Temporal Thermal Covert Channels in FPGAs Review

Dennis Binford, Charles Dailey, Tejini Murthy, Jaishil Shah, Elijah Zapalac
Department of Electrical and Computer Engineering, *Texas A&M University*
College Station, Texas
Email: dbinford@tamu.edu, charlesdailey@tamu.edu, tmurthy@tamu.edu, jaishilshah@tamu.edu, elijah.zapalac@tamu.edu

***Abstract*— Cloud FPGAs are becoming increasingly popular in the computing industry as Amazon and Microsoft, two leading cloud providers, have instances and servers that employ them. This raises concerns about attackers targeting cloud FPGAs for insidious information leaks using covert channel communication. This paper will explore the utilization of these covert channels on a physical FPGA by replicating the research performed in [1]. This method uses heat generated from a previous user to uncover sensitive information sent through an FPGA via on-off keying (OOK).**

I. INTRODUCTION

The use of FPGAs is steadily increasing in the cloud computing industry due to the ability to leverage temporal sharing. Temporal sharing allows cloud providers to assign the same FPGA to different users for computation when not in use. This sharing of hardware has opened the door to potential exploitations by malicious users of the cloud FPGAs. One such possible exploitation is the use of thermal covert channels.

The concept of this exploit is that heat from a ring oscillator (RO) heater on an FPGA can be observed by another user on the same FPGA at a later point in time with RO sensors. This secret communication between users can be used to transmit information through OOK, where a 1 is signaled by raising the FPGA temperature or a 0 is signaled by keeping the FPGA temperature cool.

II. BACKGROUND & METHODOLOGY

The goal of this project is to monitor the behavior of temporal thermal covert channels in the physical FPGA. This will be achieved by utilizing ROs to create the intended signal data. To understand and obtain the signals, the FPGA will be equipped with a thermal transmitter that will be able to increase the temperature at specific points throughout the device. There will also be a receiver capable of recording the thermal measurements that matches the transmitter. In addition to the RO circuits, Verilog will be the main HDL (hardware description language) used to configure and set up the FPGA for the test.

II.i. FPGA and Verilog

Verilog is a programming language that can work with the behavior of the digital circuits and monitor the data being processed before deploying it into the physical FPGA. Additionally, Xilinx's Vivado Design Suite will be used to synthesize and execute the bitstream on the physical FPGA.

II.ii. RO Sensor

A ring oscillator temperature sensor can be designed by putting an odd number of inverters in series with a feedback loop. This creates a period of oscillation that is twice the delay of all the elements in the loop. An AND gate can be used to allow the oscillator to be started/stopped with an enable control signal. This explained setup is shown in Figure 1.

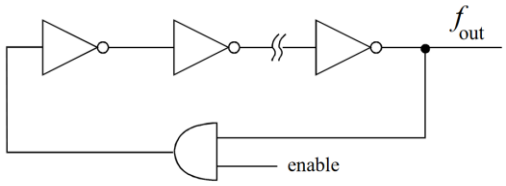

Fig 1: RO Circuit Diagram: depicts a simple RO consisting of an odd number of inverters (at least 3) and a feedback loop with an AND gate; the AND gate acts as an enabler to start or stop the RO. [2]

From past experiments, the ideal number of inverters to use is 75 [3]. This can be confirmed by testing multiple numbers of parts and observing which gives the best results.

This circuit can then be fed into a system that compares the number of oscillations of the sensor's

loop to a reference counter determined by the FPGA's internal clock. This is illustrated in Figure 2, where the RO counter can be programmed as a 20-bit counter [3].

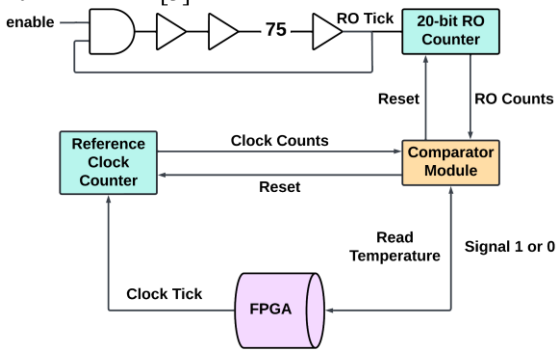

Fig 2: RO Sensor Block Diagram: a free-running RO drives a counter to be compared to the FPGA's internal clock; depending on the frequency of the 20-bit RO counter, the temperature is then deduced to signal 1 or 0; the enable control signal is controlled using a software connected to the FPGA. [1]

Temperature is a large factor in the delay through the inverters of the circuit. An increased number of gates (N) means higher accuracy but lower temperature sensitivity. This relationship is summarized in Equation 1.

$$T \uparrow \Rightarrow N \uparrow \Rightarrow d \uparrow \Rightarrow f \downarrow \qquad (1)$$

As the temperature (T) increases, so does the delay (d). The delay is then inversely proportional to the frequency (f) of the feedback, so as the delay grows, the frequency of feedback will decrease.

While the RO sensor is temperature-dependent, the other parts of the system remain unaffected, allowing the temperature to be accurately deducted based upon the comparison. From the relationship described in Equation 1, it can be determined that temperature should be increased to signal 1 when the RO counter is higher than the reference clock (indicating more delay) and decreased to signal 0 otherwise.

In order to utilize the RO sensor and communicate through the FPGA, a system that heats the circuit must also be incorporated. This can be done through a heater consisting of an array of free-running ROs as seen in Figure 3.

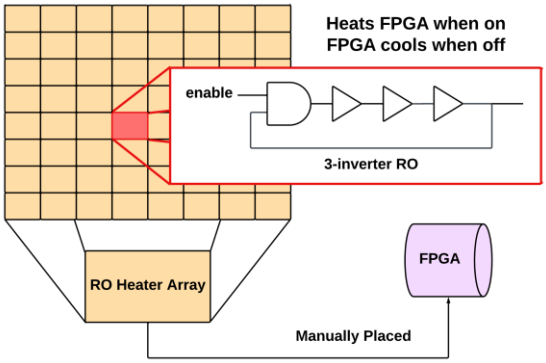

Fig 3: RO Heater Block Diagram: simple free-running ROs (using only 3 inverters) are manually placed on the FPGA to create a heater array; software connected to the FPGA controls when the heater is started or stopped. [1]

Dynamic power (P) is proportional to the frequency and number of switching elements. Furthermore, from Equation 1, frequency is inversely proportional to the number of gates (N) in an RO. Combined, this leads to the relationship in Equation 2.

$$N \downarrow \Rightarrow \frac{P}{area} \uparrow \qquad (2)$$

It follows that an RO with 3 inverters gives the highest power density since it is the smallest possible number of odd inverters in an operational system.

In addition, the RO heater should typically be manually placed on an FPGA in order to control the inverter location. This may also be constrained to a specific region on the FPGA to maximize heat density.

Using this process, multiple ROs can be manually placed on the FPGA to create the heater array.

II.iii. RO Heater

III. SIMULATION RESULTS

The method for the RO sensor discussed in II.ii. can be simulated via two tests:

The first replicates the RO functionality such that when the temperature goes up, the delay of each gate in the RO also increases. In the simulation, temperature can be programmed by a binary signal, representing a high temperature (signal 1) or low temperature (signal 0).

The second compares the RO and FPGA's clock counts with each other in order to accurately determine the relative frequency between the RO and FPGA system. This replicates how the temperature can be determined from the frequency of the RO.

III.i. Ring Oscillator

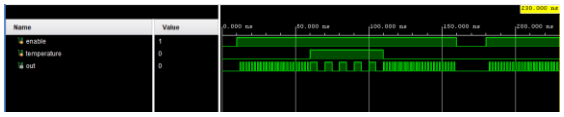

Fig 4: RO Simulation Waveforms: waveforms showing RO is on when enable is high and frequency is slower (indicating higher delay) when temperature signal is high.

Figure 4 demonstrates the results from the RO circuit programmed with Algorithm 1. It shows that when the temperature signal equals 1, the frequency of the RO clock decreases. There is also an enable switch to turn on and off the RO.

Algorithm 1: Ring Oscillator Code

```
`timescale 1ns / 1ps

module ring_oscillator #(
    parameter NUM_INVERTERS = 75
)(
    input wire enable,
    input wire temperature, // Temperature input
    output reg out
);

    reg [NUM_INVERTERS-1:0] inverters;
    integer delay;

    initial begin
        inverters = {NUM_INVERTERS{1'b0}};
        out = 1'b0;
        delay = 1; // Default delay
    end

    always begin
        // Check if enable is high; if not, hold the state
        if (enable) begin
            // Adjust the delay based on the temperature input
            if (temperature)
                delay = 5;
            else
                delay = 1;

            // Oscillation logic
            #(delay);
            inverters[0] <= ~inverters[NUM_INVERTERS-1];
            for (integer i = 1; i < NUM_INVERTERS; i = i + 1) begin
                inverters[i] <= ~inverters[i-1];
            end
            out <= inverters[NUM_INVERTERS-1];
        end else begin
            // Reset output and stop oscillation
            out <= 0;
            inverters <= {NUM_INVERTERS{1'b0}};
            #1; // Small delay to avoid glitches
        end
    end
endmodule
```

III.ii. Clock Comparator

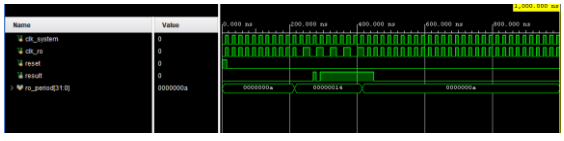

Fig 5: Clock Comparator Simulation Waveforms: waveforms showing clock comparator result to be high when RO count is less than system clock count; this indicates a high temperature when the RO frequency is slower.

Figure 5 demonstrates the results from the clock comparator circuit programmed with Algorithm 2. It shows that when the clock frequency of the RO equals the system clock frequency, the output of the comparator is 0. When the frequency of the RO decreases (indicating that temperature increases), the output of the clock comparator is 1.

Algorithm 2: Clock Comparator Code

```
`timescale 1ns / 1ps

module clock_comparator(
    input wire clk_system,        // System clock
    input wire clk_ro,            // RO counter clock
    input wire reset,             // External reset signal
    output reg result             // Result: 0 if equal, 1 if RO is slower
);

    reg [19:0] count_system;      // Counter for system clock
    reg [19:0] count_ro;          // Counter for RO clock

    parameter RESET_PERIOD = 20; // Number of cycles before reset

    // System clock counter with synchronized periodic reset
    always @(posedge clk_system or posedge reset)
        if (reset || count_system >= RESET_PERIOD)
            count_system <= 20'd0;
            count_ro <= 20'd0;  // Reset both counters when the system clock reaches the reset period
        end else begin
            count_system <= count_system + 1;
        end
    end

    // RO clock counter with periodic reset linked to system clock
    always @(posedge clk_ro or posedge reset)
        if (reset || count_system >= RESET_PERIOD)
            count_ro <= 20'd0;
        end else begin
            count_ro <= count_ro + 1;
        end
    end

    // Compare counts within each period
    always @(*)
        if (count_ro == count_system)
            result = 1'b0;  // Clocks are effectively equal within the period

        else if (count_ro < count_system)
            result = 1'b1;  // RO clock is slower within the period
    end

endmodule
```

IV. INFERENCES ON THE PROS AND CONS OF THE APPROACH USED

IV.i. Importance of Covert Thermal Channels

The use and understanding of covert thermal channels is crucial for designing countermeasures and improving the security of cryptographic algorithms on FPGAs. By utilizing these channels, engineers can identify vulnerabilities associated with side-channel attacks. These attacks exploit physical properties of a device, such as power consumption or temperature, to extract sensitive data, like cryptographic keys.

This project, in turn, demonstrates one of the most common vulnerabilities and documents how an attack is orchestrated. From this report, insights on how to resist such attacks and protect users' valuable data can further be provided.

IV.ii. Application in the Industry

Covert thermal channels on a physical FPGA can be used in various scenarios such as improving security within side-channel analysis, proper system integrity checks, and thermal management. In terms of hardware-level security, the use of covert thermal channels can be utilized to uncover sensitive information through the process of monitoring temperature spikes during certain computational activities. Many engineers working in research and security design these systems with thermal covert channels in order to find vulnerabilities in algorithms pertaining to cryptography and crucial computations. Thus, the benefit of attaching these security measures on the FPGA boards is being able to create countermeasures in the event of potential security threats.

They can also be used for monitoring the operational health of FPGA boards. The channels target the temperature of the device and can determine the current performance and longevity based on the health and power of the board. Based on certain thermal hotspots, the channels can ensure that the heat is evenly distributed across the chip, and can assess whether or not there are present abnormalities. Overheating is a sign of potential device failure, and in fields that heavily rely on smooth operation, these channels can avoid critical failure and safety in certain cases.

Covert thermal channels can also study the behavior of temperature across integrated circuits. These covert channels do not require the need for external sensors in order to monitor the thermal values of the FPGA in question. Since computational devices are becoming more powerful as technology advances, intricate cooling systems are crucial in enabling these computations to continue without thermal failure. Thus, the use of covert channels allows researchers to find a great balance between computation that keeps up with the increasing demand, with the addition of safety measures that ensure security and longevity.

IV.iii. Advantages of Using ROs

Covert thermal channels are made using thermal transducers, which convert the FPGA's temperature into a readable electrical form. These tend to be discretely placed on the board; however, this becomes difficult when there are a large number of chips. On-chip transducers avoid this issue [2].

ROs are a type of on-chip thermal transducer with the end-user in mind. The sensors are small and straightforward to code, while also providing completely digital signals. This makes use of the FPGA's general interconnection network. Furthermore, ROs can be placed anywhere on the chip, while the hardware for the thermal status of the system can be mapped directly to the FPGA [2]. This allows for ease of use.

Unlike many other synchronous heaters, RO heaters also do not require an external clock signal. This means its output is not clock-dependent and thus, will not fluctuate based on the frequency [4].

IV.iv. Limitations

While implementing the use of a covert thermal channel on a physical FPGA, there are a few limitations to be observed.

The first drawback deals with the practicality of transmitting large amounts of data by way of the covert channel on an FPGA. Temperature change propagation and stabilization on an FPGA are inherently slow due to the cooling systems (e.g. heat

sinks or fans) that a large number of FPGAs have to prevent overheating. This leads to a decreased bandwidth across the communication channel.

On the opposing side, if an FPGA is particularly nonresistant to temperature changes (e.g. not having advanced heat dissipation features), the risk of overheating arises. By intentionally generating thermal fluctuations, potential long term damage or the triggering of thermal protection systems could occur. Causing either of these situations to arise could disrupt the covert channel and likely alert the FPGA owner that something was amiss.

Another limitation to consider is the existence of uncontrolled thermal fluctuation factors. Ambient temperature fluctuations or power dissipation from normal operation of the FPGA can introduce thermal noise into the sensor, making the reliability of the communication channel difficult to maintain.

V. LESSONS LEARNED

V.i. Design and Implementation of Ring Oscillators (ROs)

Through this project, valuable experience in the design and implementation of ROs was gained. Such ROs were configured both as thermal sensors and heaters, allowing for observation of their behavior on physical FPGAs. Additionally, this project utilized skills in manually placing these circuits to optimize thermal performance. This process provided hands-on understanding of how ROs can serve as fundamental components in thermal covert channels.

V.ii. Relationship Between Temperature and Frequency

A key learning point was the relationship between temperature, delay, and frequency, as captured by Equation 1. This concept was crucial in understanding how temperature-induced delays in ROs can be exploited to create a covert communication channel. Observing and analyzing these dynamics deepened understanding of thermal properties and their impact on FPGA performance.

V.iii. Vulnerability of FPGA Systems

The project highlighted the subtle yet significant vulnerabilities of FPGA systems to thermal covert channels. These attacks exploit indirect physical properties, making them difficult to detect. Observations of how small temperature variations can be used to encode sensitive information were made. This underlines the importance of security measures in FPGA designs to counteract such risks.

V.iv. Dangers of Covert Thermal Channels

This project emphasized the dangers posed by thermal covert channels. These channels allow attackers to extract sensitive data without physical access, making them a stealthy and hard-to-detect method of compromise. This realization reinforced the necessity of designing robust countermeasures and security protocols to protect FPGA systems.

V.v. Key Takeaways

Overall, this project provided a comprehensive understanding of thermal covert channels and their implications. It also furthered technical skills in FPGA design, programming, and security analysis, paving the way for future research and innovation in securing FPGA systems against such vulnerabilities.

VI. CONCLUSION

This research into thermal covert channels on FPGA systems underscores the urgent need for developing robust proactive countermeasures to mitigate security vulnerabilities from covert thermal channels. Future research should concentrate on several key areas to fortify FPGA designs against such stealthy attacks:

VI.i. Thermal Monitoring Systems

A crucial step forward is the development of real-time thermal monitoring mechanisms capable of detecting and responding to abnormal temperature variations. These systems would require high-resolution thermal sensors integrated into FPGA architectures to continuously track temperature changes across the device. The design of these sensors should ensure minimal impact on performance while providing accurate thermal readings. When unusual patterns are detected (e.g. unexpected localized heating indicative of a potential covert communication attempt), the monitoring system should be able to trigger immediate alerts, informing security protocols to investigate or counteract the threat [5].

VI.ii. Dynamic Thermal Management

Another potential research direction is the implementation of adaptive thermal management strategies. These systems could dynamically adjust resource allocation and workload distribution across the FPGA to minimize significant temperature gradients. By evenly distributing thermal loads and preventing hotspots, dynamic management can reduce the opportunities for attackers to create effective covert thermal channels. This approach may involve software algorithms that balance tasks based on real-time thermal feedback, as well as hardware mechanisms that facilitate efficient heat dissipation. The challenge here lies in designing solutions to be both responsive and non-intrusive, ensuring they do not degrade the overall performance of the FPGA [6].

VI.iii. Anomaly Detection Algorithms

Sophisticated anomaly detection algorithms are essential for identifying covert communication attempts based on thermal signatures. Future research could explore the use of machine learning techniques to analyze temperature data and recognize patterns associated with thermal attacks. These algorithms would need to be trained on a wide range of thermal behaviors to differentiate between legitimate temperature fluctuations and those indicative of an ongoing covert channel operation. The development of these models should prioritize accuracy and speed, enabling the system to detect and respond to threats in real time. Additionally, integrating anomaly detection with other security measures, such as power analysis and side-channel monitoring, could provide a thorough defense strategy [7].

VI.iv. Hardware Redesign for Security

To enhance the resilience of FPGAs against thermal covert channels, research should also consider hardware-level design improvements. This may involve rethinking the spatial layout of critical components to minimize the impact of temperature fluctuations or incorporating materials with better thermal conductivity to dissipate heat more efficiently. Additionally, engineers could explore the use of specialized hardware blocks designed to mitigate thermal effects, such as circuits that introduce random noise into temperature data, making it harder for attackers to establish reliable covert channels [5].

REFERENCES

[1] Shanquan Tian and Jakub Szefer. 2019. "Temporal Thermal Covert Channels in Cloud FPGAs." In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '19)*, February 24-26, 2019, Seaside, CA, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3289602.3293920.

[2] Eduardo Boemo and Sergio López-Buedo. 1997. "Thermal monitoring on FPGAs using ring-oscillators." In Luk, W., Cheung, P.Y.K., Glesner, M. (eds) *Field-Programmable Logic and Applications. FPL 1997*. Lecture Notes in Computer Science, vol 1304. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-63465-7_212.

[3] Jiawen Wu. 2024. "Fall ECEN 426 Project 4." *realjw.notion.site*. https://realjw.notion.site/24-Fall-ECEN-426-Project-4-83803d0a790a4a86901e29edea3529c6.

[4] Andreas Agne, Hendrik Hangmann, Markus Happe, Marco Platzner, and Christian Plessl. 2014. "Seven Recipes for Setting Your FPGA on Fire – A Cookbook on Heat Generators." *Microprocessors and Microsystems*, Volume 38, Issue 8, Part B, 2014, Pages 911-919, ISSN 0141-9331. https://doi.org/10.1016/j.micpro.2013.12.001.

[5] Yuan, Jie, Jing Zhang, Pengfei Qiu, Xinghai Wei, and Dongxiao Liu. 2024. "A Survey of of Side-Channel Attacks and Mitigation for Processor Interconnects." *Applied Sciences* 14, no. 15: 6699. https://doi.org/10.3390/app14156699

[6] Benelhaouare, Amrou Zyad, Idir Mellal, Maroua Oumlaz, and Ahmed Lakhssassi. 2024. "Mitigating Thermal Side-Channel Vulnerabilities in FPGA-Based SiP Systems Through Advanced Thermal Management and Security Integration Using Thermal Digital Twin (TDT) Technology." *Electronics* 13, no. 21: 4176. https://doi.org/10.3390/electronics13214176

[7] Priya, B.I., Rao, P.V.R.D.P. & Parameswari, D.V.L. 2024. "Shielding secrets: developing an enigmatic defense system with deep learning against side channel attacks." *Discov Sustain* 5, 249. https://doi.org/10.1007/s43621-024-00455-4