UCLA COMPUTER SCIENCE DEPARTMENT

A BENIGN HARDWARE TROJAN ON FPGA-BASED EMBEDDED SYSTEMS

Jason Zheng, Ethan Chen, Miodrag Potkonjak
Computer Science Department
University of California, Los Angeles

FPL2012, Oslo, Norway
August 31, 2012

Outline

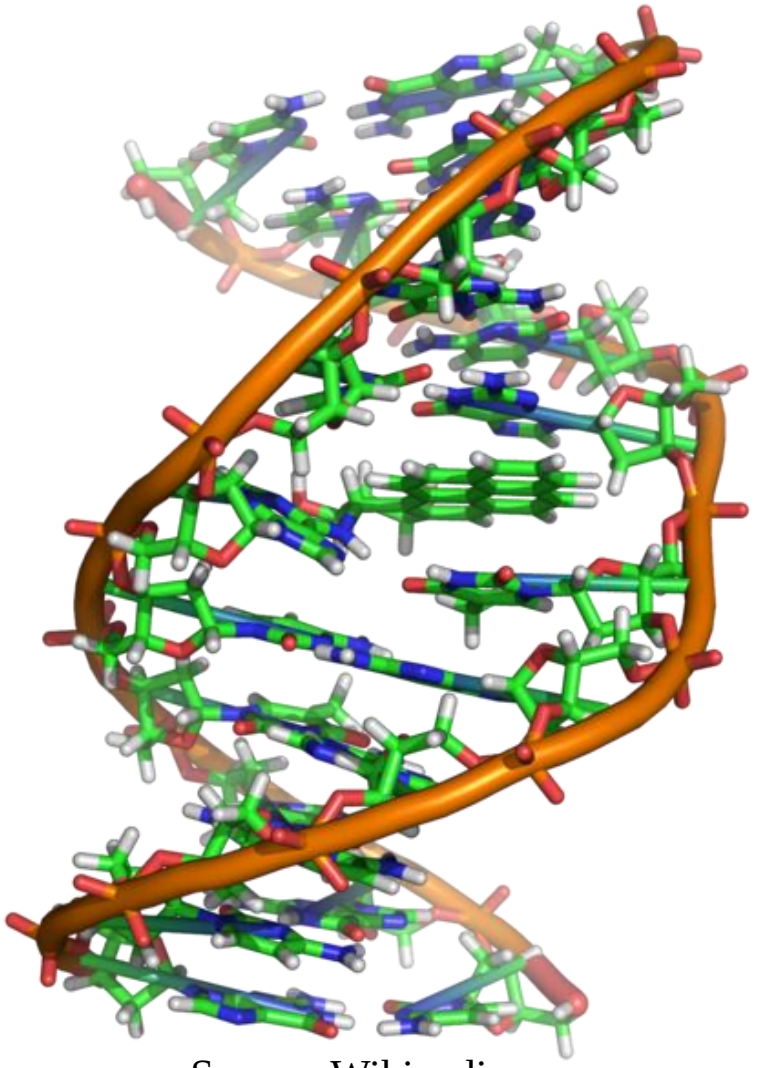
- Introduction
- Preliminaries
- Implementation
- Benchmark Results
- Summary

Benign Hardware Trojan Circuits

- **Traditionally a Hardware Trojan:**
  - Hidden structures and functionalities designed to wreak havoc in circuits
- **Security Intent**
  - Ward off security attacks: cloning, reverse engineering, code injection, etc.
- **Enabling Technology**
  - Process Variation
  - Targeted Aging (NBTI)

Key Concept – Matched Uniqueness

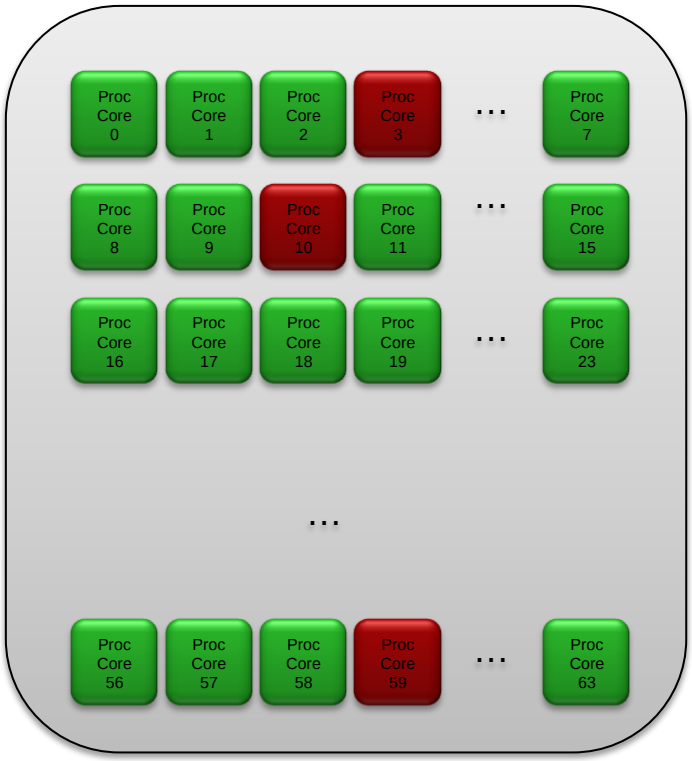
- Each instance of HW is:
  - Functionally identical
  - Unique delay signature (process variation, aging)
- Each instance of SW is matched to the HW at compilation.
- Ensures:
  - SW can be executed only on the intended HW instance.
  - HW only executes SW intended for it.



Source: Wikipedia

Example: Many-Core Tiled Processor

- A high-performance 64-core system with static schedule.
- Each instance of the chip has N disabled cores chosen by BHT.
- Compiler must know which core is disabled to produce working SW.



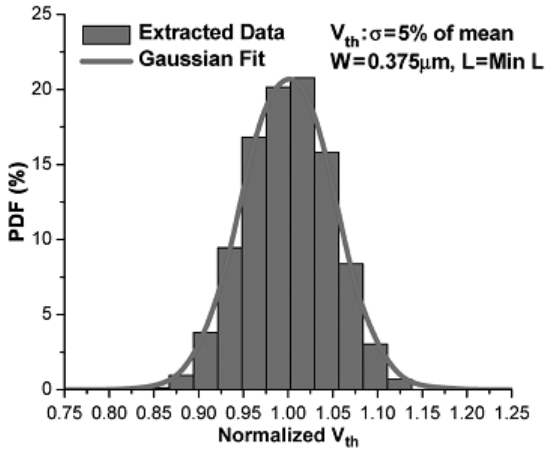
64-core Tiled Processor

Outline

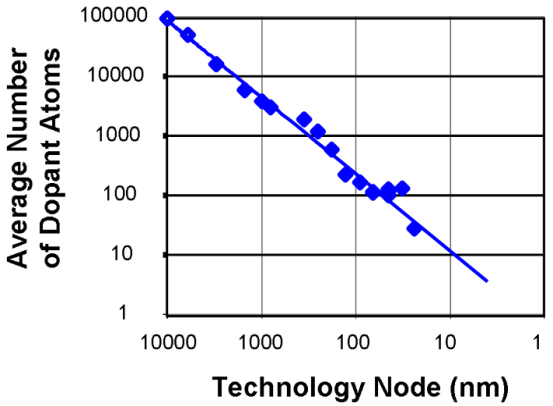
- Introduction
- Preliminaries
- Implementation
- Benchmark Results
- Summary

Process Variation

- Random and systematic factors in the VLSI fabrication process lead to intra- and inter-die variations in V_{th}, I_{off}, etc.
- As the feature size shrinks, the degree of variation increases.
- Implication: two logic gates with identical design parameters will not have identical delay.



•Fig. 1 - Normalized V_{th} variation at 65nm [1]



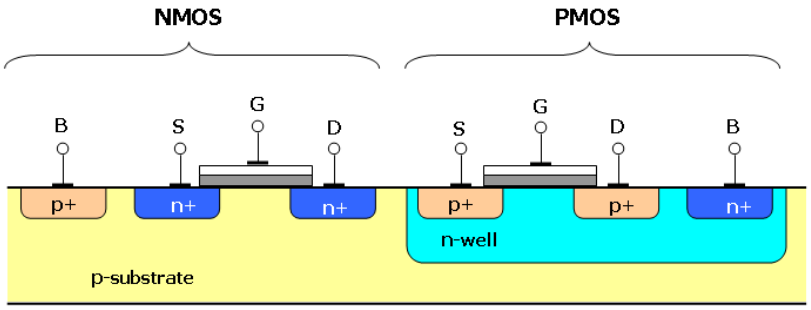
•Fig. 2 – Avg. number of dopant atoms [2]

[1] W. Zhao, F. Liu, K. Agarwal, D. Acharyya, S. R. Nassif, K. J. Nowka, Y. Cao, "Rigorous Extraction of Process Variations for 65-nm CMOS Design," *Semiconductor Manufacturing, IEEE Transactions on* , vol.22, no.1, pp.196-203, Feb. 2009.

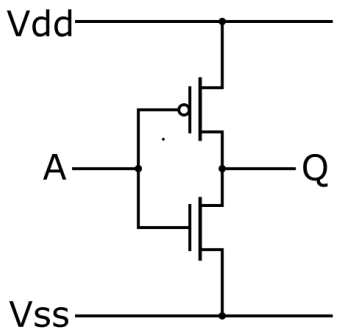
[2] K. Kuhn, C. Kenyon, A. Kornfeld, M. Liu, A. Maheshwari, W.-K. Shih, S. Sivakumar, G. Taylor, P. VanDerVoorn, and K. Zawadzki, "Managing Process Variation in Intel's 45-nm CMOS technology," Intel Tech . J., vol. 12, no. 2, pp. 93–110, Jun. 2008.

Negative Bias Temperature Instability (NBTI)

- NBTI is an aging process that primarily affects PMOS devices.
- When Vgs is negative for a prolonged period of time, interface traps are created and negatively affects threshold voltage (Vth).
- As a result, propagation delay and leakage current increases.



Source: Wikipedia

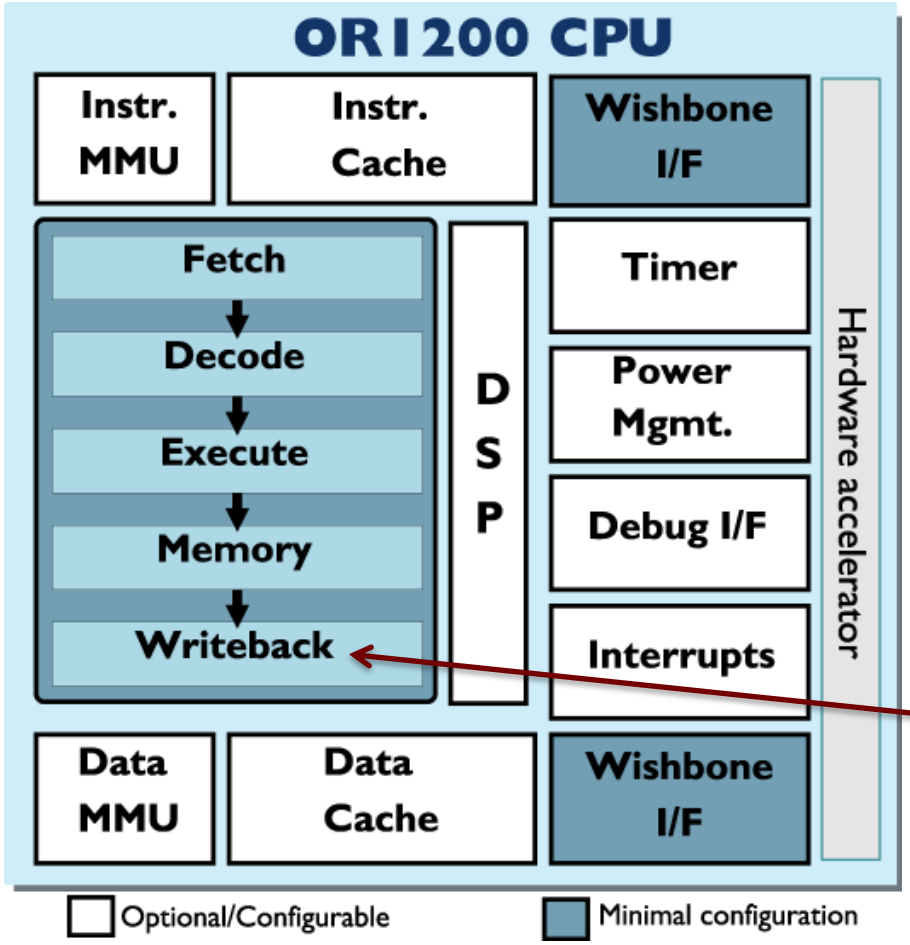


Source: Wikipedia

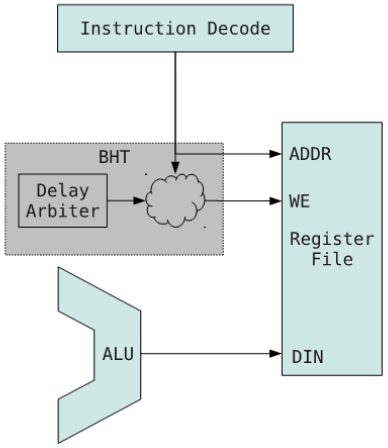
Outline

- Introduction
- Preliminaries
- Implementation
- Benchmark Results
- Summary

FPGA Implementation: OpenRISC OR1200



- 5-stage,32-bit general purpose processor.
- 32 GPRs, statically scheduled by compiler.
- Our addition: BHT embedded within the GPR write-back.



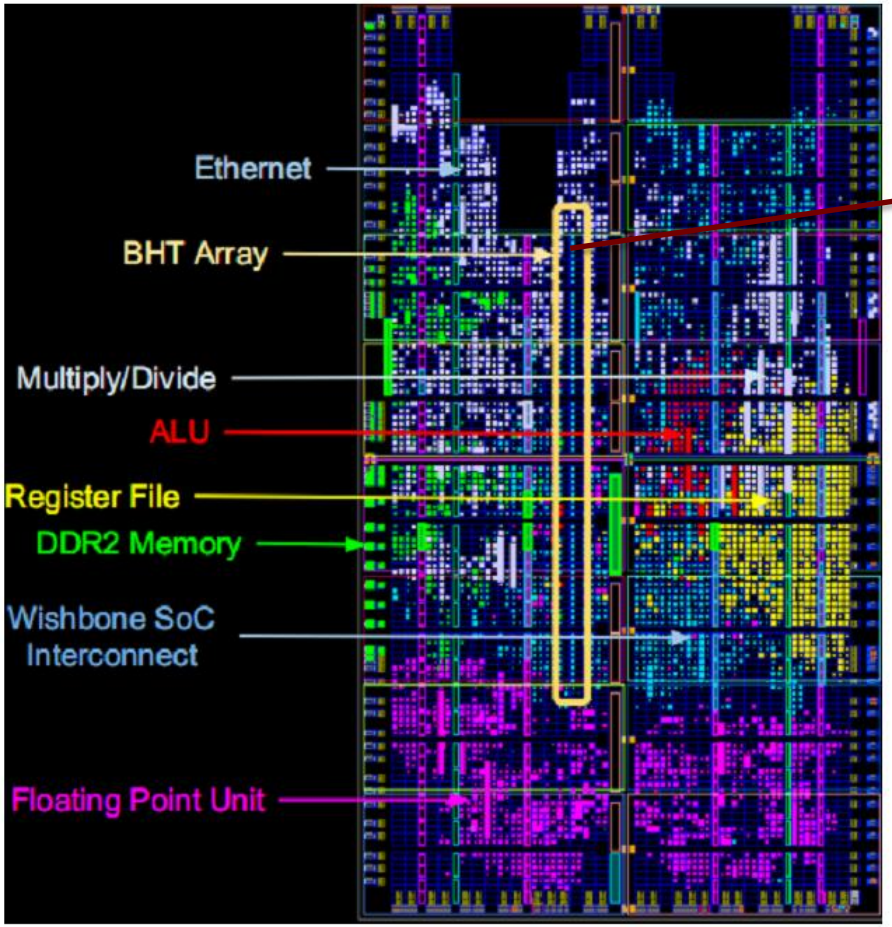
Source: Opencores.org

FPGA Implementation: Resource and Performance

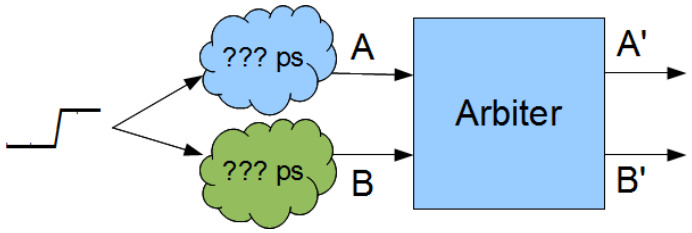
Resource	Use Count	Use Percentage
D-Flipflop	5718	10%
LUTs	10918	40%
Slices	3661	53%
BRAMs	87	37.5%
DSP48A1s	4	6%

- **Target platform:**
  - Digilent Atlys board
  - Spartan-6 FPGA (45-nm)
- **Toolchain:**
  - Xilinx ISE 13.1 toolchain
- **Clock rate: 50MHz**

FPGA Implementation: BHT Delay Logic

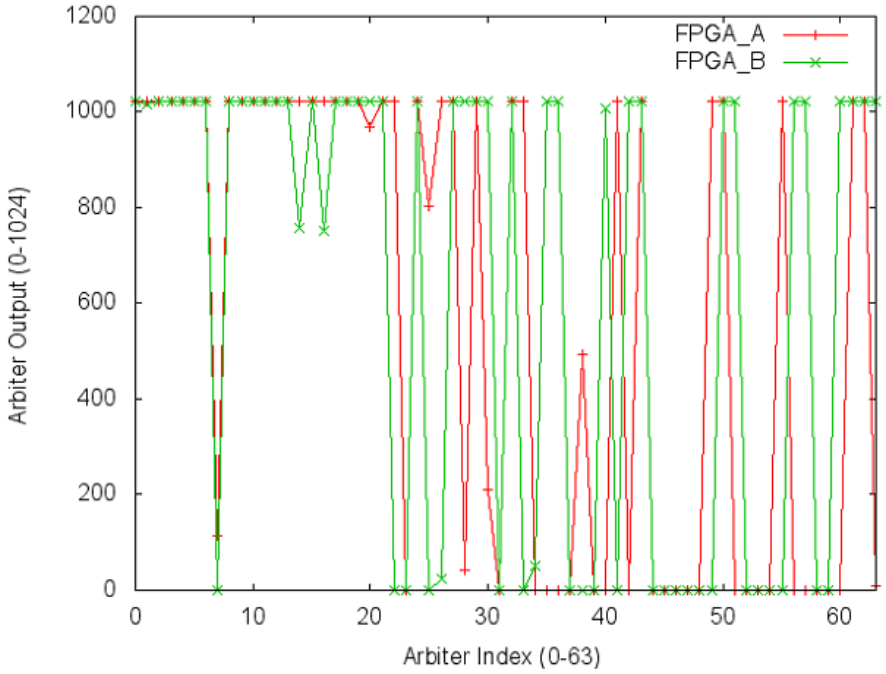


OR1200 Layout with BHT on Spartan-6 FPGA



- BHT delay arbiters measure subtle delay differences in the silicon.
- Delay signature forms at manufacturing time by Process Variation.
- NBTI can also alter the relative delay.

Process Variation



- **Figure left:**
  - 64 arbiter outputs from two Spartan-6 FPGAs over 1024 samples.
- **Arbiters 21-63**
  - Strong Process Variation influence.
  - Prime candidate for BHT
  - Stability can be improved by voting logic or digital filter.

Software Implementation

- Baseline toolchain – GCC, OR1Ksim (simulator)
- Compiler modifications
  - -mregistermask compiler flag
  - A 32-bit mask parameter passed to the compiler
  - Indicates which GPR is disabled to GPR scheduler
- Simulator modifications
  - CPU configuration flag disable_regs
  - Disables selected GPRs in simulation

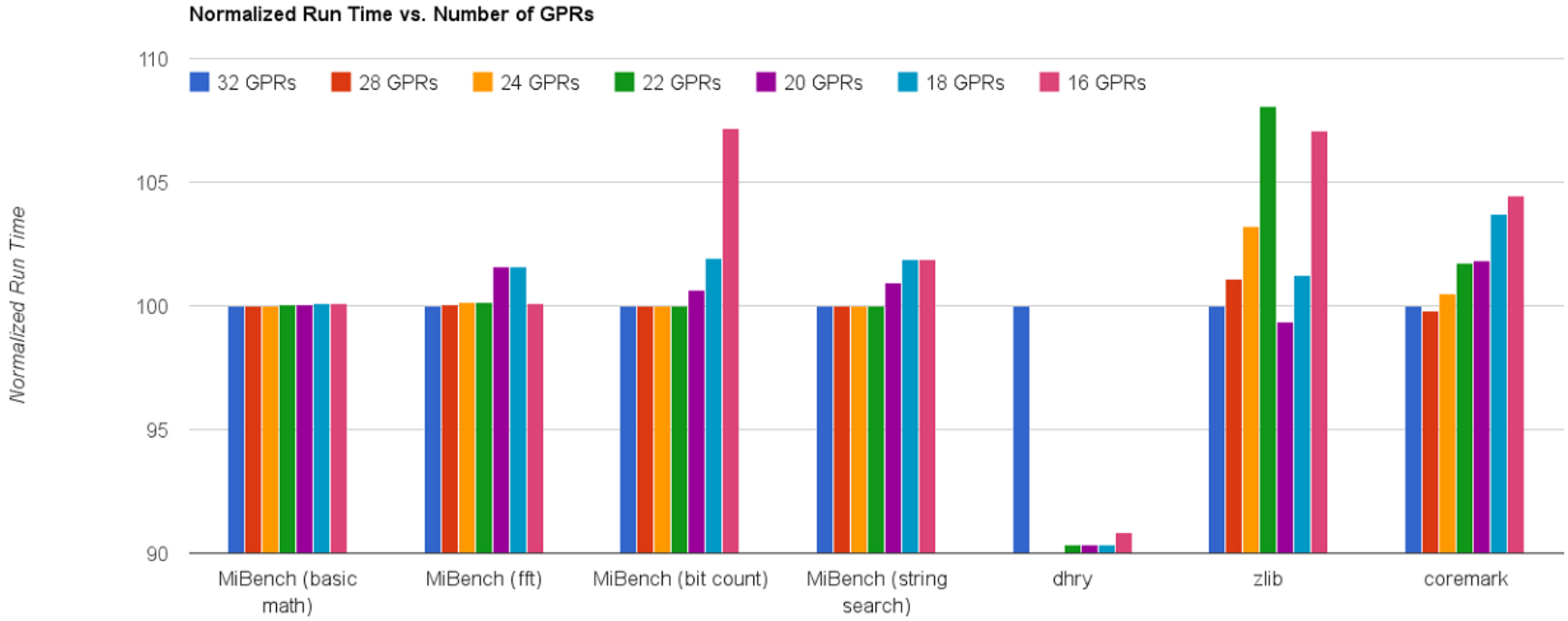
Outline

- Introduction
- Preliminaries
- Implementation
- **Benchmark Results**
- **Summary**

Benchmarks

- **Objectives**
  - Show that the OR1200 BHT modification works as expected.
  - Show that the toolchain modifications work as expected.
  - Measure runtime overhead as a result of fewer available GPRs.
- **Chosen embedded benchmarks:**
  - Dhrystone (synthetic)
  - CoreMark (synthetic)
  - MiBench
  - zlib

Benchmarks, continued



- Benchmark results from OR1200 on Spartan-6.
- # of GPRs is reduced from 32 to 16.
- Highest impact is 8% (zlib with 22 GPRs)
- Dhry anomaly probably due to compiler optimization.
- zlib non-monotonic results: change of compiler optimization strategies due to # of GPR change.

Summary

- BHT works by creating HW instances with unique delay signatures and SW instances that understand them.
- HW and SW will only work correctly when shared signatures match.
- Successfully implementation in GPR write-back logic in OR1200 General Purpose Processor.
- Synthetic and realistic benchmarks show a small overhead due to reduced number of GPRs to compiler.

Questions

- **Security Model?**
  - Now: Software copying
  - Next: Exponentially long code reversing + negligible overhead (PPUF)
- **FPGA specific?**
  - Now: Nothing
  - Next: Mapping to take maximal advantage, device aging and characterization
- **Architecture?**
  - Now: General purpose processor
  - Next: ASIC and FPGA