.1: A typical structure of an 8T SRAM bitcell for a 1R1W register file.

4.3.2.1 Write Logic

The write logic circuit is the block responsible for generating the enable signals to enable one register of the flip-flop array to allow a write operation to this register at the next clock edge. It is usually implemented as a decoding circuit whose inputs are the write enable (*wr_en*) and the write address (*wr_address*) and whose output are W enable signals. If the write

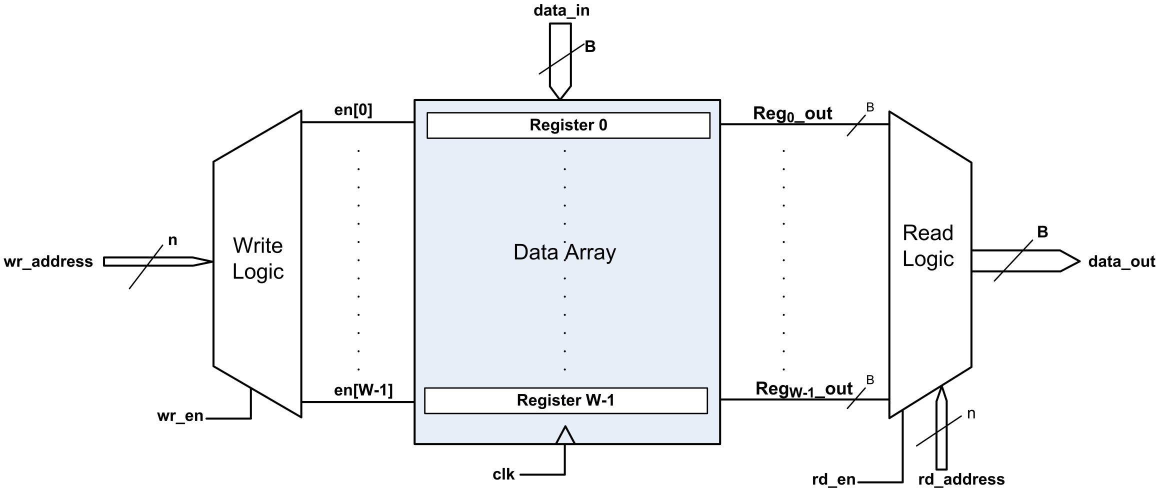


Figure 4.2: A block diagram for a standard cell based register file.

operation is enabled (by asserting the *wr_en* signal), only one of the output enable signals will be active depending on the input write address, while all the other enable signals will be inactive. In the case when no write operation is needed, all the enable signals remain inactive.

4.3.2.2 Data Array

The data array is the actual data storage elements. In the case of a flip-flop based register file, this data array is built using flip-flops. The array consists of $W * B$ flip-flops, where each B flip-flops share the same enable signal from the write logic block to form a register. To allow the write operation, two alternative options can be used.

The first one is using flip-flops with input enable signal, where the flip-flop can store a new input data at the next edge of the clock only if its enable signal is activated. Otherwise, the stored data remain unchanged after the clock edge. The implementation of such flip-flop is usually done by adding a 2-to-1 multiplexer to the input of a simple flip-flop, where the inputs of the multiplexer are the outside data and a feedback from the current output. If this option is picked to build the data array, then the same clock signal will be routed to the entire flip-flop array.

The second option is using simple flip-flops without enable, while driving their clock signals through clock gating cells (CGC), where each row of the array shares the same clock gating cell. In this case, the enable signals and the clock signal are only routed to these clock gating cells, and the outputs of these cells are used to clock their corresponding flip-flops as shown in Figure 4.3. When a register is selected for read, its corresponding clock gating cell is enabled. Hence, the pulse of the next clock cycle is passed to the selected row of flip-flops, and the input write data will be sampled and stored. At the same time, the clocks of the remaining rows remains inactive (usually tied to logic '0').

Comparing the two approaches, our synthesis results show that the second approach saves

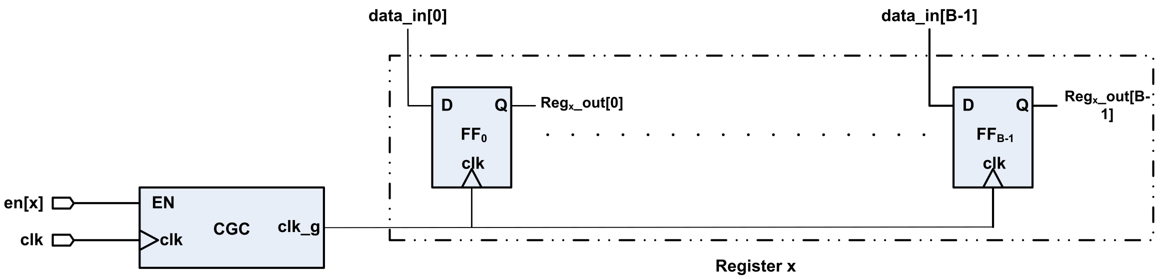


Figure 4.3: A block diagram for one of the flip-flop registers in the data array showing the usage of the clock gating cell (CGC) for the enabling of the flip-flops.

power and area. Instead of using bigger size enabled flip-flop (more than 35% bigger in area when compared to the regular flip-flop in the technology we are using), only one clock gating cell is added for every row of flip-flops. Since the area of the added clock gating cell is only slightly bigger than the area of one flip-flop, using the clock gating cell approach will save a significant area if the number of flip-flops per row is bigger than 2 (i.e. $B \geq 3$). In addition, with the clock gating approach, the main clock signal is only routed to the clock gating cells instead of being routed to the entire flip-flop array. Hence, a significant amount of clock switching power is saved when the clock gating approach is used.

4.3.2.3 Read Logic

The read logic circuit is the block responsible for extracting the data stored in one of the array registers and routing it to the output port. A specific register is selected based on the input read address ($rd\_address$) and the input read enable (rd_en)[2]. The read operation of most of the register files is asynchronous operation (i.e. the output data changes whenever the read address changes without waiting till the edge of the next clock cycle). However, some architectures insert a latching stage (flip-flops or latches) for the read address input that samples the input read address at the active clock edge at the start of the read cycle and the

[2] Note: some standard cell based register file architectures don't use an enable for the read operation and generate an output data whenever the read address changes. In our designs, we tried both approaches. For the results we are presenting in this chapter, we are using the read enable signal to have a fair comparison to the SRAM based register file that requires having this enable

sampled read address is used inside the register file to select the needed read resister. This will ensure that the read address remains stable during the read operation and eliminates any glitches or activity that could happen if all the read address bits don't arrive simultaneously. Since the read address are usually generated by some other logic inside the processing core, this can also relax the timing constraints on this logic (since not all the address bits are required to change exactlt at the same time) and allow some timing margins for the different read address bits.

The read logic block is usually implemented as B parallel W-to-1 multiplexers. This is usually preferred over using a tri-state buffers at the output of each register that can be enabled by a decoding circuit. Since it is difficult to buffer tri-state buses, significant timing degradation can happen to selected data if the buses are routed for long distance [44].

4.3.3 Latch Based Implementation

The general structure of the latch-based 1R1W register is close to that of the flip-flop described previously with the exception of using latches instead of flip-flops. Since latches are smaller in area than flip-flops, the latch based register file is expected to be smaller in area. In addition, since latches have lower leakage and internal power, their register file based one is also expecting to consume less power. However, due to the wide transparency window of the latch (half of the clock cycle for a 50% duty cycle clock signal), it has more timing restrictions when compared to a flip-flop counterpart. If the same register is being chosen for read and write operation simultaneously, the latch transparency window can make the latch in a closed loop through any external logic that can be used to feed the output of the latch back to its input. This issue can be solved by adding a restriction on the write and read input data or by adding either flip-flops or latches (transparent on the other half of the clock cycle) into the paths that has such feedback [43]. However, this will add some extra area and power overhead, in addition to affecting the timing of these external logic.

4.3.4 Pulsed Latch Based Implementation

Pulse latch (PL) based register files can provide the best compromise between the flip-flop and the latch based register files, gaining the advantages of both. As discussed in Chapter 2, since PL are latches driven by short pulsers, their timing model are very close to flip-flops. In addition, as in latch based register files, they are expected to consume less area and power compared to flip-flop based register files. The only additional overhead over the latch-based register file is the addition of the pulser. However, this overhead is very small since the same pulser will be shared between all the B latches in a register. In addition, since the pulser circuit controls the transparency of the latches through its generated pulse, it can eliminate the need the clock gating cell as shown in Figure 4.4, where the clock gating cell is replaced by the pulser circuit. In order to do this, the pulser circuit needs to be modified to allow the control of the pulse generation through the write logic.

Our proposed design of the modified pulser circuit is shown in Figure 4.5. The NOR gate is used as "gated-inverter" for the clock signal. If the enable signal (*en*) is not active (logic '0' in the diagram), the output of the gate (*clkb_g*) will remain '0' regardless of the value of the input clock signal. When the *en* signal becomes active (i.e. logic '1'), the output *clkb_g* will be the inverted version of the input clock. In this case, the pulser circuit will be similar to the basic pulser circuit discussed in Chapter 2. When the *en* signal becomes active before the rising edge of the clock, the output *clkb_g* is asserted. The delay buffers are used to

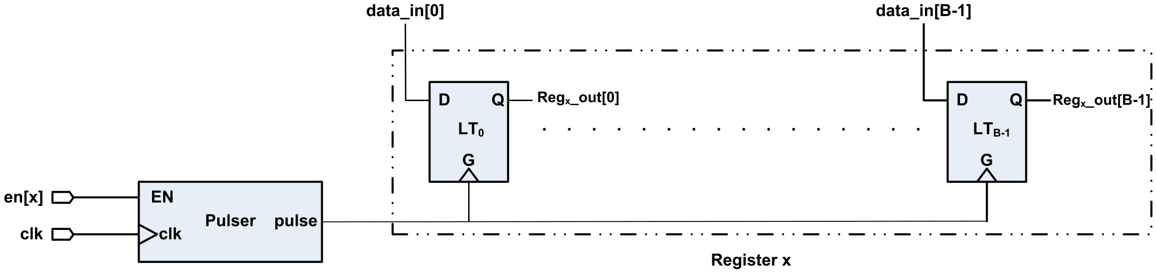


Figure 4.4: A block diagram for one of the pulsed latch registers in the data array showing the replacement of the clock gating cell by the pulser circuit.

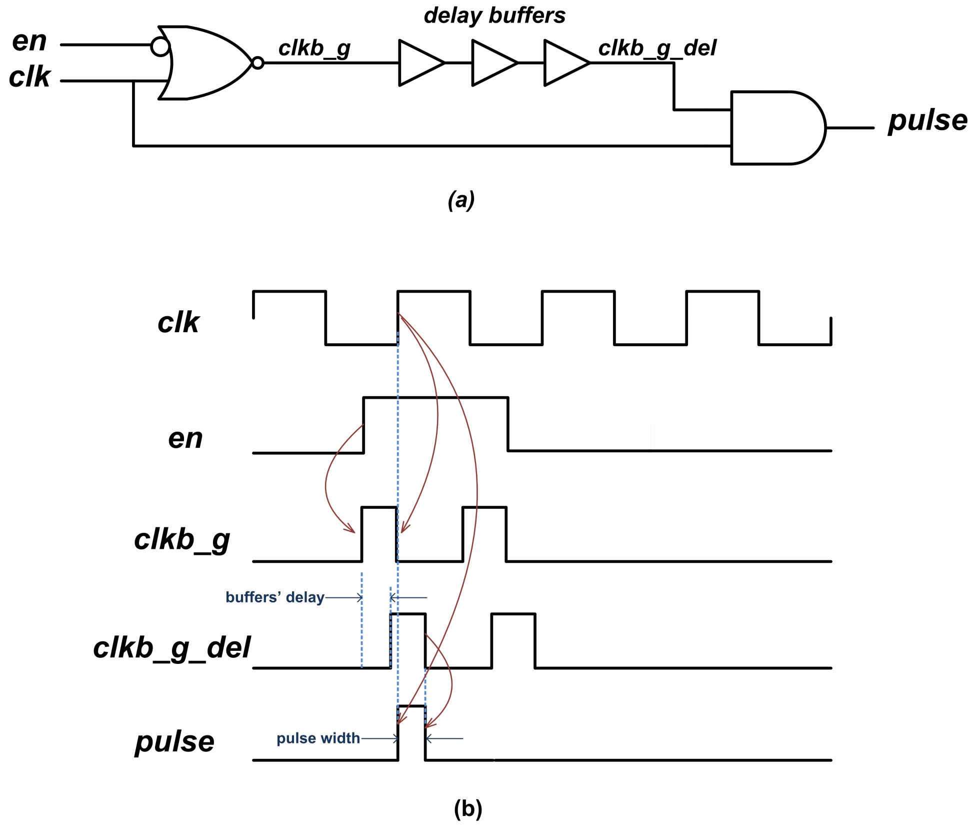


Figure 4.5: The pulser used in the pulsed latch register file (a) The proposed pulser circuit, (b) Timing diagram for the pulser operation

generate a delayed version of this inverted clock signal (*clkb_g_del*). When the clock signal is asserted, both the inputs of the AND gate will become logic '1'. Hence, the output *pulse* is generated. At the same time, the asserted clock signal will pull down the *clkb_g* signal. However, the output *pulse* will remain high till the *clkb_g_del* signal is pulled down to logic '0' after the buffers' delay. Hence, the width of the generated pulse can be controlled by controlling the delay buffers.

In addition to the above proposed design, another alternative can be used. Due to the regular structure of the data array (repeated registers of the same structure), and the lower number of pulsers compared to the number of bits (only one pulser for each register), the two clock paths of the pulser (the clock and its delayed version) can be physically splitted to two different input clocks (i.e. the pulser will have two input clocks instead of one: the clock and the delayed clock). Hence, the delay buffers can be shared between several pulsers

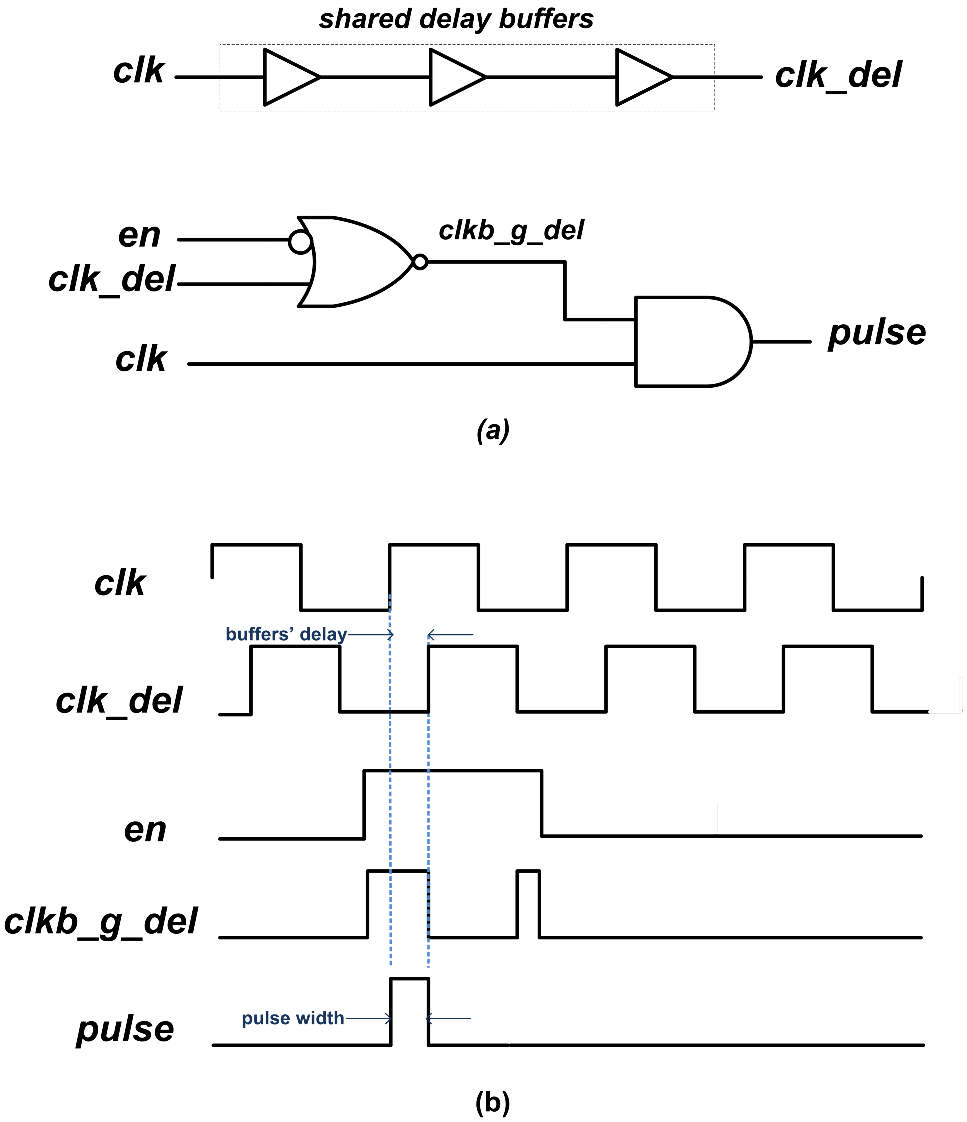


Figure 4.6: The modified pulser design with the delay buffers shared among different pulsers (a) The modified pulser circuit, (b) Timing diagram for the modified pulser operation

instead of being included in each pulser as shown in Figure 4.6. This will results in extra area and power savings. The drawback of this approach is the added complications to the place and route and the clock tree synthesis (CTS) operations. Instead of routing one clock signal to each pulser, two different clock signals will be routed with certain value of desired skew to control the pulse width. In addition, the common delay buffers need to be carefully selected and verified under different corners since any variations or failure in their operation will affect the entire register file operation, not only a single register.

The remaining of the register file blocks will remain similar to the flip-flop based ones, with the write logic generating the enable signals for the pulsers based on the write enable and write address, while the read logic is used to select the output data for reading based on the read enable and read address.

Based on the above discussion, it seems that PL based register files are keeping all the advantages of the latch based register files, and at the same time, overcoming most of its drawback. Therefore, from now on, we will drop the latch based register files form our discussions and focus on the PL based register files in addition to the SRAM and the flip-flop based ones.

4.3.5 Comparison between Different 1R1W Register File Implementations

For 1R1W register file, three different implementations were discussed: an SRAM based implementation, a flip-flop based implementation, and a pulsed latch based implementation. We have implemented the three discussed register file options using UMC 28nm technology. For the SRAM implementation, we used the two port high density register file compiler provided by Synopsys for this technology. The bitcells use high threshold voltage transistors, while the peripheral circuits use standard threshold voltage transistors.

The flip-flop and pulsed latch implementations are implemented using the standard cells libraries. To have fair comparison with SRAM, we have chosen the high threshold voltage cells for the storing elements (flip-flops and latches), while using standard threshold voltage cells for the combinational cells. For the flip-flop design, clock gating was enabled during the synthesis. All the implementations were placed and routed and the clock tree synthesis (CTS) was optimized for low power.

The SRAM bitcells are designed with minimum size to save area, while standard cells are having different design constraints (as an example, the cell height must be a multiple of

Table 4.1: The area of 1R1W SRAM bitcell, flip-flop and latch for the UMC 28nm process.

Bitcell	SRAM	Flip-Flop	Latch
Area (μm^2)	0.4	1.8	1.2

certain routing grids defined by the design tools, where most of the cells should have the same height to be able to get aligned during the place and route). Hence, storage elements formed by standard cell are expected to be much larger than the SRAM bitcells. This can be shown in Table 4.1 for UMC 28nm process. However, SRAM peripheral circuits have significant area overhead. For small size SRAM register files, their area overhead can be significantly large. The border at which the area of the SRAM based register file will be larger than the standard cell based register files will depend on the number of words and the number of bits per word, and may be as large as 1 kbit [43]. In our study, we have chosen a common 32-bits per word registers and designed few register files with different number of register up to 1024 register. Figure 4.7 shows the post place and route area for the three discussed register file implementations for different number of words. As shown, the area of the pulsed latch based register files are always smaller than their flip-flop counterparts by more than 20%. In comparison with the SRAM based register file, the rate of area growth for both flip-flop and pulsed latch register files is much higher than the SRAM ones. For W=32, the PL based register file is 17% higher than its SRAM counterpart. This number jumps to more than 2.3X for W=128 and 4.2X for W=1024.

However, the power consumption numbers look a little different. Figure 4.8 is the VCD-based post place and route power numbers for the same register files, where random data are written to random registers, and data are read from randomly chosen registers, for around 200 cycles at a clock frequency of 500 MHz. As shown, the power consumption of the pulsed latch register file is always lower than its flip-flop counterpart by around 40-50%. In comparison to SRAM, flip-flop based register files consumes less power for register file sizes of up to 64 words, while pulsed latch based register files can save power for sizes beyond 128 words. Indeed, the pulsed latch based register files save significant amount of power for such sizes. For W=32, pulsed latch register file is more than 70% lower in power than its SRAM counterpart. This number becomes 51% for W=64 and 15% for W=128.

As a summary, standard cell based 1R1W register files can be larger in area when compared to SRAM ones, but they can consume much less power for small size register files, specially, the pulsed latch based ones. For some applications, a small area overhead can be acceptable for the benefit of having lower energy per memory access, since register files are accessed very frequently. In addition, SRAM has much higher restriction on the minimum supply voltage it can work at [58]. Since SRAM bitcells use high threshold transistors to reduce leakage, they become very slow at lower voltage. Also, since these transistors are nearly of minimum sizes, they will also suffer from high variability at lower voltages. Hence, for designs that needs to operate at a lower supply voltage, standard cell based register files may be a better fit. In addition, since standard cell based register files are described in hardware description languages, this will give more control and flexibility to design teams. Also, this will ease the portability of the design to other technologies.

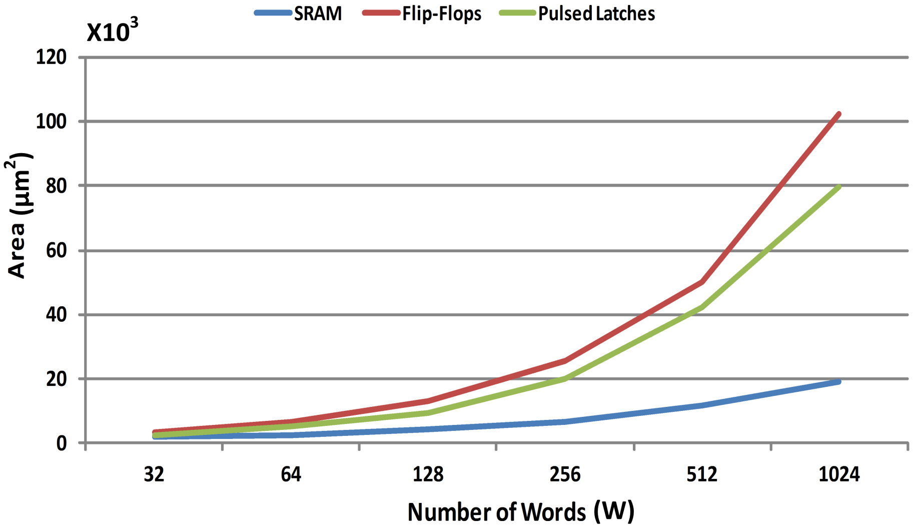


Figure 4.7: Area comparisons of the three versions of 32-bits/word 1R1W register files for different number of words

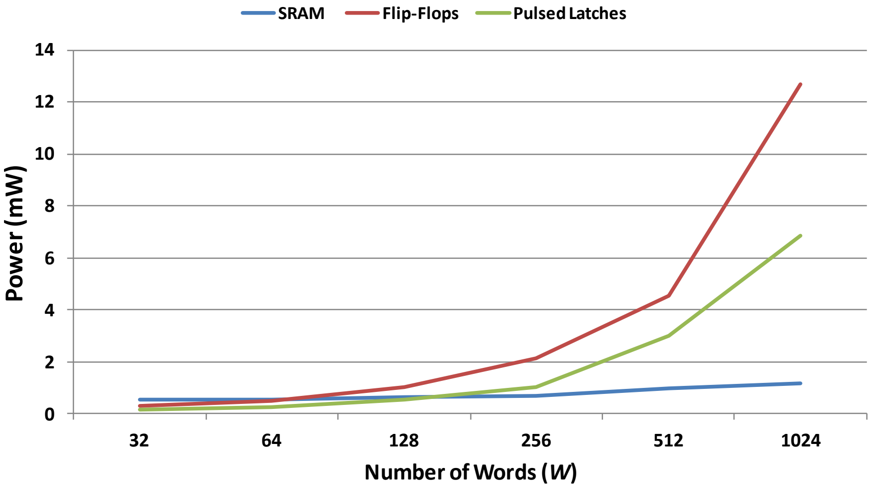


Figure 4.8: Power consumption of the three versions of 32-bits/word 1R1W register files for different number of words

4.4 Design of Multiport Register File

Multiport register files are critical components for superscalar microprocessors [61]. Since they allow simultaneous read and write operations, they enable the execution of multiple out-of-order and simultaneous multi-threading operations. For example, the register file for the Itanium microprocessor can simultaneously handle up to 12 read operations and 10 write operations [28]. Figure 4.9 shows a block diagram of a multiport register file with m read ports and j write ports. The main problem with multiport register files is that as the number of read or write ports increase, both the area and the power consumption increase significantly. In this section, we will discuss the different traditional implementations of multiport register files.

4.4.1 SRAM Based Implementation

Multiport SRAM designs have been widely used in different architectures. Multi-core processors usually require multiport data caches to handle multiple simultaneous loads and

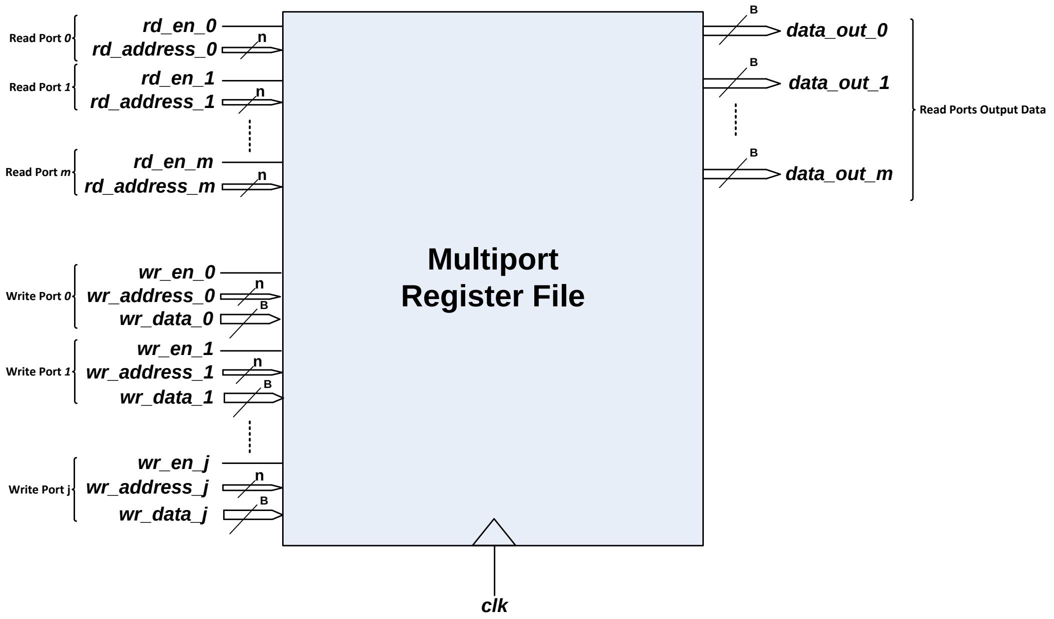


Figure 4.9: A block diagram for a multiport register file with m read ports and j write ports

stores [74], in addition to multiport register files to enable concurrent execution of multiples instructions in a single cycle [69, 51, 65, 30, 72]. A straight forward implementation of multiport SRAMs is by expanding the 8T SRAM bitcell shown in Figure 4.1 by adding more read and write access transistors for each added port as shown in Figure 4.10. Each additional read port can be provided at the cost of a read word line selector, a read bitline, and two read transistors. Each additional write port can be provided at the cost of a write word line selector, a write bitline and its compliment, and two access transistors. In addition, the bitcell crossed coupled inverters will need to be larger in size to handle the extra bitlines' load when adding more read or write access transistors. Also, with all the excess word lines and bitlines wiring, the memory size can grow quadratically with the total number of ports [24]. To save some wiring, the complementary write bitline routing across the memory array can be eliminated, and an inverter can be added in every cell to generate the complementary write bitline locally for every cell [67]. However, this added inverter will cause a significant increase in every bitcell area.

To reduce the area of multiport register files significantly, the number of physical access

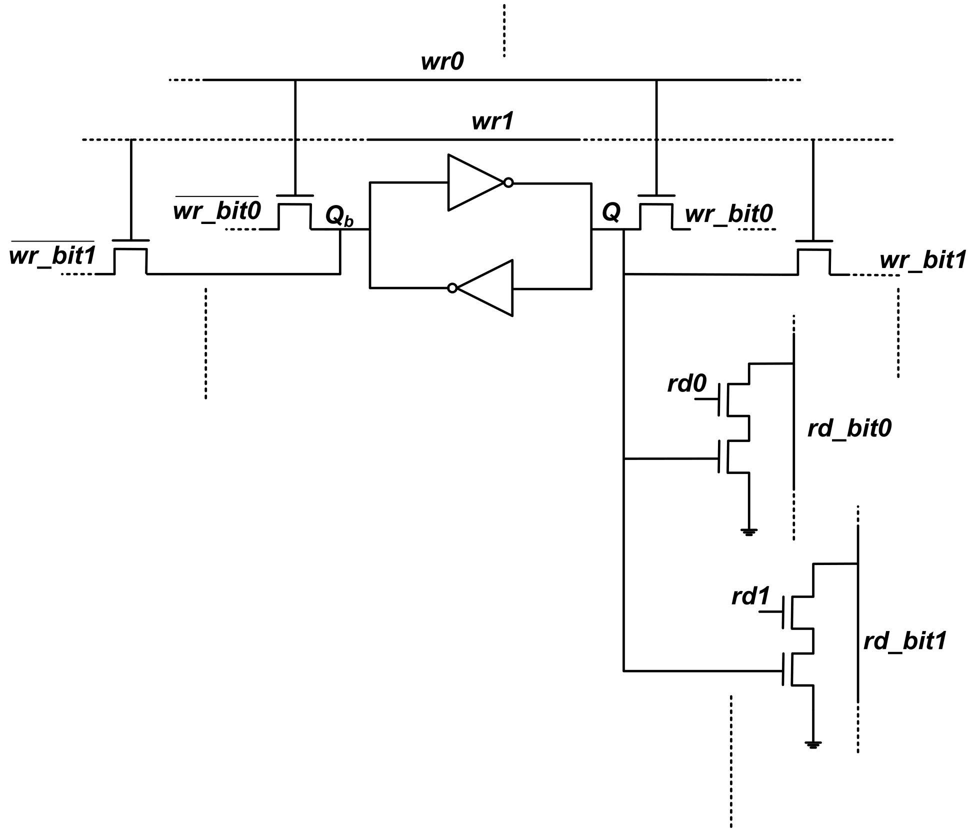


Figure 4.10: A straight forward expansion of the 8T SRAM bitcell to add more read and write ports

ports in a memory cell should be reduced. In addition to array savings, this will also save power and improve latency. Two techniques exist for reducing the register file area: time multiplexing and using multiple banks. Each technique can be applied individual or both of them can be applied at the same time.

Time multiplexing is also called double pumping. In this technique, a port can be accesses multiple times (usually twice) in the same clock cycle. As an example, a single write port can be accessed twice, one time during the first half of the clock cycle and the other during the second half [66]. This will yield to two effective write operations. Thus, the number of physical write ports will be halved. The same can be applied to the read ports, where one register can be read during the first half of the clock cycle, while the other one can be read during the second half. Another approach is using a clock frequency for the register file that is higher than the processor core. Hence, in one clock cycle of the main processor, the register file ports toggle multiple times.

In multiple banks approach (also called array duplication), the memory array is duplicated into two or more banks (copies), each with some of the read ports (half the read ports if the array is twice duplicated), while they are sharing the same write ports. Write operation always write the same data to all banks, while read can take place from any bank (depending on the chosen read port). Since doubling the number of read port per array can increase the array size by 3-4X, having two copies of the same array with half the number of read ports can provide a significant reduction in area.

Both techniques can be applied together. As an example, [24] proposes building a register file with four read and two write ports (4R2W) using two register files, each with two read and one write ports (2R1W). Array duplication is applied, where the two 2R1W sub-array share the same write port, while the two read ports from each register file form the four read ports. In addition, double pumping for write operation is applied, where the single common write port is accessed twice in the same clock cycle to allow two different write operation in one clock cycle, yielding to two effective write ports. The resulting 4R2W register file achieved 2X reduction in area compared to a traditional 4R2W register file.

4.4.2 Standard Cell Based Implementation

Standard cell based register files (flip-flop and pulsed latches) can also be extended to have multiple read and write ports. The data array structure (which is the main difference between flip-flops based and pulsed latches based register files) will not change, only the flip-flops or latches may be replaced by a higher driving cells of the same type to support the increased cell loads by the added ports. The main changes will be in the read and write logic.

The straight forward approach to add read ports is by adding additional multiplexer in parallel to the one shown in Figure 4.2. If large number of read port is needed, a large area overhead will exist due to the wiring of the all the registers' outputs to the inputs of the read ports' multiplexers, especially with large number of registers and/or large number of bits per register.

The straight forward approach for adding write ports may be a little more complicated. In addition to adding more decoding circuits to generate the enable signals for each write port, additional logic is needed to merge the different enable signals to each register to a single enable signal that can enable the clock gating cell for the flip-flop based register file or enable the pulser for the pulsed latch based one. Also, additional logic is needed for the selection of a single input data from the provided write data on each write port to go to the data input of the selected register.

Similar to SRAM, adding read or write ports will add a significant area and power overhead, in addition to performance degradation. Also, similar to SRAM, array duplication and double pumping can be used to reduce the number of physical ports. However, there will still be some significant area and power overheads associated with both approaches. In addition, there will a limit on how many ports can be obtained through pumping or duplication. The two halves of the clock cycle can be used to obtain two effective write ports through double pumping. However, obtaining more than two write ports without significant overhead may be more complicated. Similarly, using array duplication may help in providing more read ports. However, duplicating the data array adds a significant area overhead. In addition, the duplicated arrays increase the power consumption significantly, including both leakage power (due to having more bit storage elements) and dynamic power (due to writing the same data simultaneously in more than one data array).

4.5 Proposed Pulsed Latch Based Register File with Virtual Ports

As discussed in section 4.3, pulsed latch based register files can have some advantages in the implementation of register files of small to medium sizes. They can save a significant amount of power consumption, be smaller in area than flip-flop based register files, in addition to their lower latency. However, similar to other register file architectures, adding read or write ports can add significant overhead in area and power, in addition to causing performance degradation. Since pulsed latch operation depends on using pulser circuits to generate short pulses from the clock signals, these pulsers can be used to give access to the same register multiple times within the same clock cycle. In addition, some other pulsers can be arranged in groups (thus, we call them *Pulser Groups (PGs)*) to perform some necessary operations during the read or write process. As shown in Figure 4.11, this will result in having a register file with many *Virtual Ports* that are generated from much smaller number of actual physical ports using some control logic blocks for both the read and write ports. In these control logic blocks, different pulser groups are used to generate some control signals, in addition to holding some intermediate data such as internal read and write addresses and data. This is described in details in the next two subsections.

4.5.1 Adding Virtual Read Ports

As described in 4.3.2.3, the read logic that is used to generate the data output for each read port is usually implemented as parallel W-to-1 multiplexers, where the selector of these multiplexers is the read address and their outputs are the required output data. To implement a traditional register file with multiple physical read ports, these multiplexers should be repeated for each additional read port. However, in our proposed architecture, a

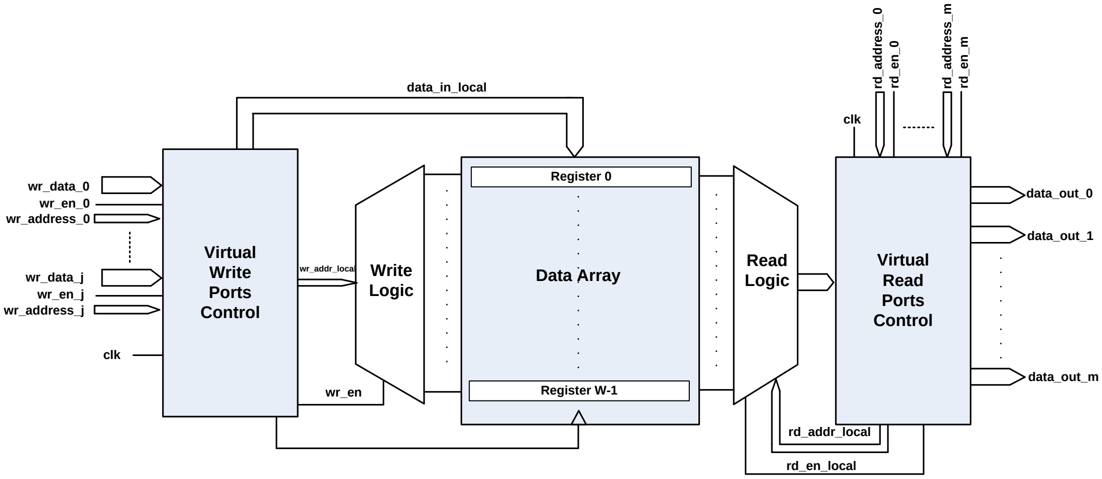


Figure 4.11: The structure of the pulsed latch based register file with virtual ports.

single multiplexer will be used multiple times in a single clock cycle to generate the needed data for each of the virtual ports. To achieve that, some logic are needed to select one of the input read addresses for one of the enabled read ports. In addition, some other logic are needed to hold the routed read data for each port. Figure 4.12 shows a detailed block diagram for our proposed read logic. It consists of five main parts, in addition to the main data multiplexer that is similar to the one in the 1R1W register file.

4.5.1.1 Internal Clocks Generator

This block is responsible for generating clock signals with different phases from the main input clock signal (*clk*). It consists of a clock gating cell that gates the *clk* signal if no read operation is needed (all read enable signals (*rd_en*)) are not asserted. If at least one of the *rd_en* is active, the *clk* signal is passed to *clk_g*, which will be, internally, the main clock signal for the read logic. The *clk_g* is used to generate other clock signals (*clk_g_del*) with different phases through some delay buffers. These *clk_g_del* signals are used by the other four blocks to control the read operation for every read port.

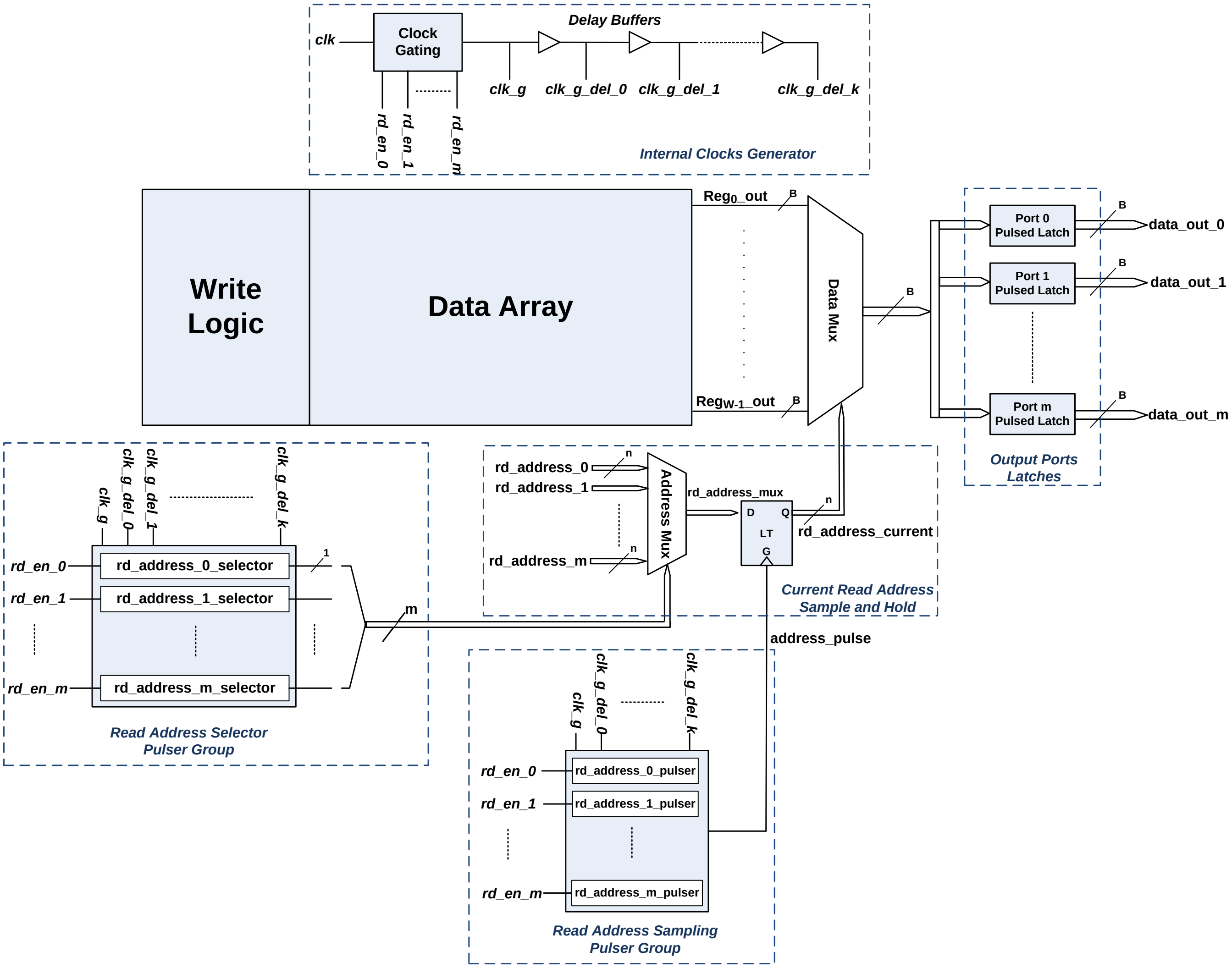


Figure 4.12: The structure of the proposed read logic that provides multiple virtual read ports.

4.5.1.2 Read Address Selector Pulser Group

This block is used to generate the control signals that are used by the *Address Mux* to select which of the input read addresses will be read next. It consists of *m* sub-blocks, one for each read port. Each sub-block is a pulser circuit of structure close to the pulser shown in Figure 4.6(a). Each pulser has an input enable signal which is the *rd_en* of its corresponding port and two clock signals from the signals generated by the internal clocks generator. Its output is a pulse that is used as one of *Address Mux* selectors.

4.5.1.3 Read Address Sampling Pulser Group

This block is used to generate the *address_pulse* signal. This signal is used to enable the read address latches of the *Current Read Address Sample and Hold* block. From functionality perspective, it can be considered as a large "pulser" circuit that can generate few pulses within one clock cycle. Similar to the read address selector pulser group, it contains m pulser circuits, each with an input enable signal and two generated clock signals. The outputs of these m pulsers are merged together to generate an output signal that is used as the enable signal for the read address latches.

4.5.1.4 Current Read Address Sample and Hold

This block is responsible for selecting one of the input read addresses and then holding it for some time that should be enough for the data multiplexer to select and route the needed register data to the output ports' latches. It is divided into two sub-blocks: the *Address Mux* and the *Address Latch*.

The *Address Mux* is a group of multiplexers that select one of the input read addresses according to the select signals generated by the read address selector pulser group. This group of multiplexers is arranged in an certain order to give the higher priority to the read address of port 0, then the read address of port 1, etc. In addition, if all the selectors are not active, the default output of the *Address Mux* will be the read address of port 0. The structure of this *Address Mux* is shown in Figure 4.13. If the selector of read port 0 is active, then port 0 address is selected. If not, then the multiplexer checks on the selector of read port 1, etc. If non of the selectors is active, then port 0 address is selected by default. The default address is selected to be that of port 0 as to reduce the setup time for read enable signals, since port 0 is the first virtual port in the clock cycle to be considered if enabled.

The *Address Latch* is responsible for capturing and holding the address of the resister that is currently being read. The current read address should be hold stable for enough time for the *Data Mux* to route the selected register data to its output and then for one of output ports' latches to store the routed data. This latch is enabled by the *address_pulse* signal. The *Address Latch* together with the *Read Address Sampling Pulser Group* represent a pulsed latch register for the current read address.

4.5.1.5 Output Ports Latches

Since only one data multiplexer is use for readout for different read ports during the same clock cycle, its output should be stored at the right time within the clock cycle to represent one of actual read port outputs for the register file. In addition, each read output should remain stable until a new read operation is performed on the same port. This is done using the *Output Ports Latches* block. The block consists of few pulsed latches of the same number of the read ports. Each pulsed latch is enabled by one of the read enable signals and triggered by one of the clock signals generated by the internal clocks generator. The triggering clock should be selected such that the correct data to be stored is ready at the output of the *Data Mux*.

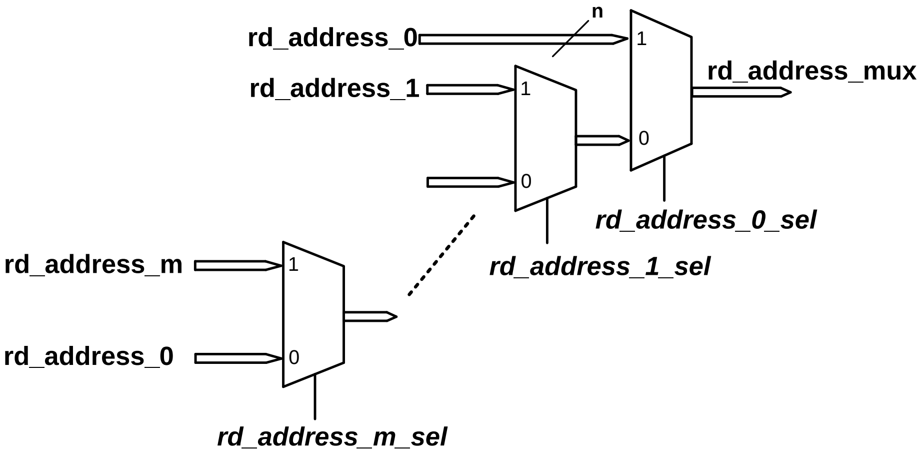


Figure 4.13: The structure of the *Address Mux* that give priority to the read ports in the required order.

4.5.2 Adding Virtual Write Ports

As described in 4.3.2.1, the write logic block is responsible for enabling the pulser of one of the registers in the data array (for a pulsed latch based register file). This is usually implemented as a decoder who is enabled by the *wr_en* signal and one of its outputs is activated based on the *wr_address*. Hence, to be able to use the same single decoder to allow more than one write operation, some control logic need to be added to select one of the write addresses and enables (one at a time), and pass it to the decoder circuit. In addition, since the entire data array has the same input data, the same control logic should select the right input data to be written and pass it to the data array at the same time it passes the current write address and current enable signal. We call this control logic the *Port Selector*. It consists of three multiplexers, one for each of the write address, write enable and write data. These multiplexers are controlled by a *Write Port Selector* block, which generates the multiplexer select signal based on the clock signals generated by the internal clocks generator.

The proposed write logic design is shown in Figure 4.14. In addition to the *Port Selector* block, another control block is needed to generate the write clock signal for the data array. Since the pulsers of the data array's registers will generate the enable pulses for the latches based on the triggering clock signal, performing multiple write operations in the same clock cycle requires a clock signal that is triggered multiple times during the same cycle. This is accomplished by the *Data Array Clock Generator*, that triggers the clock signal of the data array (*clkw_data_array*) every time a write operation is needed. The triggering event should take place after the write decoder has already enabled the needed register according to the current write address and also the current write data is already routed to the data array.

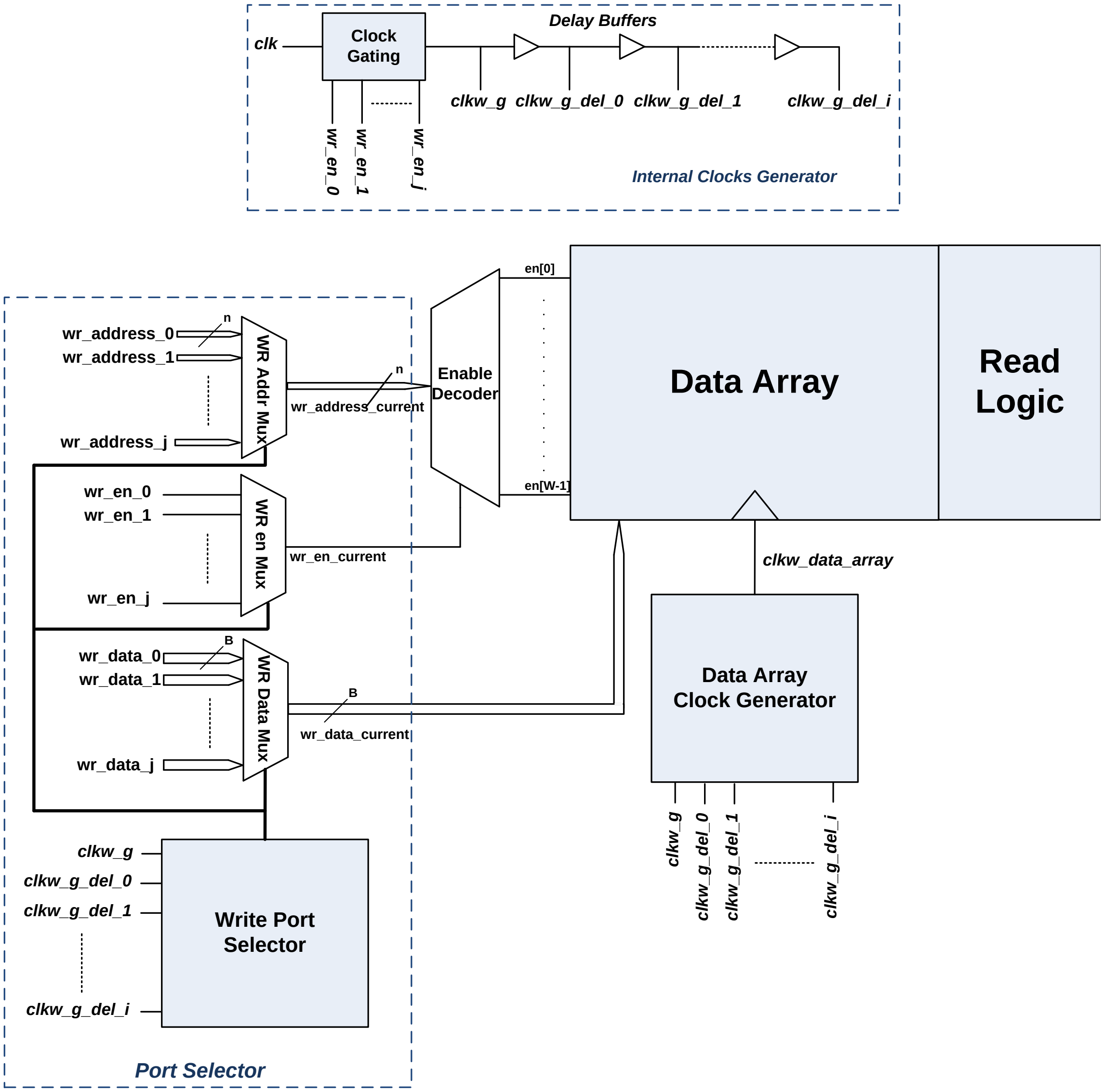


Figure 4.14: The structure of the proposed write logic that provides multiple virtual write ports.

4.6 Results

4.6.1 Register Files Circuit Designs

To be able to compare the different approaches of the implementation of multiport register files, we implemented 32-words, each with 32-bits register file with few multiport configurations (combinations of two or four read ports with one or two write ports) using all the discussed implementations. Four different multiport configuration where compared in addition to the 1R1W one: a register file with two read and one write ports (2R1W), a register file with two read and two write ports (2R2W), a register file with four read and one write ports (4R1W), and a register file with four read and two write ports (4R2W). However, the same design approaches can be used for other numbers of read or write ports.

For the SRAM implementations, we used the UMC 28nm register file compiler to generate a 1R1W instance. Then, we used CACTI [60](which is a modeling tool for dynamic and leakage power, access time, and area of caches and other types of memories) to scale the 1R1W numbers to get the needed area numbers for different physical multiport configurations. To scale the 1R1W power numbers, the average energies per read and write ports of the 1R1W were calculated using the power analysis tool and then scaled using the CACTI numbers. Then, to get the average power consumption for a selected register file activity, the scaled numbers were added based on the activity extracted from the VCD file. In addition, to compare our proposed implementations with that presented in the literature for multiport SRAM register files, we applied array duplication and double pumping to the 1R1W memory array. The memory array was duplicated two times, with a common write port, to get the 2R1W register file. Similarly, the memory array was duplicated four times, with a common write port, to get the 4R1W register files. To get two write ports, double pumping was used, where additional logic was added to perform the two write operations in the same clock cycle. The first write operation takes place within the first half of the clock cycle, while the second write operation is executed within the second half of the clock cycle. This double pumping was used with the array duplication discussed previously to get the 2R2W and the 4R2W register files.

For the standard cell based register files, different designs were implemented in RTL using the UMC 28nm libraries. All the designs were synthesized, placed and routed including the clock tree synthesis (CTS) operation. For the standard cell based register files with actual physical ports, the design was extended from the 1R1W design showed in Figure 4.2 by adding one more write decoder for each additional write port, in additional to some logic to route the enable signals and the data inputs to the data array. To add read ports, additional multiplexers were added in parallel. For each multiport configuration, two standard cell register files were implemented: one with the flip-flop based data array and the other with the pulsed latch based data array.

For the proposed pulsed latch based register file with virtual ports, the design approaches discussed in Section 4.5 were used. Figure 4.15 shows the detailed implementation of the read logic to provide two read virtual ports. For two read ports, two delayed versions of the gated clock are needed: *clk_g_del_*0 and *clk_g_del_*1. In addition, the *Read Address Selector Pulser Group* can be eliminated and the *rd_en_*1 and *clk_g_del_*0 signals can be used directly to select which read address should be read at a certain time, where the default address is that of port 0, unless the *clk_g_del_*0 and *rd_en_*1 are both '1'. The selected address is sampled by the *rd_address_current* latches whenever a pulse signal is generated by the *Read Address Sampling Pulser Group*. This pulser group has two pulser circuits whose outputs are ORed together to generate the needed *address_pulse* signal. Each of the two pulser are enabled by the read enable signals of the two read ports. The triggering clock for each pulser is chosen as to make sure that the needed read address is already selected and routed to the latches' inputs as shown in the timing diagram of Figure 4.15(b). To hold the output data, two pulsed latches were used. Again, both are enabled by the read enable signals of each port and the triggering clocks were selected as to make sure that the pulsed latches are triggered after the needed read data is selected and routed to the *data_out* signal. The pulsed latch of port 0 was designed to work at the rising edge of the *clk_g_del_*1 clock signal, while that of port 1 was designed to work at the falling edge of the *clk_g_del_*0 clock signal.

To get four read ports, the read logic is modified as shown in Figure 4.16. A third delay buffer is added to generate an additional delayed clock signal. The *Read Address Selector Pulser Group* was designed to provide three control signals for the section of the read address, together with the *rd_addr_0_pulse* signal generated by the *Read Address Sampling Pulser Group* will form a 4-bit signal for the selector of the *Address Mux* to select one of the fours read addresses. As described in Section 4.5.1.4, the ports are checked in a priority order, such that if the *rd_addr_0_pulse* signal is active, *rd_address_0* is selected. Otherwise, if *rd_addr_1_sel* is active, then *rd_address_1* is selected. If not, then if *rd_addr_2_sel* is active, *rd_address_2* is selected. Finally, if *rd_addr_3_sel* is the only active signal, *rd_address_3* is selected. If non of the fours signals are active, then *rd_address_0* is selected by default. The

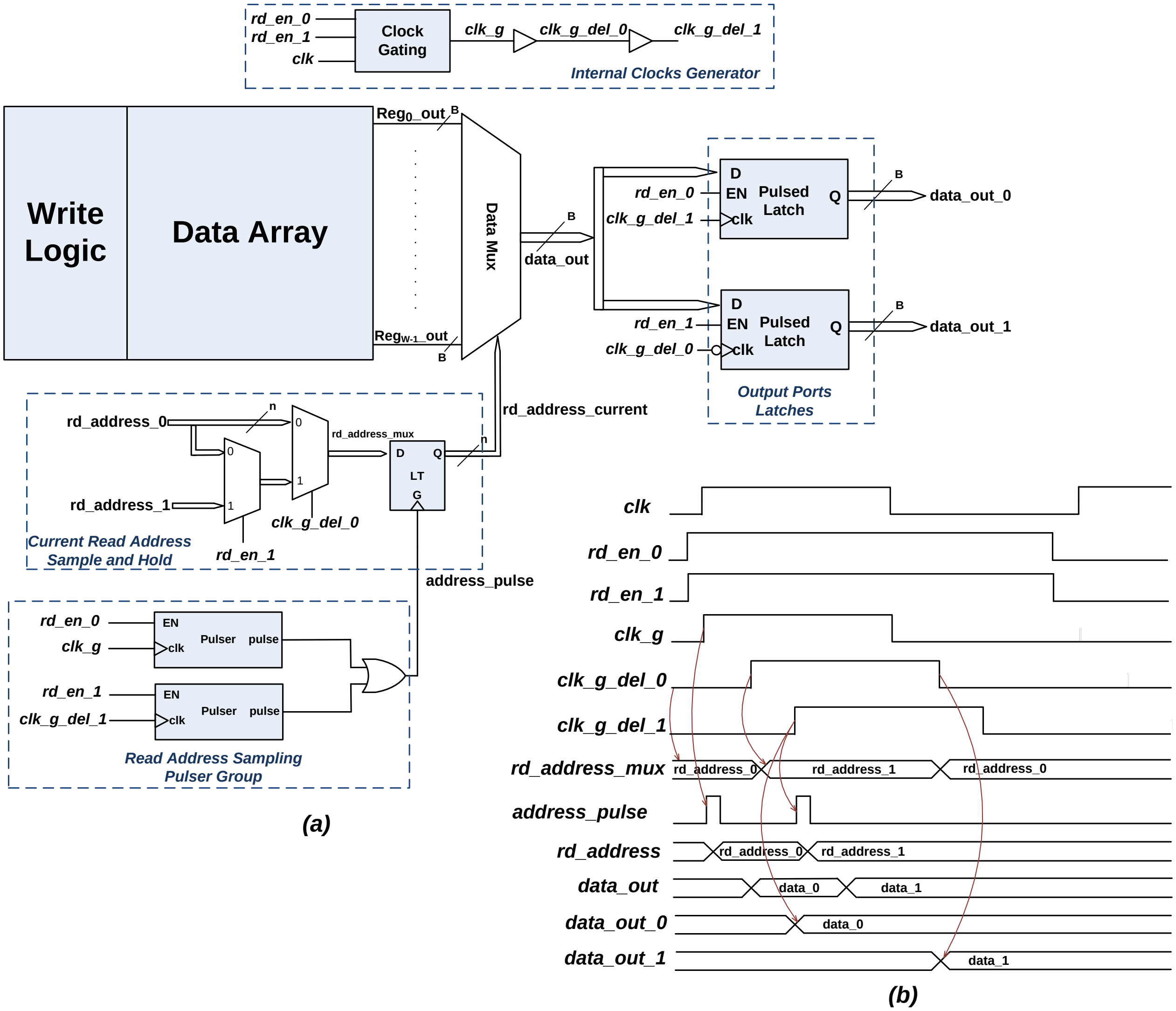


Figure 4.15: The proposed read logic structure for register files with two virtual read ports.

Read Address Sampling Pulser Group has four pulser circuits, whose outputs are ORed to generate the *address_pulse* signal. Each of the four pulsers is enabled by one of the read enable signals and two clock signals of different phases are chosen to generate the required pulse signal. One of the clock signal is used as a trigger to start the pulse, while the other one is used to end it. The two clock signals are chosen as to make sure that the needed read address is already selected and routed to the input of the address latches and kept stable for the entire pulse duration. To hold the output data for each port, four pulsed latches (two triggered at the rising edge and the other two triggered at the falling edge of their clock signal) are used. Similar to those of the two port design, each is enabled by one of the read enable signals and triggered by a suitable clock signal chosen such that the needed read data is ready at the *data_out*. The timing diagram in Figure 4.16(b) shows the entire operation of the read logic in the case of simultaneous four read operations. As shown, the read logic is designed such that all the read data should be stable at the output ports before the next clock cycle by some time that should be enough for any setup time needed by the next stage that is reading the output of the register file.

The design of the write logic to get two write ports is shown in Figure 4.17. To perform two write operations simultaneously, only one delayed clock signal is needed in addition to the gated clock signal. To select the port, three 2-to-1 multiplexers are needed for the write enable, write address, and write data. The selector of the three multiplexers is the gated clock signal *clkw_g*. Hence, each port is selected for a half clock cycle. If the selected port is enabled, then the enable decoder would enable one of the registers of the data array. The clock signal of the data array (*clkw_data_array*) is generated by the shown AND-OR structure. If the selected write port is enabled (i.e. *wr_en_current* is '1'), then a clock signal is generated to trigger the pulsed latch of the selected register. As shown in timing diagram of Figure 4.17(b), the *clkw_data_array* is triggered enough time after the selected write address and data are routed to the data array to guarantee correct write operation. As shown in the timing diagram, each half of the *clkw_g* signal is divided into two intervals. The first

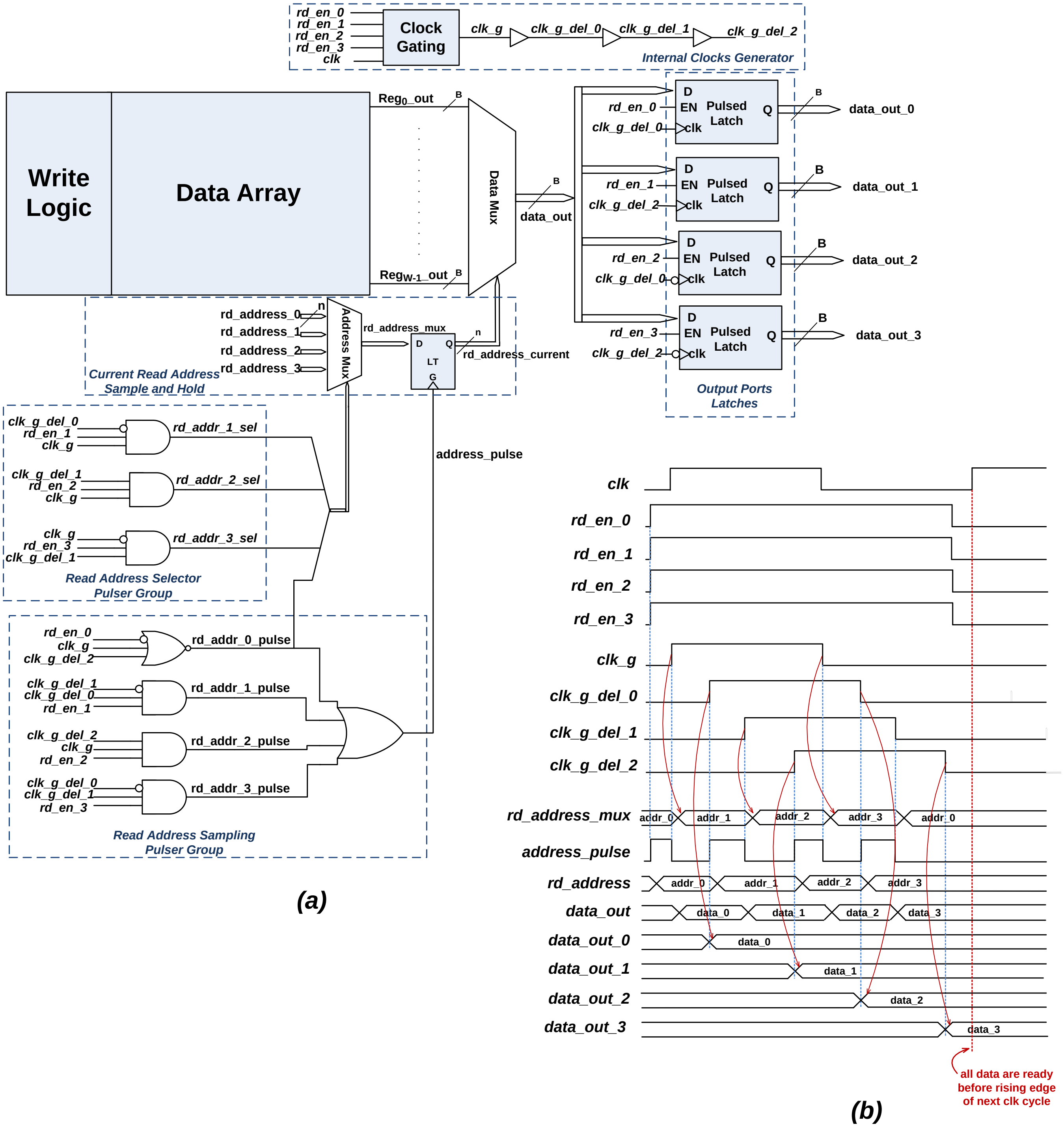


Figure 4.16: The proposed read logic structure for register files with four virtual read ports.

interval is used by the port selector to select the write port and then by the enable decoder to enable one of the register. The second interval is used to trigger the data array to capture and store the new write data.

All the designs were optimized for low area and power. The designs were synthesized using the Synopsys Design Compiler. The place and route and the clock tree synthesis were done using Synopsys IC Compiler. All the designs were tested and verified using different testbenches at a clock frequency of 500MHz and Synopsys VCS was used to do all the simulations. Synopsys Prime Time PX was used for power analysis using the VCD files generated from the post place and route simulations.

4.6.2 Register Files Area

Figure 4.18 shows the area numbers for the different register file implementations with different number of ports. As discussed in 4.3.1, the 1R1W SRAM register file has some advantages in area. However, when more read or write ports are added, the area starts to grow significantly. With each additional read port, the area increases by more than 2X, while the area increases by around 30-40% for each additional write port. This significant increase in area is due to the read and write bitlines wiring [24], and the SRAM bitcells up-sizing to ensure correct read and write operation with the increased bitline capacitance. If array duplication is applied, the register file area becomes lower when compared to the one with a single memory array. In addition, when double pumping is applied for write operations, just a small area overhead is added when compared to adding a second physical write port.

Similar to SRAM, the area of the standard cell based register file grows when read or write ports are added. However, the rate of area increase is much less when compared to that of SRAM. Except the 1R1W register file, the flip-flop based register file is always smaller than its SRAM counterpart. However, when array duplication is applied, the flip-flop based register file shows some advantages in area when the number of read ports exceeds two ports. For the pulsed latch register file with physical ports, there is a significant advantage in area for all the register file configurations (except for the 1R1W), even when array duplication

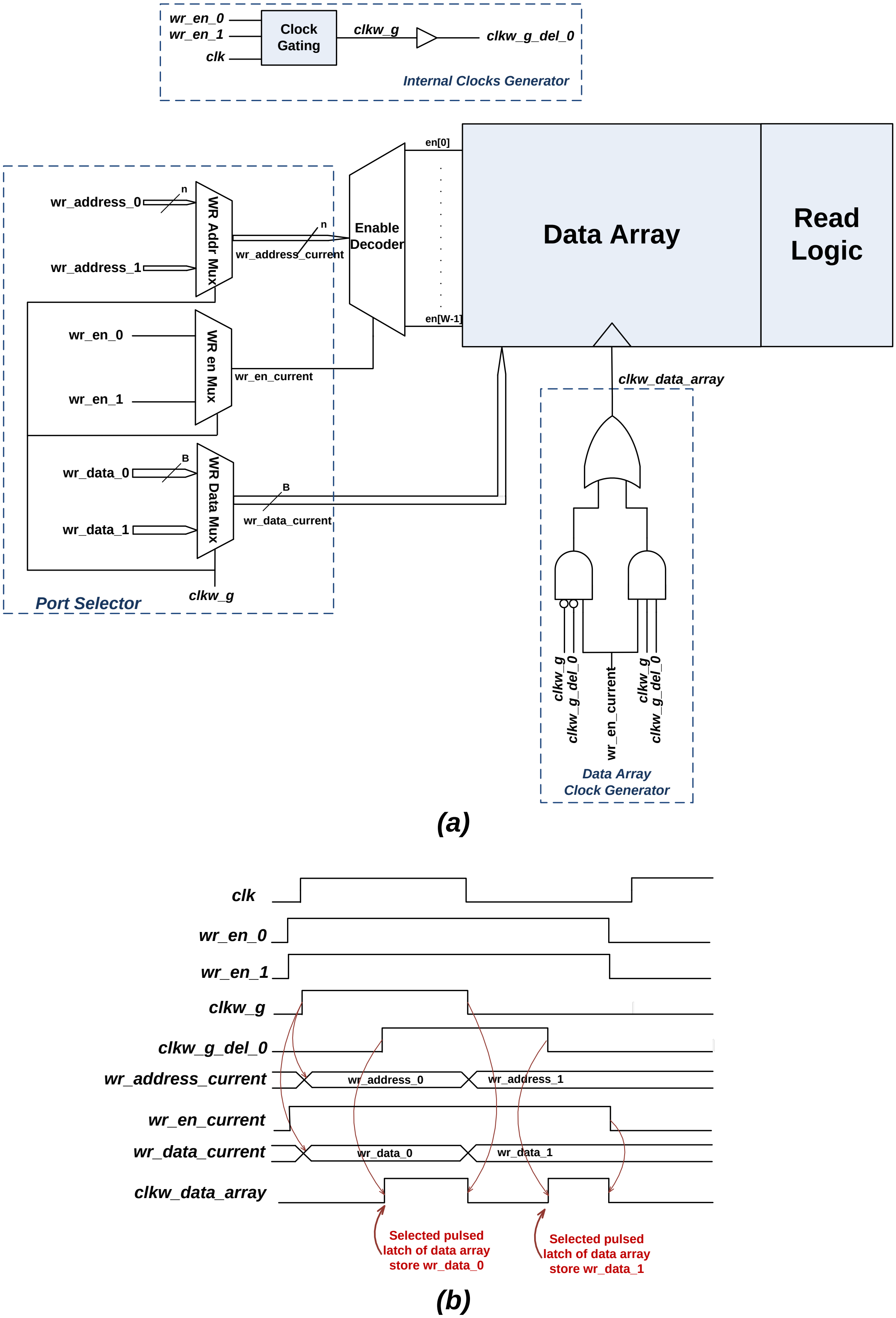


Figure 4.17: The proposed write logic structure for register files with two virtual write ports.

and double pumping are applied to SRAM. However, there is still a significant area overhead for each added port (as an example, the area of the 4R2W register file is more than 2.4X the area of the 1R1W register file).

Here comes one of the advantages of our proposed implementation. As shown in Figure 4.18, a negligible area overhead is added for every added read or write port. As an example, the area of the 4R2W pulsed latch register file with our virtual port architecture is only 20% higher than the 1R1W register file. This represents only about 15% of the area overhead of getting the same number of ports physically using pulsed latches or flip-flops, and only 3% of the area overhead when using SRAM. The same proposed approach can be used to add more read or write ports. The maximum number of virtual ports that can be utilized using single read or write ports will depend on the operating clock frequency and the used technology. Even if the needed number of ports is higher than the maximum number that can be implemented with one port, an additional physical port can be added and the same

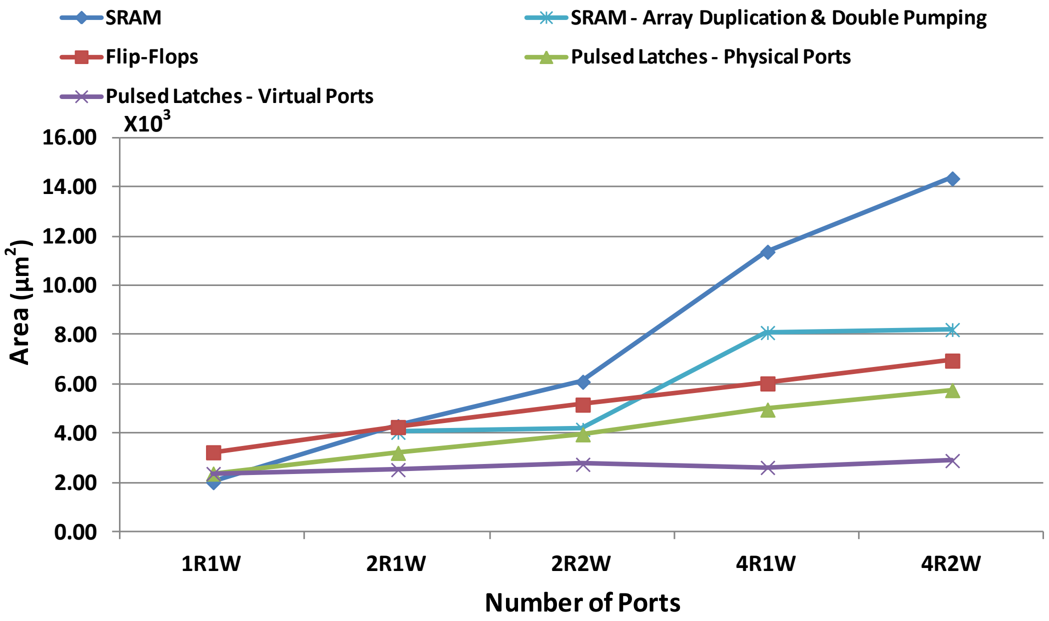


Figure 4.18: Area comparisons of the five implementations of the 32-bits x 32-words register files with different read and write ports

proposed read and write logic can be used for both ports. This will result in a huge saving in area when compared to the other implementation choices.

Also, since the area overhead of the 4R2W register file is very small when compared to that of 1R1W, the 4R2W architecture can be used globally in all the designs. At run time, the processing unit will have the freedom of utilizing as many port as needed depending on the running application. For applications that have many independent operations that can be processed in parallel, having multiple read and write ports can have an advantage in the processing time of such applications.

4.6.3 Register Files Power Consumption

To get an estimate of the average power consumed by each register file architecture, we did post-layout power analyses based on VCD activity obtained by testbenches written for each register file configuration, where random data is written to random addresses, while data is read from random addresses. This was carried output for more than 200 cycles at a clock frequency of 500 MHz. Figure 4.19 shows the obtained power consumption for the different register file implementations for different number of ports. As expected for a 1-Kbit register file size, the SRAM power consumption is always much higher than any of the standard cell based implementations. As an example, the power consumption of the 4R2W SRAM register file is more than 2.5X the power consumption of the flip-flop based or pulsed latch based register files. Even the power consumption of the SRAM register file increases slightly when array duplication and double pumping are applied.

Looking at the standard cell based implementations, the flip-flop based register file is always consuming higher power when compared to the two pulsed latch implementations. This is expected, since flip-flops power consumption are higher when compared to pulsed latches with a shared pulser. Also, flip-flops have higher input clock load, which results in higher clock network power. For the two pulsed latch implementations, the one with dedicated

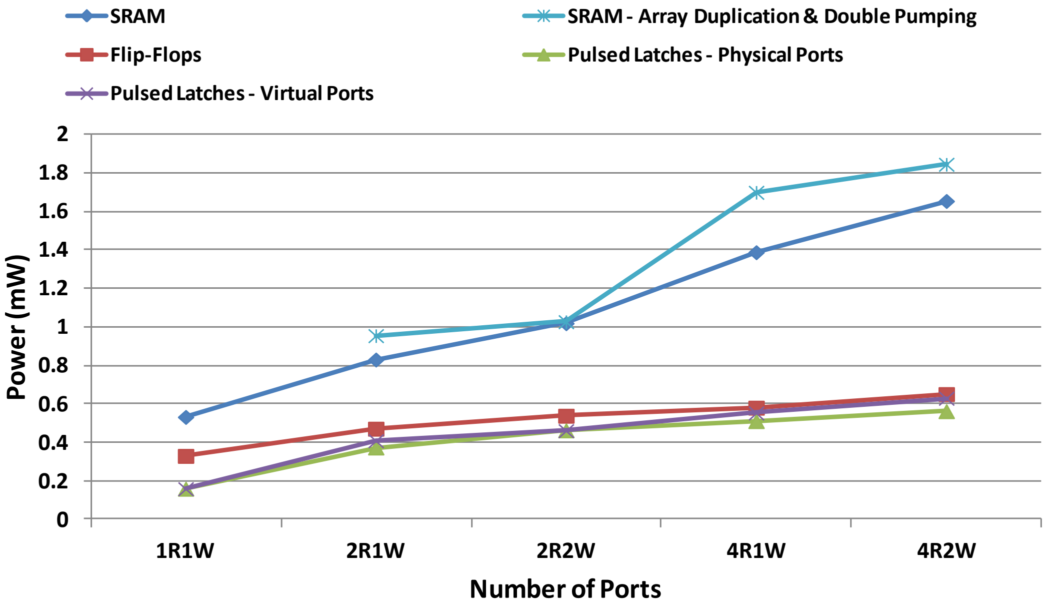


Figure 4.19: Power consumption of the five implementations of register files with different number of ports

physical port for read or write operation consumes a slightly less power than our proposed virtual port approach. However, the power overhead of the proposed approach is less than 10% than its counterpart with physical port (only 7% higher on average across the different number of ports). When compared to flip-flop based ones, the power consumption of the pulsed latch register file with virtual ports is on average 9% lower.

As described in 4.6.2, the 4R2W pulsed latch register file with virtual port can be used globally to run applications that may need lower number of ports. However, the associated power consumption overhead shouldn't be significant. Figure 4.20 shows the power consumption of the proposed 4R2W register file when running the same testbenches used in the power analysis of different number of ports. As shown, it seems that (except for the 1R1W), the power overhead is only around 2%. Even with the 87% power overhead when compared to a dedicated 1R1W pulsed latch register file, the power consumption of the 4R2W register file working as a 1R1W will be more than 10% lower in power than a flip-flop based 1R1W register file.

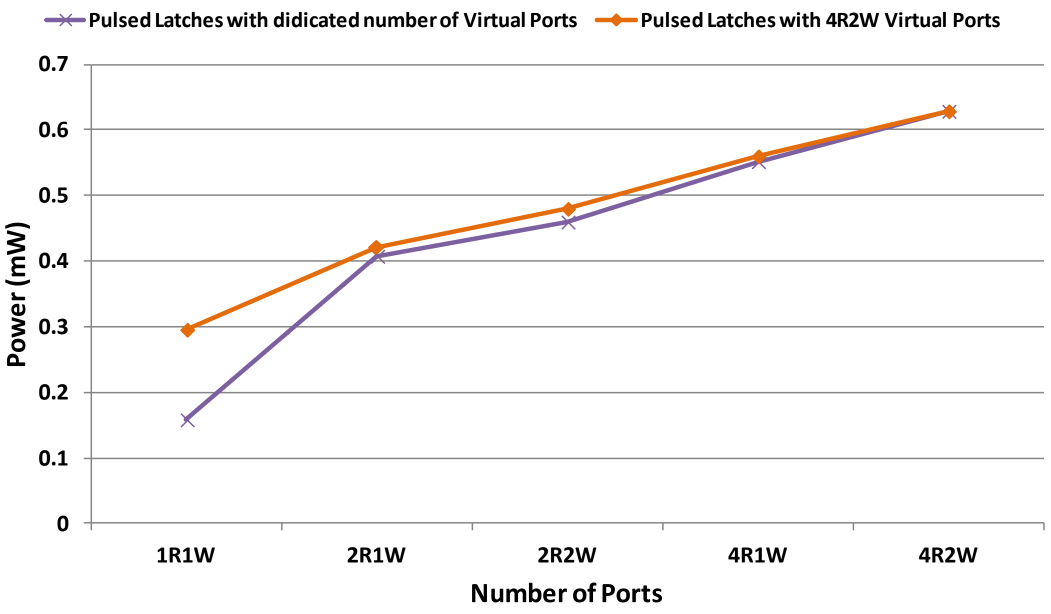


Figure 4.20: Power consumption of the pulsed latch based register file with 4R2W virtual ports when running the same testbenches of the lower number of ports.

4.7 Conclusion

In this chapter, different register file implementations were examined. The traditional implementations of 1R1W register file using SRAM, flip-flops, or latches were compared and a pulsed latch implementation was proposed to provide an efficient standard cell implementation for register files of small sizes. In addition, multiport register file architectures were discussed, including different techniques presented in the literature to achieve better area efficiency.

A novel pulsed latch implementation for multiport register files was proposed and details about adding virtual read and write port using lower number of physical ports was discussed. Several implementations of multiport register files with different number of ports were presented and compared. The proposed pulsed latch implementation with virtual ports showed huge savings in area when compared to the other implementations using SRAMs, flip-flops, or pulsed latches with physical ports. In addition, the proposed implementations are very energy efficient, consuming, on average, only 7% higher power when compared to the pulsed latch based register file with physical ports. However, it is always consuming lower power than the flip-flops or SRAM counterparts.

The 4R2W register file implemented with the proposed virtual port approach was examined to be used as a general register file implementation that can be configured at run time to run with the required number of ports according to the running application. The obtained results when the register file is configured to run with smaller number of ports show a negligible power overhead when compared to other register files with dedicated read and write ports. At the same time, the area overhead is only 20% when compared to a dedicated 1R1W pulsed latch register file.

Chapter 5

Future work

In this chapter, we will discuss some of the other work in progress, in addition to some of the future work. We will divide our discussion into two sections. In the first section, we will focus on work related directly to the pulsed latch at the circuit level, mainly focusing on the clock gating of pulsed latch, which can also be called pulse gating. In the second section, we will discuss some future work related to the usage of pulsed latch in the register file implementation, mainly focusing on some architecture level work.

5.1 Pulsed Latches at the Circuit Level

As previously discussed, the pulser circuit consumes significant amount of power. That's why it is preferred to share the pulser across a large number of latches. However, due to some issues in the placement of the pulser and the latches, this can't always be achieved. In addition, even when the pulser is shared among several latches, the percentage of power consumed by the pulser is still significant. As an example, for a 16-bit register, consisting of a pulser driving 16 latches, the pulser consumes about 35% of the total register power assuming a 50% data activity on the latches' inputs. This number goes up to around 45% of the total register power for a 20% data activity. In addition, the power share of the pulser will increase if the number of shared latches decreases. Most of this power is the switching power due to charging and discharging of the internal nodes through the pulser's delay chain. Hence, a significant power saving is expected if the pulser is turned off when no driving pulse is needed.

Although several works [35, 48, 70] have been presented in the literature trying to apply clock gating to pulsed latch circuits, several drawbacks are still existing. The main challenge in applying traditional clock gating to pulsed latch circuit is the conflict between the methodology of selecting which latches should be shared by the same pulser and which latches should be gated together. While the former one will highly depend on the placement of the latches, trying to drive the nearby latches by the same pulser, the later will depend on the gating function of each latch, which depends on the logic driving this latch. Hence, to apply the conventional clock gating techniques, the gating functions of the considered latches have to be similar as well as they should be physically placed close [35]. Since these two conditions may not occur simultaneously, either these latches will not be gated or they will be divided into smaller groups and more pulsers will be added to drive these groups. Both solutions will add some extra power overhead.

Alternatively, pulse gating can be done based on the data activity of the latches sharing the same pulser. Since the data stored in different registers are not updated in every clock cycle, gating the pulser from generating any pulse when no data update is needed could save a significant amount of power. This can be done by comparing the input data with the already stored data in the latches, if they are similar, then the pulser driving these latches should be gated. We call this technique *pulser self gating*, since the gating signal is generated inside each pulsed latch, without the need of any additional clock gating cell. A similar technique called XOR self-gating is introduced by Synopsys to be used with flip-flops with low data activity [27]. In this technique, a tree of XOR gates can be used to compare the inputs and outputs of a group of flip-flops, and the output of this tree is used to control a clock gating cell that is driving the clock signal to this flip-flops group [54]. However, this XOR tree adds a significant area and power overhead.

Figure 5.1 shows our proposed pulser self gating approach. Three more transistor are needed in each latch, in addition to a single pull up transistor for each pulser circuit. Two of the three added transistors inside the latch act as a simple comparator, comparing the latch's input and output (the transistor on the right is comparing the input D with the inverted value of the output (Q_{bar}), while the one on the left is comparing the inverted value of the input with an internal node storing the same output value). If the values of the input and output are the same, then one of these two transistors will be ON, with one side at logic '0', hence, pulling the EN signal down to logic '0'. If the input logic value is different from the output, then one of these two transistors will be ON with one side at logic '1', hence, pulling the EN up to be a weak logic '1'. The third transistor whose gate is connected to the EN signal is used to discharge the pre-charged $PulseEnable$ signal, whenever the value of the EN signal is logic '1' (i.e. when the stored value is different from the input value). This $PulseEnable$ signal is a wired-OR signal shared between all the latches. Hence, whenever any latch need to change its stored value, this $PulseEnable$ signal will be pulled down to enable the pulser circuit. The pull-up PMOS transistor is used to pre-charge the common $PulseEnable$ node and can be enabled by either the inverted version of the pulse signal or through a delayed version of the clock signal.

Initial implementation of this technique was implemented and tested with a 16-bit pulsed latch register. The initial obtained results show a significant power saving for data activity of 40% or less. The power saving can go up to more than 20% for data activity less than 25%. This proposed technique can be applied as a standalone technique or in addition to the conventional clock gating technique to achieve higher clock gating efficiency.

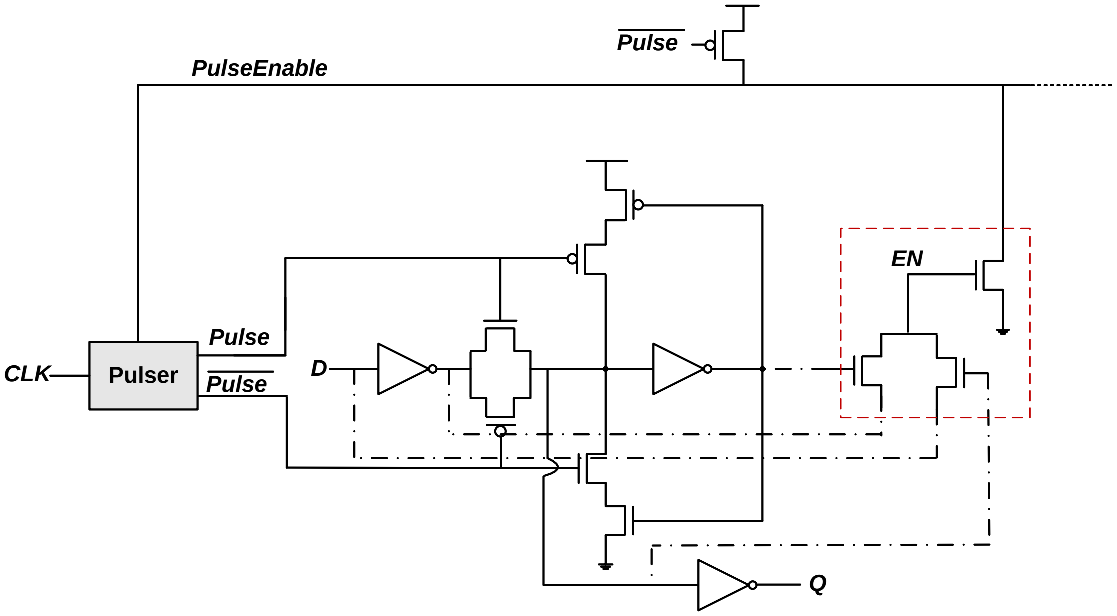


Figure 5.1: The proposed pulsed latch self gating approach.

5.2 Pulsed Latches in Register Files

Although there may be a lot of advantages of having multiport register files, their implementations used to be very costly, adding a significant power, area and latency overhead. With our proposed approach of implementing virtual multiport register files using pulsed latches, most of these overheads already diminished, allowing the addition of an extra read or write port at nearly no cost. This will open the door to few interesting questions. First, the addition of read and write ports will allow parallel access for read and write operations, which can result in executing several independent instructions in parallel. Then, which can be more energy efficient, running an application using a single port register file running at the highest possible frequency or dividing the main applications into several parallel taks using the multiport register file while running at a lower frequency? Also, which approach will give shorter execution time? What is the maximum number of ports that can be used to achieve significant benefits? Which application category can benefit from such approach? A cost function may need to be implemented to answer the previous questions, taking into account what are mentioned previously, in addition to several other factors such as the processing core maximum operating frequency and the size of the register file.

Other interesting questions also arise. What can be the maximum number of virtual ports that can be implemented for a certain timing budget? What is the maximum register file size that can be implemented with such approach while keeping all its advantages in comparison to other implementation choices? What is the effect of supply voltage scaling on the reliability of the virtual port implementation?

Also, since additional read ports are added at a minimum cost, can one of these read ports be used as a self test for the register file write operation, where the read port that starts its read operation later in the clock cycle can be used to check a data written earlier in the same cycle? Can this be used to detect any defects in the data array? Can this be used to correct any write errors that can be associated, as an example, with any PVT variations? What will be the overhead of implementing such correcting techniques?

The main conclusion from the previous discussion is that there are lots of opportunities arising from the ability of implementing multiport register file at a very small cost. Each of these opportunities needs to be carefully investigated to figure out all the gained benefits in reliability, performance and energy consumption.

Chapter 6

Conclusions and Summary

In this thesis, we have discussed the usage of pulsed latch in the implementation of low power reliable System-on-Chip. Pulse latches have been an attractive alternative for flip-flops in building sequential elements. In addition to have a timing model close to flip-flops, which make their timing analysis quite similar, they also have a smaller area, consumes less power and can achieve higher performance. However, they have some challenges that need to be addressed.

In chapter 2, we discussed the reliability of pulsed latch circuits. We proposed a methodology for evaluating the effect of process, voltage and temperature variations on pulsed latches. Using this methodology, we were able to quantify the probability of write failure taking into account both components of the circuit, the pulser and the latches. We evaluated the effect of temperature variation and noticed that the pulsed latch reliability degrades as the temperature decreases. in addition, we showed that pulsed latch are very sensitive to voltage variations and highlighted that novel techniques need to be implemented in order to apply voltage scaling without significant degradation in power or performance.

In chapter 3, two novel approaches were introduced to allow the usage of wide range voltage scaling on pulsed latch. The first approach depends on performing an additional level on voltage scaling on the pulser delay path, while the second approach depends on implementing multiple delay paths inside the pulser. Both approaches were tested and proved to allow more than 30% of voltage scaling with any degradation in the pulsed latch reliability. In addition post-layout simulations shows a minimum area and power overhead compared to traditional pulsed latch circuits.

In chapter 4, the usage of pulsed latch in the implementation of register files was widely discussed and compared with other conventional implementation options such as SRAM and flip-flops. Pulsed latch implementation of a single read and single write port register file has a significant advantage over flip-flop implementation. When compared to SRAM based register files, pulsed latch showed a competitive alternative for small size register and for low power applications. In addition, a novel implementation of multiport register file was proposed. The implementation depends on driving few virtual read and write ports from a single read and write port using some additional pulser circuits. This technique eliminates the significant area and power overhead associated with adding read and write ports. When compared to other conventional implementation, our proposed approach showed a significant advantage in both area and power. When compare to SRAM implementation, area saving that can reach more than 60% can be achieved, even when techniques like array duplication and double pumping are applied to SRAM. When compared to other standard cell implementations, area saving of more than 50% was achieved when compare to flip-flop implementation, and more than 40% when compared to a conventional pulsed latch implementation. For power, power saving of more than 60% was achieved when compared to SRAM based implementation, while power saving of around 15% was achieved when compared to flip-flop based implementation. In addition, the idea of using a general multiport register file that can be configured at run time was investigated and tested.

In chapter 5, few of the on going work in addition to some future work were discussed. At the circuit level, the idea of pulser self gating was introduced and the initial implementation and results shows a significant improvement in power consumption. At the architecture level, few ideas have been discussed that can gain from the benefits of building multiport register file at a very small cost in area and power.

Bibliography

[1] International Technology Roadmap for Semiconductors (ITRS). `http://www.itrs2.net`. Accessed: 7-Febraury-2017.

[2] Synopsys 32/28nm iPDKs. `https://www.synopsys.com/community/university-program/teaching-resources.html`. Accessed: 25-January-2017.

[3] A. Aggarwal and M. Franklin. Energy efficient asymmetrically ported register files. In *Proceedings 21st International Conference on Computer Design*, pages 2–7, Oct 2003.

[4] M. Alam, K. Roy, and C. Augustine. Reliability- and process-variation aware design of integrated circuits – a broader perspective. In *Reliability Physics Symposium (IRPS), 2011 IEEE International*, April 2011.

[5] M. Alioto, E. Consoli, and G. Palumbo. Analysis and comparison in the energy-delay-area domain of nanometer CMOS flip-flops: Part I–methodology and design strategies. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 19(5), May 2011.

[6] M. Alioto, E. Consoli, and G. Palumbo. Analysis and comparison in the energy-delay-area domain of nanometer CMOS flip-flops: Part II–results and figures of merit. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 19(5), May 2011.

[7] M. Alioto, E. Consoli, and G. Palumbo. *Flip-Flop Design in Nanometer CMOS*. Springer International Publishing, 2015.

[8] M. Alioto, E. Consoli, and G. Palumbo. Variations in Nanometer CMOS Flip-Flops: Part I –impact of process variations on timing. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(8):2035–2043, Aug 2015.

[9] M. Alioto, E. Consoli, and G. Palumbo. Variations in Nanometer CMOS Flip-Flops: Part II –energy variability and impact of other sources of variations. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(3):835–843, March 2015.

[10] R. Balasubramonian, S. Dwarkadas, and D. H. Albonesi. Reducing the complexity of the register file in dynamic superscalar processors. In *Proceedings. 34th ACM/IEEE International Symposium on Microarchitecture. MICRO-34*, pages 237–248, Dec 2001.

[11] T. Baumann, J. Berthold, T. Niedermeier, T. Schoenauer, J. Dienstuhl, D. Schmitt-Landsiedel, and C. Pacha. Performance improvement of embedded low-power microprocessor cores by selective flip flop replacement. In *33rd European Solid State Circuits Conference, ESSCIRC 2007*, Sept 2007.

[12] S. Bernard, M. Belleville, J.-D. Legat, A. Valentian, and D. Bol. Ultra-wide voltage range pulse-triggered flip-flops and register file with tunable energy-delay target in 28 nm UTBB-FDSOI. *Microelectronics Journal*, 57:76 – 86, 2016.

[13] K. Bernstein, D. Frank, A. Gattiker, W. Haensch, B. Ji, S. Nassif, E. Nowak, D. Pearson, and N. Rohrer. High-performance CMOS variability in the 65-nm regime and beyond. *IBM Journal of Research and Development*, 50(4.5), 2006.

[14] D. Bol, J. D. Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J. D. Legat. Sleepwalker: A 25-MHz 0.4-V sub-mm^2 7-μW/MHz microcontroller in 65-nm LP/GP CMOS for low-carbon wireless sensor nodes. *IEEE Journal of Solid-State Circuits*, 48(1):20–32, Jan 2013.

[15] F. Botman, J. de Vos, S. Bernard, F. Stas, J. D. Legat, and D. Bol. Bellevue: A 50MHz variable-width SIMD 32bit microcontroller at 0.37v for processing-intensive wireless sensor nodes. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1207–1210, June 2014.

[16] J. Chen, L. T. Clark, and T. H. Chen. An ultra-low-power memory with a subthreshold power supply voltage. *IEEE Journal of Solid-State Circuits*, 41(10):2344–2353, Oct 2006.

[17] D. Chinnery and K. Keutzer. *Closing the Gap Between ASIC & Custom: Tools and Techniques for High-Performance ASIC Design.* Kluwer Academic Publishers, 2002.

[18] D. Chinnery and K. Keutzer. *Closing the Power Gap Between ASIC & Custom: Tools and Techniques for Low Power Design.* Springer New York, 2007.

[19] L. Clark, E. Hoffman, J. Miller, M. Biyani, Y. Liao, S. Strazdus, M. Morrow, K. Velarde, and M. Yarch. An embedded 32-b microprocessor core for low-power and high-performance applications. *IEEE Journal of Solid-State Circuits*, 36(11), Nov 2001.

[20] E. Consoli, G. Palumbo, J. Rabaey, and M. Alioto. Novel class of energy-efficient very high-speed conditional push-pull pulsed latches. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 22(7), July 2014.

[21] J. L. Cruz, A. Gonzalez, M. Valero, and N. P. Topham. Multiple-banked register file architectures. In *Proceedings of 27th International Symposium on Computer Architecture (IEEE Cat. No.RS00201)*, pages 316–325, June 2000.

[22] D. C. Daly, L. C. Fujino, and K. C. Smith. Through the looking glass – the 2017 edition: Trends in solid-state circuits from ISSCC. *IEEE Solid-State Circuits Magazine*, 9(1):12–22, winter 2017.

[23] S. Dhong, R. Guo, M. Z. Kuo, P. L. Yang, C. C. Lin, K. Huang, M. J. Wang, and W. Hwang. A 0.42v Vccmin ASIC-compatible pulse-latch solution as a replacement for a traditional master-slave flip-flop in a digital SOC. In *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*, pages 1–4, Sept 2014.

[24] G. S. Ditlow, R. K. Montoye, S. N. Storino, S. M. Dance, S. Ehrenreich, B. M. Fleischer, T. W. Fox, K. M. Holmes, J. Mihara, Y. Nakamura, S. Onishi, R. Shearer, D. Wendel, and L. Chang. A 4R2W register file for a 2.3GHz wire-speed POWER^{TM} processor with double-pumped write operation. In *2011 IEEE International Solid-State Circuits Conference*, pages 256–258, Feb 2011.

[25] A. K. Djahromi, A. M. Eltawil, and F. J. Kurdahi. Exploiting fault tolerance towards power efficient wireless multimedia applications. In *2007 4th IEEE Consumer Communications and Networking Conference*, pages 400–404, Jan 2007.

[26] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. In *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ISCA '11, pages 365–376, New York, NY, USA, 2011. ACM.

[27] J. Ezroni. Advanced dynamic power reduction techniques: XOR Self-Gating. Technical report, Synopsys White Paper, April 2011.

[28] E. S. Fetzer, D. Dahle, C. Little, and K. Safford. The parity protected, multithreaded register files on the 90-nm itanium microprocessor. *IEEE Journal of Solid-State Circuits*, 41(1):246–255, Jan 2006.

[29] S. Gangadharan and S. Churiwala. *Constraining Designs for Synthesis and Timing Analysis: A Practical Guide to Synopsys Design Constraints (SDC).* Springer Publishing Company, Incorporated, 2013.

[30] M. Golden, S. Hesley, A. Scherer, M. Crowley, S. C. Johnson, S. Meier, D. Meyer, J. D. Moench, S. Oberman, H. Partovi, F. Weber, S. White, T. Wood, and J. Yong. A seventh-generation x86 microprocessor. *IEEE Journal of Solid-State Circuits*, 34(11):1466–1477, Nov 1999.

[31] R. Goldman, K. Bartleson, T. Wood, K. Kranen, V. Melikyan, and E. Babayan. 32/28nm educational design kit: Capabilities, deployment and future. In *Microelectronics and Electronics (PrimeAsia), 2013 IEEE Asia Pacific Conference on Postgraduate Research in*, Dec 2013.

[32] M. Horowitz, P. Chow, D. Stark, R. T. Simoni, A. Salz, S. Przybylski, J. Hennessy, G. Gulak, A. Agarwal, and J. M. Acken. MIPS-X: a 20-MIPS peak, 32-bit microprocessor with on-chip cache. *IEEE Journal of Solid-State Circuits*, 22(5):790–799, Oct 1987.

[33] H. Kaeslin. *Digital Integrated Circuit Design: From VLSI Architectures to CMOS Fabrication.* Cambridge University Press, Cambridge, 007 2007.

[34] A. Khajeh, A. Gupta, N. Dutt, F. Kurdahi, A. Eltawil, K. Khouri, and M. Abadir. TRAM: A tool for temperature and reliability aware memory design. In *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, April 2009.

[35] S. Kim, I. Han, S. Paik, and Y. Shin. Pulser gating: A clock gating of pulsed-latch circuits. In *Proceedings of the 16th Asia and South Pacific Design Automation Conference*, ASPDAC '11, pages 190–195, Piscataway, NJ, USA, 2011. IEEE Press.

[36] R. Kumar, K. C. Bollapalli, R. Garg, T. Soni, and S. P. Khatri. A robust pulsed flip-flop and its use in enhanced scan design. In *Computer Design, 2009. ICCD 2009. IEEE International Conference on*, pages 97–102, Oct 2009.

[37] R. Kumar and G. Hinton. A family of 45nm IA processors. In *2009 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*, pages 58–59, Feb 2009.

[38] M. Lanuzza, R. De Rose, F. Frustaci, S. Perri, and P. Corsonello. *Impact of Process Variations on Pulsed Flip-Flops: Yield Improving Circuit-Level Techniques and Comparative Analysis*, pages 180–189. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[39] X. Liang and D. Brooks. Mitigating the impact of process variations on processor register files and execution units. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 39, pages 504–514, Washington, DC, USA, 2006. IEEE Computer Society.

[40] H. Mahmoodi, S. Mukhopadhyay, and K. Roy. Estimation of delay variations due to random-dopant fluctuations in nanoscale CMOS circuits. *IEEE Journal of Solid-State Circuits*, 40(9), 2005.

[41] S. M. Martin, K. Flautner, T. Mudge, and D. Blaauw. Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads. In *Proceedings of the 2002 IEEE/ACM International Conference on Computer-aided Design*, ICCAD '02, pages 721–725, New York, NY, USA, 2002. ACM.

[42] T. McConaghy, P. Drennan, K. Breen, J. Dyck, and A. Gupta. *Variation-Aware Design of Custom Integrated Circuits: A Hands-on Field Guide*. Springer, 2012.

[43] P. Meinerzhagen, C. Roth, and A. Burg. Towards generic low-power area-efficient standard cell based memory architectures. In *2010 53rd IEEE International Midwest Symposium on Circuits and Systems*, pages 129–132, Aug 2010.

[44] P. Meinerzhagen, S. M. Y. Sherazi, A. Burg, and J. N. Rodrigues. Benchmarking of standard-cell based memories in the Sub-V_T domain in 65-nm CMOS technology. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 1(2):173–182, June 2011.

[45] S. Mukhopadhyay, H. Mahmoodi-Meimand, and K. Roy. Modeling and estimation of failure probability due to parameter variations in nano-scale SRAMs for yield enhancement. In *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*, June 2004.

[46] S. D. Naffziger, G. Colon-Bonet, T. Fischer, R. Riedlinger, T. J. Sullivan, and T. Grutkowski. The implementation of the itanium 2 microprocessor. *IEEE Journal of Solid-State Circuits*, 37(11):1448–1460, Nov 2002.

[47] F. Oboril and M. B. Tahoori. Extratime: Modeling and analysis of wearout due to transistor aging at microarchitecture-level. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, pages 1–12, June 2012.

[48] S. Paik, I. Han, S. Kim, and Y. Shin. Clock gating synthesis of pulsed-latch circuits. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 31(7):1019–1030, July 2012.

[49] S. Paik, G.-J. Nam, and Y. Shin. Implementation of pulsed-latch and pulsed-register circuits to minimize clocking power. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2011.

[50] S. Paik and Y. Shin. Pulsed-latch circuits to push the envelope of asic design. In *SoC Design Conference (ISOCC), 2010 International*, pages 150–153, Nov 2010.

[51] J. Pille, D. Wendel, O. Wagner, R. Sautter, W. Penth, T. Froehnel, S. Buettner, O. Torreiter, M. Eckert, J. Paredes, D. Hrusecky, D. Ray, and M. Canada. A 32kb 2R/1W l1 data cache in 45nm soi technology for the power7^{TM} processor. In *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, pages 344–345, Feb 2010.

[52] Y. Pu, X. Zhang, J. Huang, A. Muramatsu, M. Nomura, K. Hirairi, H. Takata, T. Sakurabayashi, S. Miyano, M. Takamiya, and T. Sakurai. Misleading energy and performance claims in sub/near threshold digital systems. In *Computer-Aided Design (ICCAD), 2010 IEEE/ACM International Conference on*, Nov 2010.

[53] J. Rabaey. *Low Power Design Essentials.* Integrated Circuits and Systems. Springer, 2009.

[54] J. S, M. Rao, J. Srinivas, P. Vishwanath, U. H, and J. Rao. Clock gating for power optimization in asic design cycle theory amp; practice. In *Low Power Electronics and Design (ISLPED), 2008 ACM/IEEE International Symposium on*, pages 307–308, Aug 2008.

[55] S. Shibatani and A. H. Li. Pulse-latch approach reduces dynamic power. *EE Times*, 18 July; 2006.

[56] Y. Shin and S. Paik. Pulsed-latch circuits: A new dimension in ASIC design. *IEEE Design Test of Computers*, 28(6):50–57, Nov 2011.

[57] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. De Micheli. Dynamic voltage scaling and power management for portable systems. In *Proceedings of the 38th Annual Design Automation Conference*, DAC '01, pages 524–529, New York, NY, USA, 2001. ACM.

[58] 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.

[59] M. B. Taylor. Is dark silicon useful? Harnessing the four horsemen of the coming dark silicon apocalypse. In *DAC Design Automation Conference 2012*, pages 1131–1136, June 2012.

[60] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi. Cacti 5.1. Technical report, HP Laboratories, April 2008.

[61] J. H. Tseng and K. Asanović. Banked multiported register files for high-frequency superscalar microprocessors. In *Proceedings of the 30th Annual International Symposium on Computer Architecture*, ISCA '03, pages 62–71, New York, NY, USA, 2003. ACM.

[62] A. Venkatraman, R. Garg, and S. P. Khatri. A robust, fast pulsed flip-flop design. In *Proceedings of the 18th ACM Great Lakes Symposium on VLSI 2008, Orlando, Florida, USA, May 4-6, 2008*, pages 119–122, 2008.

[63] N. Verma and A. P. Chandrakasan. A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy. *IEEE Journal of Solid-State Circuits*, 43(1):141–149, Jan 2008.

[64] J. Warnock, Y. Chan, H. Harrer, S. Carey, G. Salem, D. Malone, R. Puri, J. Zitz, A. Jatkowski, G. Strevig, et al. Circuit and physical design of the zenterprise EC12 microprocessor chips and multi-chip module. *IEEE Journal of Solid-State Circuits*, 2014.

[65] J. Warnock, D. Wendel, T. Aipperspach, E. Behnen, R. A. Cordes, S. H. Dhong, K. Hirairi, H. Murakami, S. Onishi, D. C. Pham, J. Pille, S. D. Posluszny, O. Takahashi, and H. Wen. Circuit design techniques for a first-generation cell broadband engine processor. *IEEE Journal of Solid-State Circuits*, 41(8):1692–1706, Aug 2006.

[66] D. F. Wendel, R. Kalla, J. Warnock, R. Cargnoni, S. G. Chu, J. G. Clabes, D. Dreps, D. Hrusecky, J. Friedrich, S. Islam, J. Kahle, J. Leenstra, G. Mittal, J. Paredes, J. Pille, P. J. Restle, B. Sinharoy, G. Smith, W. J. Starke, S. Taylor, A. J. V. Norstrand, S. Weitzel, P. G. Williams, and V. Zyuban. Power7^{TM};, a highly parallel, scalable multi-core high end server processor. *IEEE Journal of Solid-State Circuits*, 46(1):145–161, Jan 2011.

[67] N. Weste and D. Harris. *CMOS VLSI Design: A Circuits and Systems Perspective.* Addison Wesley, 2011.

[68] M. Wirnshofer. *Variation-Aware Adaptive Voltage Scaling for Digital CMOS Circuits.* Springer Netherlands, 2013.

[69] H. Yan, Y. Liu, D. h. Wang, and C. h. Hou. A low-power 8-read 4-write register file design. In *2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, pages 178–181, Sept 2010.

[70] Z. H. Yang and T. Y. Ho. Timing-aware clock gating of pulsed-latch circuits for low power design. In *2013 International Symposium onVLSI Design, Automation, and Test (VLSI-DAT)*, pages 1–4, April 2013.

[71] H. E. Yantir, S. Bayar, and A. Yurdakul. Efficient implementations of multi-pumped multi-port register files in fpgas. In *2013 Euromicro Conference on Digital System Design*, pages 185–192, Sept 2013.

[72] C. Zang, S. Imai, and S. Kimura. Duplicated register file design for embedded simultaneous multithreading microprocessor. In *2005 6th International Conference on ASIC*, volume 1, pages 90–93, Oct 2005.

[73] H. Zhang, X. Chen, N. Xiao, and F. Liu. Architecting energy-efficient STT-RAM based register file on gpgpus via delta compression. In *Proceedings of the 53rd Annual Design Automation Conference*, DAC '16, pages 119:1–119:6, New York, NY, USA, 2016. ACM.

[74] Y. Zhao, J. Li, and K. Mohanram. Multi-port FinFET SRAM design. In *Proceedings of the 23rd ACM International Conference on Great Lakes Symposium on VLSI*, GLSVLSI '13, pages 293–298, New York, NY, USA, 2013. ACM.

[75] V. Zyuban and P. Kogge. The energy complexity of register files. In *Proceedings. 1998 International Symposium on Low Power Electronics and Design (IEEE Cat. No.98TH8379)*, pages 305–310, Aug 1998.