What is Fault Injection?

Fault injection attacks intentionally

cause errors in a system in order to

compromise the security of the system

Non-invasive Fault Injection Attacks

- **Clock Glitching**
  - Burst of double clock speed – timing critical
  - Requires knowledge gained from side-channel attack
  - Prevent flip-flops from latching correct data
  - Prevent security fuses from setting properly
  - Could cause skipping instructions

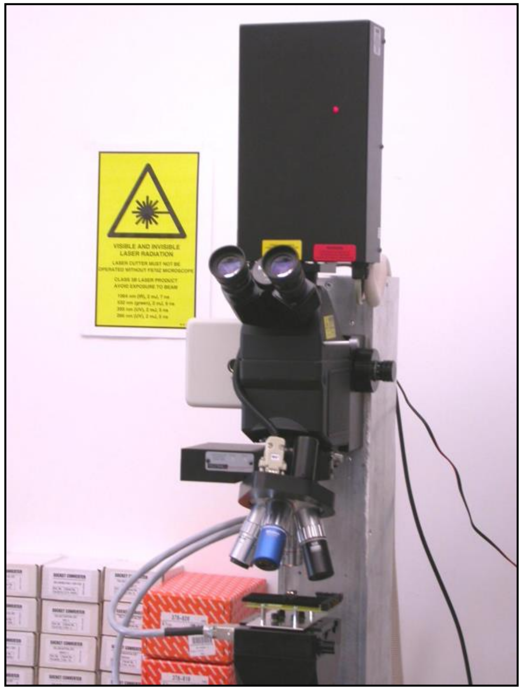
- **Voltage Glitching**
  - Burst of high or low voltage – timing critical
  - Requires knowledge gained from side-channel attack
  - Force VDD < VTH
  - Prevent security fuses from setting properly
  - Change control logic outputs
  - Change memory amplifier outputs

Semi-invasive Fault Injection Attacks

- **Local Heating**
  - High power laser is used to selectively heat small areas
  - Hot enough to change VTH but not hot enough to damage
  - Trial and error with location is used to determine glitches
- **Flash Glitching**
  - Magnified camera flash can cause mass glitching
  - Tinfoil masks created to cause selective glitching
  - Trial and error with location and timing is used to determine glitches
- **Laser Glitching**
  - Infrared laser is used to selectively glitch small areas
  - Trial and error with location and timing is used to determine glitches
  - Process is more precise than Flash Glitching

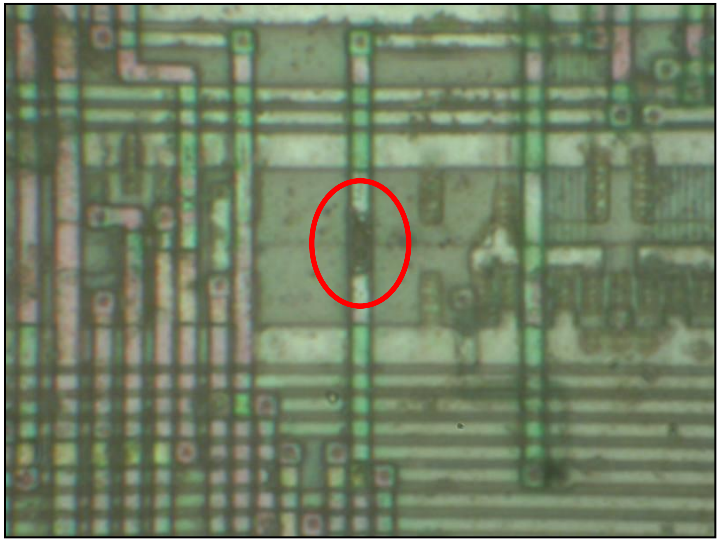
IC Modification

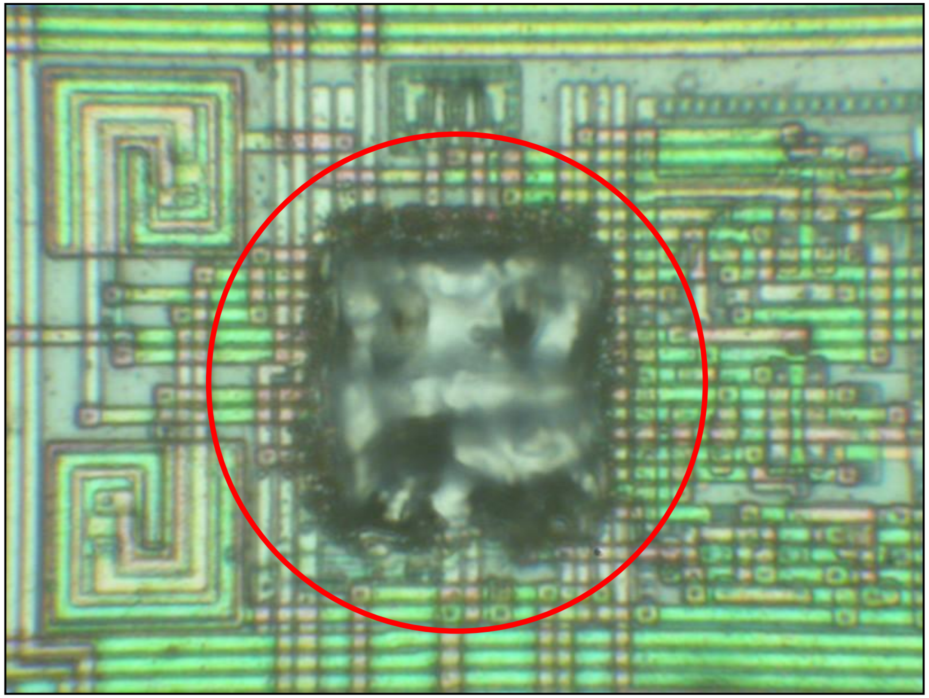
- **Laser Cutting**
  - Not completely destructive
  - Selective exposure of lower layers
  - Selectively disconnect nets
- **Test Point Creation**
  - Cut test points into IC
  - More spots for micro probing below top layer
  - See more signals on more nets
- **Wire Bonding**
  - Use laser cutting to expose net
  - Cut test point for bonding target
  - Modify circuit paths as needed



Source: Skorobogatov. Semi-Invasive Attacks. Page 85

Example of Laser Cutting



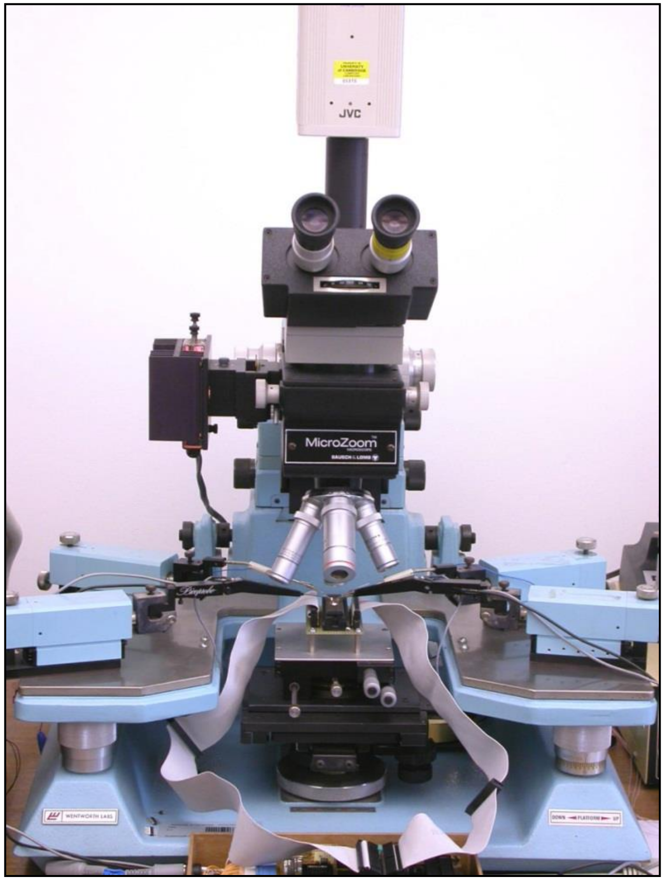


Source: Skorobogatov. Semi-Invasive Attacks. Page 88

Micro Probing

- **Eavesdropping**
  - Listen to control lines
  - Listen to data bus
  - Full bypass of all protections
- **Signal Injection**
  - Insert control signals
  - Modify memory contents
  - Forcefully bypass security controls
- **Fault Injection**
  - High voltage between two probes
  - Destroy transistors
  - Destroy traces

Sample Micro Probing Station



Source: Skorobogatov. Semi-Invasive Attacks. Page 84

Countermeasures

Clock Glitching

- **Internal oscillator for bootloader code**
- **Internal oscillator for secure functions**
- **Make security fuses faster than control logic**
- **Asynchronous logic**

Voltage Glitching

- **Internal brownout reset**
- **Different voltage threshold for security fuses**

IC Modification

- **Metal protection layers on top**
- **Critical signals routed on top of important targets**
- **Tamper sensors in metal layers**

Micro Probing

- **Tamper sensors in metal layers**
- **Small transistor size**
- **Internal shielding**
- **Top level shielding**
- **Security through obscurity**
- **Glue Logic**

Memory Attacks

- **UV Protection**
- **Temperature lockout sensors**
- **Tamper sensors to detect decapsulation**
- **Close proximity between security fuses and memory**

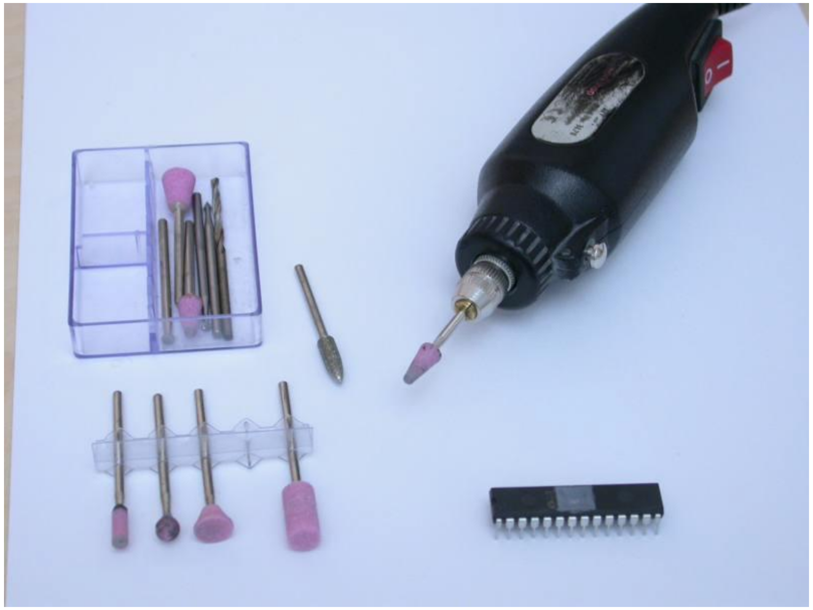
Optical Glitching

- **Protective metal layers to block optical penetration**
- **Tamper sensors in metal layers**
- **UV Protection**
- **IR Protection**
- **Proximity of security fuses and control logic to memory**

Practical Fault Injection Attacks

Step 1: Backside Decapsulation

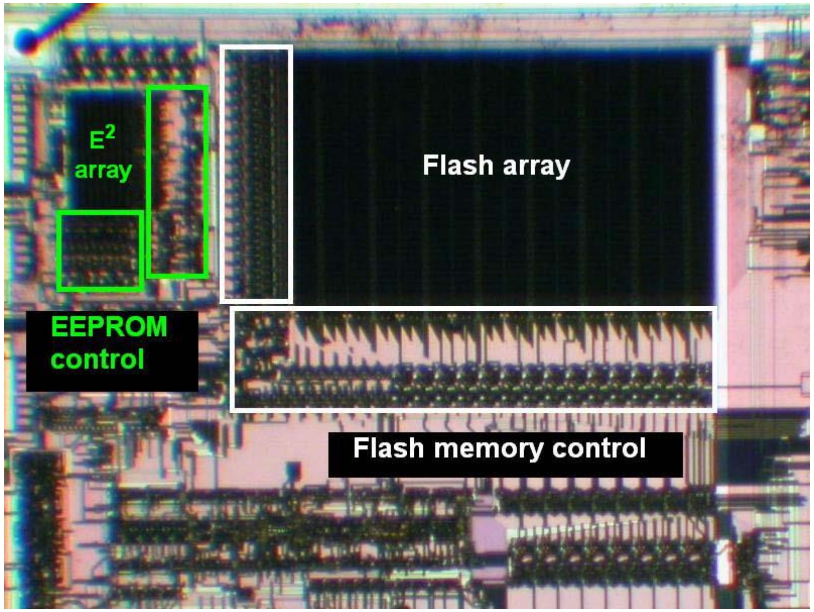
- Use dremel tool to remove backside of outer casing
- Clean surface of exposed substrate material
- Install the IC upside-down to a test interface board



Source: Skorobogatov. Semi-Invasive Attacks. Page 75

Step 2: Backside Imaging

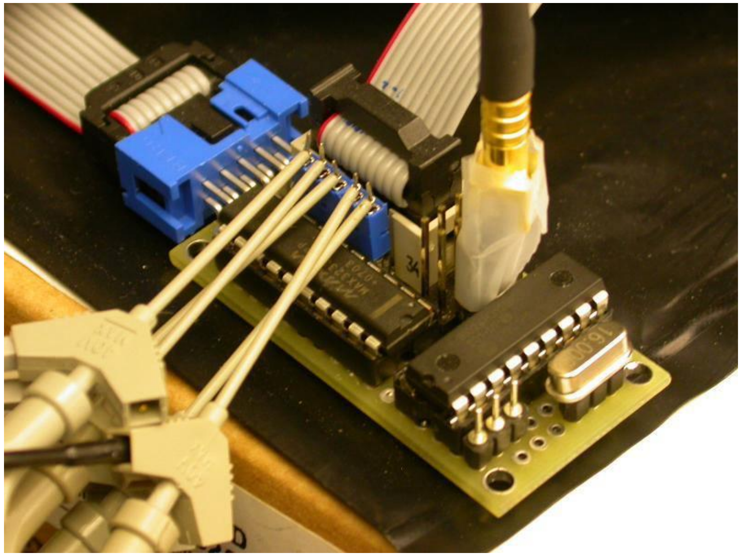
- Use 1000nm infrared light and an optical microscope
- Identify the location of the EEPROM/FLASH memory
- Identify the locations of the memory control logic
- Determine memory bus width



Source: Skorobogatov. Optical Fault Masking Attacks. Page 4

Step 3: Side Channel Attack

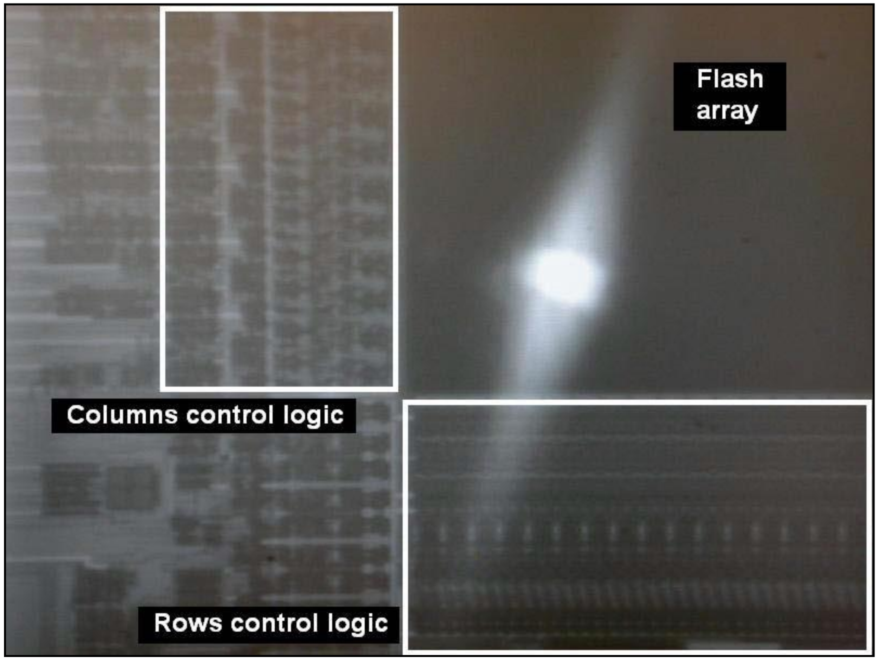
- Set up a power analysis attack using a 10ohm sense resistor
- Perform a Verify function on a dummy input
- Monitor transient current to reverse engineer the process
- Determine packet size of Verify function



Source: Skorobogatov. Flash Memory Bump Attacks. Page 7

Step 4: Laser Glitching Location

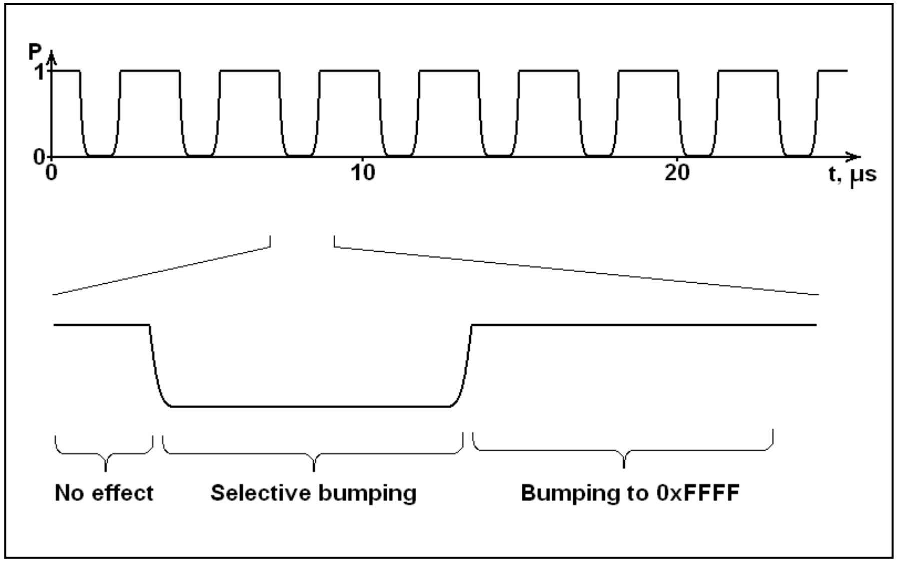
- Set Verify to a pattern of all ‘1’ or all ‘0’
- Find a location in the memory control logic to attack
- Keep trying until your verify pattern succeeds



Source: Skorobogatov. Flash Memory Bump Attacks. Page 5

Step 5: Laser Glitching Timing

- Configure Laser timing to attack all but one block
- Verify that your timing delivers repeatable results
- Maximum unmasked length is the data bus width
- The fewer bits you can unmask at a time the better



Source: Skorobogatov. Flash Memory Bump Attacks. Page 12

Step 6: Brute Force Attack

- Perform a brute force attack on the first unmasked segment
- Unmask the next segment and repeat
- Repeat until all segments are determined

- Example: Verification of a 1024 bit memory on an 8-bit bus
- Traditional Brute Force = 2^1024 Combinations
- Bump Attack = 128*2^8 = 2^15 Combinations

- Example: Verification of a 16384 bit memory on a 16-bit bus
- Traditional Brute Force = 2^16384 Combinations
- Bump Attack = 1024*2^16 = 2^26 Combinations