

Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report (DRAFT)

---



Project Title: **A High-radix Online Arithmetic Verification System**

Student: **Zifan Wang**

CID: **01077639**

Course: **EEE4**

Project Supervisor: **Dr. James J. Davis**

Second Marker: **Dr. Christos Bouganis**

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Background</b>	<b>2</b>
<b>3</b>	<b>Requirements Capture</b>	<b>2</b>
<b>4</b>	<b>System-level Design</b>	<b>2</b>
4.1	Testbench Architecture . . . . .	2
4.2	User Interface . . . . .	2
<b>5</b>	<b>Hardware Design Choices</b>	<b>2</b>
5.1	Randomiser . . . . .	2
5.2	Driver . . . . .	3
5.3	Monitor . . . . .	4
5.4	Scoreboard . . . . .	4
<b>6</b>	<b>Software Design Choices</b>	<b>4</b>
6.1	Code Structure . . . . .	4
<b>7</b>	<b>Hardware Implementation</b>	<b>4</b>
7.1	Project Hierarchy . . . . .	4
7.2	Randomiser . . . . .	4
7.3	Driver . . . . .	4
7.4	Monitor . . . . .	5
7.5	Scoreboard . . . . .	5
<b>8</b>	<b>Software Implementation</b>	<b>5</b>
<b>9</b>	<b>System Integration</b>	<b>5</b>
9.1	Qsys . . . . .	5
<b>10</b>	<b>Testing</b>	<b>5</b>
<b>11</b>	<b>Results</b>	<b>5</b>
<b>12</b>	<b>Evaluation</b>	<b>5</b>
<b>13</b>	<b>Conclusion</b>	<b>5</b>
<b>14</b>	<b>Further Work</b>	<b>5</b>
<b>15</b>	<b>User Guide</b>	<b>5</b>

## Abstract

Nice abstract

# 1 Introduction

# 2 Background

# 3 Requirements Capture

# 4 System-level Design

## 4.1 Testbench Architecture

## 4.2 User Interface

# 5 Hardware Design Choices

## 5.1 Randomiser

With relatively low effort, random testing can provide significant coverage and discover relatively subtle errors [7]. LFSRs are a reliable way of generating pseudorandom numbers quickly with low cost [10].

### 5.1.1 LFSR Configurations

To compare, we can examine an 8-bit LFSR with taps on bit [7,5,4,3].

*Fibonacci* – Classical option, easier to write and scale in hardware.

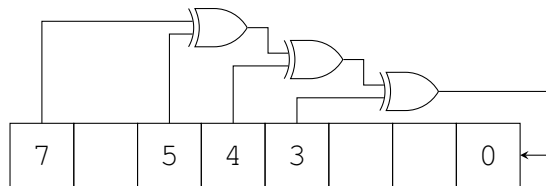


Figure 1: Fibonacci Configuration

*Galois* – Harder to write if variable length is desired, but faster in hardware.

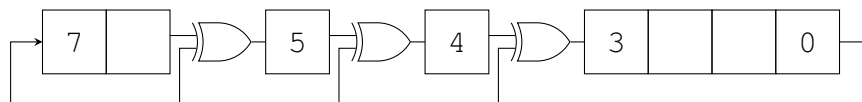


Figure 2: Galois Configuration

*Others* – Other LFSR configurations such as Xorshift [15] exists, but they are much slower in hardware.

### 5.1.2 Randomiser Structure

*Horizontal* – Easy to build, easy to test with.

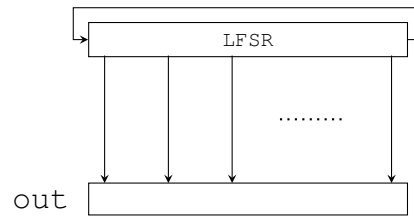


Figure 3: Horizontal Structure

*Vertical* – Nice randomness, more scalable, need to seed all the LFSRs differently.

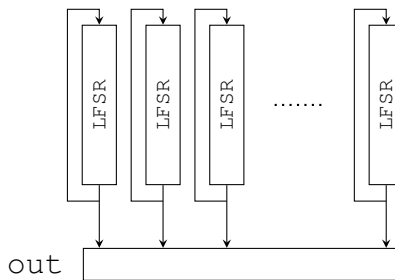


Figure 4: Vertical Structure

## 5.2 Driver

### 5.2.1 Dual Driver System

One driver focusses on fast stress tests, The other allows handwritten tests to coexist with random tests. They can be switched in software.

## 5.3 Monitor

## 5.4 Scoreboard

# 6 Software Design Choices

## 6.1 Code Structure

# 7 Hardware Implementation

## 7.1 Project Hierarchy

## 7.2 Randomiser

## 7.3 Driver

### 7.3.1 Delay Tester

I built a delay tester to find out the delay of the DUT. With a 3-bit counter as shown in the timing diagram, it can measure this delay for up to 8 clock cycles.

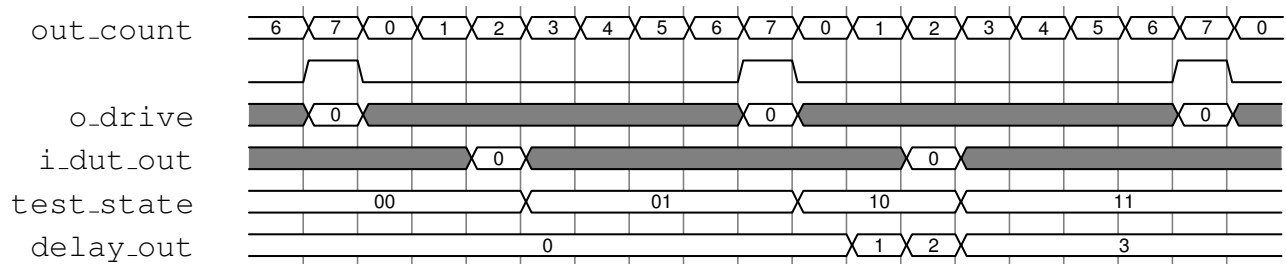


Figure 5: 3-bit Delay Tester FSM

Testing with 0 is safe since LFSR will never output 0.

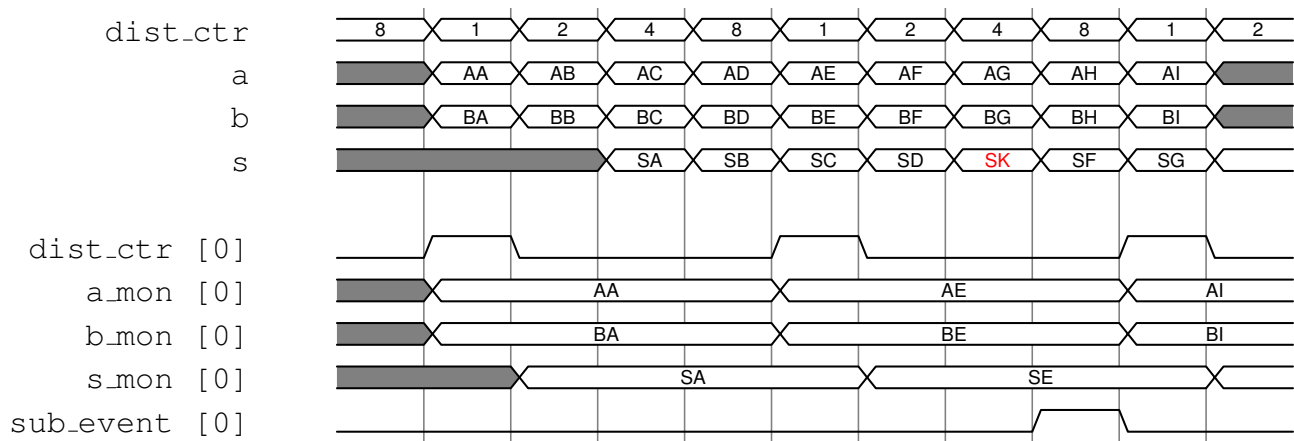


Figure 6: Distributed Monitoring System

### 7.3.2 Switching system

## 7.4 Monitor

### 7.4.1 Sub Monitors

## 7.5 Scoreboard

# 8 Software Implementation

# 9 System Integration

## 9.1 Qsys

# 10 Testing

# 11 Results

# 12 Evaluation

# 13 Conclusion

# 14 Further Work

# 15 User Guide

## References

- [1] I. Ahmed, S. Zhao, J. Meijers, O. Trescases and V. Betz, “Automatic BRAM Testing for Robust Dynamic Voltage Scaling for FPGAs”, *Int. Conf. on Field-Programmable Logic and Applications*, 2018.
- [2] A. Amin, W. Shinwari, “High-Radix Multiplier-Dividers: Theory, Design, and Hardware”, *IEEE Trans. Comput.*, vol. 1, no.8, 2008.
- [3] R.P. Brent, “A Regular Layout for Parallel Adders”, *IEEE Trans. Comput.*, vol. C-31, pp. 260-264, 1982.
- [4] B. Catanzaro, and B. Nelson, “Higher Radix Floating-Point Representations for FPGA-Based Arithmetic”, *Proceedings of the 51st Annual Design Automation Conference*, 2005.
- [5] L. Chen, F. Lombardi, P. Montuschi, J. Han and W. Liu, “Design of Approximate High-Radix Dividers by Inexact Binary Signed-Digit Addition”, *Proceedings of the on Great Lakes Symposium on VLSI*, 2017.
- [6] R. Duncan, “A Survey of Parallel Computer Architectures”, *Computer*, vol. 23, pp. 5-16, 1990.
- [7] J.W. Duran, “An Evaluation of Random Testing”, *IEEE Trans. on Software Engineering*, vol. SE-10, no. 4, pp. 438-444, 1984.
- [8] M.D. Ercegovic, “On-line Arithmetic: An Overview”, *28th Annual Technical Symposium*, pp. 86-93, International Society for Optics and Photonics, 1984.
- [9] M.D. Ercegovic, and T. Lang, “Digital Arithmetic”, Morgan Kaufmann, 2003.
- [10] S. Hazwani, et al, “Randomness Analysis of Pseudo Random Noise Generator Using 24-bits LFSR”, *Fifth Int. Conf. on Intelligent Systems, Modelling and Simulation*, 2014.
- [11] P. Kornerup, “Reviewing High-Radix Signed-Digit Adders”, *IEEE Trans. Comput.*, vol.64, no. 5, pp. 1502-1505, 2015.
- [12] H. Li, J.J. Davis, J. Wickerson and G.A. Constantinides, “ARCHITECT: Arbitrary-precision Constant-hardware Iterative Compute”, *Int. Conf. on Field-Programmable Technology*, 2017.
- [13] T. Lynch, and M.J. Schulte, “A High Radix On-line Arithmetic for Credible and Accurate Computing”, *Journal of Universal Computer Science*, vol. 1, no. 7, pp. 439-453, 1995.
- [14] T. Lynch, and M.J. Schulte, “Software for High Radix On-line Arithmetic”, *Reliable Computing*, vol. 2, no. 2, pp. 133-138, 1996.
- [15] G. Marsaglia, “Xorshift RNGs”, *Journal of Statistical Software*, 2003.
- [16] H.R. Srinivas, and K.K. Parhi, “High-Speed VLSI Arithmetic Processor Architectures Using Hybrid Number Representation”, *J. of VLSI Sign. Process.*, vol. 4. pp. 177-198, 1992.
- [17] K. Shi, D. Boland, and G.A. Constantinides, “Accuracy-Performance Tradeoffs on an FPGA through Overclocking”, *Proc. Int. Symp. Field-Programmable Custom Computing Machines*, pp. 29-36, 2013.

- [18] K. Shi, D. Boland, E. Stott, S. Bayliss, and G.A. Constantinides, “Datapath Synthesis for Over-clocking: Online Arithmetic for Latency-Accuracy Trade-offs”, *Proceedings of the 13th Symposium on Field-Programmable Custom Computing Machines*, pp. 1-6, ACM, 2014.
- [19] O. Šćekić “FPGA Comparative Analysis”, *University of Belgrade*, 2005.
- [20] A.F. Tenca, and M.D. Ercegovac, “Design of high-radix digit-slices for on-line computations”, 2007.
- [21] K.S. Trivedi, and M.D. Ercegovac, “On-line Algorithms for Division and Multiplication”, *IEEE Trans. Comput.*, vol. C-26, no. 7, pp. 667-680, 1977.
- [22] P. Whyte, “Design and Implementation of High-radix Arithmetic Systems Based on the SDNR/RNS Data Representation” *Edith Cowan University*, 1997.
- [23] Y. Zhao, J. Wickerson, and G.A. Constantinides, “An Efficient Implementation of Online Arithmetic”, *Int. Conf. on Field-Programmable Technology*, 2016.
- [24] Accellera Systems Initiative, “Universal Verification Methodology 1.2 User’s Guide”, 2015.
- [25] Altera Corporation, “Cyclone V SoC Development Board Reference Manual”, 2015.
- [26] Altera Corporation, “Memory System Design”, *Embedded Design Handbook*, 2010.
- [27] Altera Corporation, “Introduction to Altmemory IP”, *External Memory Interface Handbook: Reference Material*, vol. 3, 2012.
- [28] Altera Corporation, “Phase-Locked Loop Basics, PLL”.
- [29] Altera Corporation, “Creating Qsys Components”, 2018.
- [30] Altera Corporation, “Cyclone V Hard Processor System Technical Reference Manual”, 2018.
- [31] Imperial College “An Ethics Code”, *Imperial College Research Ethics Committee*, 2013.
- [32] Intel Corporation, “Cyclone V SoC Development Kit and Intel SoC FPGA Embedded Development Suite”.
- [33] Intel Corporation, “Introduction to Intel FPGA IP Cores”, 2018.
- [34] Intel Corporation, “Avalon Interface Specifications”, 2018.
- [35] RocketBoards.org, “GSRD 14.1 User manual”, 2015.
- [36] Xilinx, Inc, “Zynq-7000 All Programmable SoC”, 2018.
- [37] Xilinx, Inc, “ZedBoard (Zynq Evaluation and Development) Hardware User’s Guide”, 2012.