Non-Deterministic Timers for Hardware Trojan Activation
(Or How a Little Randomness Can Go the Wrong Way)

Frank Imeson

Electrical and Computer Engineering

August 8th, 2016

Collaborators: Saeed Nejati (Waterloo), Siddharth Garg (NYU) and Mahesh V. Tripunitara (Waterloo)
1 Intro
   ■ Attack Model
   ■ Background

2 Deterministic Timers
   ■ Hardware
   ■ Single Stage D-Timer
   ■ Dual Stage D-Timer

3 ND-Timers
   ■ Single Shot Timer
   ■ ND-Timer

4 TRNG

5 Clock Jitter
   ■ Ring Oscillators
   ■ Sampling

6 Results
   ■ Hardware
   ■ 24 Hour Experiment

7 Detection
   ■ Hardware
   ■ Simulation
   ■ Analysis

8 Summary
case (display_state)
  UPDATE : begin
    seg00_reg <= seg00;
    seg01_reg <= seg01;

    // update leds
    if (count00[0]) begin
      state <= UPDATE;
    end
  default : begin
    on00 <= 0;
    count00 <= 0;
    display_state <= UPDATE;
  end
endcase
```haskell
case(display_state)
  UPDATE : begin
    seg00_reg <= seg00;
    seg01_reg <= seg01;

    // update leds
    if (count00[0]) begin
      state <= UPDATE;
    end
  endcase
```

Background

*Hardware modifications are permanent!*

Trojans:
- Privilege escalation [5]
- Leaking private information [3]
- Sabotage [4]

Activation Types:
- Time Trigger [2]
- Data Trigger [5]
## Deterministic Timers

1. **Intro**
   - Attack Model
   - Background

2. **Deterministic Timers**
   - Hardware
   - Single Stage D-Timer
   - Dual Stage D-Timer

3. **ND-Timers**
   - Single Shot Timer
   - ND-Timer

4. **TRNG**

5. **Results**
   - Clock Jitter
   - Ring Oscillators
   - Sampling

6. **Detection**
   - Hardware
   - 24 Hour Experiment
   - Simulation
   - Analysis

7. **Summary**
Hardware Constraints

- No access to Wall time (guarded)
- Power on time
- Power is not persistent
- Cannot save state off chip (guarded)
- NV-memory can be used on chip
Hardware Constraints

- No access to Wall time (guarded)
- Power on time
- Power is not persistent
- Cannot save state off chip (guarded)
- NV-memory can be used on chip
- NV-memory technology, e.g., Flash, has limited write durability
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
- Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1.
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1.
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Can be defeated with power cycling [7]

- Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
- Can be defeated with power cycling [7]

• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Can be defeated with power cycling [7]

IEEE Symposium on Security and Privacy
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Can be defeated with power cycling [7]

• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1

• Can be defeated with power cycling [7]

IEEE Symposium on Security and Privacy
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Can be defeated with power cycling [7]

IEEE Symposium on Security and Privacy
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Can be defeated with power cycling [7]

IEEE Symposium on Security and Privacy
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Can be defeated with power cycling [7]

IEEE Symposium on Security and Privacy
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Can be defeated with power cycling [7]

IEEE Symposium on Security and Privacy
• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1

• Can be defeated with power cycling [7]

• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1

• Can be defeated with power cycling [7]

Single Stage D-Timer

- Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
- Can be defeated with power cycling [7]

Single Stage D-Timer

- Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
- Can be defeated with power cycling [7]

• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1

• Can be defeated with power cycling [7]

• Deterministic Timer (D-Timer) will trigger at a specified number of clock cycles with probability 1
• Can be defeated with power cycling [7]

Dual stage timer stores 2\textsuperscript{nd} timer in NV memory
• Dual stage timer stores $2^{nd}$ timer in NV memory
• Dual stage timer stores $2^{nd}$ timer in NV memory
• Dual stage timer stores $2^{nd}$ timer in NV memory
• Dual stage timer stores $2^{nd}$ timer in NV memory
• Dual stage timer stores $2^{nd}$ timer in NV memory
Dual stage timer stores $2^{nd}$ timer in NV memory.
Dual stage timer stores $2^{nd}$ timer in NV memory
Dual stage D-Timer

- Dual stage timer stores $2^{nd}$ timer in NV memory
• Dual stage timer stores 2\textsuperscript{nd} timer in NV memory
- Dual stage timer stores $2^{nd}$ timer in NV memory
• Dual stage timer stores $2^{nd}$ timer in NV memory
• Dual stage timer stores 2\textsuperscript{nd} timer in NV memory
Dual stage timer stores 2\textsuperscript{nd} timer in NV memory
• Dual stage timer stores $2^{nd}$ timer in NV memory
• Dual stage timer stores $2^{nd}$ timer in NV memory
- Dual stage timer stores $2^{nd}$ timer in NV memory
- Dual stage timer stores $2^{nd}$ timer in NV memory
• Dual stage timer stores $2^{nd}$ timer in NV memory
• Dual stage timer stores $2^{nd}$ timer in NV memory
• This design defeats the power cycling
Dual stage timer stores $2^{nd}$ timer in NV memory
This design defeats the power cycling
Dual stage timer stores $2^{nd}$ timer in NV memory

This design defeats the power cycling

Vulnerable to power cycling with period $\leq VSW$ (Volatile State Window)
Deterministic Timer Drawbacks

Dual Stage D-Timers require frequent writes to NV. Given a write durability $w$ and a timer that counts up to $k$ we need $m \geq \frac{k}{w}$ bits.

A one year timer:

- $VSW = 1$ second
- $m \geq 3153$ NV-bits
- $VSW = 52$ minutes
- $m = 14$ NV-bits
ND-Timers

1. Intro
   - Attack Model
   - Background

2. Deterministic Timers
   - Hardware
   - Single Stage D-Timer
   - Dual Stage D-Timer

3. ND-Timers
   - Single Shot Timer
   - ND-Timer

4. TRNG

5. Results
   - Clock Jitter
   - Ring Oscillators
   - Sampling

6. Detection
   - Hardware
   - 24 Hour Experiment
   - Simulation
   - Analysis

7. Summary
Single Shot Timer

- Not easy to control
- Chance of early detection
- Large Variance
• Not easy to control
• Chance of early detection
• Large Variance
Single Shot Timer

- Not easy to control
- Chance of early detection
- Large Variance
Single Shot Timer

- Not easy to control
- Chance of early detection
- Large Variance


- Not easy to control
- Chance of early detection
- Large Variance
• Not easy to control
• Chance of early detection
• Large Variance
• Not easy to control
• Chance of early detection
• Large Variance
• Not easy to control
• Chance of early detection
• Large Variance
Single Shot Timer

- Not easy to control
- Chance of early detection
- Large Variance
- Small VSW
Statistical Comparison

<table>
<thead>
<tr>
<th></th>
<th>Early</th>
<th>Expected</th>
<th>Late</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-Timer</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>SS-Timer</td>
<td>39%</td>
<td>39%</td>
<td>22%</td>
</tr>
</tbody>
</table>

Figure: The ND-Timer$_1$ was designed with an expectation of 1000. The corresponding deterministic timer is shown for comparison.
ND-Timer: Counts $\kappa$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance

$$\text{Rand X} \quad \text{NV Timer} \quad \text{Trigger} \quad \text{Timer} \quad \text{Rand X} \quad \text{Power}$$

\[ + \quad + \quad + \quad \cdots = \text{Waveform} \]
ND-Timer: Counts $\kappa$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance

$\text{Rand X} \quad \text{NV Timer}$

$\text{Trigger}$

$\text{Rand X}$

$\text{Power}$

$+ + + = \text{Peak}$
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance

Frank Imeson (University of Waterloo) Non-deterministic Timer Based Attacks on Computer Hardware
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts \( k \) Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts \( k \) Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance

Frank Imeson (University of Waterloo)
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance

Frank Imeson (University of Waterloo)
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance

Frank Imeson (University of Waterloo) Non-deterministic Timer Based Attacks on Computer Hardware 12
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
ND-Timer: Counts $k$ Independent Random Events

- Controllable
- Small Variance
- Small VSW
Figure: The ND-Timer was designed with an expectation of 1000. The corresponding deterministic timer and single event is shown for comparison.
ND-timers break the trade-off between size of NV-memory and susceptibility to periodic power cycling.

- **Accuracy** $E(N) = \frac{k}{p}$
- **Precision** $\frac{\sigma}{E(N)} = \frac{1}{\sqrt{k}}$

<table>
<thead>
<tr>
<th></th>
<th>D-Timer01</th>
<th>D-Timer02</th>
<th>ND-Timer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected Trigger</td>
<td>1y</td>
<td>1y</td>
<td>1y</td>
</tr>
<tr>
<td>99% Confidence Interval</td>
<td>±0m</td>
<td>±0m</td>
<td>±12d</td>
</tr>
<tr>
<td>VSW</td>
<td>52m</td>
<td>27.6µs</td>
<td>27.6µs</td>
</tr>
<tr>
<td>NV-Bits</td>
<td>14</td>
<td><strong>114Mbits</strong></td>
<td>14</td>
</tr>
</tbody>
</table>

**Table:** Performance and resource comparison of each timers.

Frank Imeson (University of Waterloo) Non-deterministic Timer Based Attacks on Computer Hardware
1 Intro
   - Attack Model
   - Background
2 Deterministic Timers
   - Hardware
   - Single Stage D-Timer
   - Dual Stage D-Timer
3 ND-Timers
   - Single Shot Timer
   - ND-Timer
4 TRNG
5 Results
   - Hardware
   - 24 Hour Experiment
6 Detection
   - Hardware
   - Simulation
   - Analysis
7 Summary

Clock Jitter
Ring Oscillators
Sampling
The generation of random numbers is too important to be left to chance.

Robert R. Coveyou

- True random sources exist in the physical world
  - Temperature
  - Timing
  - Power
  - ...
Clock Jitter

• Clock jitter contains true unbiased randomness [1]
• Easy to construct (minimum of 3 gates)

![Diagram of clock jitter circuit with time and voltage axis](image-url)
Sample ROs with a slower clock to allow for multiple separate transitions.
Use multiple ROs to fill the time spectrum with transitions. Down sample results 1024 times to remove bias and ensure *true* randomness. [8]
Summary

- 16 ROs
- 80 gates
- 1 bit/1024 cycles
- Passes NIST

Sample and reset every 1024 cycles

Decimate 1024 times
1 Intro
   ■ Attack Model
   ■ Background
2 Deterministic Timers
   ■ Hardware
   ■ Single Stage D-Timer
   ■ Dual Stage D-Timer
3 ND-Timers
   ■ Single Shot Timer
   ■ ND-Timer
4 TRNG
5 Results
   ■ Clock Jitter
   ■ Ring Oscillators
   ■ Sampling
6 Detection
   ■ Hardware
   ■ Simulation
   ■ Analysis
7 Summary
Experimental Hardware Setup:

- Altera DE4 development board
- Simulated power cycling and NV memory
- Two different 24h timer experiments
- Each experiment repeated 15 times
- Both experimental trials passed the $\chi^2$ test for expected distributions.
Figure: Measured and predicted CDFs for the two 24 hour triggers. Trigger one (red) has a standard deviation of 2.84 minutes, while trigger two (blue) the other has a standard deviation of 16 minutes.
Detection

1. Intro
   - Attack Model
   - Background
2. Deterministic Timers
   - Hardware
   - Single Stage D-Timer
   - Dual Stage D-Timer
3. ND-Timers
   - Single Shot Timer
   - ND-Timer
4. TRNG
5. Results
   - Hardware
   - 24 Hour Experiment
6. Detection
   - Hardware
   - Simulation
   - Analysis
7. Summary
Figure: Hardware overhead comparison for two data trigger and two timer attacks. Black represents the amount used by the trigger and grey represents the amount used by the Trojan.
Simulation: Does not have randomness!

Figure: A simulation results from Modelsim shown in the top diagram for ND-timer. Bottom diagram shows a typical pattern for rand_bit gathered from FPGA.
UCI & FANCI:

- Looks for unused or nearly unused logic (circuitry).
- Redundant circuitry could be replaced with a wire.
- Flagged as suspicious.
Logic Analysis: Unused Circuit Identification

UCI & FANCI:

- Looks for unused or nearly unused logic (circuitry).
- Redundant circuitry could be replaced with a wire.
- Flagged as suspicious.
- Known methods to defeat UCI [6] and FANCI [9].
Summary

ND-Timers:

- Defeat power cycling defences.
- Use less resources than D-Timers.
- Controllable.
- Do not simulate.
_Evaluation of random number generators on FPGAs._  

A case study in hardware trojan design and implementation.  

Experiences in hardware trojan design and implementation.  

Phillip Jones, and Joseph Zambreno.  
Circumventing a ring oscillator approach to fpga-based hardware trojan detection.  

Breakthrough silicon scanning discovers backdoor in military chip.  
Defeating uci: Building stealthy and malicious hardware.  

Silencing hardware backdoors.  

Analysis and enhancement of random number generator in fpga based on oscillator rings.  

Detrust: Defeating hardware trust verification with stealthy implicitly-triggered hardware trojans.  