

Side channel attacks: countermeasures

Ingrid Verbauwhede

Ingrid.verbauwhede-at-esat.kuleuven.be

K.U.Leuven, COSIC



Acknowledgements:
Patrick Schaumont,
Kazuo Sakiyama,
Kris Tiri,
Lejla Batina

Outline & Goal

- Part I: side channel attacks
- Part II: countermeasures
 - Algorithm
 - Architecture
 - Circuits

Remember: Embedded Security

NEED BOTH

- **Efficient Implementation**

- Within power, area, timing budgets
- Public key: 1024 bits RSA on 8 bit μC and 100 μW
- Public key on a passive RFID tag



- **Trustworthy implementation**

- Resistant to attacks
- Active attacks: probing, power glitches, JTAG scan chain
- Passive attacks: monitor electromagnetic radiation

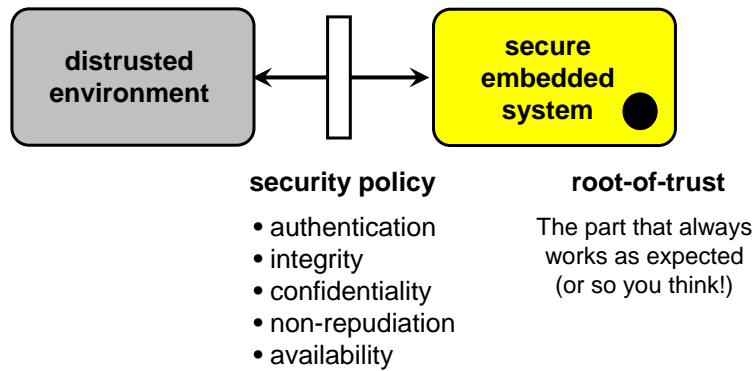


Security as design dimension

Secure design methods

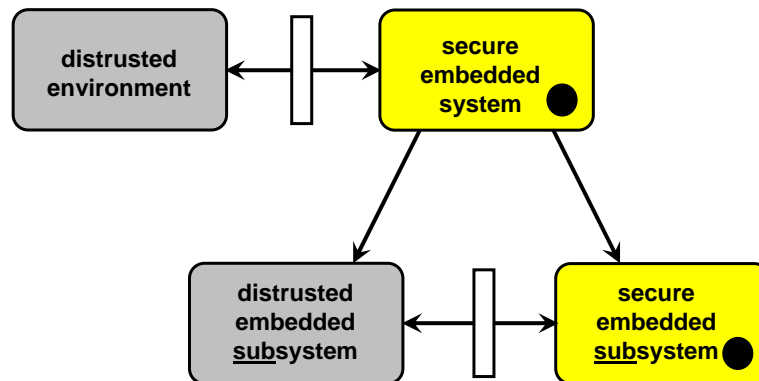
The starting point: Root of Trust

- A secret in a box by itself is useless (useless like a key without a matching door-lock)
- So a secure system contains at least two parts:



[slide courtesy: P. Schaumont]

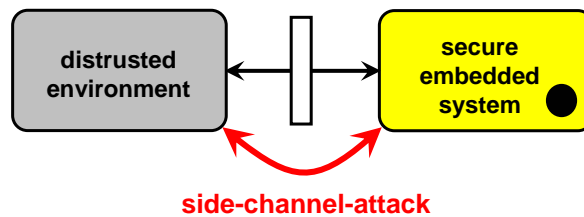
Design Step: Secure partitioning



[slide courtesy: P. Schaumont]

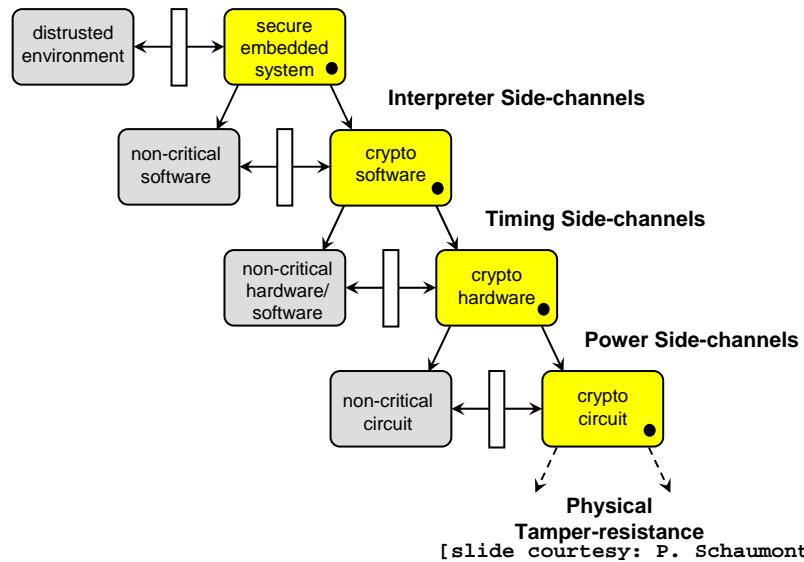
What drives secure partitioning?

- Minimize footprint of the root-of-trust
 - Reduce the physical size of the root-of-trust
 - Make the root-of-trust dynamic and short-lived
- Minimize the risk for *side-channel-attacks*
 - A side-channel-attack is an interface into the root-of-trust around the security policy



[slide courtesy: P. Schaumont]

Hierarchical approach



Countermeasures: Classification

- According to their application level:
 - Hardware countermeasures
 - Algorithmic (Software) countermeasures
 - Protocol countermeasures
- According to their applicability:
 - Algorithm dependent
 - Algorithm independent

Hardware Countermeasures

- Are typically algorithm independent
- Different methods:
 - Add (increase) noise
 - Introduce random delays
 - Use special logic styles
- Typically try to target the link between the side channel leakage and the data which is processed

Algorithmic Countermeasures

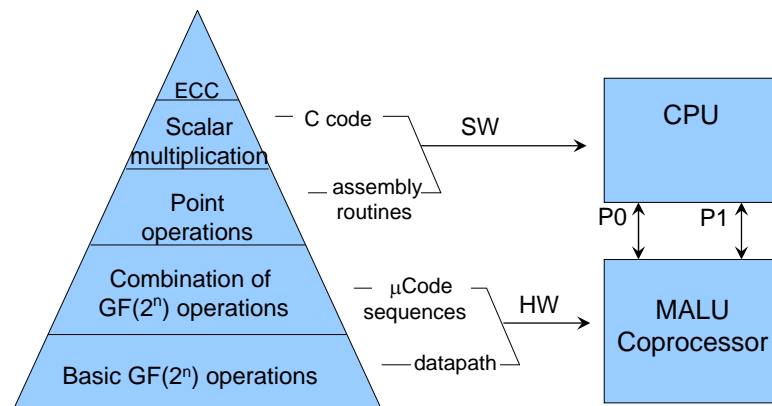
- Idea: intermediate values are randomized (blinded, masked), hence they target the link between the model and the data which is processed
- Are typically implemented in software
- Some are very much algorithm dependent:
 - Asymmetric Cryptography: Blinding
 - Symmetric Cryptography: Masking
- Some are less dependent:
 - Randomization of sequence of instructions
 - Indistinguishable instructions
- Some are independent:
 - Random dummy instructions (similar to random delays)

Protocol Countermeasures

- Put restrictions on the amount of executions of an algorithm
- Basis scenario of session key establishment for smart cards:
 - Master Key MK
 - Random Value R_i
 - One-Way Function F
 - Derivation of the i^{th} session key: $Sk_i = F(MK, R_i)$
 - If one session key is compromised, neither the master key, nor the other session keys are compromised

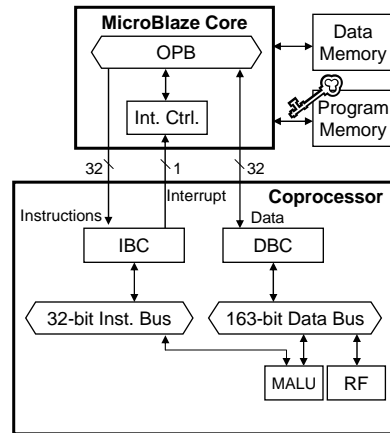
Architecture design

Public Key: security partitioning

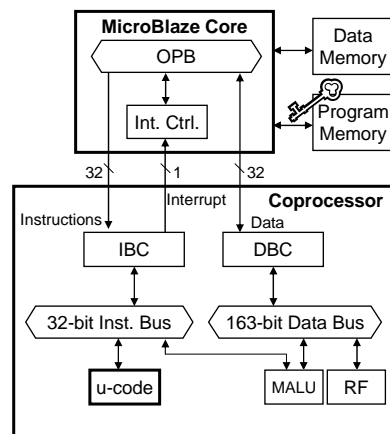


Where is boundary between SW and HW?

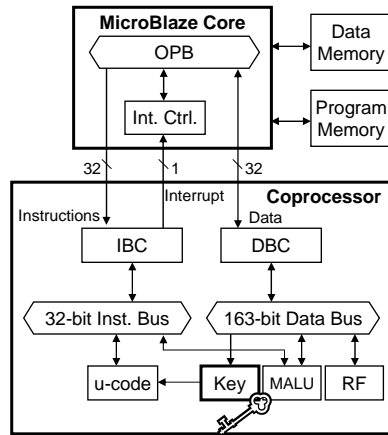
[CHES 2005]



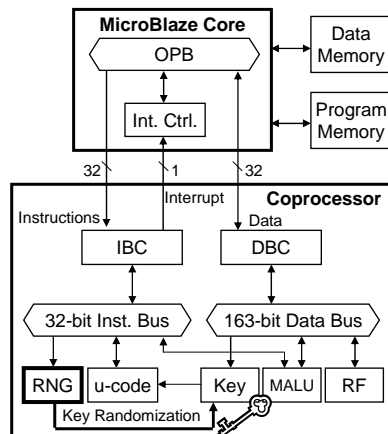
Option 1: 163 bit data path



Option 2: 163 bit data path & instruction sequence decode



Option 3: 163 bit data path & instruction sequence decode & local storage of key



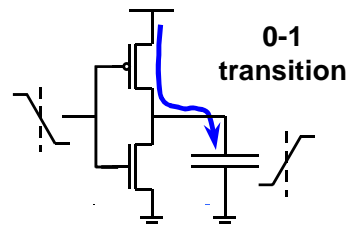
Option 4: 163 bit data path & instruction sequence decode & local storage of key & Key randomization

Circuit level countermeasures

Intro to Static CMOS

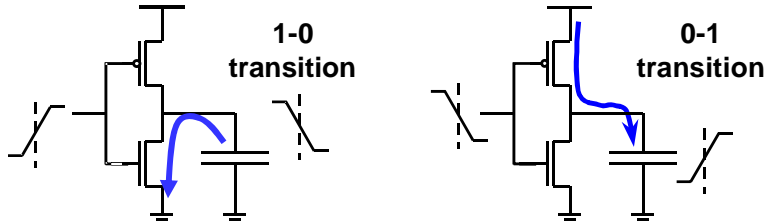
- Consumes power when output makes a 0 to 1 transition
- Most popular circuit style!

| IN | OUT |
|-----|-----------|
| 0→0 | 0 |
| 0→1 | discharge |
| 1→0 | charge |
| 1→1 | 0 |



Duplicate logic

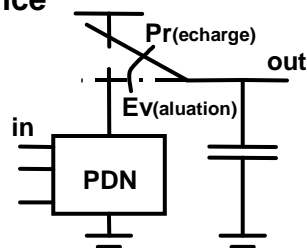
- As suggested by famous cryptographers . . .



| IN | $\overline{\text{IN}}$ | OUT | $\overline{\text{OUT}}$ |
|-----|------------------------|-----------|-------------------------|
| 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

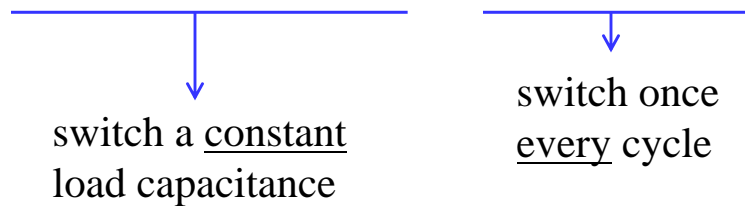
- Dynamic logic breaks input **sequence**



| IN | OUT _{Pr} | OUT _{EV} | Charge |
|-----|-------------------|-------------------|-----------|
| 0→0 | 1 | 1 | 0 |
| 0→1 | 1 | 0 | discharge |
| 1→0 | 1 | 1 | 0 |
| 1→1 | 1 | 0 | discharge |

Transition independent power consumption ...

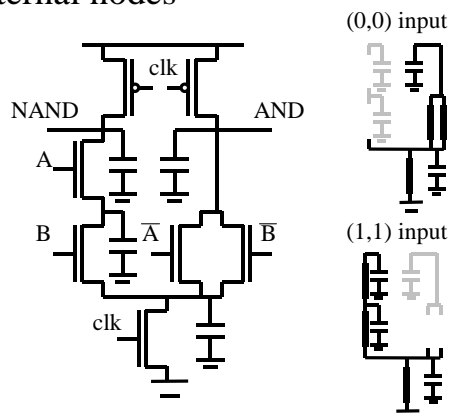
- ...doesn't create any side channel information
- When logic values are measured by charging and discharging capacitances, we need to use a fixed amount of energy for every transition



Dynamic and Differential logic ...

- is necessary but not sufficient
- Balance differential output nodes
- (Dis)charge all internal nodes

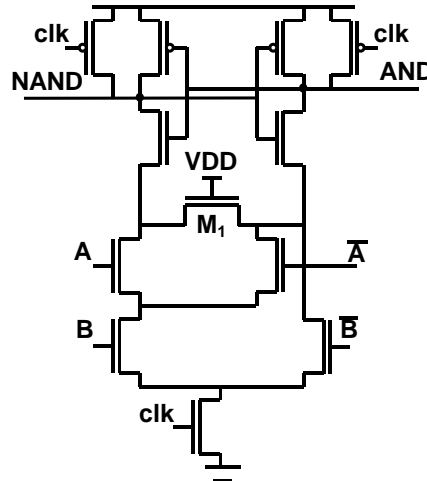
→
E.g. DCVSL
is not
sufficient



[Tiri, ESSCIRC02]

Sense Amplifier Based Logic charges each cycle a constant load

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

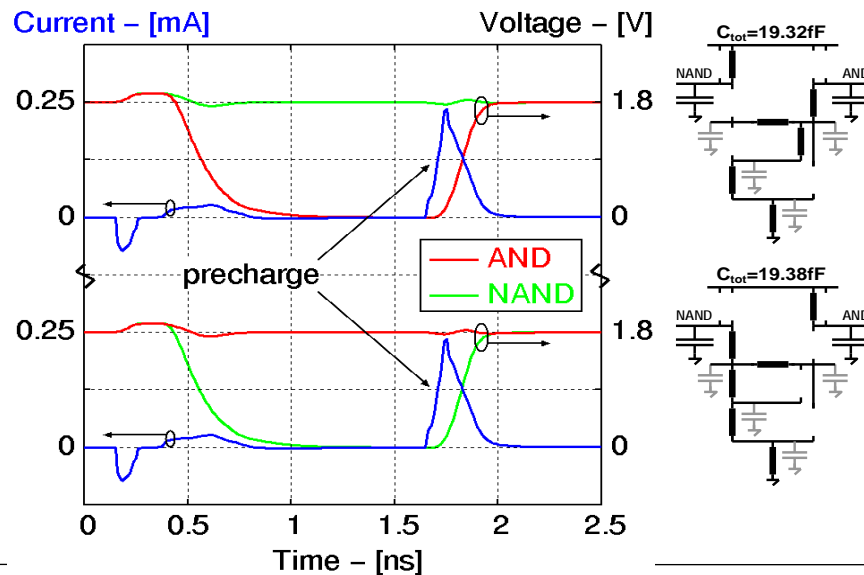


KUL - COSIC

Tutorial – part II - 25

Chennai, December 2007

Sense Amplifier Based Logic



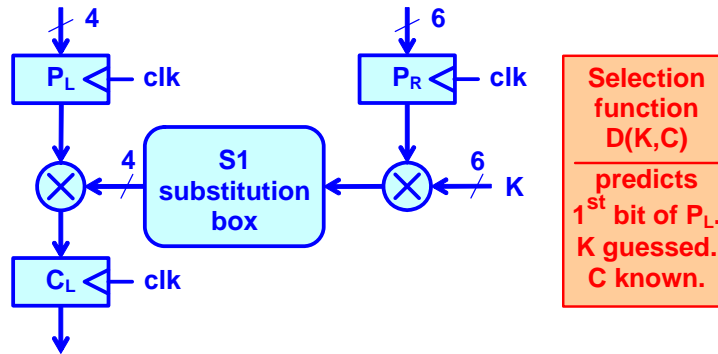
KUL - COSIC

Tutorial – part II - 26

Chennai, December 2007

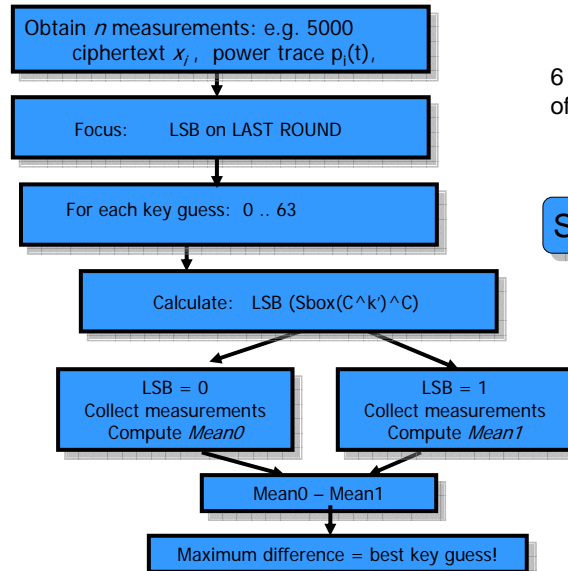
Experimental setup

- DPA on module of last round DES



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."

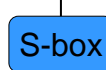
HSPICE DES Sbox in SABL



Classical 1-bit DPA

6 bits of right half of the 'plaintext' for this round

6 bits of round key



4 left bits

4 cipher bits

5000 measurements *
time window t *
256 key guesses

Implementation details

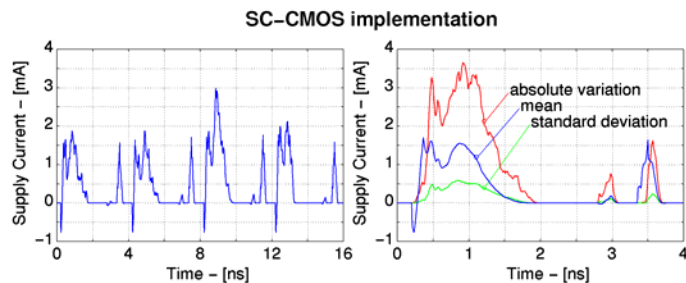
- Same circuit; two implementations.
- Difference in logic style:
 - static CMOS
 - SABL
- 0.18 μm , 1.8V CMOS technology
- 5000 encryptions
- Hspice with 10ps simulation step

KUL - COSIC

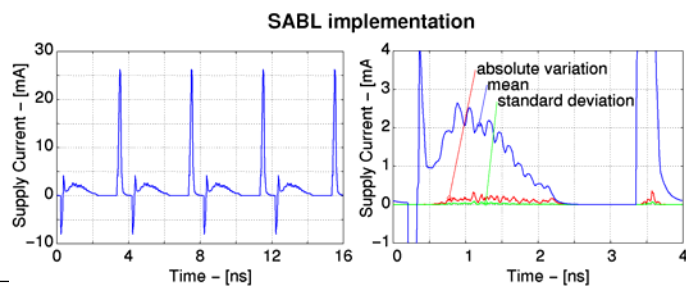
Tutorial – part II - 29

Chennai, December 2007

Supply current profile



- irregular
⇒ input
dependent



- regular
⇒ input
independent

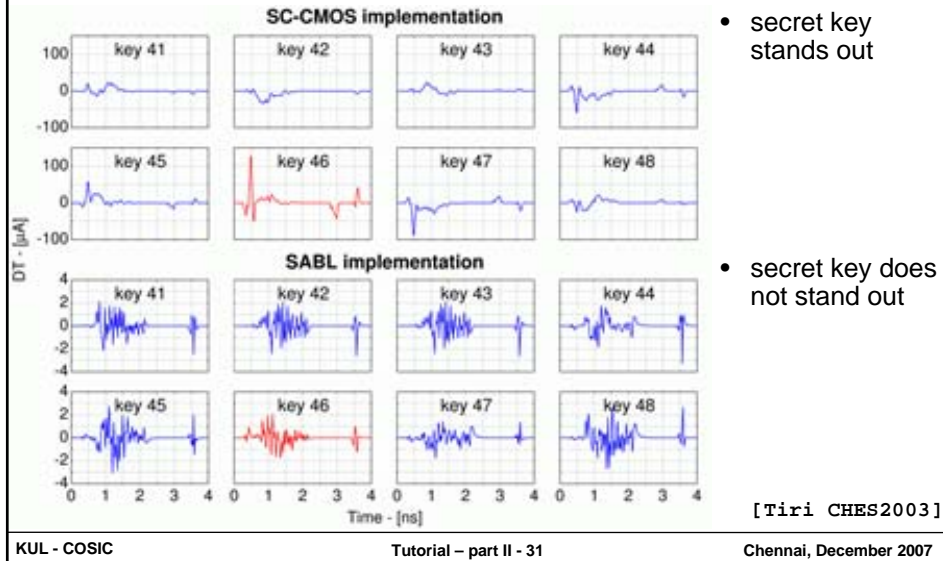
[Tiri CHES2003]

KUL - COSIC

Tutorial – part II - 30

Chennai, December 2007

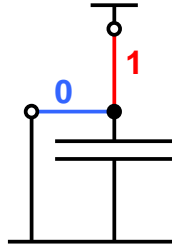
DPA – differential trace



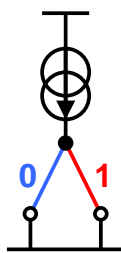
Measurements to disclosure



Logic families



- Voltage Mode Logic
 - fixed amount of charge
 - including events in which gate does not change state



- Current Mode Logic
 - perfect current source
 - major drawback: static power consumption

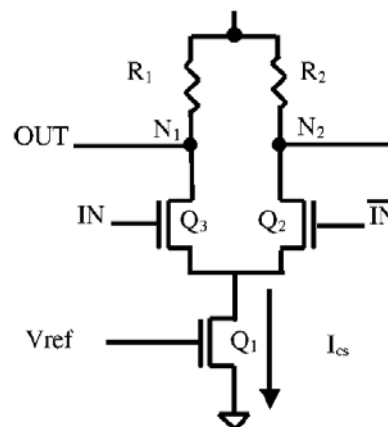
KUL - COSIC

Tutorial – part II - 33

Chennai, December 2007

MOS Current Mode Logic

- SABL is effective BUT requires full custom approach
- Other full custom approach: dynamic current mode logic (DyCML)
- Start from MOS Current Mode Logic (MCML):
- Current is CONSTANT
- Left or right branch depends on the data



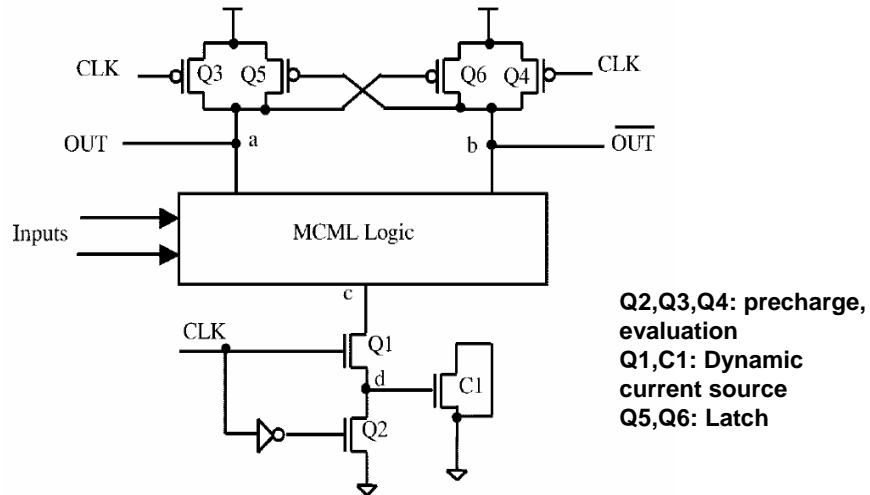
[Allam, Elmasry, JSSC2001]

KUL - COSIC

Tutorial – part II - 34

Chennai, December 2007

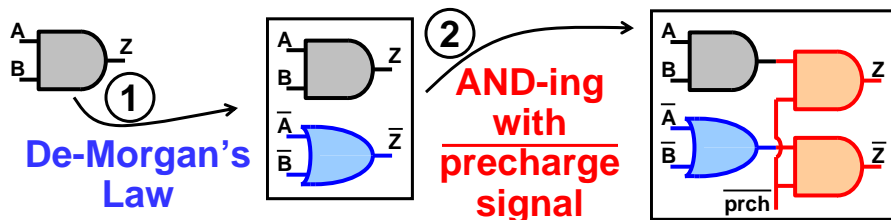
Dynamic Current Mode Logic (DyCML)



Evaluation by Mace et al. for side-channel resistance

[Allam, Elmasry, JSSC2001]

Solution based on *Standard cells*



- false output

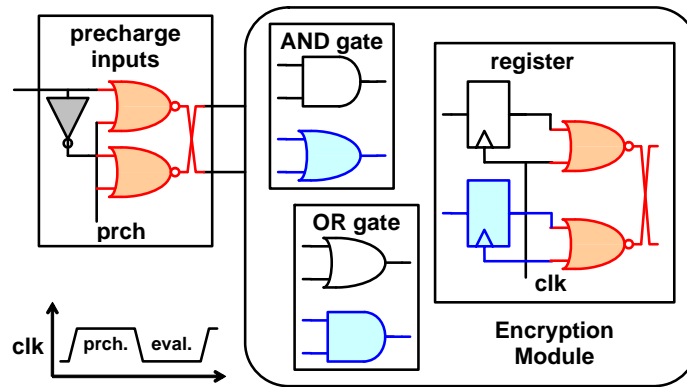
- with false inputs

- precharge 1:
outputs are 0

- precharge 0 - evaluation:
1 output is 1

Wave Dynamic Differential Logic

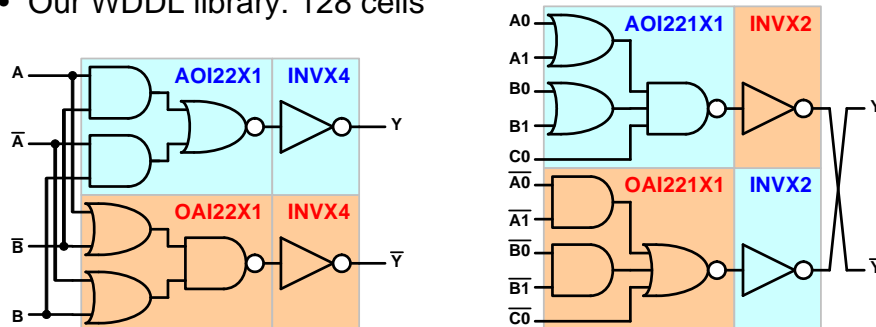
- Restrict library to AND, OR gate
 - input 0 \Rightarrow output 0
 - no precharge operator



[Tiri, DATE2004]

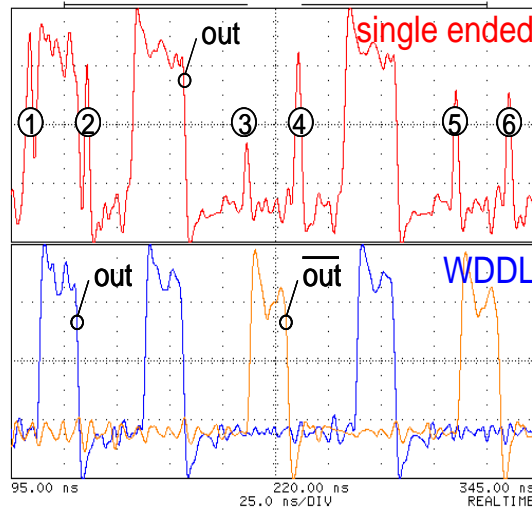
WDDL library

- 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



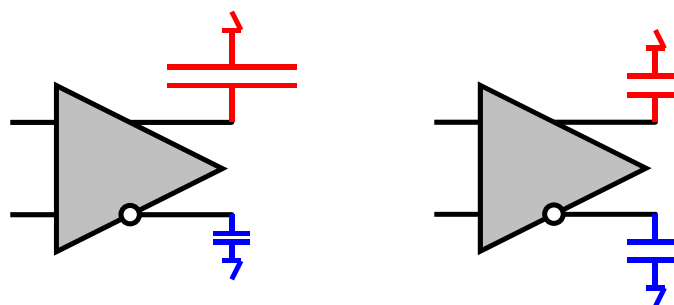
KUL - COSIC

Tutorial – part II - 39

Chennai, December 2007

Unbalanced capacitive loads

- For constant power consumption:
[constant load capacitance](#).
- Match loads at differential outputs.

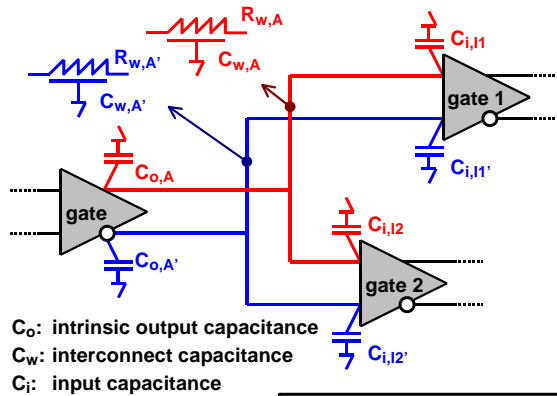


KUL - COSIC

Tutorial – part II - 40

Chennai, December 2007

Load capacitance breakdown

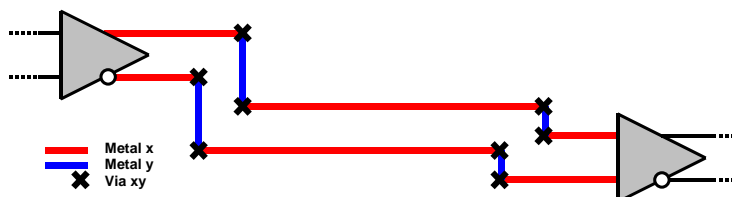


- Intrinsic caps.: matched
- Interconnect: dominant (Moore's law)
- Balancing interconnect: crucial

$$\begin{aligned}
 CA &= CA' \\
 C_{o,A} + C_{w,A} + C_{i,I1} + \dots C_{i,Ik} \\
 &= C_{o,A'} + C_{w,A'} + C_{i,I1'} + \dots C_{i,Ik'} \\
 C_{w,A} &= C_{w,A'}
 \end{aligned}$$

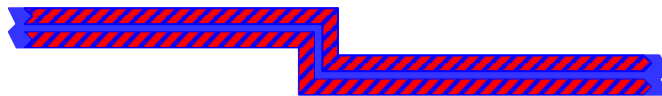
Place & Route approach

- 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

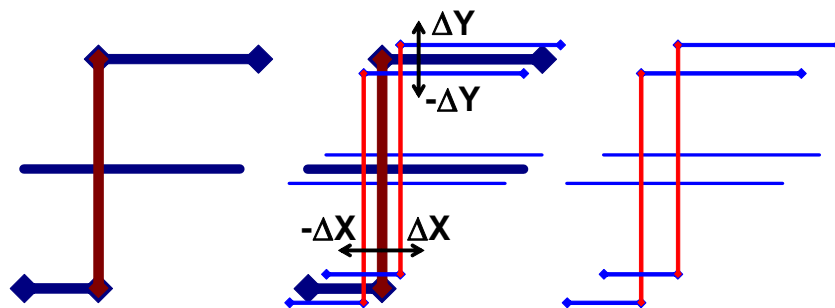
- Available via gridless/shape-based routers.
 - only few critical signals (e.g. clock)
 - experiment with 200 pairs:
8 hours CPU, 1000 conflicts, 100 open nets.
- Gridded routers avoid wires in parallel.
- Trick router: “fat”-wire routing.
 - Abstract differential pair as one single fat wire.
 - Route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



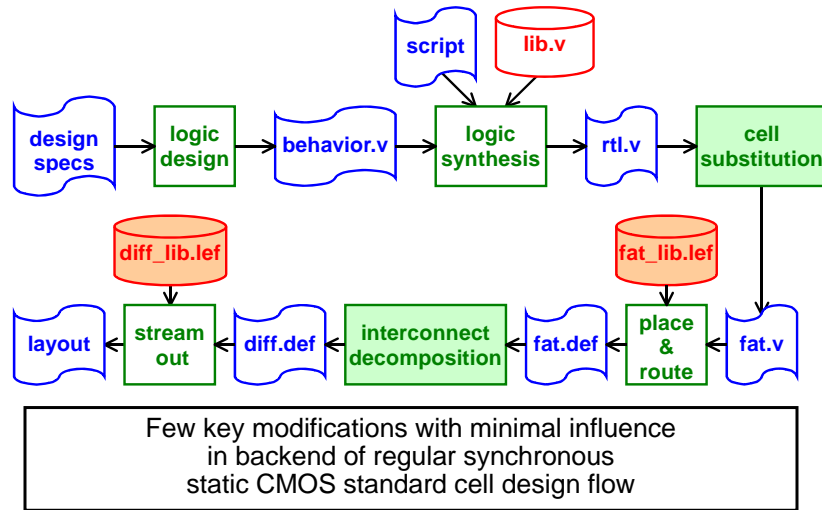
[Tiri, CARDIS2004]

Fat wire decomposition

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



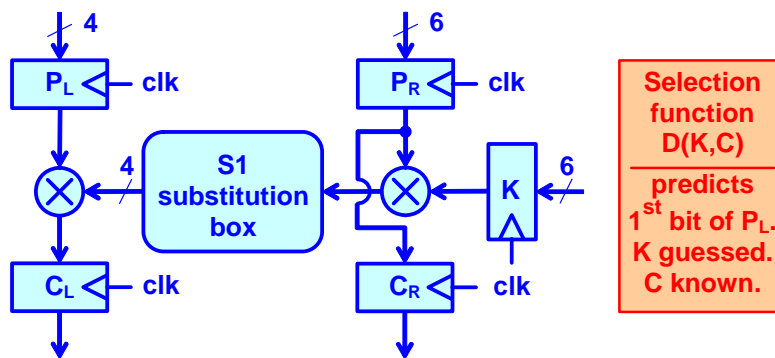
Secure digital design flow



[Tiri, TCAD2006]

Experimental setup

- DPA on module of last round DES



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

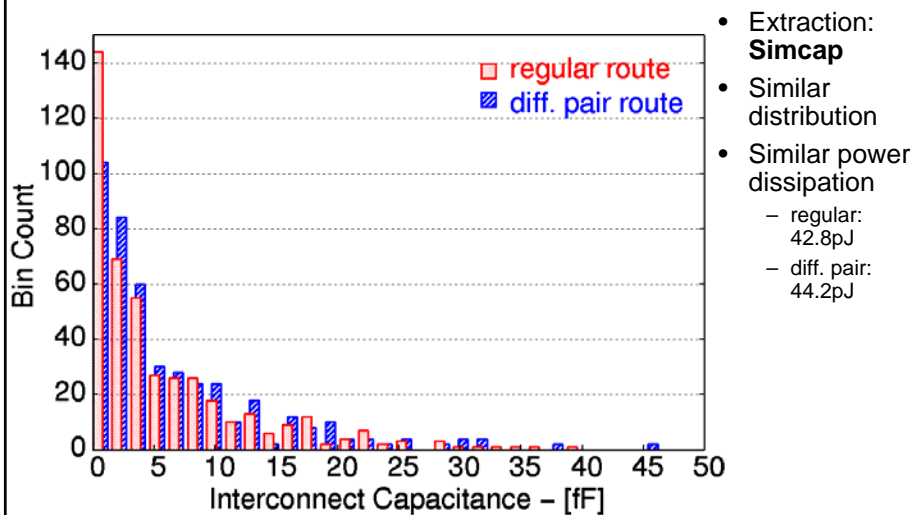
- Same circuit; two implementations.
- Difference in routing:
 - regular route (without constraints) – 8 sec. CPU
 - differential pair route – 3 sec. CPU
- Same floorplan.
 - aspect ratio 1, row utilization 0.8
- Toolflow:
 - Place & route: Silicon Ensemble 5.3
 - Layout-to-netlist (extraction parasitics): Virtuoso
 - Power traces (transient simulation): Hspice
- 2000 encryptions.

KUL - COSIC

Tutorial – part II - 47

Chennai, December 2007

Absolute interconnect capacitances

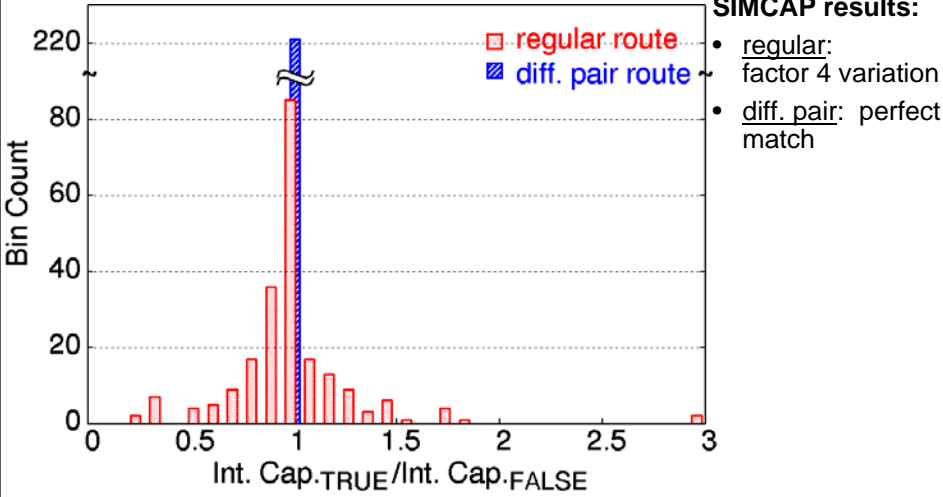


KUL - COSIC

Tutorial – part II - 48

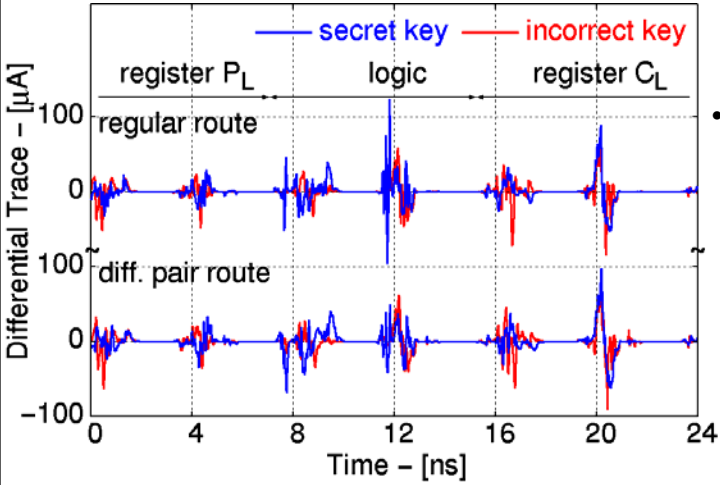
Chennai, December 2007

Matching precision



- SIMCAP results:**
- regular: factor 4 variation
 - diff. pair: perfect match

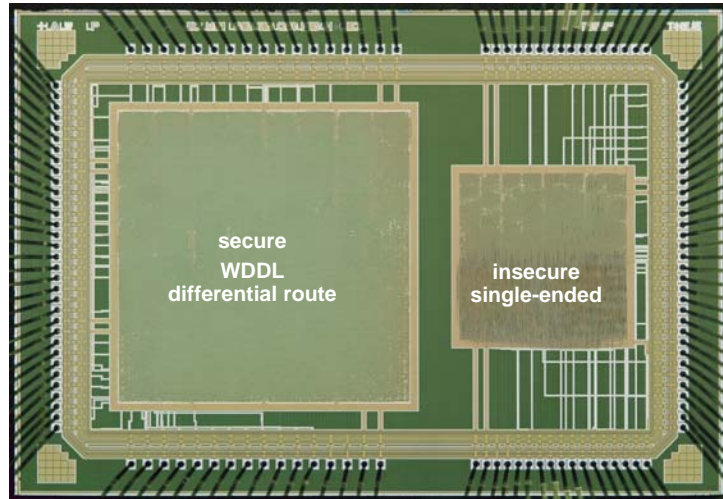
DPA – differential trace



- regular: DPA works despite a mere 2% power variation.
- diff. pair: effective reduction of peaks secret key.

Prototype IC – ThumbPodII

- AES, controller, fingerprint processor.



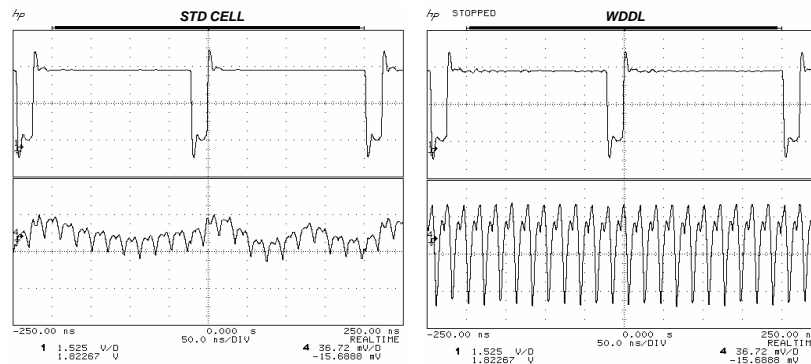
KUL - COSIC

Tutorial – part II - 51

Chennai, December 2007

Circuit techniques to address SCA

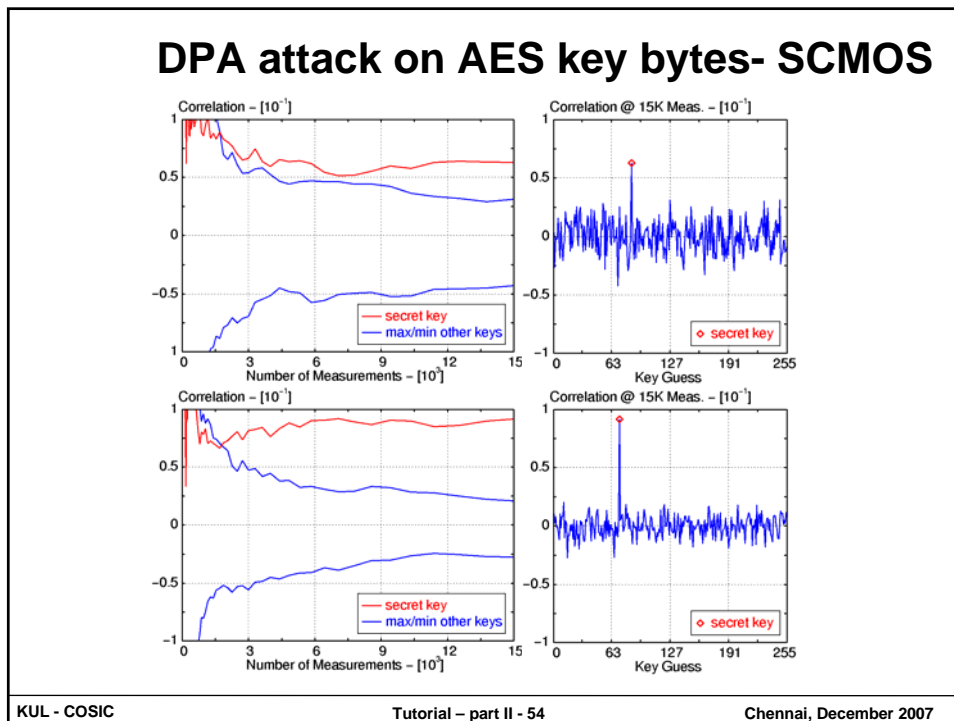
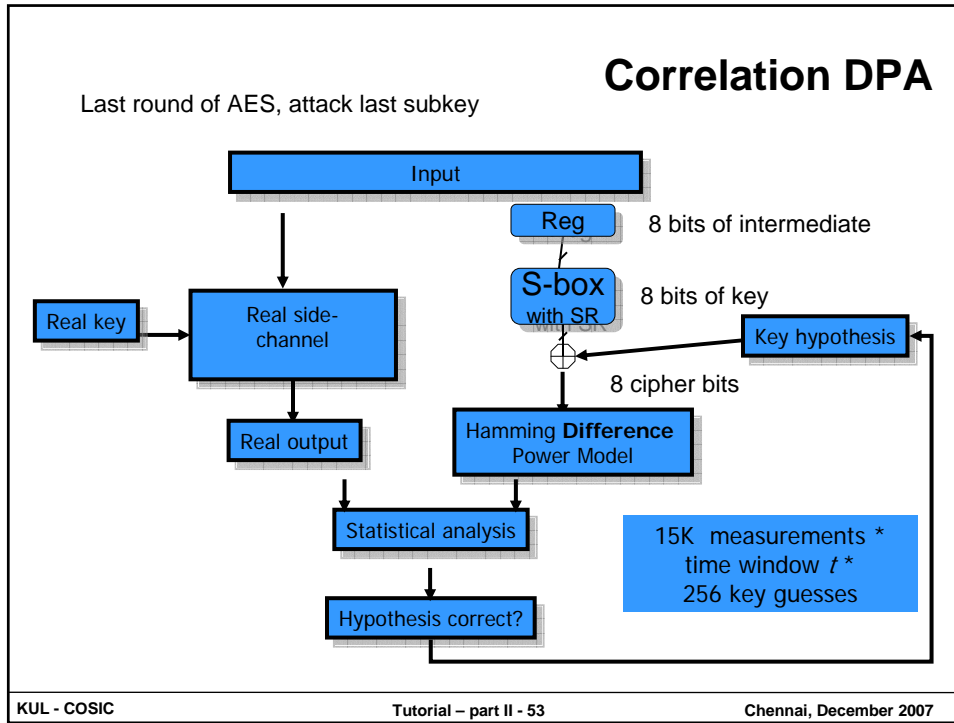
- Standard cells: break AES with 8000 encryptions
- Special cells (build from standard cells): over 1.5M encryptions and still not broken



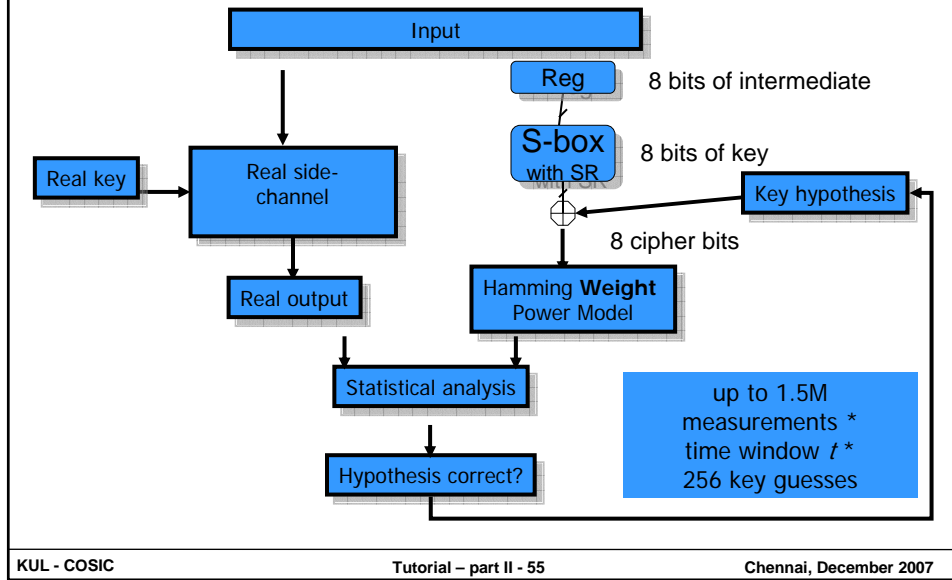
KUL - COSIC

Tutorial – part II - 52

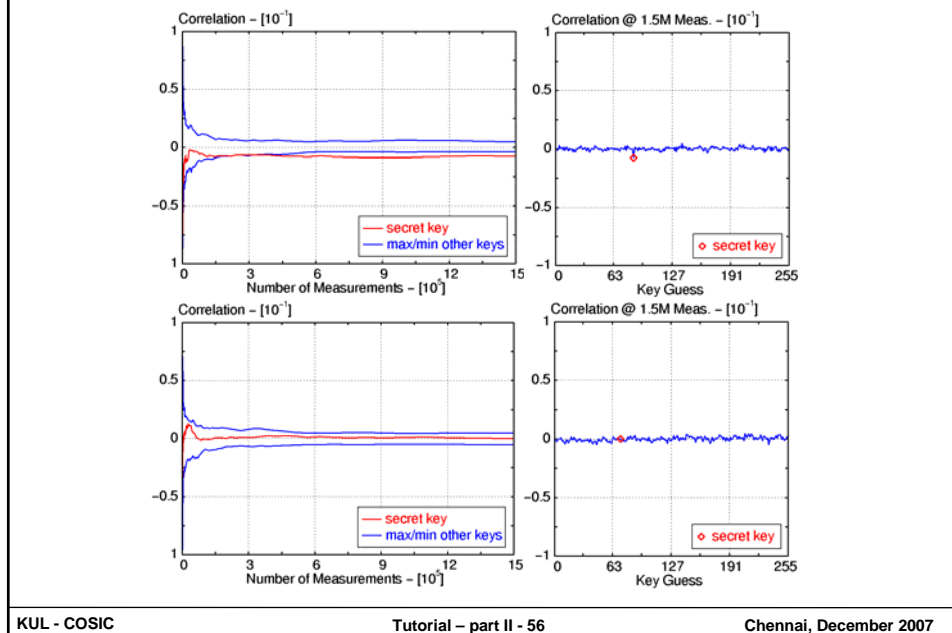
Chennai, December 2007



Correlation DPA



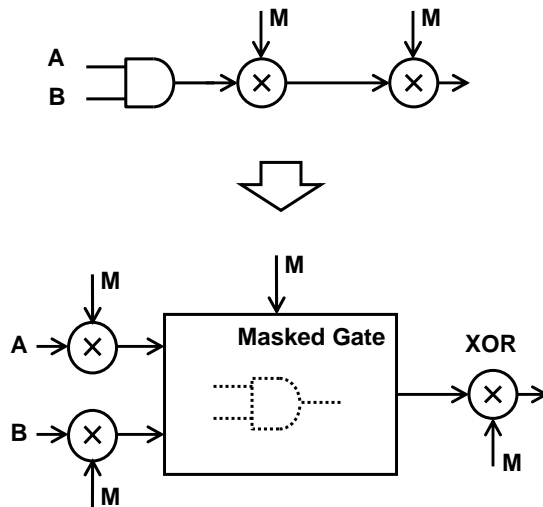
DPA attack on WDDL



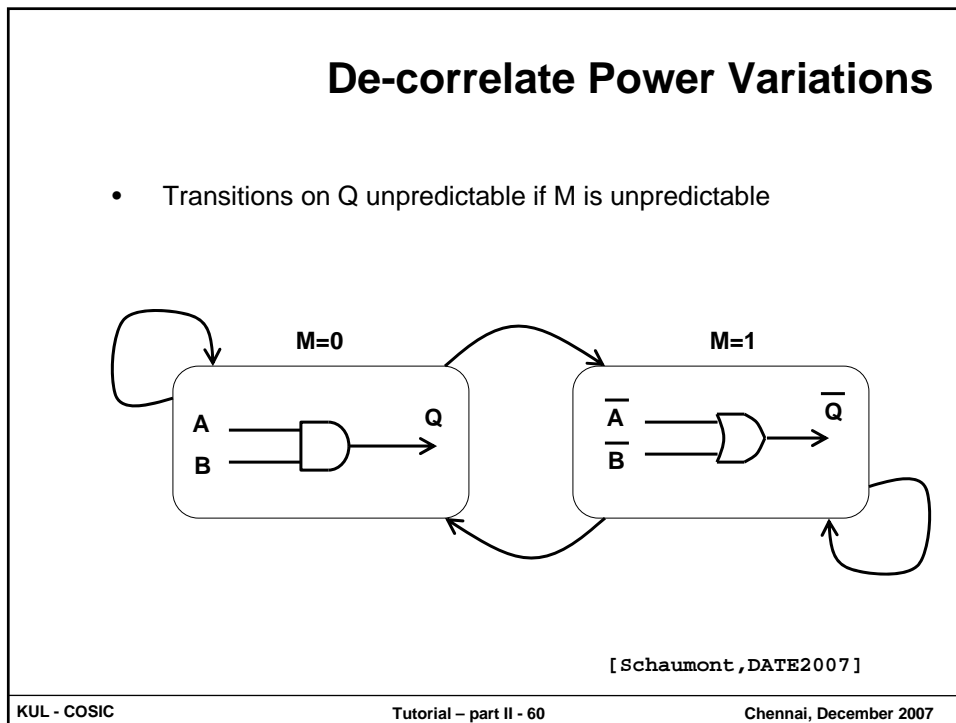
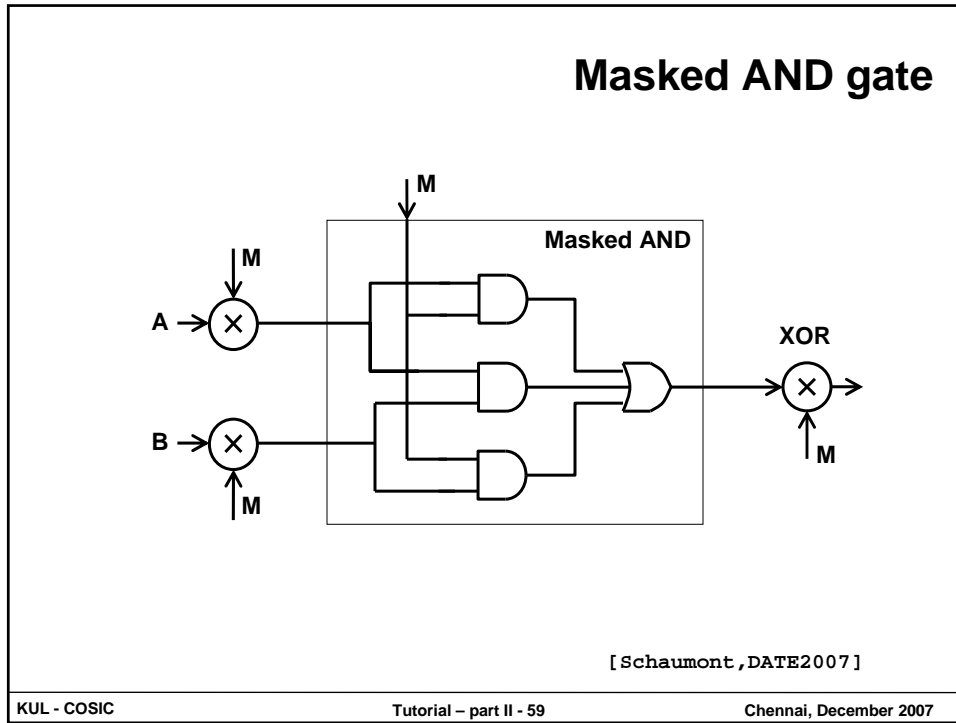
Alternative: masking

- Alternative to 'constant' power design
 - Parasitic capacitances, process variations, etc. don't guarantee *perfect* balancing. (Perfect security does not exist...)
- De-correlate power variations: masking

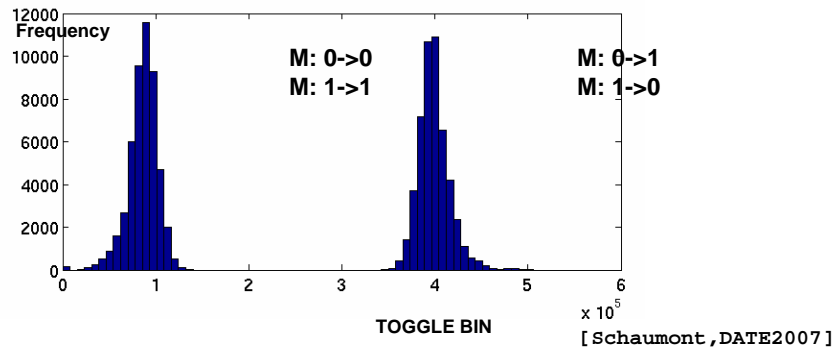
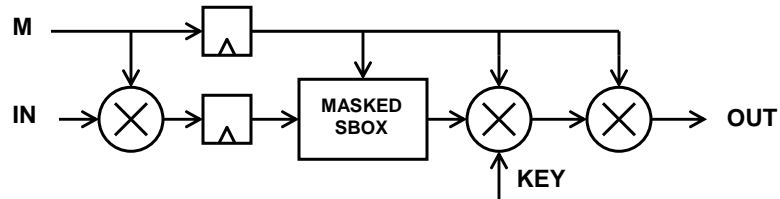
Masked Gates



[Schaumont, DATE2007]



Analysis for Masked SBOX

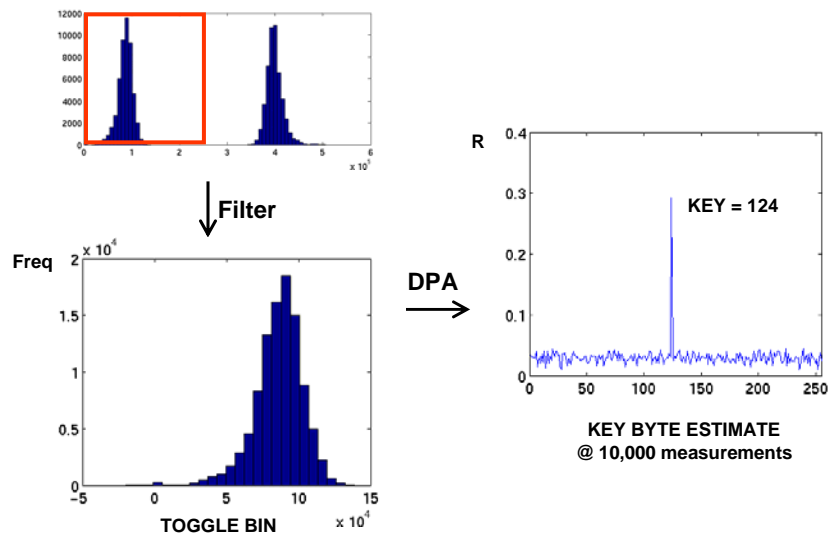


KUL - COSIC

Tutorial – part II - 61

Chennai, December 2007

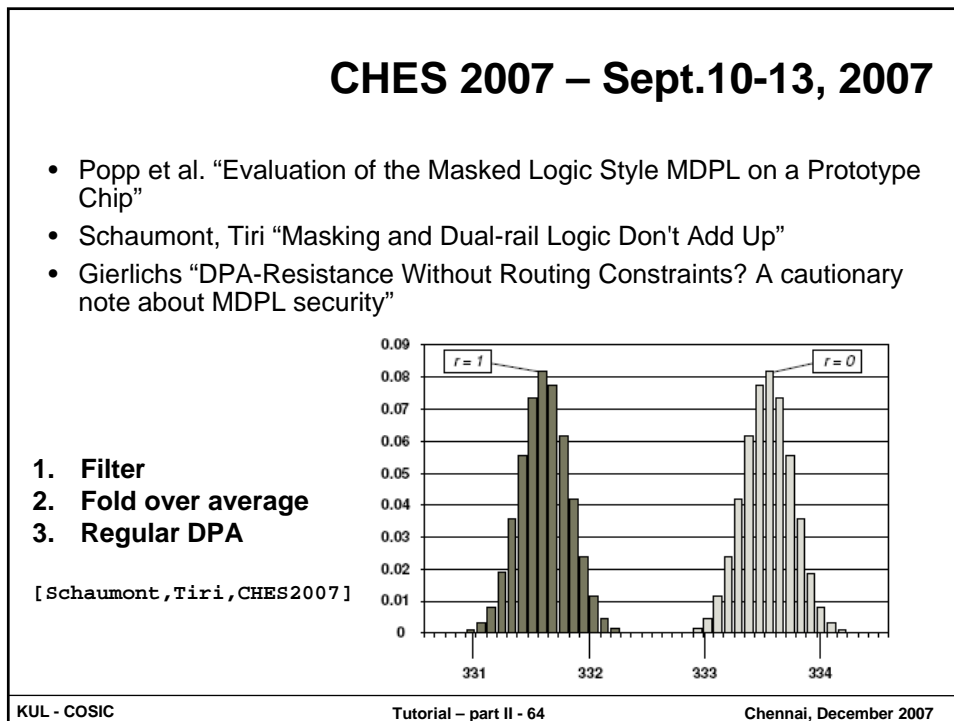
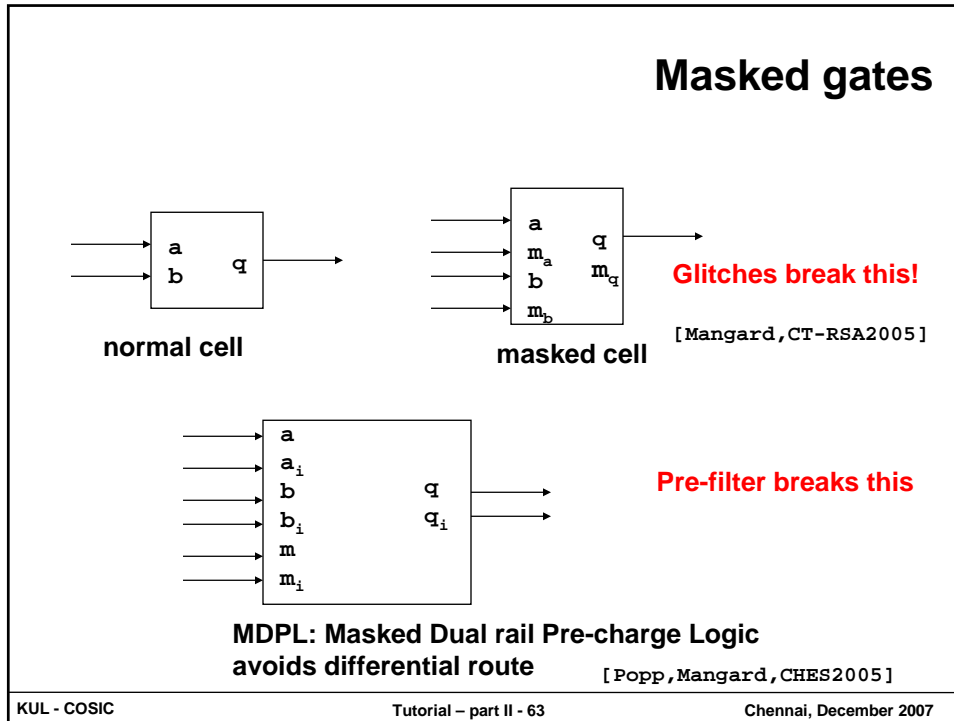
DPA straightforward after filtering



KUL - COSIC

Tutorial – part II - 62

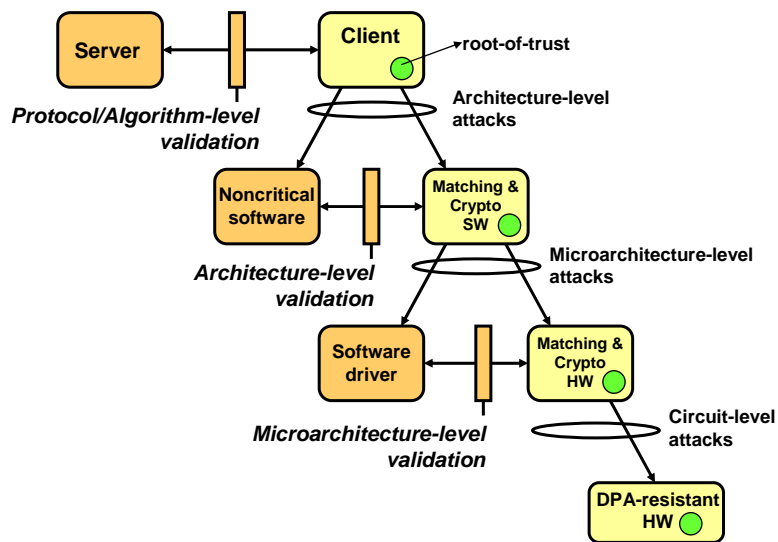
Chennai, December 2007



Security Partitioning

- Side channel protection = extra cost
- Part of design methodology
- Only secure parts needs protection against side channel attacks!
- In Thumpbod example: over 90% of calculations are on non-secure embedded processor with DFT co-processor
- Only AES calculations & matching needs SCA resistant implementation.

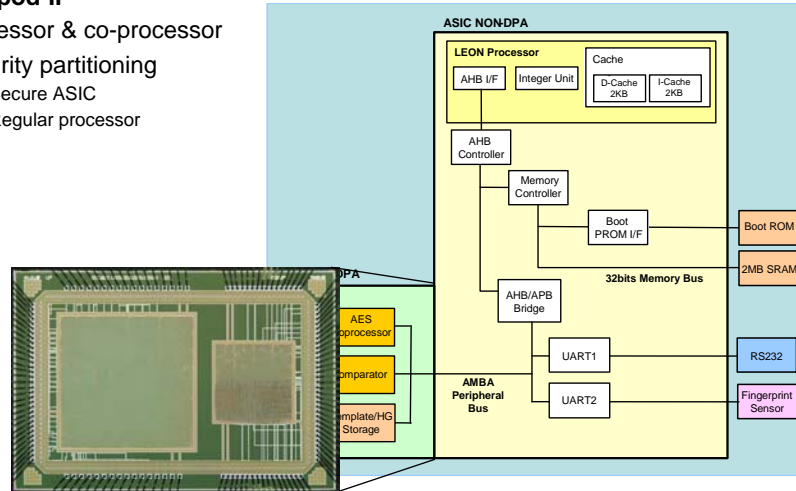
Mapping Algorithms - Security Partitioning



Security partitioning

Thumbpod-II

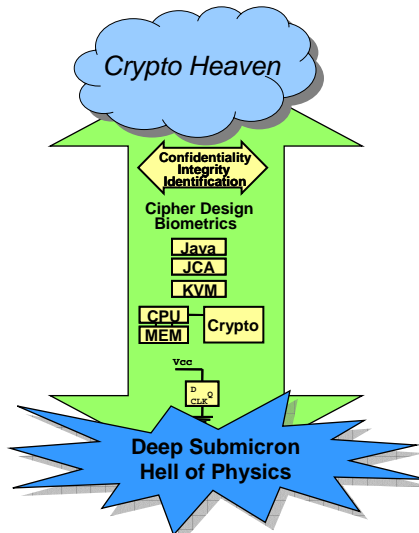
- Processor & co-processor
- Security partitioning
 - Secure ASIC
 - Regular processor



Conclusion

- Secure design means
 - Efficient implementation of security functions (not topic of this tutorial)
 - Secure implementation
- Side-channel attacks and countermeasures
 - Arms-race
 - Perfect security does not exist
 - Make it too expensive to attack
 - Algorithm security still orders harder than side-channel attacks
- Systematic design methods to design for security

Future: Technology Aware Security



[Modified after H. De Man]

*Security is as strong as
the weakest link!*

*Link crypto heaven to
deep sub-micron hell*

1. Architectures & design methods

Trusted platforms, modules and security partitioning methods

2. Ultra Low Power Security:

Arithmetic, 'light-weight crypto', co-processors, RFID,

3. Technology Aware:

Secure memories, process variations, robust, reliable, side-channel secure