

Hardware Design for Cryptographers



KU LEUVEN

Ingrid Verbauwhede
ingrid.verbauwhede-at-esat.kuleuven.be

KU Leuven, ESAT- COSIC
Computer Security and Industrial Cryptography

Acknowledgements:
Current and former PhD students



KUL - COSIC

Summer School- 1

Šibenik Croatia, June 2016

Outline

- Given: cryptographic algorithm
- Request: “efficient” hardware design
- AND “secure” hardware
- This lecture about Efficiency
- The cost of side-channel resistance is addressed in other lectures

KUL - COSIC

Summer School- 2

Šibenik Croatia, June 2016



Design Parameters

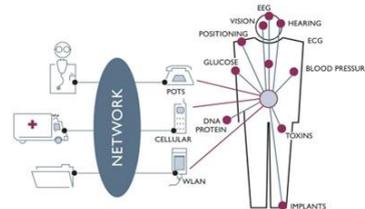
Embedded security:
Area, delay, power, energy,
physical security

KUL - COSIC

Summer School- 3

Šibenik Croatia, June 2016

Embedded crypto everywhere



IMEC: Human++ project

Cheney's defibrillator was modified to prevent hacking



STORY HIGHLIGHTS
 - "I worried that someone could kill me," Doctor tells Cheney
 - Cheney, 72, suffered his first of five heart attacks in 1970 -- at
 (CNN) -- Cautious doctors replacing former Vice President Dick Cheney's heart defibrillator in 2007 modified it so it couldn't be hacked by terrorists who might try to kill him, Cheney told CBS's Sangny Gupta in an interview that aired Sunday night on CBS' "60 Minutes."

KUL - COSIC

Summer School- 4

Šibenik Croatia, June 2016

Swarm at the edge of the cloud



[Source photograph: J. Rabaey: A Brand New Wireless Day]

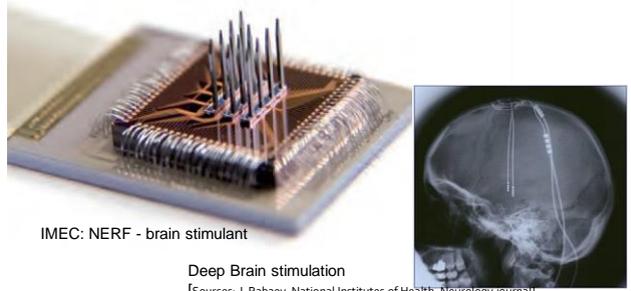
KUL - COSIC

Summer School- 5

Šibenik Croatia, June 2016

Cyber-physical system

“Networked embedded devices interacting with the environment”
E.A. Lee after H. Gill (NSF)



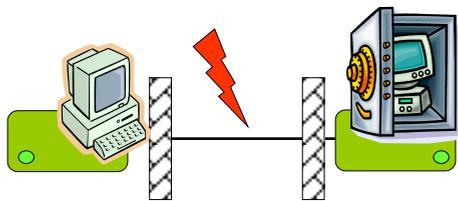
[Sources: J. Rabaey, National Institutes of Health, Neurology journal]

KUL - COSIC

Summer School- 6

Šibenik Croatia, June 2016

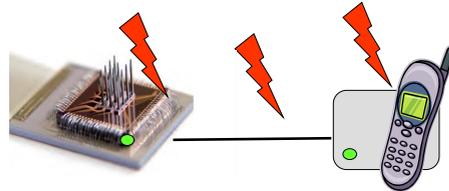
Embedded Security



Old Model (simplified view):

- Attack on channel *between* communicating parties
- Encryption and cryptographic operations in *black* boxes
- Protection by strong mathematic algorithms and protocols

Embedded Security



New Model (also simplified view):

- Attack channel *and* endpoints
- Encryption and cryptographic operations in *gray* boxes
- Protection by strong mathematic algorithms and protocols
- Protection by secure implementation

Need secure *implementations* not only algorithms

KUL - COSIC

Summer School- 7

Šibenik Croatia, June 2016

KUL - COSIC

Summer School- 8

Šibenik Croatia, June 2016

Embedded Security

NEED BOTH



- Efficient, light-weight Implementation
 - Within power, area, timing budgets
 - Public key: 1024 bits RSA on 8 bit μ C and 100 μ W
 - Public key on a passive RFID tag
- Trustworthy implementation
 - Resistant to attacks
 - Active attacks: probing, power glitches, JTAG scan chain
 - Passive attacks: side channel attacks, including power, timing and electromagnetic leaks



KUL - COSIC

Summer School- 9

Šibenik Croatia, June 2016

Cost definition

- Area
- Time: throughput versus latency
- Power, Energy
- Physical Security
- NRE (Non Recurring Engineering) cost

KUL - COSIC

Summer School- 10

Šibenik Croatia, June 2016

AREA

Area

- ASIC = Application Specific Integrated Circuit
 - Gate count
 - Unit = NAND gate = 4 transistors
- FPGA = Field Programmable Gate Array
 - Unit is LUT, flip-flops
- Embedded micro-controllers
 - Memory size = program size + data size

KUL - COSIC

Summer School- 11

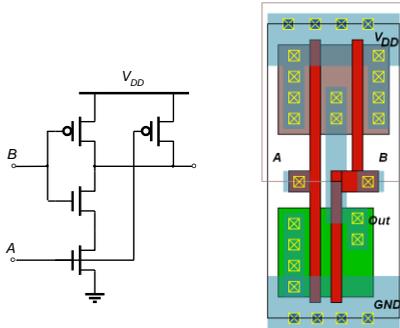
Šibenik Croatia, June 2016

KUL - COSIC

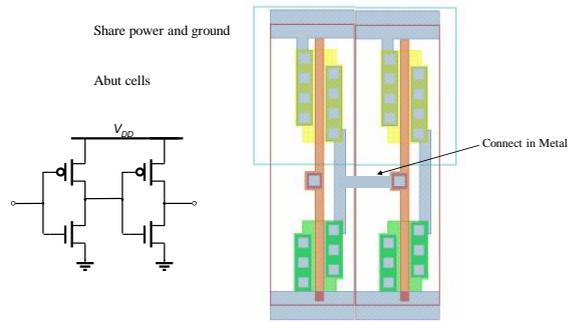
Summer School- 12

Šibenik Croatia, June 2016

One Standard cell NAND gate

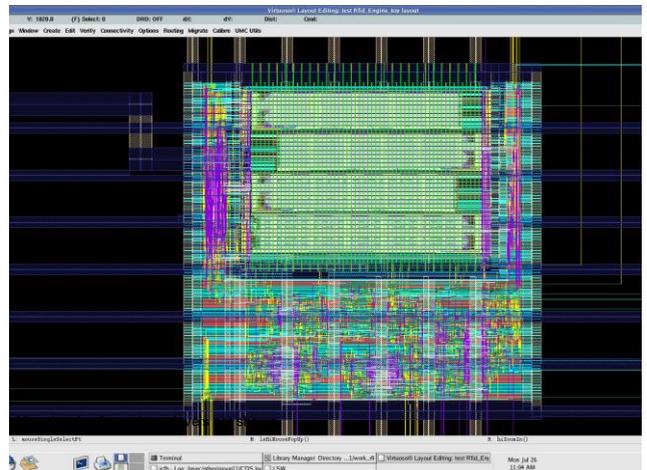
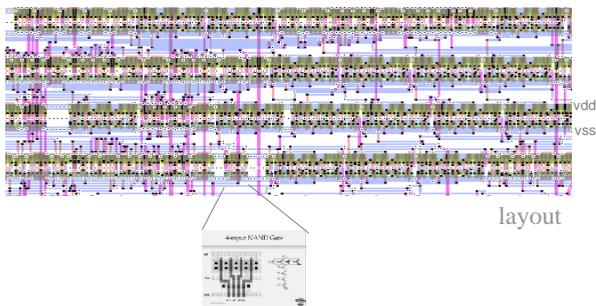


Two Inverters



Slide courtesy: J. Rabaey

Standard Cell Zoom In



TIME



Clock frequency versus sample frequency
Throughput versus latency
REAL-TIME

Real-time, throughput, latency

- Throughput = associated with **application**
 - Amount of data processed per time unit
 - Video: Gbits/sec, Internet: Gpackets/sec
 - **Real-time sample rate**: HW has to work as fast as application dictates
- Latency = associated with **application**
 - Delay from input to output
 - Measure of reaction speed or turn-around time
 - Low latency: brakes of a car, memory encryption
- High throughput and low latency don't go together

Clock Frequency

Clock frequency is a property of the hardware!
= 1 / max (longest combinatorial path)
= 1 / (critical path)

- Extremely high throughput (Radar or fiber optics)
 - One operator (= hardware unit, e.g. adder, shifter, register)
 - for each operation (= algorithmic, e.g. addition, multiplication, delay)

⇒ clock frequency = sample frequency

- Most designs: time multiplexing

clock frequency ≠ sample frequency

$\frac{\text{clock frequency}}{\text{sample frequency}} = \text{number of clock cycles available for the job}$

Example: AES variations

- There is only one AES 128 algorithm
- There are multiple AES hardware implementation options.
- Basic operations:
 - Byte-sub: non-linear operation on every byte
 - Shift-row: Circular shifting of bytes in each row
 - Mix-column: multiplying the round data with a fixed polynomial
 - Add-key: XORing the round data and round key

AES sequential

AES: one cycle per round

- Sample frequency < clock frequency
- Example: clock is 200 MHz, sample rate is 10MHz, 128 bits per sample
- Requested throughput: min 1.28 Gbits/sec

$$10 \text{ Msamples/s} * 128 \text{ bits/sample} = 1.28 \text{ Gbits/s}$$

- Puts HW limit: one AES in 20 clock cycles or less

Solution:

Parallel datapath, sequential execution

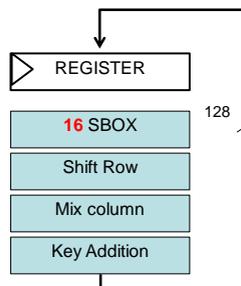
AES: one cycle per round

Solution:

- One 128 bit datapath
- 11 clock cycles to finish AES (plus load and store clock cycle)
- 200 MHz clock

Throughput: ??

Latency: ??



AES: one cycle per round

Solution:

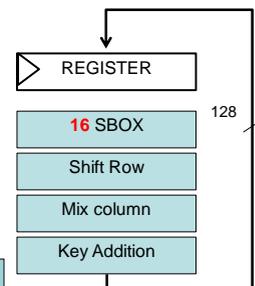
- One 128 bit datapath
- 11 clock cycles to finish AES (plus load and store clock cycle)
- 200 MHz clock

Throughput: ??

$$200 \text{ Mcycles/s} * 128 \text{ bits/sample} / 11 \text{ cycles/sample} = 2.33 \text{ Gbits/s}$$

Latency: ??

$$11 \text{ cycles/sample} / 200 \text{ Mcycles/s} = 55 \text{ ns/sample}$$

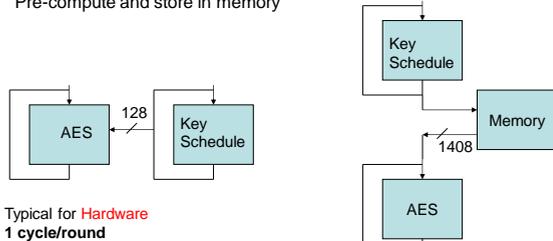


Efficiency - adapt HW platform to application

Simple example: Key Schedule for round key

Two options:

- On the “fly” = just in time processing
- Pre-compute and store in memory



Typical for **Hardware**
1 cycle/round

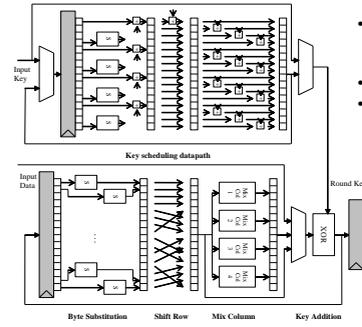
Typical for **Software**
Minimum around **10 cycles/byte + bandwidth**

KUL - COSIC

Summer School- 25

Šibenik Croatia, June 2016

AES Core



- Key schedule in parallel with data path
 - 128-bit data and key
 - One round implementation with minimum delay
 - One cycle per round
 - Direct implementation of Byte Substitution phase
- 11 cycles one encryption

Slide credit: Alireza Hodjat

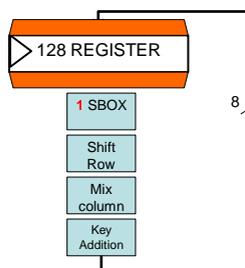
KUL - COSIC

Summer School- 26

Šibenik Croatia, June 2016

AES compact: one SBOX

- Goal: low area
- Exercise: (clock is 200MHz)
 - Min number of clockcycles?
 - Throughput ?
 - Latency?



- Overhead: muxes and control logic

KUL - COSIC

Summer School- 27

Šibenik Croatia, June 2016

AES compact: one SBOX

- Goal: low area
- Exercise: (clock is 200MHz)
 - Min number of clockcycles?
 - Throughput ?
 - Latency?

$$128 \text{ bits/round} * 11 \text{ rounds/sample} / 8 \text{ bits/cycle} = 176 \text{ cycles/sample}$$

$$128 \text{ bits/sample} * 200 \text{ Mcycles/s} / 176 \text{ cycles/sample} = 145 \text{ Mbits/s}$$

$$176 \text{ cycles/sample} / 200 \text{ Mcycles/s} = 880 \text{ ns/sample}$$

- Overhead: muxes and control logic

KUL - COSIC

Summer School- 28

Šibenik Croatia, June 2016

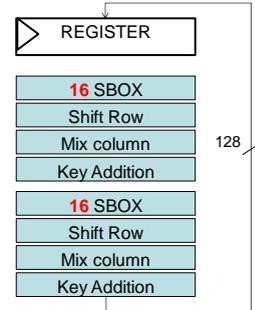
AES parallel and pipeline

AES parallel = loop unrolling

- Unroll twice, now 6 CC
- **IF** same clock THEN
- double throughput

- latency: ?

- (overhead ignored)
= expensive



AES parallel = loop unrolling

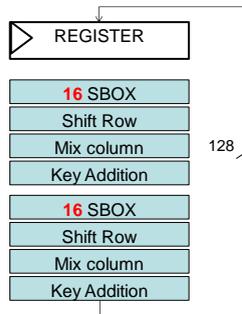
- Unroll twice, now 6 CC
- **IF** same clock THEN
- double throughput

**128 bits/sample * 200 Mcycles/s
/ 6 cycles/sample = 4.27 Gbits/s**

- latency: ?

**6 cycles/sample / 200 Mcycles/s
= 30 ns/sample**

- (overhead ignored)
= expensive



AES unroll & pipeline

- 1 clock cycle per AES
- 11 data samples in pipeline
- 200 MHz clock
- Throughput: ??

- Latency: ??



AES unroll & pipeline

1 clock cycle per AES
 11 data samples in pipeline
 200 MHz clock
 Throughput: ??

128 bits/sample * 200 Mcycles/s / 1 cycles/sample = 25.6 Gbits/s

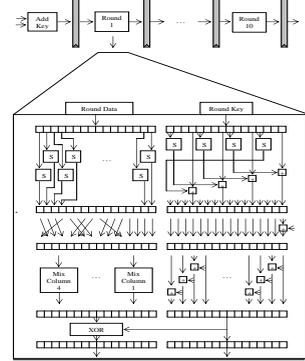
Latency: ??

11 cycles/sample / 200 Mcycles/s = 55 ns



AES loop enroll AND pipelined

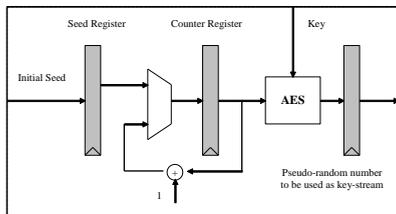
In combination with AES key schedule



Slide credit: Alireza Hodjat

Non feedback modes of operation!!

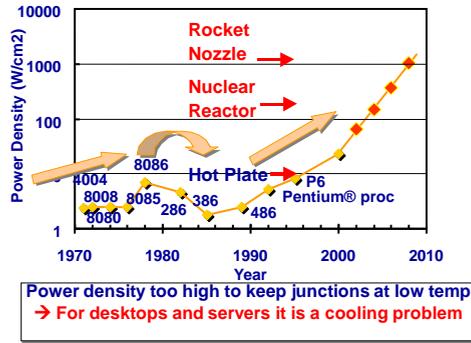
- Example counter mode of operation
- Does not work with CBC or other feedback modes of operation



POWER



Power density



Courtesy, Intel

KUL - COSIC

Summer School- 37

Šibenik Croatia, June 2016

Cooling

- Cooling for the very small and the very large



Deep Brain stimulation
[Sources: J. Rabney, National Institutes of Health, Neurology journal]



[San Francisco Chronicle online, source Google]

KUL - COSIC

Summer School- 38

Šibenik Croatia, June 2016

Power and Energy are not the same!

- Power = $P = I \times V$ (current x voltage) (= Watt)
 - instantaneous
 - Typically checked for cooling or for peak performance
- Energy = Power x execution time (= Joule)
 - Battery content is expressed in Joules
 - Gives idea of how much Joules to get the job done

Low power processor \neq low energy solution !

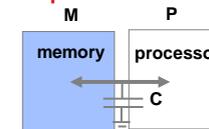
KUL - COSIC

Summer School- 39

Šibenik Croatia, June 2016

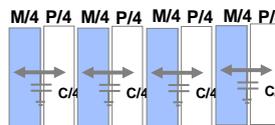
Heat and parallelism

Reduce power = reduce WASTE !!



Power
(Heat)

$$P_{\text{mono}} = CV^2f \text{ (Watt)}$$



$$4 (C/4)V^2(f/4) = P_{\text{mono}}/4$$

but since $f \sim V$

can be even $P_{\text{mono}}/4^3$

TREND: MULTI-CORE!!

KUL - COSIC

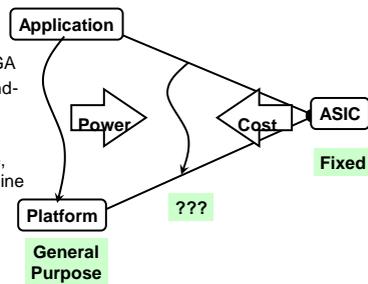
Summer School- 40

Šibenik Croatia, June 2016

Match between algorithm & architecture

Close the gap:

- Dedicated HW: ASIC
- Programmable HW: FPGA
- Custom instructions, hand-coded assembly
- Compiled code
- JAVA on virtual machine, compiled on a real machine



Energy – flexibility trade-off

AES 128bit key 128bit data	Throughput	Power	Figure of Merit (Gb/s/W)
0.18mm CMOS	3.84 Gbits/sec	350 mW	11 (1/1)
FPGA [1]	1.32 Gbit/sec	490 mW	2.7 (1/4)
Intel ISA for AES	32 Gbit/sec	95 W	0.34 (1/33)
ASM StrongARM [2]	31 Mbit/sec	240 mW	0.13 (1/85)
Asm Pentium III [3]	648 Mbits/sec	41.4 W	0.015 (1/800)
C Emb. Sparc [4]	133 Kbits/sec	120 mW	0.0011 (1/10.000)
Java [5] Emb. Sparc	450 bits/sec	120 mW	0.0000037 (1/3.000.000)

[1] Amphion CS5230 on Virtex2 + Xilinx Virtex2 Power Estimator

[2] Dag Arne Osvik: 544 cycles AES – ECB on StrongArm SA-1110

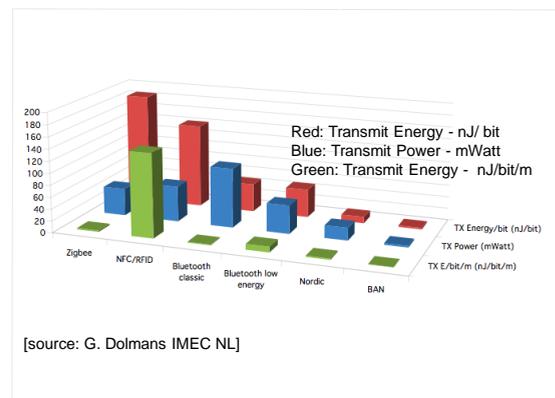
[3] Helger Lipmaa: PIII assembly hand-coded + Intel Pentium III (1.13 GHz) Datasheet

[4] gcc, 1 mW/MHz @ 120 Mhz Sparc – assumes 0.25 μ CMOS

[5] Java on KVM (Sun J2ME, non-JIT) on 1 mW/MHz @ 120 Mhz Sparc – assumes 0.25 μ CMOS

Communication & computation

Back of the envelope



1 micro Joule

Transmission:

- 300 bits in BAN
- 11 bits Bluetooth
- 3 bits Zigbee

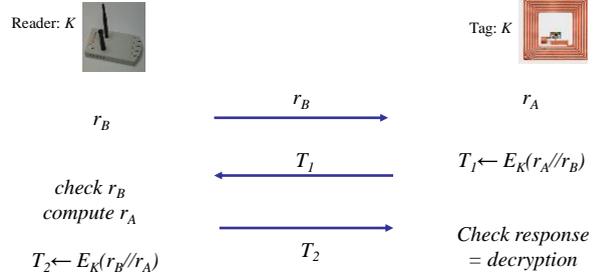
Encryption:

- 11000 bits AES
- 500 bits SHA3 hash
- 1/5 of one point multiplication



Ignores receive buget (= listening)
Ignores "overhead" of adding authentication bits, etc.

Mutual Authentication Symmetric shared key



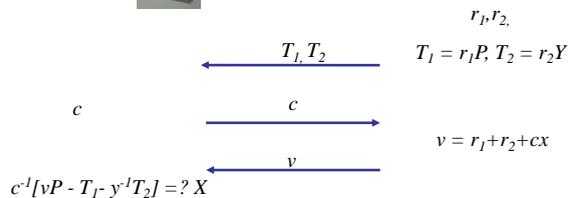
Tag: two AES encryptions, one transmission over Bluetooth
128 bit Bluetooth + 2 x AES \approx 10 microJoule

ECC based randomized Schnorr

Reader: $y, X = xP$



Tag: $x, Y = yP$



Tag: two point multiplications, two transmissions over BAN
Crypto dominates \approx 4 microJoule + 1 microJoule



Conclusions

- Time has many faces: real-time, throughput, latency, clock frequency, critical path, ...
- Power is not same as energy !
- Glitches = wasteful transitions = only occur in static CMOS, not possible in dynamic logic
- Energy – flexibility trade-off
- Depending on the algorithm/protocol and the wireless communication standard, either the crypto computation or the transmission dominates in terms of energy