Low Power Embedded Security

Ingrid Verbauwhede
K.U.Leuven - ESAT - SCD/COSIC

With thanks to:
EMSEC and COSIC/HW team members

E: ingrid.verbauwhede@esat.kuleuven.be
www.esat.kuleuven.be/cosic
www.emsec.ee.ucla.edu

Outline

- Embedded security
- Extra optimization goal: time – area – energy – security
- Security as strong as the weakest link
- Bottom-up:
  - Circuits & logic styles
  - Micro-architecture
  - HW & SW
  - Algorithms & protocols

Embedded Security: Motivation

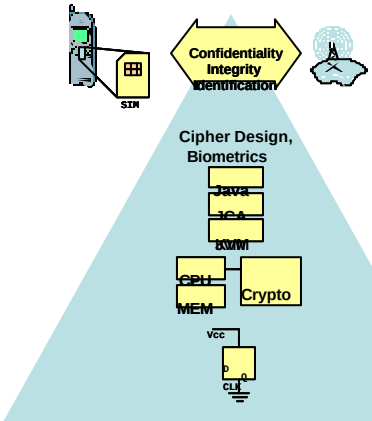
- Ambient intelligence
- PDA's, cell phones, smart cards, gadgets..
- Distributed, communicating, devices

- Secure ?
- Low Energy ?
- Distributed security ?

New York Times (1/24/05):
"A Virus Writer Tests the Limits in Cell phones"
LosAngeles Times (10/14/06):
"Federal Data Theft Found to Affect Millions:
Data Theft at Agencies Not as Uncommon as Hoped"

Embedded Security Pyramid

- **Security is as strong as the weakest link!**



Protocol: Wireless authentication protocol design

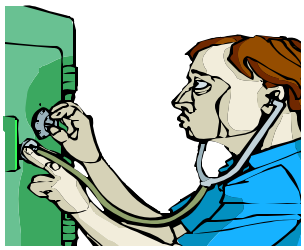
Algorithm: Embedded fingerprint matching algorithms, crypto algorithms

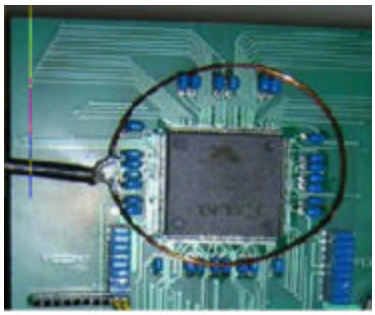
Architecture: Co-design, HW/SW, SOC

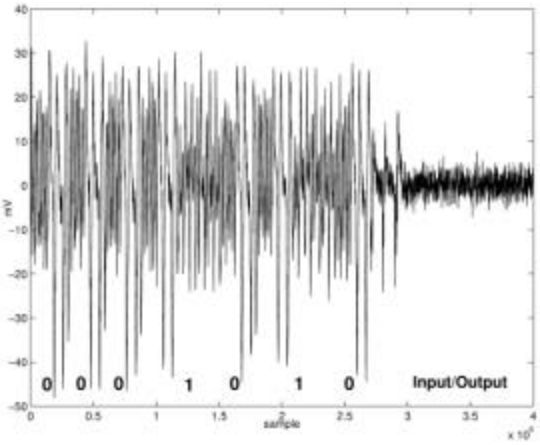
Micro-Architecture: co-processor design

Circuit: Circuit techniques to combat side channel analysis attacks

Side Channel Attacks







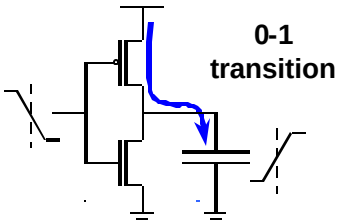
Side channel attacks

- Based on observation of the embedded device: smart-card, RFID tag, FGPA, ASIC, embedded micro-controllers, ….
- Observe: timing, power (= current), electro magnetic variations
- Simple attacks: one or a few measurements, visual inspection often sufficient
- Differential attacks: build a model of the behavior (e.g. the timing or current consumption) and correlate the measurement(s) with the model
- Higher order attacks, template attacks, combined attacks: as countermeasures improve, attacks become more complex
- Countermeasures:
  - At circuit & logic level
  - Micro-architecture level
  - SW level
  - Algorithm level

Foundation: Intro to Static CMOS

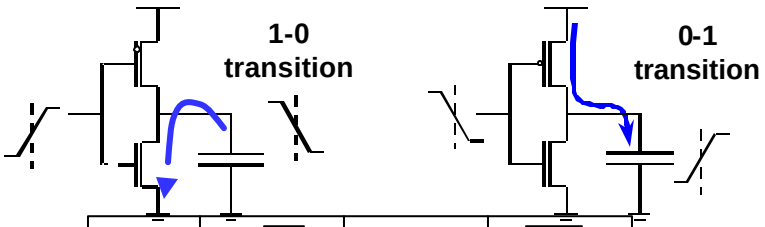
- Consumes power when output makes a 0 to 1 transition

IN	OUT
0→0	0
0→1	discharge
1→0	charge
1→1	0



Duplicate logic

- As suggested by famous cryptographers . . .

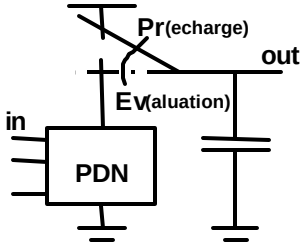


IN	$\overline{IN}$	OUT	$\overline{OUT}$
0→0	1→1	0	0
0→1	1→0	discharge	charge
1→0	0→1	charge	discharge
1→1	0→0	0	0

Dynamic logic

- Dynamic logic breaks input **sequence**
- or hamming distance

IN	OUT_{Pre}	OUT_{EV}	Charge
0→0	1	1	0
0→1	1	0	discharge
1→0	1	1	0
1→1	1	0	discharge



- [Side note: no need to do the reset/precharge with a clock. Can also be done in asynchronous logic or with explicit reset data.]

Transition independent power consumption …

- …doesn't create any side channel information

- When logic values are measured by charging and discharging capacitances, we need to use a fixed amount of energy for every transition

switch a constant load capacitance

switch once every cycle

[Side note: in principle can also be obtained by current mode logic. But extremely hard to realize in practice.]

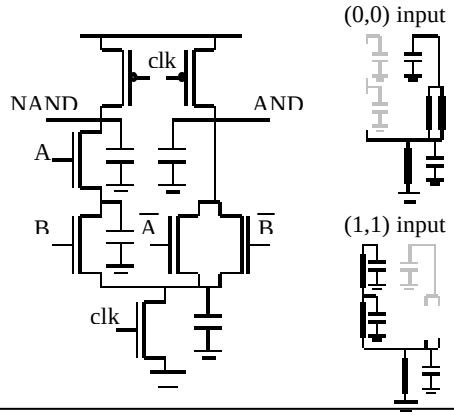
Dynamic and Differential logic …

- is necessary but not sufficient

→Balance differential output nodes

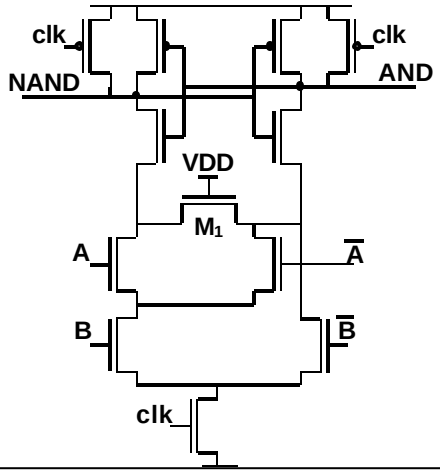
→(Dis)charge all internal nodes

E.g. DCVSL is not sufficient

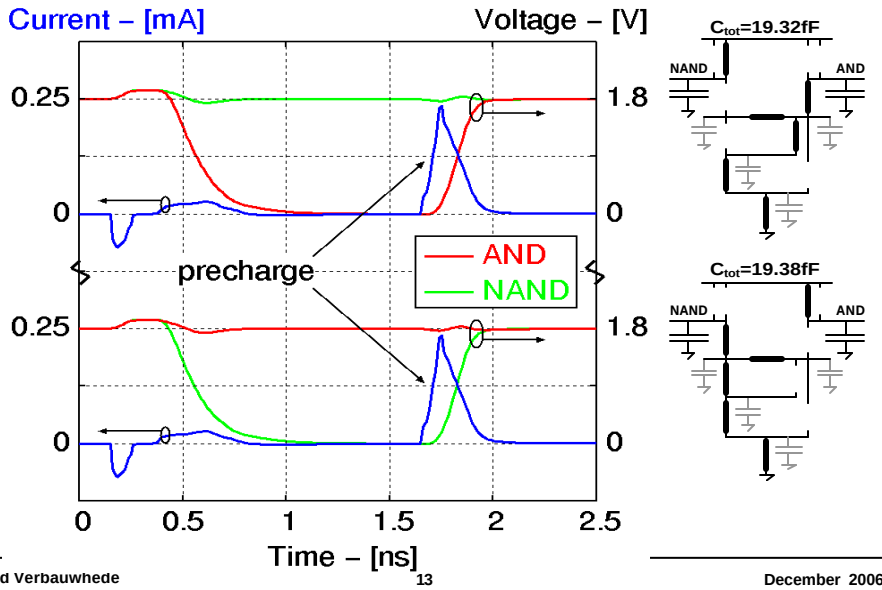


Sense Amplifier Based Logic **charges each cycle a constant load**

- Balanced input and output nodes
- All internal nodes connect to an output

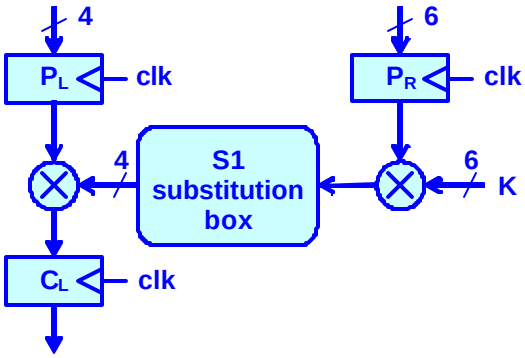


Sense Amplifier Based Logic



Experimental setup

- DPA on module of last round DES



Selection function D(K,C)

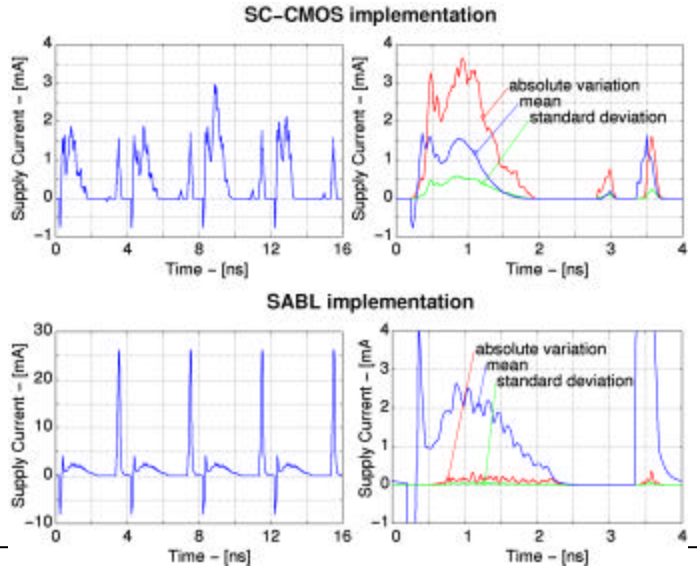
predicts 1st bit of P_L. K guessed. C known.

DPA: *"Power measurements are partitioned over 2 sets based on guess of secret key. Difference between typical supply currents of sets has noticeable peaks if guess was correct."*

Implementation details

- Same circuit; two implementations.
- Difference in logic style:
  - static CMOS
  - SABL
- 0.18µm, 1.8V CMOS technology
- 5000 encryptions
- Hspice with 10ps simulation step

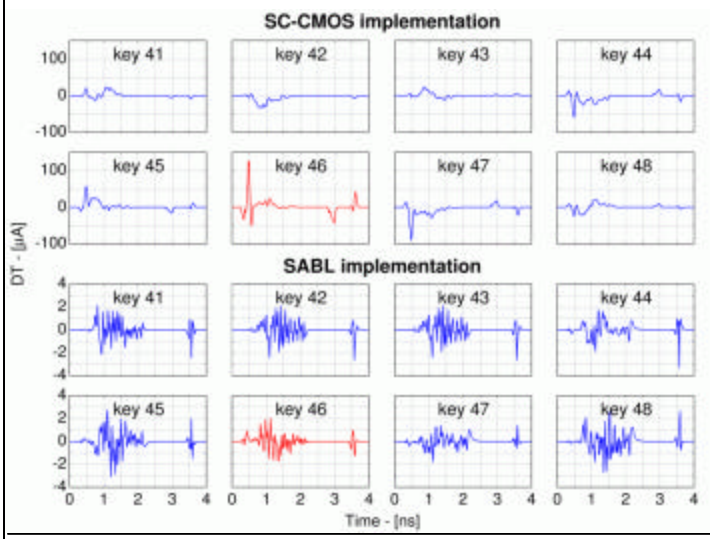
Supply current profile



- irregular ⇒ input dependent
- regular ⇒ input independent

[Tiri CHESS2003]

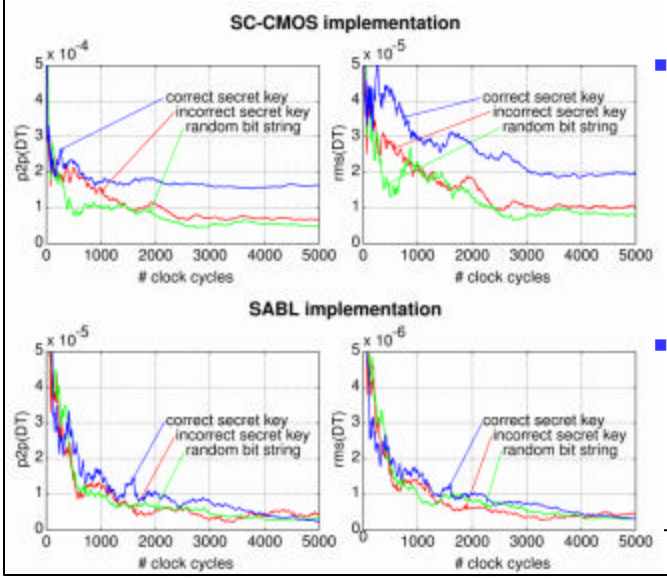
DPA – differential trace



- secret key stands out
- secret key does not stand out

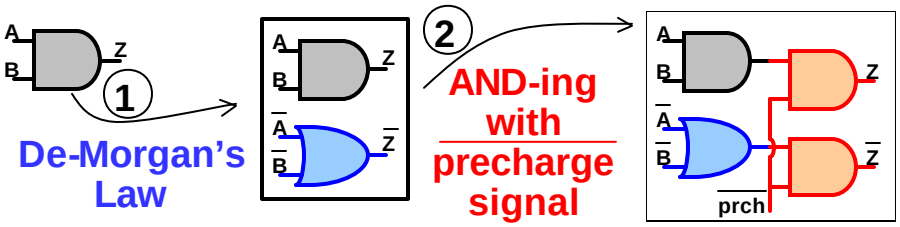
[Tiri CHESS2003]

Measurements to disclosure



- 200 cycles sufficient to disclose key
- transient response has died out

Standard building blocks

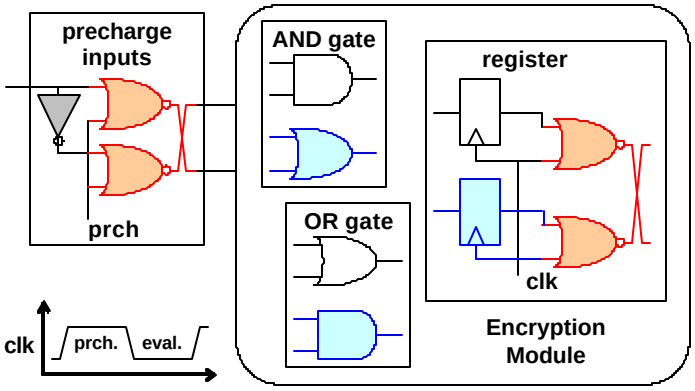


- false output
- with false inputs

- precharge 1: outputs are 0
- precharge 0 - evaluation: 1 output is 1

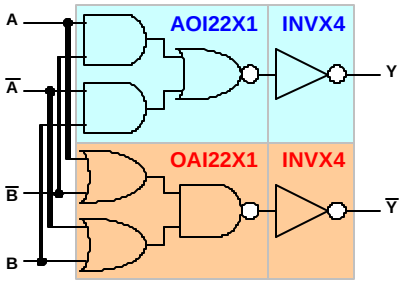
Wave Dynamic Differential Logic

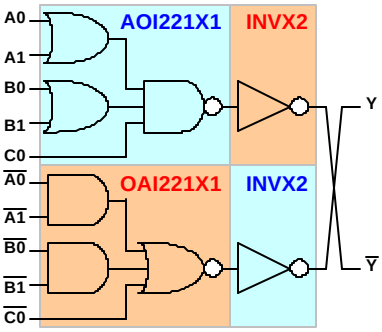
- Restrict library to AND, OR gate
  - input 0 ⇒ output 0
  - no precharge operator



WDDL library

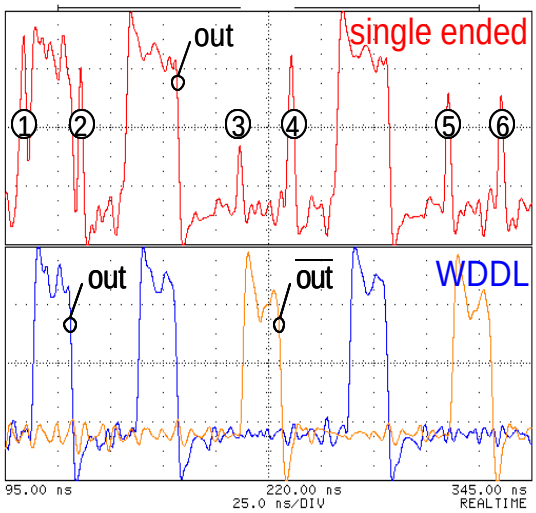
- All functions of and2, or2 operator
- In addition: inverted input, output signals
- XOR2X4: OAI221X2:
- Our WDDL library: 128 cells





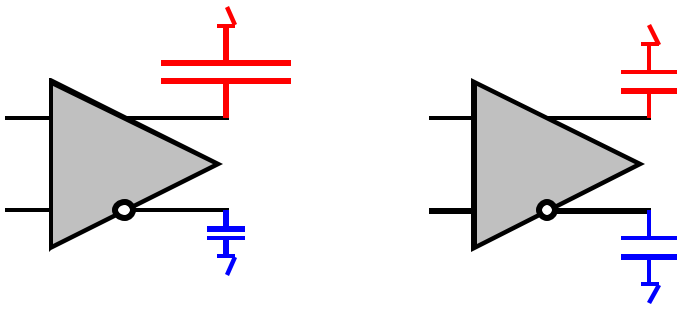
Experimental results

- Measurement results for FPGA test circuit

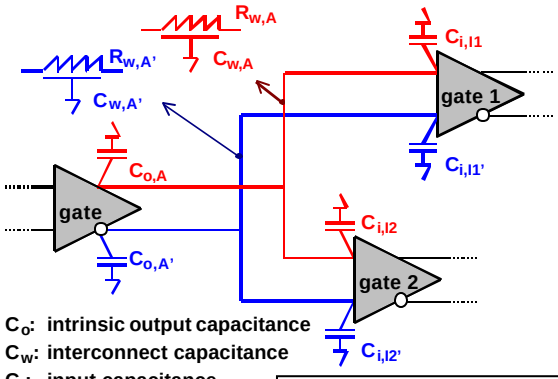


Unbalanced capacitive loads

- For constant power consumption: constant load capacitance.
- Match loads at differential outputs.



Load capacitance breakdown



C_o: intrinsic output capacitance
C_w: interconnect capacitance
C_i: input capacitance

- Intrinsic caps.: matched
- Interconnect: dominant (Moore's law)
- Balancing interconnect: crucial

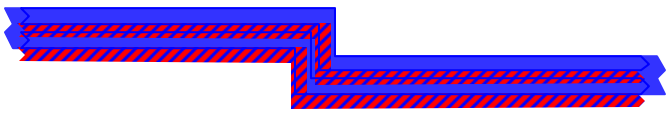
$$CA = CA'$$
$$Co,A + Cw,A + Ci,I1 + \ldots Ci,Ik = Co,A' + Cw,A' + Ci,I1' + \ldots Ci,Ik'$$
$$Cw,A = Cw,A'$$

Place & Route approach

- Parallel routes (adjacent tracks, same layer) balance geometric distances, parasitic effects
- Resistance: equal vias, wire segments
- Capacitance (to other layers):
  ideally same environment
  exact if every other layer is a power plane

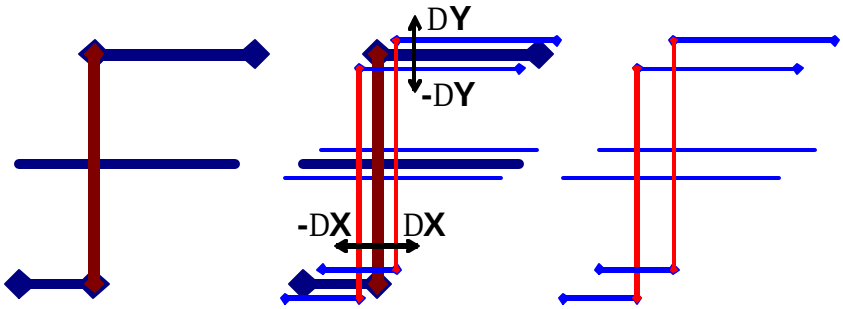
Differential pair routing

- Available via gridless/shape-based routers.
  - only few critical signals (e.g. clock)
  - experiment with 200 pairs:
    8 hours CPU, 1000 conflicts, 100 open nets.
- Gridded routers avoid wires in parallel.
- We propose "fat"-wire routing.
  - Abstract differential pair as one single fat wire.
  - Route with fat wire; then decompose into pair.

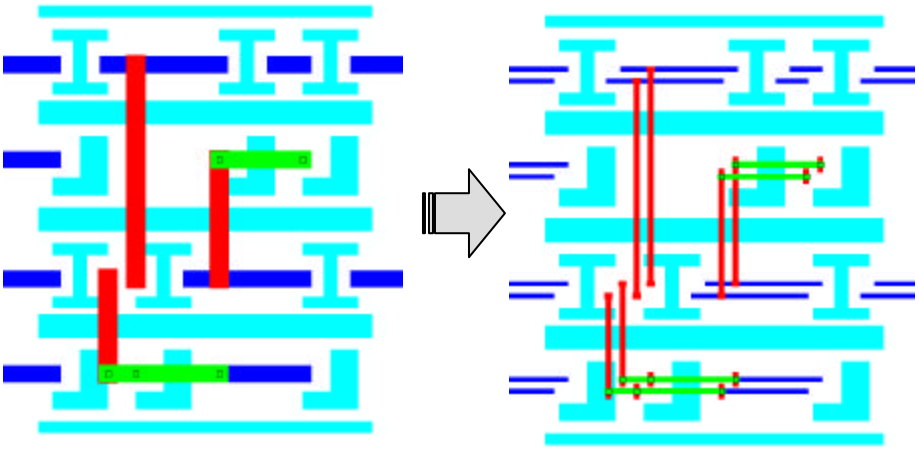


Fat wire decomposition

1. Duplicate fat wire.
2. Slide apart copies.
3. Reduce to normal width.



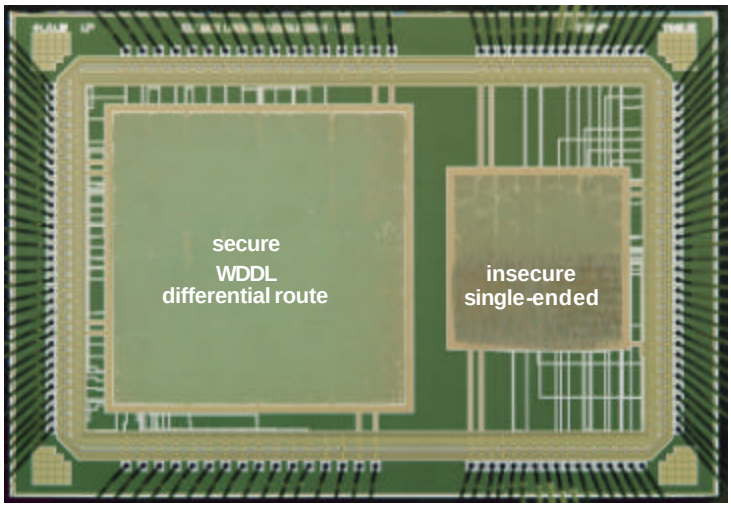
Design example



- Two normal wires replace each fat wire.

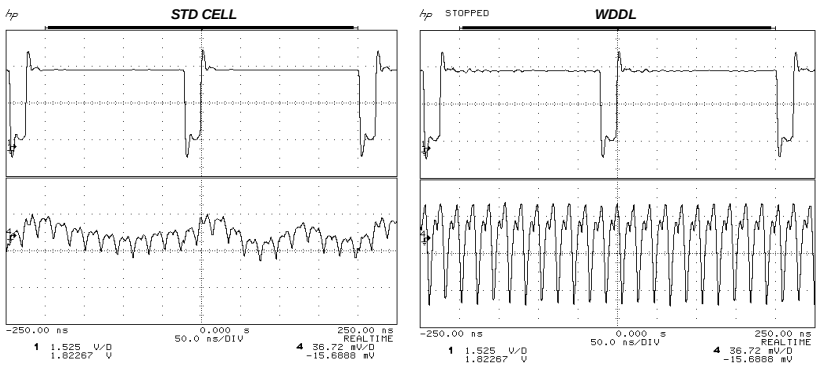
Prototype IC – ThumbPodII

- AES, controller, fingerprint processor.

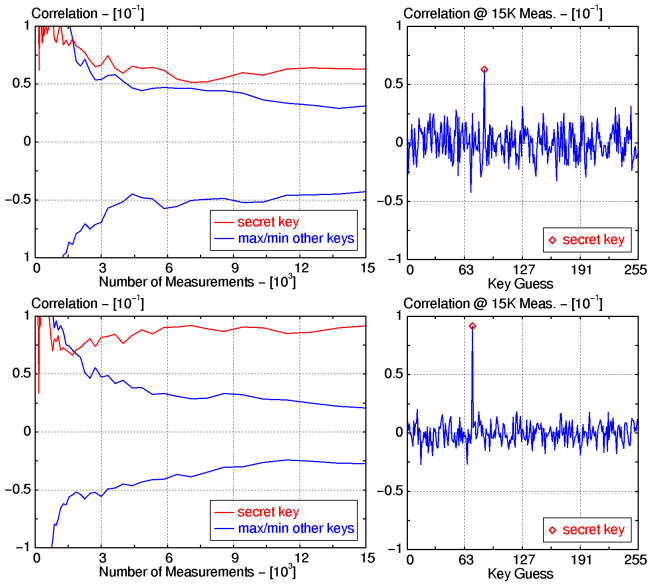


Circuit techniques to address SCA

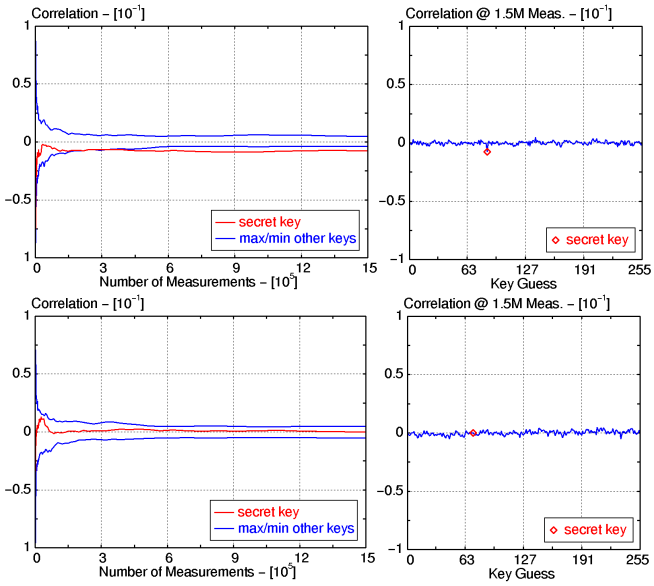
- Standard cells: break AES with 8000 encryptions
- Special cells (build from standard cells): over 1.5M encryptions and still not broken



DPA attack on AES key bytes- SCMOS



DPA attack on WDDL

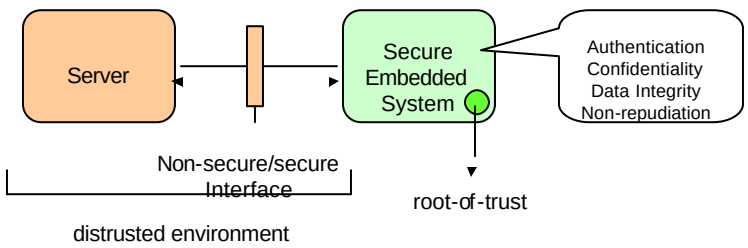


Security partitioning

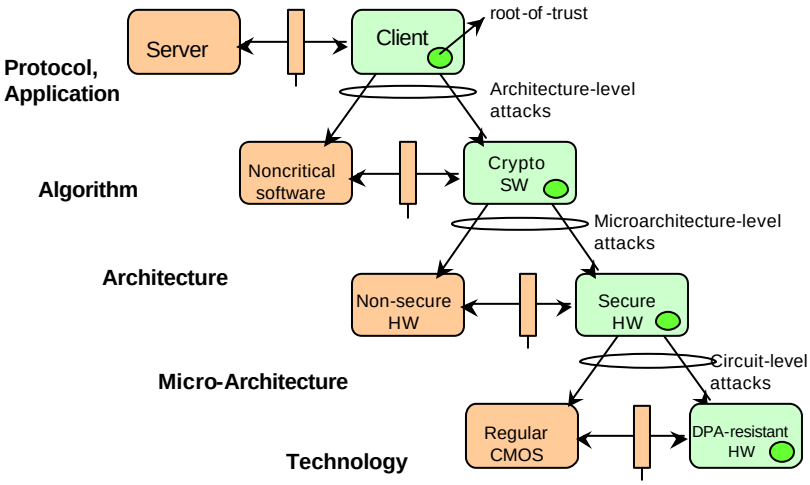
- Security at circuit level:
  - Back-end flow “automated” creates secure circuits
- But:
  - Area + energy cost
- Security partitioning

Embedded Security

- Security in an embedded system



Systematic Design Method: tree of trust

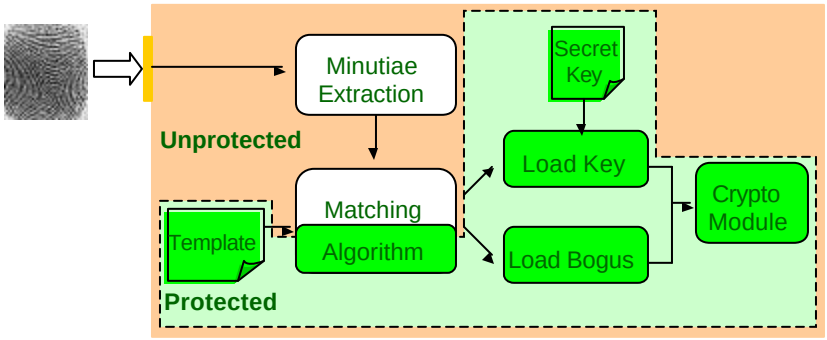


Example Application: ThumbPod

- Intelligent secure keychain device that recognizes owner biometrically
- Components:
  - Microcontroller with memory
  - Fingerprint sensor
  - Biometric signal processing
  - Security processing
- Communication: IR and USB
- Applications:
  - Secure credit cards, secure memory, access control, etc.

[UCLA work]

Security Partitioning

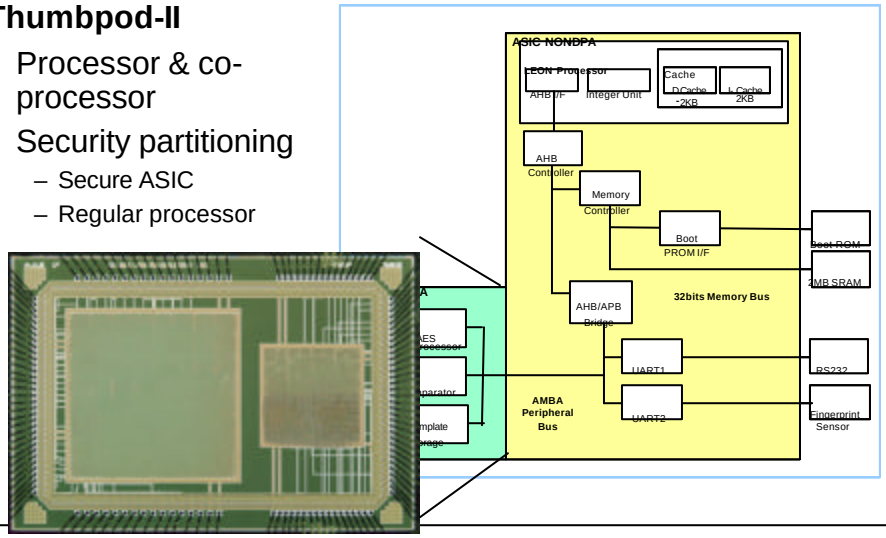


- **Only the sensitive template and the corresponding processes need to be protected.**
- **Fuzzy vault avoids storage of sensitive material but also a price in terms of performance and cost**

Security partitioning for Thumpod-II

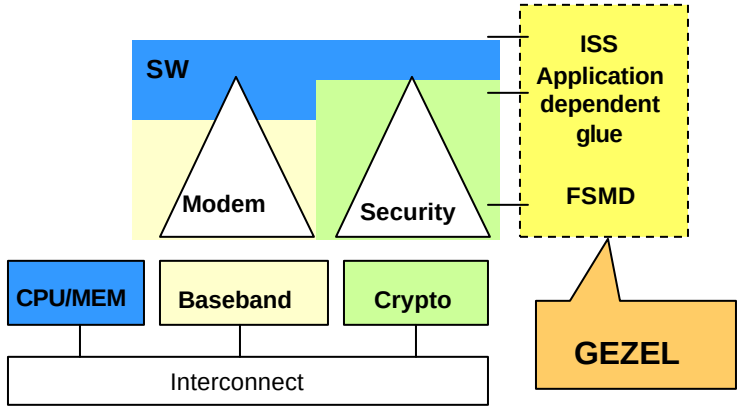
Thumbpod-II

- Processor & co-processor
- Security partitioning
  - Secure ASIC
  - Regular processor

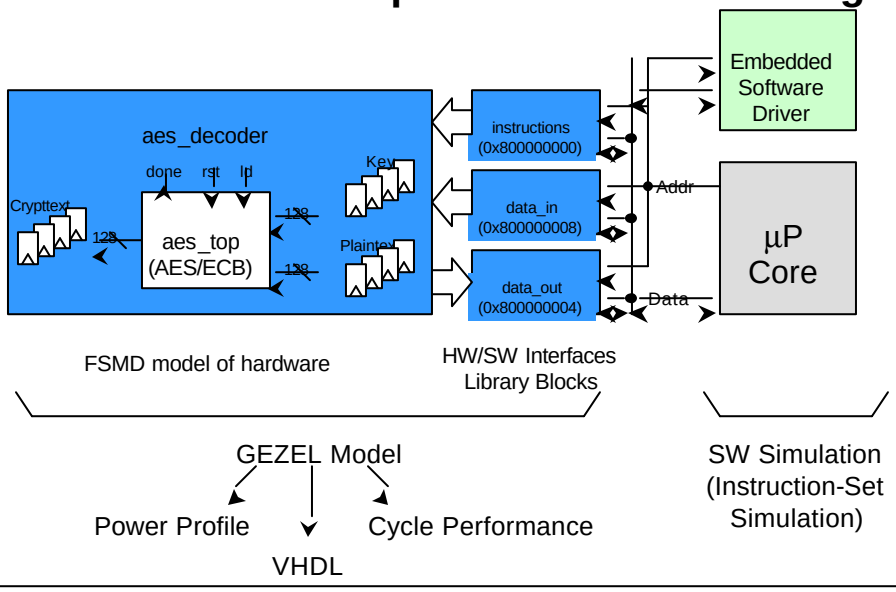


Support for security partitioning

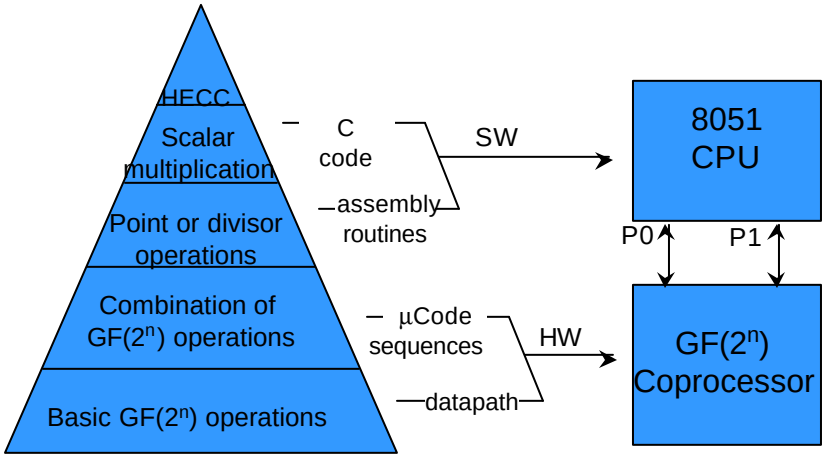
Hardware – Software co-design
SOC = embedded CPU with programmable co-processors



Example of a GEZEL codesign



Public Key: ECC/HECC HW/SW co-design

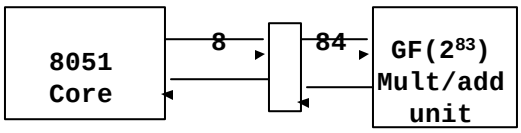


GEZEL based design [CHES 2005]

HW/SW for HECC

- 8051 8-bit micro controller
- With/without hardware acceleration

From the Tiny



3300 LUTs
820Bytes RAM
12KBytes ROM

+ 480 LUTs
+ 100Bytes RAM

192 s
@ 12MHz

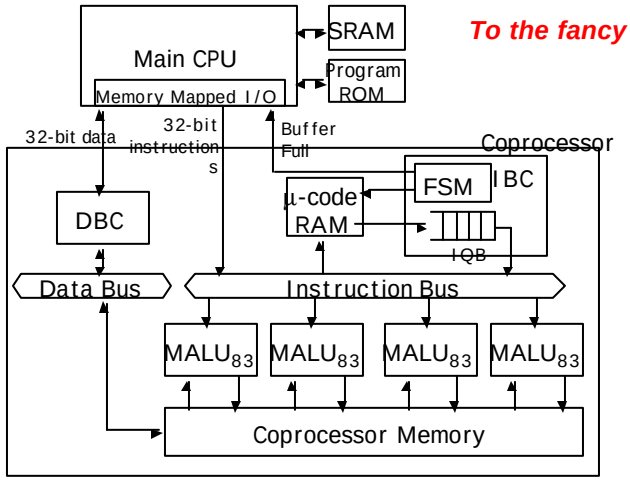
X 80

2.5 s
@ 12MHz

[Ches2005]

HW/SW for superscalar co-processor

- Superscalar
- ARM CPU



To the fancy

[Ches2006]

Embedded Security

Protocol: Security – Communication – Computation trade-off

Algorithm: Security partitioning

Architecture: RINGS & Gezel, HW/SW Co-design

Micro-Architecture: co-processor design

Circuit: WDDL & Diff routing, Secure memory

Embedded Security is NOT a point solution!