Flash Memory ‘Bumping’ Attacks

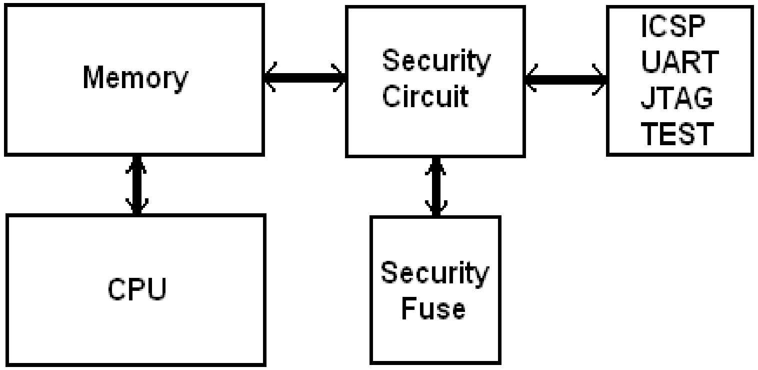
Sergei Skorobogatov

http://www.cl.cam.ac.uk/~sps32 *email: sps32@cam.ac.uk*

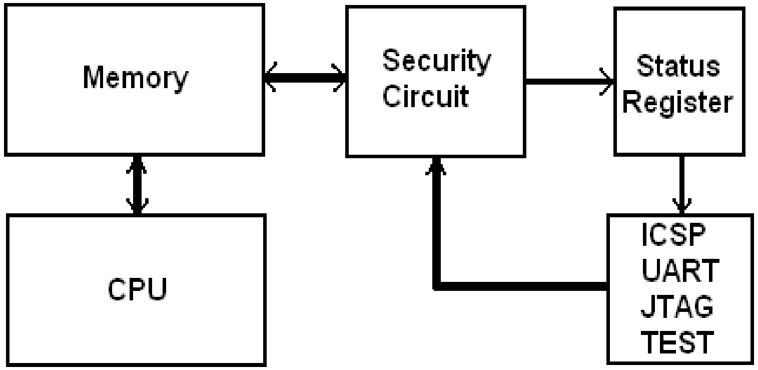
UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Introduction

- Data protection with integrity check
  - verifying memory integrity without compromising confidentiality
  - How secure is "No Readback" solution?



Readback access controlled by security fuse



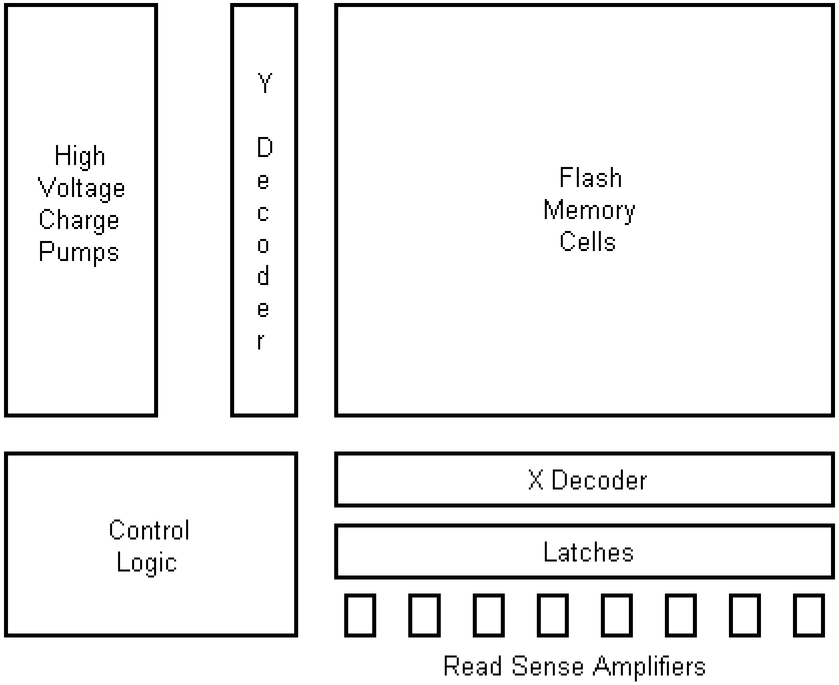
No Readback access
only secure verification

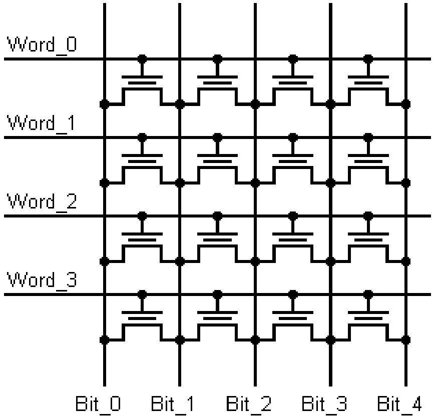
Introduction

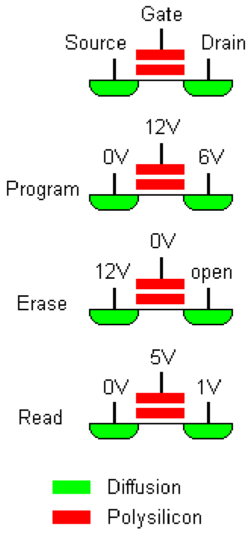
- Flash memory prevails
  - usually stores IP, sensitive data, passwords and encryption keys
  - widely used in microcontrollers, smartcards and some FPGAs
  - non-volatile (live at power-up)
  - reprogrammable
- How secure is Flash memory storage?
  - used in smartcards and secure memory chips
  - used in CPLDs by Xilinx and believed to be highly secure
  - used in secure FPGAs by Actel and claimed virtually unbreakable
- Vulnerabilities of Flash memory found during my research
  - power glitching influence on data read from memory (Web2000)
  - optical fault injection changes data values (CHES2002)
  - laser scanning techniques reveal memory contents (PhD2004)
  - data remanence allows recovery of erased data (CHES2005)
  - optical emission analysis allows direct data recovery (FDTC2009)

Background

- Flash memory structure
  - – high voltages required for operation
  - – narrow data bus
  - – dedicated control logic

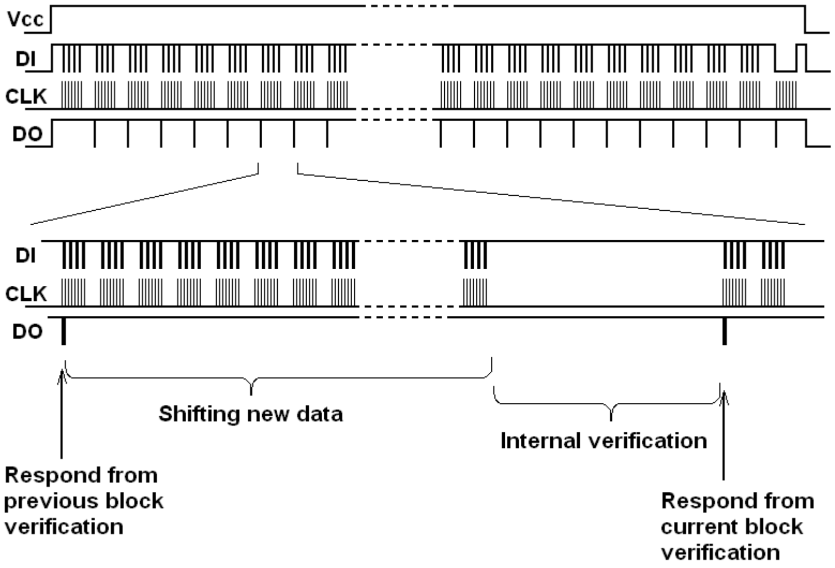


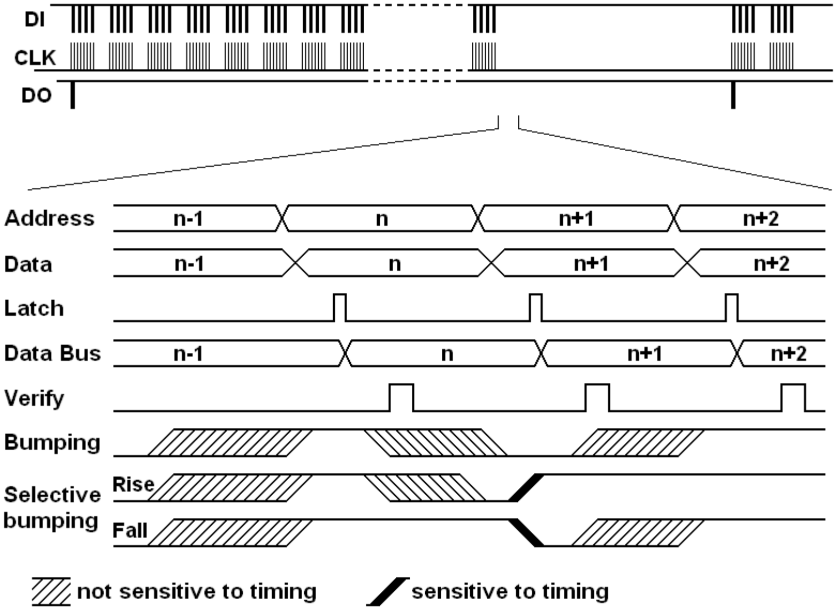




Background

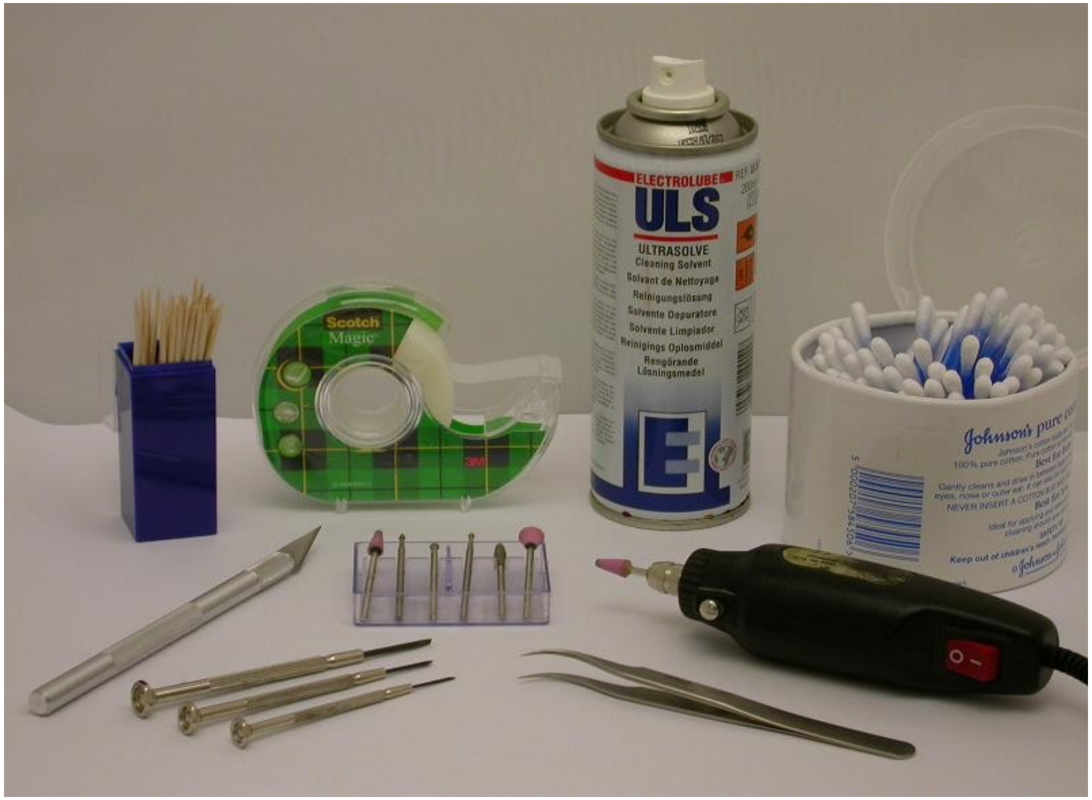
- 'Bumping' is a certain type of physical attack on door locks
- Memory 'Bumping attacks' is a new class of fault injection attacks aimed at internal integrity check procedure in the chip
  - 'bumping' is aimed at blocks of data down to bus width
  - 'selective bumping' is aimed at individual bits within the bus

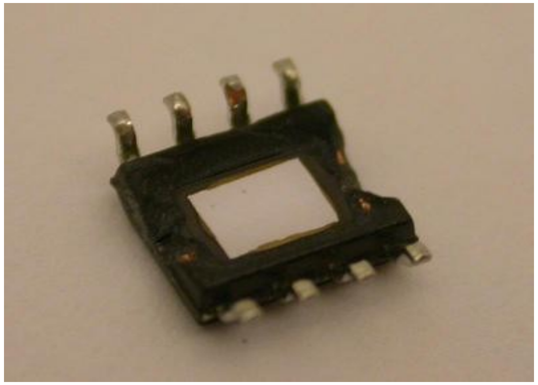

Vcc
DI
CLK
DO
DI
CLK
DO
Shifting new data
Internal verification
Respond from previous block verification
Respond from current block verification

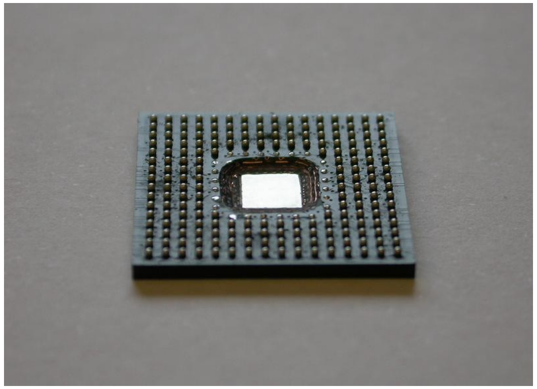


Experimental setup

- Sample preparation for modern chips (<0.5μm and >2M)
  - only backside approach is effective
  - it is very simple and inexpensive
  - no chemicals are required

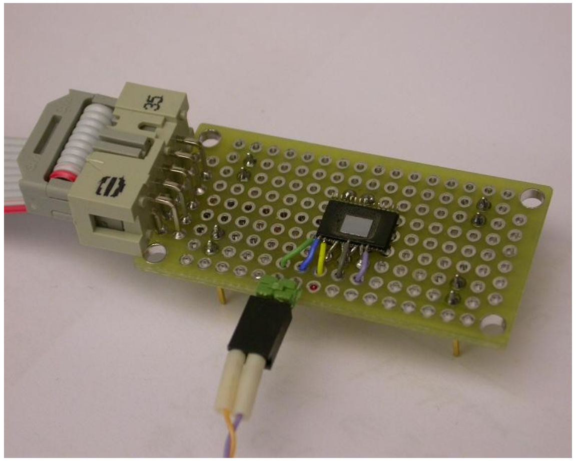


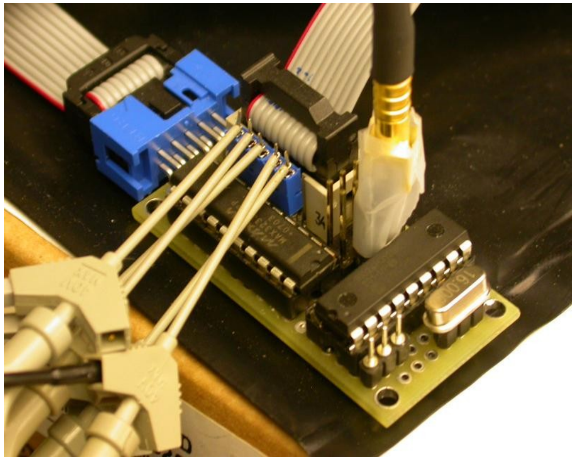


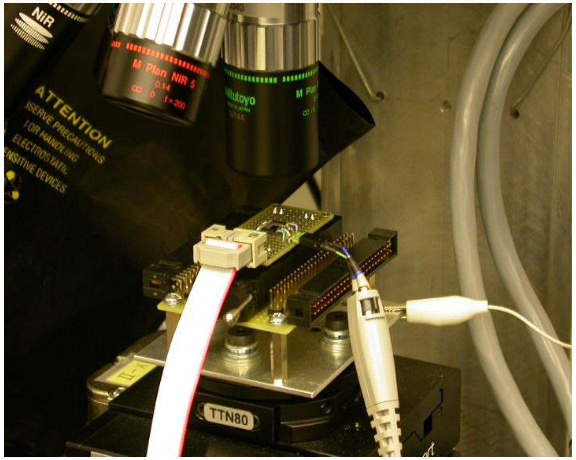


Experimental setup

- NEC 78K/0S µPD78F9116 microcontroller with 16kB Flash
  - memory access via bootloader for Erase, Write, Verify, Blank Check
  - 0.35µm process with 3 metal layers
- Optical fault injection attack
  - 1065nm laser diode module with output power up to 100mW
  - NIR objective lens with 20× magnification



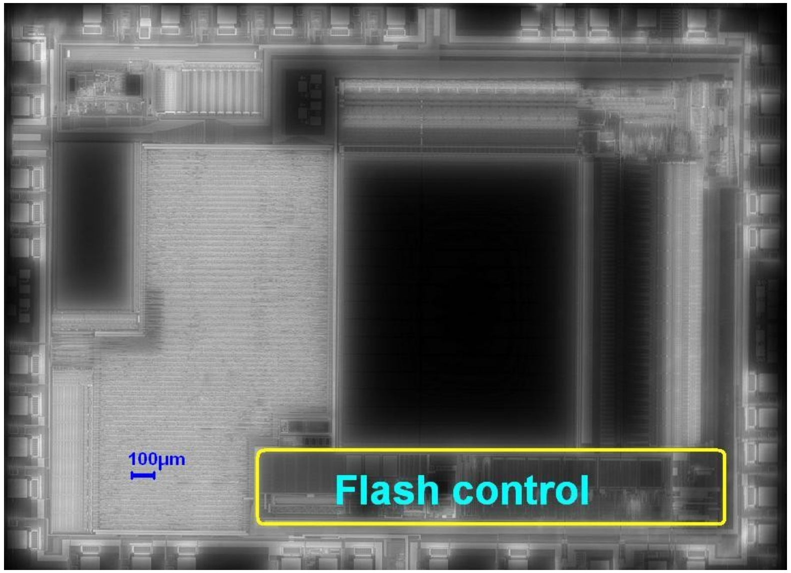


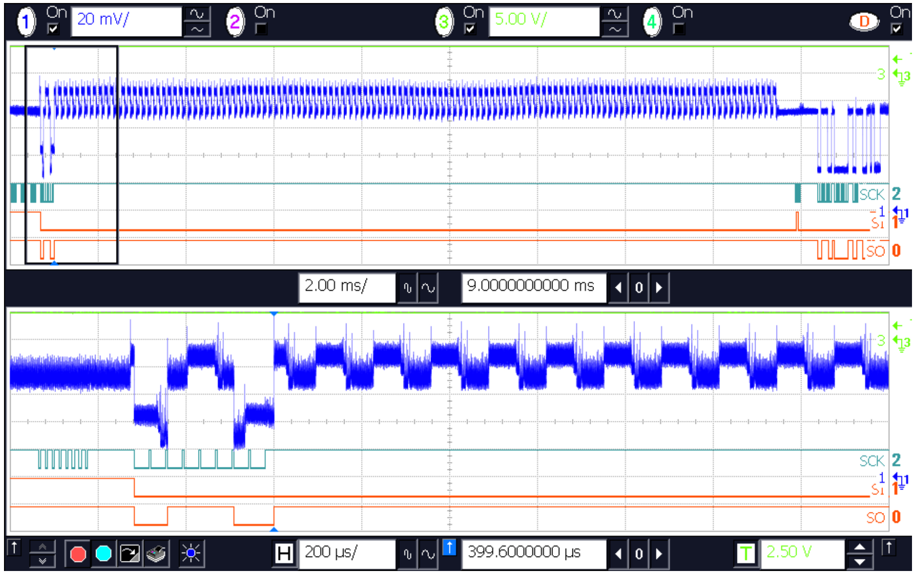

NIR
M Plan NIR 5
Mitutoyo
ATTENTION
OBSERVE PRECAUTIONS FOR HANDLING ELECTROSTATIC SENSITIVE DEVICES
TTN80

Results for bumping

- Locating Flash and active areas is easy (laser scanning)
- SPI interface for data transfer and SPA for timing analysis
- Memory matches all '0' when the laser is switched on
- Verification result is available only after all bytes are compared
- Data extraction time: 10 hours per block, or 2 months per chip

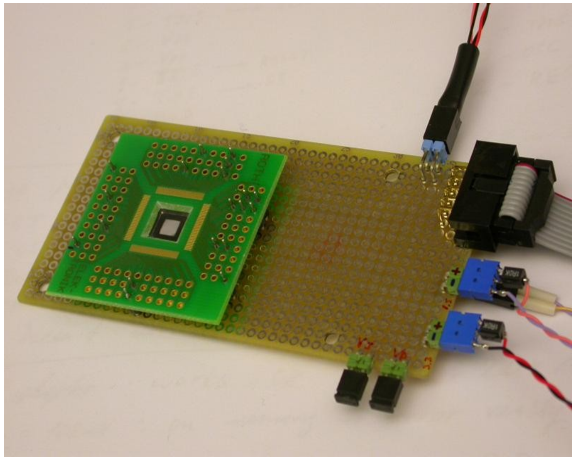
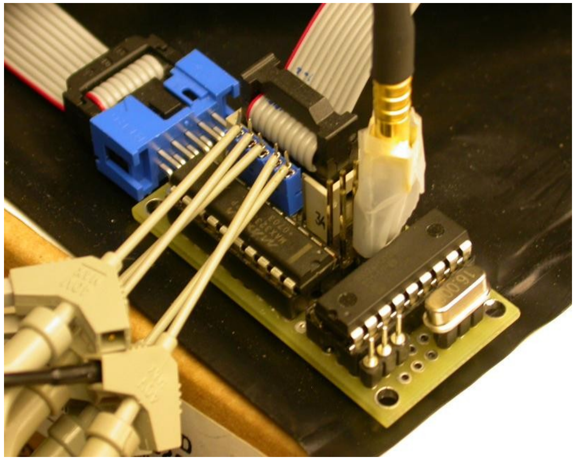
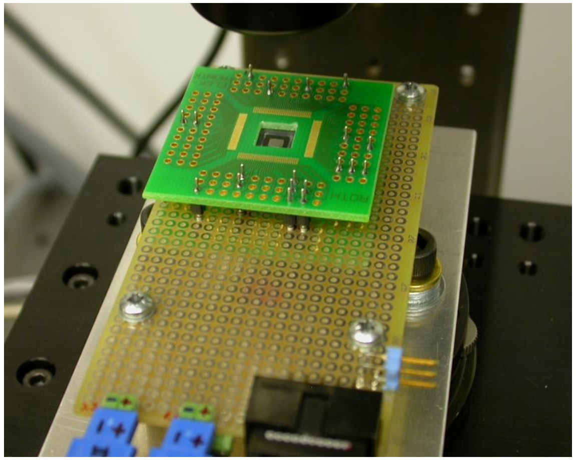
2^7 attempts per byte, 128 bytes per block, 128 blocks, 2s per cycle





Experimental setup

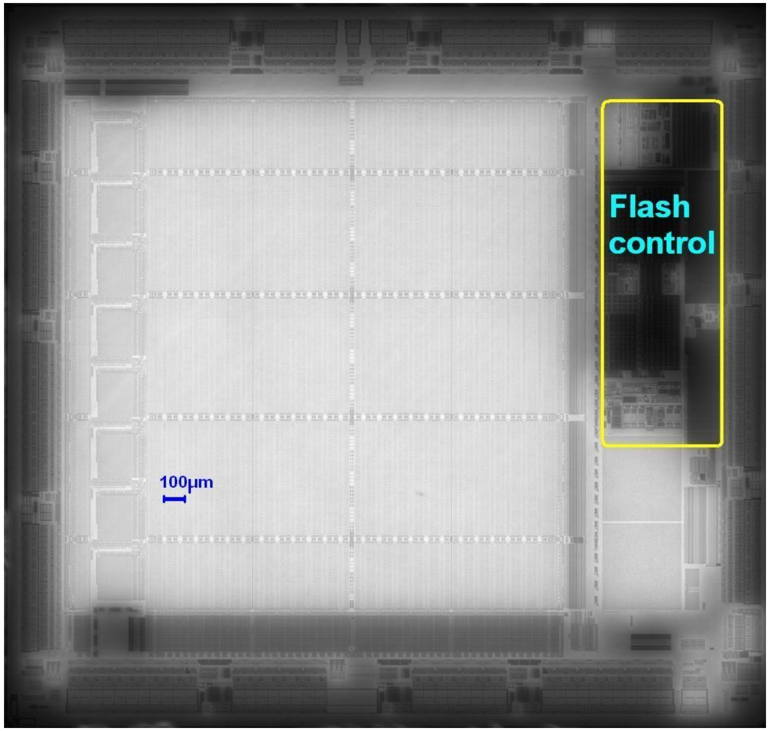
- Actel ProASIC3 Flash-based A3P250 FPGA
  - memory access via JTAG for Erase, Program and Verify operations
  - 0.13µm process with 7 metal layers, limited information is available
  - "...*offer one of the highest levels of design security in the industry*"
  - "*There is NO readback mechanism on PA3 devices*"
  - soon after introduction of optical fault attacks I warned Actel about possible outcomes for Flash technology, but they showed no interest
- Same optical fault injection attack setup

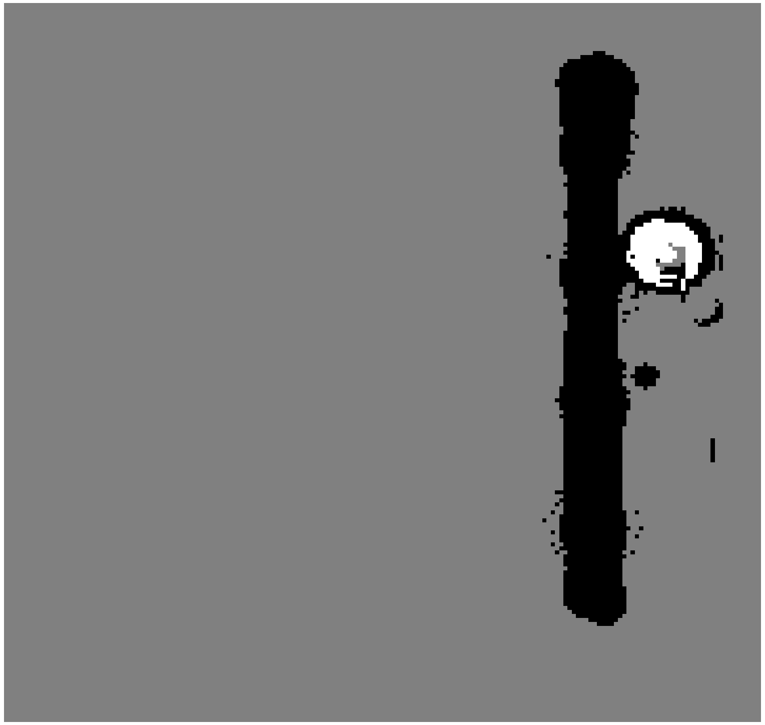




Results

- Locating Flash and active areas is easy (laser scanning)
- JTAG interface for data transfer
- Finding sensitive locations with exhaustive search (20µm)

  black – data corrupted, white – matching all '1'

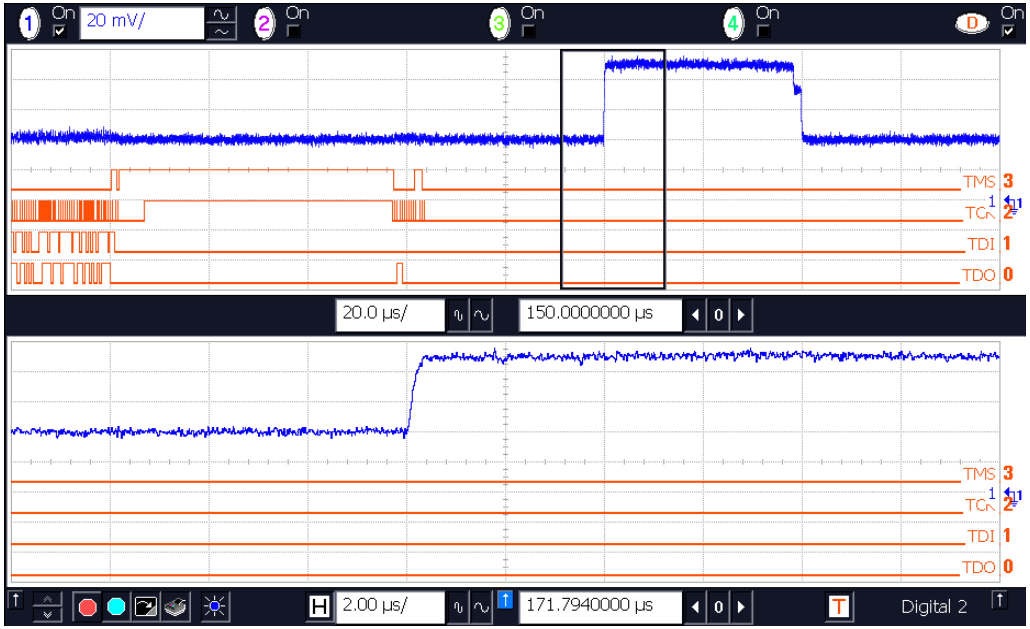




Results for bumping

- Using SPA for timing analysis: cannot detect data timing
- Verification result is available after each block of 832 bits
- 2300 blocks per array, 26 of 32-bit words per block
- Data extraction time: 18 years per block, 40000 years/chip

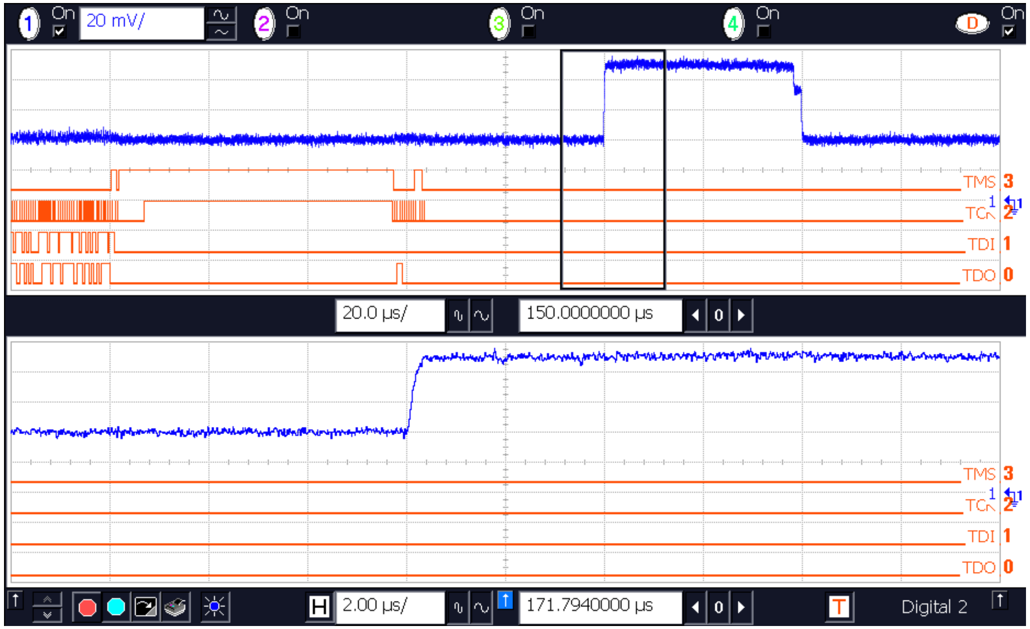
2^{31} attempts per word, 26 words per block, 10ms per cycle



Results for selective bumping

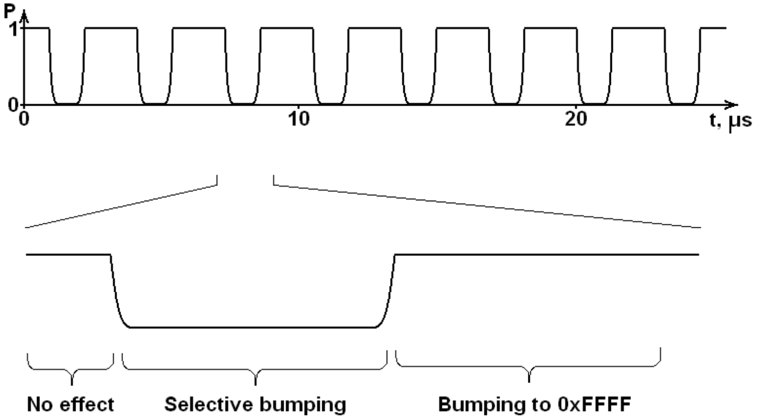
- Using SPA results as a time reference
  - block verification 40µs, 26 of 32-bit words per block, 1.5µs/word
- Laser switching time was changed in 25ns steps
  - searching for single '0' bit, then two '0' and so on until passed
- Data extraction time: 30 minutes per block, 50 days/chip

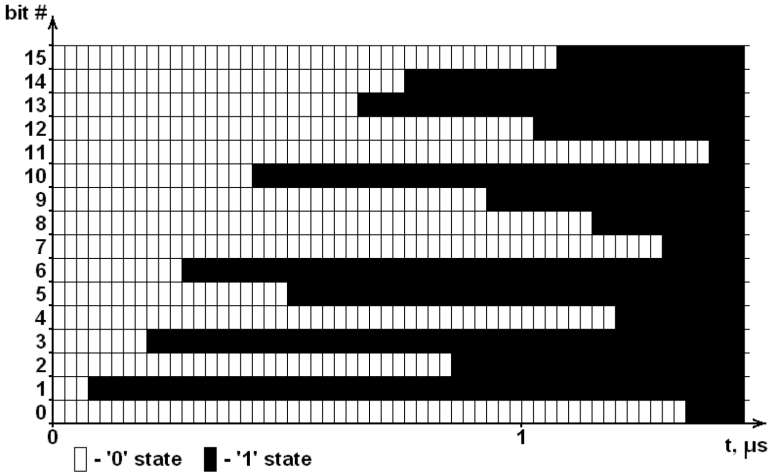
2^{13} attempts per word, 26 words per block, 10ms per cycle



Experimental setup

- Analysis of the selective bumping phenomenon using a secure microcontroller with AES authentication
  - not in production yet, supplied under NDA
  - hardware setup was supplied by industrial sponsor
  - chip was supplied pre-programmed with a test AES key
- Non-invasive power supply glitching attack was used
  - bumping: 2^{15} attempts per 16-bit word, 100ms cycle, 8 hours for AES key
  - selective bumping: 2^{7} attempts per 16-bit word, 2 minutes for AES key





Attack time on 128-bit block

- Without any improvements: brute force search

  requires on average 2^{127} attempts
- Bumping: down to bus width

  8-bit bus: $2^{7} \times 16 = 2^{11}$ attempts

  16-bit bus: $2^{15} \times 8 = 2^{18}$ attempts

  32-bit bus: $2^{31} \times 4 = 2^{33}$ attempts
- Selective bumping: down to single bit in limited steps

  8-bit bus: $(1+8+7+6+5+4+3+2+1)\times\frac{1}{2}\times16 \approx 2^{8}$ attempts

  16-bit bus: $(1+16+15+...+2+1)\times\frac{1}{2}\times8 \approx 2^{9}$ attempts

  32-bit bus: $(1+32+31+...+2+1)\times\frac{1}{2}\times4 \approx 2^{10}$ attempts
- In a real attack the complexity could be higher due to the granularity of the delay time and timing jitter

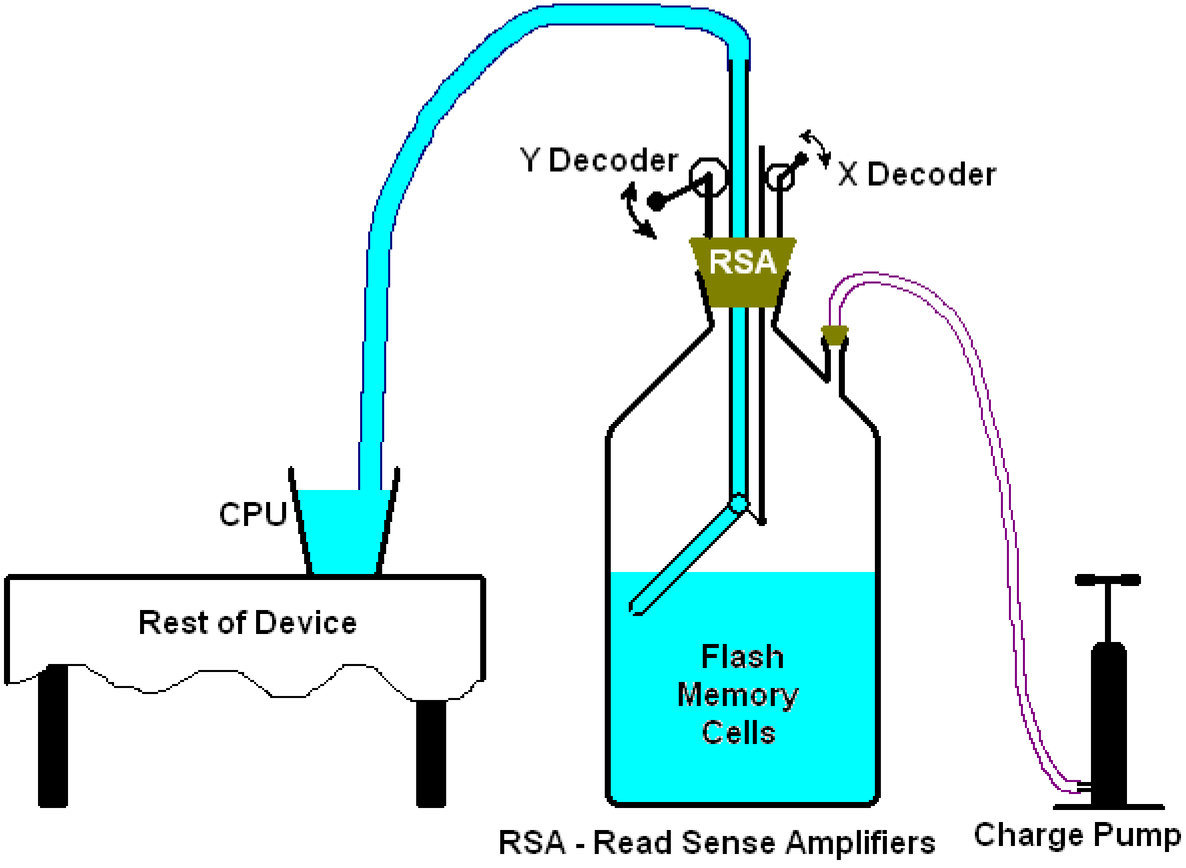
  32-bit bus: $(1+32+31+...+2+1)\times\frac{1}{2}\times4\times8\times4 \approx 2^{15}$ attempts

Limitations and countermeasures

- Slow process
  - depends on the implementation of data verification or authentication
- Precision timing is not necessary
  - slowly increase the delay until the effect is observed
- Selective bumping attacks have partial repeatability
  - between words in the row and between memory rows
- Fault attacks can be carried out with glitching or optically
  - optical attacks on modern chips require backside approach
- Precise positioning for optical attacks is not necessary
- Encryption and redundancy check make analysis harder
- Asynchronous circuits could make the attack more problematic as bumping requires predictable timing
- Understanding the core of a problem is vital

Why Flash memory fails?

- Flash memory in a nutshell – for better understanding
  - can you see the bottleneck(s)?



Improvements and Future work

- Security with no readback is not the only one in ProASIC3
  - passkey access protection, AES encryption, security fuses
- Moving away from semi-invasive attacks toward using non-invasive attacks like in the last example with AES key extraction from the secure microcontroller
  - easier to setup for deep-submicron chips
  - faster to get the result
  - pose larger threat to the hardware security
- Using data remanence effect for bumping through threshold voltage adjustment
  - S. Skorobogatov: Data Remanence in Flash Memory Devices, CHES-2005, LNCS 3659, pp.339–353
- Testing other chips for strength against firmware and secret key extraction

Conclusions

- Bumping attacks are dangerous and can compromise the security in chips – evaluation and protection is necessary
- Backside approach helps in modern chips, it is simple to do and does not require expensive optics and precise positioning
- Bumping attacks can be used for partial reverse engineering to understand internal data paths and chip structure
- If you do not want to get screwed talk to experts in academia; there are solutions for increasing the security of chips