Dynamic Reconfiguration using Crystalline Oxide Semiconductor Technology in a Multi-Context Field Programmable Gate Array

Master’s thesis at The Institute of Technology at Linköping University by Nora Björklund

LiTH-ISY-EX--15/4823--SE
Linköping 2015
Dynamic Reconfiguration using Crystalline Oxide Semiconductor Technology in a Multi-Context Field Programmable Gate Array

Master’s thesis at The Institute of Technology at Linköping University by

Nora Björklund

LiTH-ISY-EX--15/4823--SE

Supervisor: Takeshi Aoki
Semiconductor Energy Laboratory Corporation, Ltd.

Examiner: Atila Alvandpour
ISY, Linköping University

Linköping, 9 februari 2015
Dynamic reconfigurable FPGAs was described in the 1990's by Bolotski [1], Tau [2], DeHon [3, 4, 5] and Trimberger [6]. The idea was to expand the FPGA's function space temporally instead of spatially, and in doing so allowing reuse of the FPGA's functional resources in time, increasing the utilization rate of the functional resources.

Many DPGA designs today are based on the "Time-Multiplexed FPGA" that Trimberger et al. described in 1997 now more commonly called Multi-Context FPGA, in which memory bits are added to every configuration memory to create configuration contexts that the FPGA can switch between. The dominating memory technology used in FPGAs and DPGAs is SRAM; a volatile memory technology that uses a relatively large area and also have an excessive power consumption. Because of the increase of configuration bits in DPGAs, the SRAM memory drawbacks imposes larger effects on its design.

In recent years new memory technologies have been implemented in a broad range of applications, out of which DPGAs are one. Among these technologies, implementation with crystalline IGZO FETs have been argued to overcome several of the earlier mentioned drawbacks in a DPGA. The memory technology is based on a hybrid-process of CMOS and crystalline-IGZO, with IGZO material stacked on top of the CMOS to save area; further, it has an extremely low off-state power which reduces off-state leakage and is used to create small, non-volatile memory cells.[7]

In this thesis a way to enable dynamic reconfiguration in a CAAC-IGZO-based MC-FPGA is presented. A routing switch is presented and implemented to solve a problem in a reference design relating to boosting on the routing switches' configuration memories. The proposed routing switch is non-volatile and can reduce area by about 38%, and increase performance by 37% at a driving voltage of 1.5V compared to a SRAM-based routing switch.
Abstract

Dynamically reconfigurable FPGAs was described in the 1990’s by Bolotski [1], Tau [2], DeHon[3, 4, 5] and Trimberger [6]. The idea was to expand the FPGA’s function space temporally instead of spatially, and in doing so allowing reuse of the FPGA's functional resources in time, increasing the utilization rate of the functional resources.

Many DPGA designs today are based on the "Time-Multiplexed FPGA" that Trimberger et al. described in 1997 now more commonly called Multi-Context FPGA, in which memory bits are added to every configuration memory to create configuration contexts that the FPGA can switch between. The dominating memory technology used in FPGAs and DPGAs is SRAM; a volatile memory technology that uses a relatively large area and also have an excessive power consumption. Because of the increase of configuration bits in DPGAs, the SRAM memory drawbacks imposes larger effects on its design.

In recent years new memory technologies have been implemented in a broad range of applications, out of which DPGAs are one. Among these technologies, implementation with crystalline IGZO FETs have been argued to overcome several of the earlier mentioned drawbacks in a DPGA. The memory technology is based on a hybrid-process of CMOS and crystalline-IGZO, with IGZO material stacked on top of the CMOS to save area; further, it has an extremely low off-state power which reduces off-state leakage and is used to create small, non-volatile memory cells.[7]

In this thesis a way to enable dynamic reconfiguration in a CAAC-IGZO-based MC-FPGA is presented. A routing switch is presented and implemented to solve a problem in a reference design relating to boosting on the routing switches’ configuration memories. The proposed routing switch is non-volatile and can reduce area by about 38%, and increase performance by 37% at a driving voltage of 1.5V compared to a SRAM-based routing switch.
Abstract

[26/09/14 - Kanagawa, Japan]

I wish to express my great appreciation to all the various people around me that have made this project and internship at SEL successful.

I want to start by expressing my gratitude to Shunpei Yamazaki who have offered an incredibly generous internship and also to Edward Fleetwood and Sweden Japan Foundation that helped out with all the matters before the internship.

I very sincerely want to thank my mentor and supervisor at SEL, Mr. Takeshi Aoki (I could not have been assigned anyone better) who have helped me both at work and outside, answering my questions and making sure I am alright. Further, I want to thank my advising professor at Linköping University Atila Alvandpour who have trusted in me doing well and helped out when needed.

Finally I would also like to thank the CAD3 group members who I have worked together with, Ms. Mika Tatsumi and Ms. Yoko Otake who have gone to great lengths to make all the bureaucracy related to studying/working in Japan go smoothly, all the members of the circle activities (basketball, running and bicycling) with whom I have had a really great time, Ms. Yukiko Tabata and her family who I truly felt at home with, Ms. Yuriko Yano who was a great guide and company in visiting the Tsunami-struck Tohoku-region and all the other SEL members that have made my time here in Japan very precious.

SELの皆様、どうもありがとうございました。
機会があれば、また会いましょう！
Contents

Notation ix

1 Introduction 1
   1.1 Purpose and Goal ............................................. 2
   1.2 About SEL ...................................................... 2
   1.3 Demarcation .................................................... 2
   1.4 Publications and Patents .................................... 2
   1.5 Thesis Outline ............................................... 3

2 Theory 5
   2.1 Semiconductor technology .................................... 5
      2.1.1 CAAC-IGZO ................................................. 5
      2.1.2 Crystalline IGZO-CMOS Hybrid-process .................. 6
   2.2 Field Programmable Gate Arrays .......................... 7
      2.2.1 Logic Element ............................................ 7
      2.2.2 Routing Switch .......................................... 8
      2.2.3 Configuration Memories ................................... 8
      2.2.4 Dynamically Programmable Gate Arrays ............... 10
   2.3 Capacitive Coupling and Boosting Effects ............... 11
   2.4 Threshold levels, Threshold Voltage Drop and Effects of Boosting on a Routing Switch .................................................. 13

3 Previous Work and Where it Faltered 15
   3.1 Previous FPGA and its Components .......................... 15
      3.1.1 Logic element ............................................ 16
      3.1.2 Routing Switch .......................................... 17
      3.1.3 FPGA routing ............................................. 17
   3.2 Measurements and Simulations to Visualize Error after DR .... 17
      3.2.1 Simulation of Schematics ............................... 17
      3.2.2 Chip Measurements ...................................... 20

4 Methods and Workflow 23
   4.1 Workflow .................................................... 23
4.1.1 LSI design and Verification Tools ........................................... 24
4.2 Main Motivations and Design Factors ........................................ 25
4.3 Test Models ........................................................................... 26
  4.3.1 Ring Oscillator with Routing Switch ..................................... 26
  4.3.2 MC-FPGA ................................................................. 29
4.4 Measurements ........................................................................ 30
  4.4.1 Measurements to verify basic functionality ............................ 30
  4.4.2 Performance Measurements ............................................... 30
  4.4.3 Area Measurements ......................................................... 30
  4.4.4 Energy Usage .................................................................. 31
  4.4.5 Measurements on the produced chip .................................... 31

5 Implementation ........................................................................... 33
  5.1 Routing Switch Implementation .............................................. 33
    5.1.1 Reference switches ......................................................... 34
  5.2 Inv-RS ................................................................................ 34
  5.3 Inv-Mux-RS ......................................................................... 36
    5.3.1 53-stage RO layouts ....................................................... 37
    5.3.2 Produced TEG ............................................................... 38

6 Results and Discussion ................................................................. 45
  6.1 Basic behavior ...................................................................... 45
    6.1.1 Boosting Effect and its Relation to the Memory Size ............ 45
    6.1.2 Even Execution ............................................................ 45
    6.1.3 Dynamic Reconfiguration ............................................... 47
  6.2 Performance .......................................................................... 48
    6.2.1 Comparison of maximum frequency in MC-FPGA ............ 50
  6.3 Area usage ........................................................................... 52
  6.4 Power and Energy Usage ....................................................... 53
  6.5 Chip measurements on TEG ................................................... 54

7 Conclusions ............................................................................... 55
  7.1 Future Work .......................................................................... 55

Bibliography .................................................................................. 57
## Notation

### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAAC</td>
<td>C Axis Aligned Crystal</td>
</tr>
<tr>
<td>IGZO</td>
<td>Indium Gallium Zinc Oxide</td>
</tr>
<tr>
<td>OS</td>
<td>Oxide Semiconductor</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Array</td>
</tr>
<tr>
<td>DPGA</td>
<td>Dynamically Programmable Gate Array</td>
</tr>
<tr>
<td>ASIC</td>
<td>Application Specific Integrated Circuit</td>
</tr>
<tr>
<td>MC</td>
<td>Multiple Context</td>
</tr>
<tr>
<td>DR</td>
<td>Dynamic Reconfiguration</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>LE</td>
<td>Logic Element</td>
</tr>
<tr>
<td>RS</td>
<td>Routing Switch</td>
</tr>
<tr>
<td>CM</td>
<td>Configuration Memory</td>
</tr>
<tr>
<td>TEG</td>
<td>Test-Element-Group</td>
</tr>
</tbody>
</table>
Among existing LSI families, the programmable logic device family is particularly interesting due to its flexible nature and fast development time. Recently, they are becoming more interesting with research enabling swift partial- or full run-time reconfiguration of the device to reuse existing hardware, achieving greater function space in time[4, 6]. This is of course intriguing when discussing devices handling different sequential tasks with high performance requirements. There a dynamic FPGA could handle the sequential tasks without demanding the large area that conventional FPGA’s need, which would shorten interconnect length with better performance as result, and by being re-programmable also gives way for cheap circuit improvements and changes compared to the much more expensive to develop ASIC. However, a major drawback and the major task at hand concerning today’s reconfigurable FPGAs is their high power consumption and large interconnect area usage. The interconnects in the common MC-FPGAs includes large wire and routing switch arrays, where the routing switches consists of pass-gates and SRAM memories. SRAM memories are volatile and uses an excess of power and area. A study from 2003 [8], showed that the SRAM configuration memories in a conventional FPGA stood for as much as 38% of the total static leakage. To deal with the drawbacks of SRAM, several new non-volatile memory technologies have been emerging in recent years for use in FPGA devices. One is crystalline IGZO, an oxide semiconductor (OS) that can be used to reduce area and off-state current leakage while still showing good performance.
1.1 Purpose and Goal

The goal of this thesis is to enable stable dynamic reconfiguration in a non-volatile MC-FPGA based on a hybrid-process of crystalline IGZO and CMOS, by changing components in a MC-FPGA device previously developed at SEL [9, 7] that shows differentiating results caused by voltage boosting effects in the routing switches.

1.2 About SEL

The work of this thesis took place at SEL (short for Semiconductor Energy Laboratory Co. Ltd.) in Atsugi, Japan during an internship between Feb 18 - Oct 3. SEL is a research and development company that conducts research in a broad spectrum of areas, from flexible batteries and displays to LSI applications with novel materials. SEL do not produce any products, instead they lease their patents and technology to other companies. The research is executed similarly to academic research, and SEL produces aside from patents several research and conference papers every year.

1.3 Demarcation

This thesis deals mainly with the proposed solution and evaluation of it. Limitations in both time and in chip area made some measurements that could have improved the result-section impossible to perform and is left for future work.

1.4 Publications and Patents

The thesis project resulted in the publications below.


1.5 Thesis Outline

The report has been divided into the following chapters.

Chapter 2 Describes theory and important concepts to explain the reasons for chosen implementation and results in the thesis.

Chapter 3 A description of the former FPGA which this project is based on is given. The chapter also describes the problems encountered in the former design that disabled it from performing correct dynamic reconfiguration.

Chapter 4 Describes the project work-flow and the methods with which the dynamically reconfigurable MC-FPGA was realized.

Chapter 5 Explains the routing switch implementation.

Chapter 6 Results from simulations and measurements are presented and explained.

Chapter 7 Finally, conclusions drawn from the project and for future work is retold in chapter.
This chapter should provide a relevant theoretical background so that the reader can understand the purpose and the results in this thesis, the reader is expected to have some prior knowledge in basic electrical engineering. In section 2.1 the semiconductor technology, with focus on CAAC-IGZO which have been used in key parts of the project is introduced, and in section 2.2 a general description of FPGAs is given. The terminologies of boosting are explained in section 2.3, and Threshold Voltage and voltage drops are explained in section 2.4.

2.1 Semiconductor technology

Semiconductor development began much earlier than with the invention of the transistor in late 1947 [10]; however, the transistor can be seen as the spark that ignited the flame of our digital age. The most common semiconductor technology, MOSFET, have been a vital part in this master thesis, however special focus will be directed to the second semiconductor technology used, Crystalline IGZO also called CAAC-IGZO which is one of SEL’s main materials for LSI research.

2.1.1 CAAC-IGZO

The first findings of the CAAC-IGZO material was made by Noboru Kimizuka and Takahiko Mohri in 1985 when they successfully synthesized a crystal from the homologous compound given by the general formula \( \text{InGaO}_3(\text{ZnO})_m \) [11]. In 2009 researchers from SEL managed to develop a thin film of the CAAC-IGZO material. The name \( c \)-axis-aligned-crystal comes from the material’s hexagonal crystal structure along its \( c \)-axis, as can be seen in figure 2.1a-2.1b. Transistors made from the CAAC-IGZO material display an incredibly low leakage current on the scale of \( \text{yA} \left(10^{-24}\right) \) which is far lower than for any other transistor reported.
Comparisons of the $I_D-V_G$ characteristics for the CAAC-IGZO FET and an nMOS FET can be seen in figure 2.2 [12]. In the figure the measurement tool’s measurement limit was larger than the leakage current of the CAAC-IGZO FET which is why the results from simulation and measurement appears different for small $V_g$. However, the leakage current was successfully measured and reported by Sekine et al. in 2011 [13].

![Crystal structure](image1.png) ![Alignment along the c-axis](image2.png)

**Figure 2.1:** Figures showing physical characteristics of the CAAC-IGZO material.

![CAAC-IGZO $I_d-V_g$ characteristics](image3.png) ![NMOS $I_d-V_g$ characteristics](image4.png)

**Figure 2.2:** $I_d-V_g$ characteristics for CAAC-IGZO and NMOS. The blue line represents the measurement results and the red the simulation. The measurement limit of the tool used is $10e-14 S$

### 2.1.2 Crystalline IGZO-CMOS Hybrid-process

The LSI applications that are developed by SEL use a so-called hybrid-process technology where CAAC-IGZO FETs are stacked on top of MOSFETs as in figure 2.3. This enables a device using it to utilize the stability of CMOS design
2.2 Field Programmable Gate Arrays

Implementing a function by custom ASIC is costly both concerning development time and actual cost. However, ASICs have good performance, use little energy and take up small space on a chip. Implementing and executing the same task on a general purpose processor shorten development time significantly, although performance is bad and energy usage high. FPGAs on the other hand, are programmable hardware that can be electrically programmed to a digital circuit using a hardware description language (HDL) like VHDL or verilog. The hardware consists of LEs that can be programmed into simple logic functions which are placed in a type of mesh (figure 2.4a). These cells are connected to each other by RSs to form more complex circuits and are then connected to appropriate outputs (as the example in figure 2.4b). FPGAs have much better performance than the general purpose processor, they have a short development time, cost less to implement and if needed, most of the existing FPGAs can be reconfigured which gives designers the ability to improve and change the design. [29, 30, 3]

2.2.1 Logic Element

The most basic building block in an FPGA is the LE. There can exist a variety of special function LEs in large FPGA devices, however, for this thesis only the most basic is needed. To enable the FPGA to realize a large amount of complex circuits, the LEs needs to efficiently implement several basic logic functions. A typical LE design can be seen in figure 2.5. This block implements functions such as 2-4
2.2.2 Routing Switch

The routing switches perform a very simple function, connecting (or disconnecting) one LE to another, or to an I/O. They are nonetheless vital in the FPGA and the routing algorithm can make a huge difference in final performance. As seen in figure 2.6 they consist of a simple pass-gate and a configuration memory. In the configuration memory, which only contain one bit in a conventional FPGA, the information whether two modules are to connect or not is written.

2.2.3 Configuration Memories

To configure the FPGA a programming technology is used. Statical technologies such as fuse and anti-fuse have been used (anti-fuse is still used commercially), however, nowadays the dominating FPGA programming technology is configuration by SRAM memories. SRAM memories are reconfigurable, and since they are CMOS circuits they show good switching properties. Although, there are
2.2 Field Programmable Gate Arrays

**Figure 2.6:** A routing switch made with a configuration memory and a pass-gate, depending on the information in the CM the RS will either connect or disconnect for example an LE with an LE or LE with an I/O.

several problems concerning SRAM such as large area usage, high energy consumption and volatility that leaves room for new memory technologies such as CAAC-IGZO based memories, MRAM[32], RRAM[33] and FLASH[34]. These are all non-volatile technologies that address area usage by being stackable through different processes and by using less components than SRAM. In this thesis, attention will be drawn to the improvements that can be made by using CAAC-IGZO based memories in the FPGA components compared to SRAM based, and what drawbacks we are to expect. A CAAC-IGZO based RS uses an IGZO memory like the one in figure 2.7a made with a CAAC-IGZO FET and a capacitance for the memory, to form the switch the memory node is connected to an nMOS pass-gate to connect input with output as in figure 2.6. The capacitance is constructed with the CAAC-IGZO FETs gate and source/drain layers and creates similarly to a FLASH memories a floating gate (figure 2.8). A RS made with SRAM memories is built similarly and is using a memory like the one in figure 2.7b. The transistor count of the SRAM based RS is 6 silicon transistors while the CAAC-IGZO only needs one silicon transistor, one IGZO-transistor and a capacitance.

**Figure 2.7:** Two types of configuration memories designed with different programming technologies.
Figure 2.8: Comparison of a capacitance based floating node memory and a flash memory that has a floating gate.

2.2.4 Dynamically Programmable Gate Arrays

Conventional FPGAs execute a static and spatial computation that is configured from an off-chip memory. Off-chip memories come with large delays and reconfiguration of the FPGA will usually take a long time, longer than would be necessary for usage in a dynamically changing reconfiguration. The amount of LEs in the FPGA design can therefore be seen as a measurement of how much computation capacity the FPGA has. A dynamically reconfigurable FPGA on the other hand, that can change configuration partially over time and for which configurations are loaded from an on-chip memory have both a spatial and a temporal function space which can increase the LE utilization efficiency. One of the first dynamically reconfigurable FPGAs described, called dynamically programmable gate array (DPGA) [1, 3], was a device combined from an FPGA and a SIMD-processor \(^1\) to create multiple configuration contexts that allows for performance improvements in general-purpose machines for specific tasks. That is, instead of using fixed-function ASIC blocks limited to one function only, to use a dynamic-function blocks with decreased performance loss compared to a general purpose processors. This type of FPGA, with multiple contexts, was again in 1997 described by Trimberger et al.; however, instead of combining the FPGA with a SIMD processor the contexts were created by introducing multiple-context CMs in the design’s RSs and LEs (see figure 2.9)[6].

Accordingly, an FPGA with \(n\) contexts has configuration memories with \(n\) bits instead of one, plus a configuration selector to choose the active configuration. The FPGA then has one active context at a time, and the inactive contexts are allowed to be reconfigured from an off-chip memory to the on-chip CMs which allows for a large temporal function space.

For example, say that there are \(n\) sequential tasks to run on an FPGA figure 2.10a. The conventional FPGA in this example can fit four task configurations at a time (for simplicity’s sake all tasks needs an equal amount of LEs), to fit all the \(n\) tasks

\(^1\)SIMD stands for Single Instruction Multiple Data, and is a processor architecture that on several processing units process the same instruction on multiple data at once [35, p. 630]
2.3 Capacitive Coupling and Boosting Effects

Capacitive coupling can be intended in an analog circuit design, however it can also be a side-effect of a design choice. It is important to be aware of potential capacitive couplings and how they will affect the performance of the circuit. Consider the following simplified example for the circuit in figure 2.11. Here we can see that a parasitic capacitance, $C_p$, couples node $n_0$ to node $n_1$. In a situation where $n_0$ first is charged to logic 1 by applying voltage $V_H$ to both WL and BL while IN is kept at logic 0 by applying $V_L$ as in figure 2.12a the circuit can be represented by figure 2.12b. The voltage over $C_p$ is

$$|\Delta V| = |n_0 - n_1| = 2.5V,$$

(2.2)

when the capacitance is charged. Any voltage drop over $M_0$ is in this example ignored.

If $M_0$ after this is turned off (figure 2.13a), the two cases in figure 2.13b and

---

**Figure 2.9:** Spatial function space can be seen along the x-y plane and is increased by increasing the number of LEs. Temporal function space can be seen along x-axis and is increased by adding CMs.
2.13c can occur. For figure 2.13b, the voltage on \( n_0 \) will change accordingly with equation 2.3.

\[
n_0 = |\Delta V| + n_1 = 2.5 + 2.5V = 5V.
\]  

(2.3)

Meanwhile in the case of figure 2.13c \( n_0 \) would be as in equation 2.4.

\[
n_0 = |\Delta V| + n_1 = 2.5 + 0V = 2.5V
\]  

(2.4)

Reversely, if the signal \( \text{IN} \) in figure 2.12a had been \( V_H \) when node \( n_0 \) is charged to \( V_H \), then the situation would be as in figure 2.14a-2.14b, and voltage over \( C_p \)
2.4 Threshold levels, Threshold Voltage Drop and Effects of Boosting on a Routing Switch

![Diagram](image1)

**Figure 2.12:** (a) Writing $V_H$ to the node $n_0$ while $IN$ is $V_L$. (b) Simplified circuit representation of scenario in (a), ignoring voltage drop over the transistors.

![Diagram](image2)

**Figure 2.13:** (a) Upper transistor is turned off leaving $n_0$ floating. (b) and (c) are simplified circuit representations of when $IN$ is high and low respectively.

would be,

$$|\Delta V| = |n_0 - n_1| = 2.5 - 2.5V = 0V. \quad (2.5)$$

Which would lead to either a normal voltage level or a decreased level on $n_0$ when $M_0$ is turned off and $V_H$ (equation 2.6) or $V_L$ (equation 2.7) is applied to $IN$ respectively.

$$n_0 = |\Delta V| + n_1 = 2.5 + 0V = 2.5V \quad (2.6)$$

$$n_0 = |\Delta V| + n_1 = 0V + 0V = 0V \quad (2.7)$$

2.4 Threshold levels, Threshold Voltage Drop and Effects of Boosting on a Routing Switch

When designing circuits other than CMOS a factor that has to be taken into consideration is that nMOS cannot pass a strong one, and similarly pMOS cannot pass a strong zero due to the physical properties of the transistors.
Figure 2.14: (a) Writing $V_H$ to node $n_0$ while IN is $V_H$. (b) Simplified circuit representation of (a).

In figure 2.15 a common representation of an nMOS transistor is shown. The threshold voltage, $V_T$, is the voltage level on $V_{GS}$ where strong inversion occurs and is expressed as in equation 2.8.

$$V_T = V_{T0} + \gamma(\sqrt{(-2)\theta_F + V_{SB}}) - \sqrt{2\theta_F}$$  

(2.8)

Figure 2.15: Model of a typical transistor and its related voltages.

With these regions in mind it is easier to explain why the nMOS transistor yields a strong 0 and a weak 1 when active. An active nMOS transistor where the voltage $V_H$ (logic 1) is applied to $V_G$ has a conductive path between source and drain as long as $V_{GS} > V_T$. However, as soon as the voltage level $V_S > V_G - V_T$ we are back in cutoff region where the conductive path is cutoff and $V_S$ ceases to increase its voltage level.

---

2 nMOS is a MOSFET type transistor, its positions for source and drain is decided by the node with lowest potential which is always the source for nMOS.

3 Strong inversion for a nMOS transistor is when its channel, made from weakly doped p-type silicon, inverts to n-type silicon and a conductive path between source and drain is formed. For explanations of p-doped and n-doped silicon refer to appendix AAA.
To motivate the RS implementation later described in chapter 5, a description of previous work on a dynamically reconfigurable MC-FPGA using CAAC-IGZO based programming technology and why it could not perform a stable DR is described in this section.

3.1 Previous FPGA and its Components

As presented by both Kozuma [9] and Okamoto [7], MC-FPGA prototypes with two contexts, containing 20 LEs and 20 I/Os was developed and manufactured at SEL (figure 3.1). The prototype FPGA could successfully perform initial configuration of both contexts, and also switch context in one clock cycle [7]. In [21] an FPGA implemented with CAAC-IGZO technology could operate without refresh of the memories consecutively for more than eight days without performance loss.
Previous Work and Where it Faltered

3.1.1 Logic element

The FPGA’s LEs (figure 3.2) include 4-input LUTs, carry-chain and also support for two configuration contexts. In the figure the red squares represent where configuration contexts are held. Partial results that needs to be used even after having switched context can be saved by the Save/Load signals in the nonvolatile register (NV Reg).

Figure 3.1: The previously developed 2-context FPGA where LEs have been highlighted in green and RS areas in orange.

Figure 3.2: Logic element for the CAAC-IGZO MC-FPGA.
3.1.2 Routing Switch

The RSs (figure 3.3) similarly to the LEs contain two contexts. In [9] it was shown that the RS displayed excellent switching properties because of boosting effects caused by parasitic capacitances that created a capacitive coupling between the RS’s input and its memory pass-gate’s gate, boosting the voltage level on the floating node memory. However, performing DR on an inactive context will affect the boosting differently depending on the task that the active context is performing because they share input (IN). This causes uneven results, and depending on the size of the memory capacitance, the FPGA could even fail to execute correctly.

![Figure 3.3: Reference RS using CAAC-IGZO-based memory to store configurations.](image)

3.1.3 FPGA routing

The routing of the LEs, RSs and I/Os is visualized in figure 3.4. Every logic element connects to several RSs which is why keeping the RSs as small as possible is important. The larger the FPGA is made (that is the more LEs it is implemented with) makes the routing increasingly complicated and area demanding.

3.2 Measurements and Simulations to Visualize Error after DR

To visualize when error occurs, and how the boosting affects the performance of the RSs, one chip measurement of the manufactured FPGA and a SPICE simulation of the FPGA on schematics level are explained below.

3.2.1 Simulation of Schematics

The task execution in figure 3.5 was configured and performed on the FPGA. First an initial configuration takes place where context 0 is configured to a shifter circuit as the one shown in figure 3.6a. Then context 1 is dynamically reconfigured...
Figure 3.4: FPGA routing, * indicates that it includes an array, for example 0* means an array from 0-9.

to a ring oscillator circuit in (3.6b). In Simulation A below, the shifter uses a pulse-signal as input and shifts it from input 1 on I/O1 to input 6 on I/O6 as in figure 3.6c. This generates a low input to the RSs during DR of context 1. In Simulation B, the shifter uses a step-signal (see figure 3.6d) that instead generates a high input signal to the RSs during DR.

Figure 3.5: Tasks executed to show characteristics of DR in previously developed FPGA.

Simulation A in figure 3.7 displays the output signals from I/O0-I/O6 in which we can see the pulse shifting from I/O0 to I/O6 after which the signals all go low, we can also see the RO’s oscillating output from I/O6 after the context is switched. Further the memory signal from context 1 of one of the intermediate RSs from the RO is shown in red. Here we can clearly see that the boosting raises
3.2 Measurements and Simulations to Visualize Error after DR

The two test circuits that the FPGA was configured to test its functionality.

Figure 3.6: The two test circuits that the FPGA was configured to test its functionality.

the voltage on every high input and the recedes on low. The context signal shows when the context switch occurs, and configuration data shows when the DR’s data transfer starts.

Simulation B in figure 3.8 similarly displays I/O0-I/O6 where instead of a pulse we now use a step signal inducing high input signals to the RSs. Just like A, B shows when data starts transferring during DR, the context signal, and most importantly the memory voltage level that instead of increasing the voltage during high input decreases on every low input, degrading the RS’s performance. This is most clear when comparing the output frequency from the RS in Simulation A and B, where the oscillating frequency on I/O6 is 21MHz in A and 13MHz in B.

Figure 3.7: Simulation A: The FPGA is fed with a pulse signal during DR to its active shifter configuration. The pulse creates low input signals to RSs in the routing while the FPGA is dynamically reconfiguring. This causes boosting effects which improve the switching and gives a faster output frequency from the RO.
Figure 3.8: Simulation B: Here the FPGA is fed a step signal during a shifter circuit making the input signals to the LEs high. The signal is still high during DR which causes the CMs to have reverse boosting effect when they are used. This causes the degraded output frequency during the RO task.

3.2.2 Chip Measurements

In similar procedure as for the schematic simulations measurements using the same configurations were executed on an actual chip with memory capacitance size 184fF. Measurement A (using a pulse-signal as input like Simulation A) in these measurements can be seen in figure 3.9 and Measurement B (similarly to Simulation B, using a step-signal to produce high input to the RSs) in figure 3.10.

Figure 3.9: Output visualized on an oscilloscope from measurements on the manufactured FPGA in the setting of Simulation A. I/O6 in RO-mode have an output frequency of about 10.8MHz.
Figure 3.10: Output visualized on an oscilloscope from measurements on the manufactured FPGA in the setting of Simulation B. I/O6 in RO-mode have a frequency of about 8.5MHz.
This chapter describes first the workflow of the project, in what order tasks were performed and also the production flow of VLSI circuits. Section 4.2 describes the motivations behind the models and methods used throughout this project. Section 4.3 describes models used to evaluate the RS design. The last section (4.4) describes the measurements that was used for the evaluation of the proposed solution.

### 4.1 Workflow

The general work-flow of the thesis project is represented by the chart in figure 4.1. In the first stage of the project the MC-FPGA described in chapter 3 was studied and the problems regarding reconfiguration was visualized in both measurements and simulations. The study led to design ideas on how to enable stable dynamic reconfiguration in the FPGA. These ideas were formalized and implemented both in test circuits and in the FPGA design. In the last part of this project one solution was evaluated extensively.

![Figure 4.1: Project Workflow.](image)
4.1.1 LSI design and Verification Tools

Design of large circuits is usually divided into three main abstraction levels. RTL (Register Transfer Level), Gate Level and Transistor Level as described in the sections below. Verification by simulation and measurements are extremely important because of the high production costs involved in chip design.

Figure 4.2: Starting with functional description, translated to verilog code until measurements on a produced chip, the flow to validate a LSI solution to enable DR in a MC-FPGA.

### RTL

RTL (Register Transfer Level) is a high-level abstraction of a circuit design used to simulate high-level behavior, such as signal flows. RTL designs are usually partitioned into blocks that describe a specific units behavior. From these RTL blocks a more detailed description can be produced on lower abstraction levels. In this project the RTL design was written in verilog code and simulated using nc-verilog simulator.

### Gate

Gate level is an abstraction level where the logic function of the blocks made for the RTL design are described. To verify the behavior of these they are simulated with their material parameters in a SPICE (Simulation Program with Integrated Circuit Emphasis) simulator. In this project SmartSpice was used to verify behavior of small circuit designs, and FastSPICE was used to verify the behavior of the whole FPGA chip put together.
4.2 Main Motivations and Design Factors

For the CAAC-IGZO FET technology to have impact it needs to be able to compete with the other existing research technologies in the field. Concerning FPGAs there are several other competitors, both academic and industrial, that launches similar ideas but with other materials. So far, the SRAM programming technology have by far been dominating the production due to the already existing manufacturing process. Hence, for a new technology to succeed it need to address the following difficulties.

**Power Consumption:** A major problem with CMOS-/SRAM based RS solutions in MC-FPGAs is that they consume great amounts of power due to the SRAM memories and also have major static leakage issues [38]. In an evaluation of leakage power in a 90nm FPGA the SRAM CMs consumed 38% of the total leakage power [8]. The CAAC-IGZO technology can potentially make a huge difference with its extremely low leakage power. And can, when used as memory technology, provide a non-volatile, low-leakage solution.

**Speed/Switching Characteristics:** One of the main reasons why CMOS is used is that it displays very good speed and switching characteristics. CAAC-IGZO can achieve equal or even better speed and switching properties as SRAM due to the boosted memory (as explained in 2.3) however this is also the reason why it could not handle DR. A solution including CAAC-IGZO must overcome this problem while still achieving good switching speed. The boosting is needed to improve the voltage level on the charged CM (as the one in figure 3.3). Due to the threshold voltage, without the boosting effects voltage on the memory would be,

\[ V_{\text{memory}} \leq V_H - V_T. \]  

\hspace{1cm} (4.1)

This would yield a lower switching capability on the RS pass-gate compared to for example the SRAM that uses CMOS and would therefore have the
Methods and Workflow

voltage level of $V_H$ on the memory node. If on the other hand boosting is achieved as was explained in 2.3, then a voltage level greater than $V_H$ could be gained on the memory node that would prevent a threshold voltage drop on the pass-gate transistor and could also help preventing a voltage drop over the pass-gate.

**Area usage:** SRAM includes at least 6 transistors which is a lot compared to many of the new technologies. In a device such as the MC-FPGA that uses a great amount of configuration memories the size of the memory is a big factor to the final chip size. The fact that CAAC-IGZO FETs are stackable makes it even more attractive. A CAAC-IGZO- / MOSFET hybrid-process design area is ideally only dependent on how many MOSFET there are in the design because the full CAAC-IGZO design could be stacked fully on top of the MOSFETs.

**Manufacturing Process:** As stated above, the manufacturing process is a major factor to which technologies will come to dominate future LSI technology. A technology is only as good as the cost of its implementation with industrial glasses on. However, a manufacturer friendly technology might pose a good next option.

Further, the goal is to enable runtime reconfiguration of an inactive context without interrupting the active task. That means that a solution where we stop execution of the FPGA to send a low signal to the reference RS in order to overcome the problem of reverse boosting would not be acceptable.

### 4.3 Test Models

The models used throughout the project are presented in this section.

#### 4.3.1 Ring Oscillator with Routing Switch

To get a general idea of the proposed RS’s behavior, performance, and power-consumption compared to other RSs, a RO-model was used.

![n-stage Ring Oscillator](image)

*Figure 4.3: n-stage Ring Oscillator.*
4.3 Test Models

**Inv-Mux-RS-RO (IMR)**

An inverter-multiplexer-RS(Inv-Mux-RS)-RO model (see the RO-element figure 4.4) is primarily used to compare performance of the referenced CAAC-IGZO-RS, the proposed CAAC-IGZO-RS and an SRAM-RS. The multiplexer is needed to simulate FPGA behavior in the referenced switch. A low or a high input is chosen with the multiplexer to the RS during DR of context\textsubscript{1} to create boosting or reverse boosting on the reference switch. A signal chart to simulate DR is shown in figure 4.5.

![Diagram of RO element consisting of a multiplexer a routing switch and an inverter.](image)

**Figure 4.4:** RO element consisting of a multiplexer a routing switch and an inverter.

<table>
<thead>
<tr>
<th>Context\textsubscript{1} activity</th>
<th>idle</th>
<th>DR</th>
<th>active</th>
<th>idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>WL\textsubscript{0}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WL\textsubscript{1}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CXT\textsubscript{0}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CXT\textsubscript{1}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTR\textsubscript{0}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTR\textsubscript{1}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4.5:** Signal chart for DR in the RS with a “pull-down”-node.
Inv-RS-RO (IR)

The Inv-Mux-RS-RO seemed like a bad idea for power evaluation and a model excluding the multiplexer was created (see figure 4.7) with a RO element consisting of an inverter and a RS. Later we also realized that this model would be important for performance measurements as well. This model however, cannot be used with the reference CAAC-IGZO-RS because input-signal state cannot be chosen during DR. A signal scheme used with the test-bench can be seen in figure 4.6.

Figure 4.6: Signal chart for DR in the RS with a “pull-down“-node.

Figure 4.7: RO element consisting of an inverter and a RS.
4.3.2 MC-FPGA

The MC-FPGA model in figure 4.8 is used as a high level design of the MC-FPGA. The modules that the FPGA consist of are explained below. The MC-FPGA design and structure is created before by SEL. By substituting/including the proposed solution, behavior of the proposed solution in the MC-FPGA can be simulated.

**Figure 4.8:** High-Level block overview of the FPGA design.

**Clock Generators**

There are two clock generators in the design, the configuration controller clock generator is used for the incoming data clock and produces an internal data clock signal for configuration use. The second is the clock generator that is used for the system timing.

**Configuration Controller**

The configuration controller is the global controller of the system and takes control signals from the outside to time them and produce the internal control signals. It is using Altera’s passive serial configuration [39].

**Bit- and Word Driver**

The bit and word drivers drives the signals that enables reconfiguration of the configuration memories in the FPGA. Configuration data is written by the word driver opening the word lines that are connecting to the configuration memories sequentially, and while one word-line is open relevant data is sent on the bit-lines and stored in the configuration memories. When the data has been stored the the word driver opens the next word-line and again data is sent on the bit-line.
Logic Element Array

The logic element array is simply put an array of the logic elements and their connecting routing switches.

I/O Array

Is similarly to the Logic Element array an array of the programmable I/Os and their routing switches.

4.4 Measurements

Measurements that was used to verify behavior and evaluate the proposed RS are described below.

4.4.1 Measurements to verify basic functionality

To decide if the proposed solution succeeds, verification whether it has uniform behavior irrespective of if the signal input to the RS is high or low during DR is needed. Such can be done both in the single switch model where signals representing such flows can be fed to the switch, or in the Inv-Mux-RS model where this can be verified by setting the DR-signal high.

4.4.2 Performance Measurements

To get an idea of the performance of a RS, simulation where the RS is tested in either of above RO circuits can be performed. A comparison of the average period $T_{avg}$ from a RO with OS-RSs and SRAM-RSs can act as a measurement of how well the OS-RS performs compared to the commercially accepted SRAM. In the equation below, $T_{avg}$ is calculated from the period between the $m^{th}$ and $n^{th}$ rising edge, written $T_{m-to-n} = T_{m} - T_{n}$, divided by n minus m.

$$T_{avg} = \frac{1}{n-m} T_{m-to-n}, n > m.$$  \hspace{1cm} (4.2)

The average period in a RO is a good measurement because it is directly related to the propagation delay through one RO-element stage, where the propagation time, $t_{p}$, can be derived from the RO-period with,

$$t_{p} = \frac{T}{2N} \hspace{1cm} (4.3)$$

here N is the number of stages in the RO, and 2 is needed since a full period have both a transition from low-to-high and from high-to-low.

4.4.3 Area Measurements

A very simple area measurement was performed by measuring the smallest rectangular area that can encompass the whole transistor layout of the RS. Such a measurement is of course affected of how well the designer routed the design
and not based on an analytic model, however in this thesis it can be used as a quick estimate the size of especially the CAAC-IGZO-based circuits.

### 4.4.4 Energy Usage

The energy usage over a period can be calculated with:

\[
E = \int_{T_0}^{T_1} V_{dd} i(t) dt.
\]

(4.4)

Where \( i(t) \) is the current as a function of time, \( V_{dd} \) the driving voltage and \( T_0 - T_1 \) the time span over which we want to measure the energy used.

### 4.4.5 Measurements on the produced chip

Two of the layouts that was created during this thesis and are described in section 5.3.2 was manufactured. Testing of the chip was done by connecting pins to a pattern-generator that had been prepared with an appropriate signal pattern for the used signals, and voltages was connected to appropriate power-sources. The output from the chip was connected to an oscilloscope and a spectrum analyzer, from which spectrum data could be saved and later plotted.

![Figure 4.9: Pad-frame for the two circuits that was manufactured.](image)

**Output Frequency**

To clearly visualize the frequency a real-time spectrum analyzer was used. An average of 10 peaks was visualized at a time as in figure 4.11 to get a more stable
Figure 4.10: Measurement setup used to test the manufactured TEG.

View of the output frequency. The value from the center of the span was used as average frequency data with the outer sides to show the spectrum it was within.

Figure 4.11: Averaged peak from measurement on 53-stage OS-RS-based RO.
The final implementation is presented in this chapter. There were other ideas before this implementation to solve the problem caused by the boosting was chosen. However, the other designs all had issues either regarding not using the positive boosting, increasing the current leakage, being to unstable, or interrupting the dynamic reconfiguration.

In the following sections the implementation of the routing switch is described and shown in figures. The TEG circuit implementation is also explained.

5.1 Routing Switch Implementation

The final design that was implemented and tested extensively in different types of test-circuits during this project can be seen in figure 5.1. To still achieve good boosting, without adding any transistors that lead to static leakage issues a “pull-down”-node was created on the input of the reference switch (figure 3.3) with a nMOS transistor as pass-gate and an OS transistor for the pull-down mechanism (M₃ and M₄ in figure 5.1). The function of the “pull-down”-node is visualized in the signal chart in figure 5.2 where we can see how during DR transistor M₃ and M₂ turns off for the inactive context and electrically isolates node n₁. Between time t₁ and t₂ when writing to the configuration memory occurs, node n₁ is pulled to ground by the OS transistor M₄. Hence, when the context is active after configuration only positive boosting can occur on the output signal since the environment when DR occurs always is low for all contexts, independent of task execution. In the figure the theoretical effect of boosting on node n₀ is shown in blue.

The RS design was implemented first in schematics with memory capacitances of
5.1 Implementation

Figure 5.1: RS designed to avoid reverse boosting and enable uniform performance on a dynamically reconfigurable MC-FPGA.

<table>
<thead>
<tr>
<th>Context activity</th>
<th>idle</th>
<th>DR</th>
<th>active</th>
</tr>
</thead>
<tbody>
<tr>
<td>WL₀</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n₁</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CXT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n₀</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.2: Signal chart for DR in the RS with a “pull-down”-node.

Size 4fF and 184fF and two contexts. Layout of CAAC-IGZO RSs can be seen in figure 5.3 for memory capacitance size 184fF and figure 5.4 for 4 fF capacitance size.

5.1.1 Reference switches

To evaluate the proposed RS, a SRAM-RS and the RS discussed in [21], in this thesis referred to as reference OS-RS, was used for comparisons. The schematics of each can be seen in figure 5.5 and 5.6.

5.2 Inv-RS

The Inverter-RS circuit described in section 4.3.1 was only implemented as a schematic where a RS of the proposed RS-type, a SRAM-type, and the reference
Figure 5.3: Layout of RS with a memory capacitance of 184fF.

type was implemented with an inverter as The schematics and the layout of the inverter can be seen in figure 5.8.
Another circuit used when testing was an inverter-mux-RS circuit. Two variants were used; in schematics a 5-stage RO based on the INV-MUX-RS as basic block was evaluated; in the layout a 53-stage RO was made and used for measurement purposes. The multiplexer design in both schematics and layout can be seen in figure 5.9.

**Figure 5.4:** Layout of RS with a memory capacitance of 4fF.

### 5.3 Inv-Mux-RS
5.3 Inv-Mux-RS

The layouts that was implemented can be seen in figure 5.10a-5.10d. Out of these only figure 5.10b and 5.10d was actually produced.
Figure 5.7: The RO element created with an inverter, a multiplexer and a RS. If combined repeatedly they can create a type of ring oscillator. (a) shows a version with 4fF capacitance memory, (b) a version with 184fF.

5.3.2 Produced TEG

A chip with the 53-stage small-OS-RO and SRAM-RO circuit was produced. Due to resource and time limitations the two other 53-stage RO circuits could not be
included. Output pads were also too few and power measurements had to be abandoned. In figure 5.11-5.12 the produced chip can be seen. (a) in both figures shows the full TEG layout and (b) is a close-up to show the switch routing.

Figure 5.8: Design of the inverter.
Figure 5.9: Design of the multiplexer.
Figure 5.10: 53-stage RO layouts.
5 Implementation

(a) CAAC-IGZO-based RS in 53-stage IMR

(b) 50x

Figure 5.11
5.3 Inv-Mux-RS

(a) SRAM-based RS in 53-stage IMR

(b) 50x

Figure 5.12
Results and Discussion

The circuit that enables uniform and safe DR in a CAAC-IGZO MC-FPGA has been evaluated through various simulations and measurements. In this chapter the results describing its basic behavior, performance, area usage and energy consumption are presented.

6.1 Basic behavior

This section contains results for basic behavior of the RS, such as; how the boosting effect looks, if the switch works as it is supposed, and how the memory size affects its behavior.

6.1.1 Boosting Effect and its Relation to the Memory Size

The boosting effects was visualized in simulation with smart-spice. In figure 6.1 the voltage level on the memory node, \( n_0 \) is shown for a capacitance of 184fF and 4fF respectively. Figure 6.8 shows average results for minimum value (Min), maximum value (Max) and peak-to-peak value of the boosted signal on node \( n_0 \) for memory sizes of 4fF to 184fF of the RS in the 5stage Inv-Mux-RS RO model.

6.1.2 Even Execution

Simulations of the FPGA circuit that show if and how uniformly the proposed switch executes where performed in this section. The proposed RS was designed to achieve stable results independently of the task executed during DR, which is why *simulation A* and *simulation B* described in section 3.2.1 was performed, but this time with the proposed RS instead of the referenced one. In figure 6.3 the signal flow of the simulations are visible; how the FPGA dynamically recon-
Results and Discussion

(a) Boosting effect on memory node $n_0$ and resulting output voltage in RS with memory capacitance of 184fF

(b) Boosting effect on memory node $n_0$ and resulting output voltage in RS with memory capacitance of 4fF.

Figure 6.1: Memory voltage level showing boosting during a oscillating input signal in a IGZO-RS with a pull-down node.

Figure 6.2: Boosting signal results in a 5-stage Inv-Mux-RS RO for different memory sizes.

figures context 1 to hold the RO circuit. And on the last signal the boosting on a memory node from a RS’s context1 is shown. This node and output[6] are both visible in a zoomed in version in figure 6.4 where simulation A (low input during
6.1 Basic behavior

DR) here named low, and simulation B here named high are compared. As clearly is visible in the result both the memory nodes voltages over time have the same shape, and the output frequency is for both about 18MHz. Comparing this to the same simulation that was performed in chapter 3 the results in this section indicate that the proposed switch has even behavior irrespective of the input during DR.

![Simulation A](image1)

![Simulation B](image2)

**Figure 6.3:** (a) FPGA gate simulation with the proposed 184fF capacitance memory OS-RS for high input during DR. (b) FPGA gate simulation with the proposed 184fF capacitance memory OS-RS for low input during DR.

### 6.1.3 Dynamic Reconfiguration

To confirm that the MC-FPGA can change context properly a simulation is executed where the FPGA can switch between the following three tasks:

**Task 0:** A incremental shifter that on one clock period shifts the incoming signal to the next output. Is given a pulse-signal from the outside.
Results and Discussion

(a) Output signals for high respective low input signal to RS during DR are almost of identical frequency (both of them 18MHz) during simulation in a RO circuit configuration (figure 4.3) with 184fF capacitance memory.

(b) FPGA gate simulation with the proposed 184fF capacitance memory OS-RS for low input during DR.

Figure 6.4: Zoomed in on output[6] and memory node of context[1] between stage 6 and 0 in a 7-stage RO.

Task 1: A decremental shifter that on one clock period shifts the signal to the prior output. Is given a pulse signal from the outside.

Task 2: A divider that divides a signal for the next output. Is given a oscillating signal.

In figure 6.5 results from a simulation where context[0] holds Task 0, and context[1] Task 1 as initial configuration can be seen. First context[0] is executed, then context[1], followed by context[0] again. During the second run of context[0] reconfiguration of context[1] is started (see the data clock that starts ticking after the signal config goes low). After configuration is finished a final switch to context[1] is done, and we can clearly see that context[1] no longer executes a decrement-shifter but a divider. This simulation result, together with the result that boosting is even using the proposed RS suggest that DR is successful.

6.2 Performance

Performance was evaluated using both simulations and measurements on the actual chip. Two types of performance sweeps were performed. One where driving voltage was swept between 1.2V and 2.5V, and one where overdrive voltage was applied to the context-select signal to see if delay could be reduced enough to make the proposed RS faster than a SRAM-RS. In the driving voltage sweep in
Figure 6.5: Dynamic reconfiguration. In this figure simulation dynamic reconfiguration in a 20 LE large MC-FPGA with two contexts have been executed.

Figure 6.6: An overdrive voltage of 0.3V was applied to both the word-line and the context-signal to improve the writing to the capacitance memory and reduce the delay caused by the pass-transistors. Note also that the simulation is done without the use of a multiplexer. Judging from the voltage sweep-results the proposed switch has better performance than the SRAM RS for low voltages making it well suited for low power applications.

As seen in the results in figure 6.7 of when the context-signal is overdriven, the shape of the curve is different if a multiplexer is used in between the elements or if not, even if signal voltages applied are the same. Because of this differentiating behavior it is hard to say which switch has the better performance at for example an overdrive voltage of 0.3 V on the context signal. We can see that the simulation and the measurement results have similar curves for the 53-stage IMR-switches although the measured output frequency is lower than in the simulation. However, there can be no claims of superior performance for any specific setting in general because it is affected by the multiplexer which is not an element used in the MC-FPGA design. Instead it will be a lesson for future work to either do a similar simulation and measurement on a circuit using FPGA-similar circuitry, maybe a RO made with LEs programmed like multiplexers and RS in between the inverters. This to judge if the performance for the IGZO-RS is better for the higher loads and resistances of additional elements of if it is simply the multi-
Results and Discussion

Figure 6.6: Frequency at different driving voltages.

Table 6.1: Output frequency from MC-FPGA [MHz].

<table>
<thead>
<tr>
<th>New Routing Switch</th>
<th>Old Routing Switch</th>
<th>SRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOP SMALL</td>
<td>TOP (H/L)</td>
<td></td>
</tr>
<tr>
<td>17.6</td>
<td>18.6</td>
<td>21.5/24</td>
</tr>
<tr>
<td>23.7</td>
<td>x / 26</td>
<td></td>
</tr>
</tbody>
</table>

6.2.1 Comparison of maximum frequency in MC-FPGA

The same simulations that were done on the TEG could not be performed in the MC-FPGA circuitry as too many signals are sharing voltage source. However, simple simulations without sweep could be performed. The results comparing the output frequency on the RO in simulation A (from section 3.2.1) for the MC-FPGA with the proposed switch (4fF and 184fF), the reference switch (4fF and 184fF) for both reconfiguration on high and low input to RSs, and the SRAM-switch noted in table 6.1. x in the table means failure to execute.
6.2 Performance

(a) Simulation with multiplexer element between the inverter and the RS.

(b) Measurement with multiplexer element between the inverter and the RS.

(c) Simulation without multiplexer.

**Figure 6.7:** Output frequency for different overdrive voltages on the context signal. The performance is affected differently in the two RS of the multiplexer. (a) and (b) show the behavior in simulation and in measurements when a multiplexer is inserted between the inverter and the RS. While (c) show a sweep without the multiplexer.
Results and Discussion

Table 6.2: Area usage of final layout.

<table>
<thead>
<tr>
<th></th>
<th>New Routing Switch</th>
<th>Old Routing Switch</th>
<th>SRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TOP (µm²)</td>
<td>SMALL (µm²)</td>
<td></td>
</tr>
<tr>
<td>2 contexts</td>
<td>947.2</td>
<td>931.5</td>
<td></td>
</tr>
<tr>
<td>1 context</td>
<td>412.7</td>
<td>358.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1080.0</td>
<td>567.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>421.2</td>
<td>344.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>544.5</td>
<td></td>
</tr>
</tbody>
</table>

\(^a\) Includes wires
\(^b\) The routing is designed for the TOP-module and therefore not designed especially for SMALL, which is why it has such large area overhead.

6.3 Area usage

Area of the RS was measured for one context from the layouts in figure 5.3 and 5.4. The results from the measurement is presented in table 6.2. A normalized graph of how their size compare to the SRAM-based RS (measured from the SRAM layout used in figure 5.10d) is presented in figure 6.8.

Figure 6.8: Area comparison with SRAM-RS.
6.4 Power and Energy Usage

One of the major benefits from a CAAC-IGZO-based FPGA that makes it a good option instead of an SRAM-based is the low leakage characteristics that enables us both to make the FPGA non-volatile and also lowers static power consumption drastically. This has been discussed in several earlier papers [21, 9, 13], and the proposed switch does not change that behavior because the inserted pull-down transistor is a CAAC-IGZO and will therefore minimize leakage to ground.

The major change with the proposed switch regarding energy consumption, is the extra write-energy needed by the pull-down node during reconfiguration of a context. We use equation 4.4 to estimate the total write power over the two IGZO-transistors whose gates are connected to WL (as in figure 6.9), where $V_{dd} = 2.5V$, $i(t) = i_{mem} + i_{off}$, $T_0 = 200ns$ and $T_1 = 1680ns$. To get the worst case scenario we used a high signal on the input to the bottom pass-transistor before reconfiguration started. The integral over the current is estimated in SPICE simulations. The total energy used to charge the memory is 334fJ, out of which 187fJ was used to charge the memory and 147fJ to pull node $n_1$ to ground. Calculating the theoretical value that it takes to charge a 4fF capacitance gives us:

$$E_{charge} = CV^2 = 4 \times 2.5^2 \times F \times V^2 = 25fJ$$

Judging from this, the number we got seems quite large. However, in this calculation the gate insulator capacitance of the pass-transistor is ignored. Since we are using fairly big dimensions on our transistors the affect they have becomes important to include. In the following $A$ is the capacitor area which can be calculated with the transistors width and length, $\kappa$ is the dielectric constant of the

![Figure 6.9: Current simulation to evaluate write energy over the two CAAC-IGZO-transistors.](image-url)

$$E_{charge} = CV^2 = 4 \times 2.5^2 \times F \times V^2 = 25fJ$$

[36].
material, $\varepsilon_0$ is the permittivity of free space, and $d$ the thickness of the capacitor oxide insulator.

$$C = \frac{1}{d} \kappa \varepsilon_0 = \frac{1}{d} \ast (L \ast W) \ast (\kappa \ast \varepsilon_0) =$$

$$= \frac{1}{10 \ast 10^{-9} m} \ast (0.5 \ast 15 \ast 10^{-12} m^2) \ast (4.1 \ast 8.854 \ast 10^{-12} F/m) =$$

$$\approx 27.23 \ast 10^{-15} F$$

(6.2)

To charge one such capacitance would need:

$$E_{\text{charge}} = CV^2 = 170 fJ$$

(6.3)

And the total of these would become:

$$E_{\text{mem}} = 170 + 25 fJ = 195 fJ$$

(6.4)

which is much closer to the value that SPICE simulation revealed.

6.5 Chip measurements on TEG

This section is included to explain results from the produced TEG and why they were not used to a greater extent in this result section. There were some successful results, however because of problems in the design it is hard to use them to show anything with credibility.

The TEG that was produced contained one SRAM-based 53-stage IMR and one CAAC-IGZO-based 53-stage IMR. The circuit was initially used to compare between the referenced design, the proposed and the SRAM, where the multiplexer was important to decide the behavior of the referenced switch. However, there was not chip area enough which is why only two designs could be implemented, and there was not enough time to change them into IR circuits. Because time was scarce, the different behavior of the IR circuit and the IMR circuit as shown in figure 6.7 was not detected until the chip already was produced. We also noticed problems with the measurement equipment together with the chip, such as an unstable ground contact. There was also occurrences of different performance, especially in the SRAM-design, depending on the order that power sources was turned on in. And finally we found that dynamic reconfiguration of the SRAM and the CAAC-IGZO design disturbed the active task for our routing implementation.
Conclusions

DYNAMIC RECONFIGURATION was enabled in a multi-context FPGA by using a new routing switch that electrically isolates a context that is inactive from the other contexts in the FPGA to be able to achieve uniform boosting by pulling the input-node to ground. Even if the proposed design uses two additional transistors compared to the reference design it does not need any extra area. This is because parts of the transistors in the design are CAAC-IGZO FETs that can be stacked on top of the MOSFET. Further, the proposed switch is using 38% less area than SRAM, which is the most used FPGA programming technology. Because there is three transistors on the proposed RS’s critical path compared to the two that both the reference RS and SRAM-based RS, its performance is decreased. However for low input voltages, less than 1.8V, the switch achieves greater speed than the SRAM-based switch if an overdrive voltage of 0.3V is applied to the context signal. And at 1.5V it is as much as 37% faster than the SRAM-RS. The construction of using a IGZO-FET for the pull-down transistor in the RS helps keeping the low-leakage properties that earlier proposed CAAC-IGZO FPGA's have had, and the biggest change is the energy needed to reconfigure a RS memory. By SPICE simulations the write energy was evaluated to be 334fJ, that with regards to the technology sizing is an understandable value, where 187fF is used to charge the memory and 147fF to set the pull-down node to ground.

7.1 Future Work

For any technology to be chosen instead of pure MOSFET implementation it needs to have merits in all/or most of the following areas; Performance, Power Consumption, Area and Implementation Cost. In the case of FPGAs and MC-FPGAs all of these matters, and especially the three first needs to be compared
Conclusions

to several other technologies aside from CMOS. To be able to design for the MC-FPGA we need to know how well it performs in the abovementioned areas, and we need a standardized way to measure it so that comparisons can be made to state-of-the-art technology from literature. Furthermore, the technology limitations must also be taken into consideration to be able to predict and argue for its future. Defining such design rules for future development of CAAC-IGZO based FPGAs would be one step to follow-up this project.

Another more practical step would be to manufacture test units of a small MC-FPGA using the proposed switch to further evaluate behavior of it. Preferably a smaller technology sizing can also be used in the near-future. The issues with the new routing switch, such as decreased performance compared to the earlier version, can in such a setting be further examined.


[33] Young y. Liauw, Zhiping Zhang, Wanki Kim, Abbas El Gamal, and S. Simon Wong. Nonvolatile 3d-fpga with monolithically stacked rram-based configu-


Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/

© Nora Björklund