

A Novel Double-Data-Rate AES Architecture Resistant against Fault Injection

P. Maistri, P. Vanhauwaert, R. Leveugle

TIMA Laboratory – Grenoble – FRANCE



TIMAlaboratory

Outline

- Motivation and objectives
- Current detection countermeasures
- The DDR approach: pros and against
- The AES implementation
 - The reference design
 - DDR issues: alignment and synchronization
 - Operation modes
- Robustness evaluation
- Conclusions

Motivation

- Fault attacks are one of the most effective ways to break a cryptosystem
 - AES can be broken with 2 well-located faults (Piret-Quisquater, CHES 2003)
- Offline error detection can not guarantee enough protection against the attacks
- Current detection countermeasures are expensive and/or have poor efficiency against realistic attacks
- The error detection scheme must be efficient against both natural and intentional faults

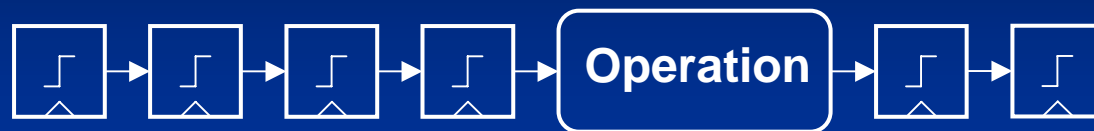
Concurrent Error Detection Schemes

- Based on spatial redundancy:
 - Circuit duplication
- Based on information redundancy:
 - Error detecting codes: parity (Bertoni et al. TC 2003), non-linear cubic codes (Karpovsky et al., DSN 2004)
- Based on temporal redundancy:
 - Computation of the inverse process (e.g., decryption) with additional (possibly existing) hardware (Karri et al., 2001)
 - Computation of the inverse process with the same hardware, for involution ciphers only (Joshi et al., CHES 2004)
 - Repetition of the same process, exploiting a pipeline (Wu and Karri, DFT 2001)

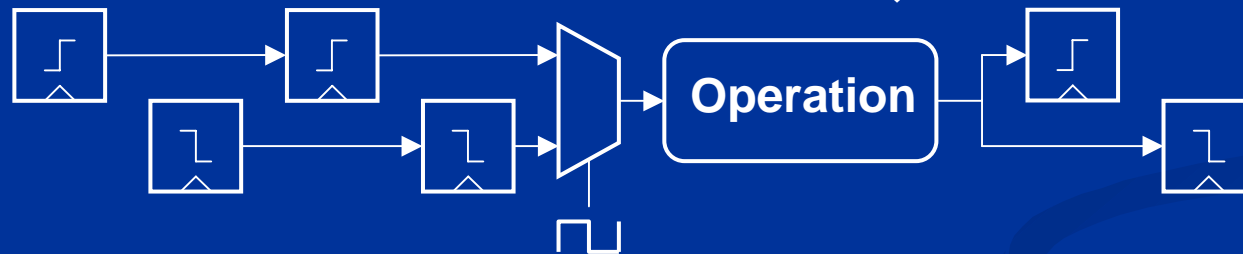
What is not good so far...

- Error codes for AES are either expensive (non-linear networks) or inefficient against malicious faults (parity)
- Spatial/information redundancy may increase correlation with power consumption and EM emissions, thus favoring side-channel attacks
- Temporal redundancy:
 - Process repetition involves performance overhead
 - Pipeline implementation requires fast system clock and significant area overhead (+50%), but ...
 - ... the global system may work at reduced frequency, thus affecting the global throughput

Double-Data-Rate Computation



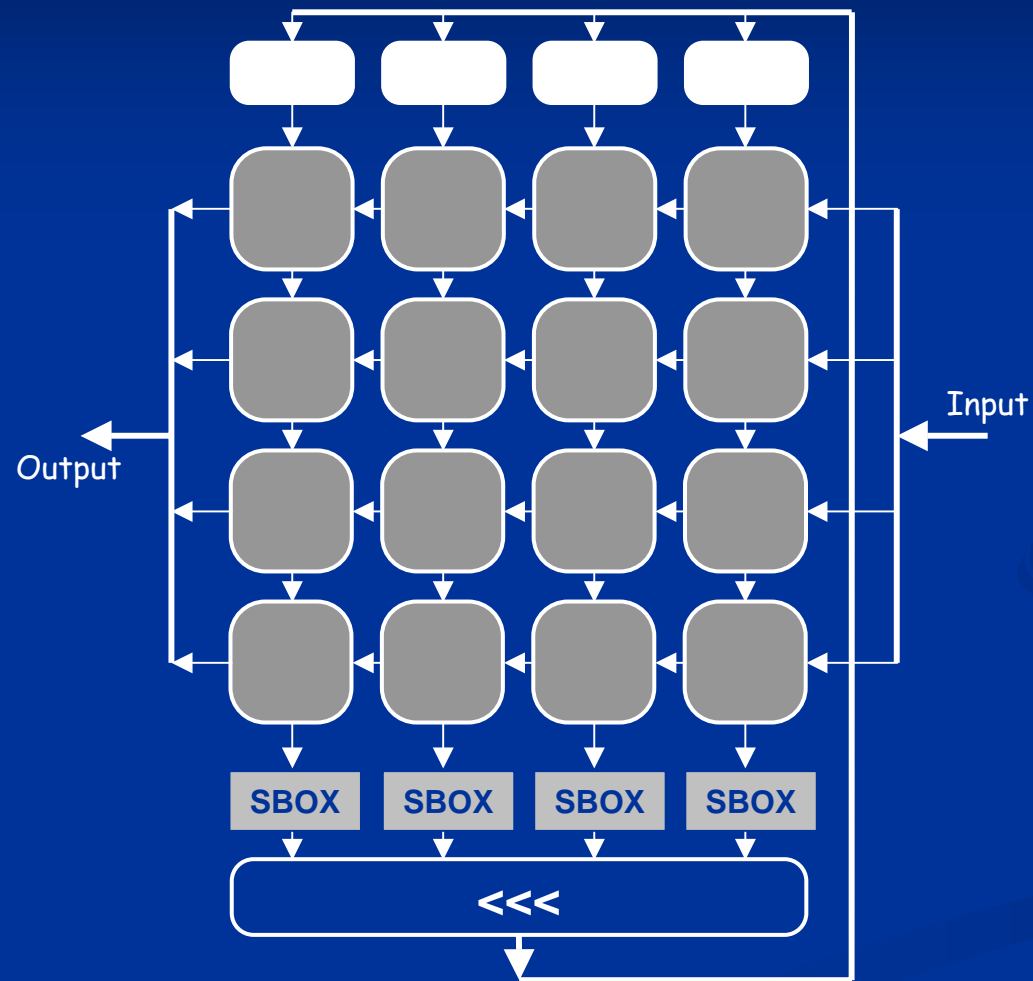
4 clock cycles to compute *Operation* for all input bytes



2 clock cycles to compute *Operation* for all input bytes

- ✓ Twice the throughput at the same frequency
- ✓ Small area overhead for DDR logic
- ✓ Increased parallelism
- ✗ More complex routing, thus lower max frequency
- ✗ Error detection requires additional overhead
- ✗ Design may require synchronization “bubbles”



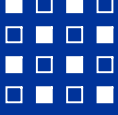
AES – The architecture



- 32-bit data-path
- 4 Substitution Boxes
- 16 GF Multipliers for MixCol
- 3 clock cycles per round
- On-the-fly key unrolling

- MixColumns, AddRoundKey and State
- 2-stage SBox
- Register layer
- Combinatorial logic
- 8-bit signal
- 32-bit signal

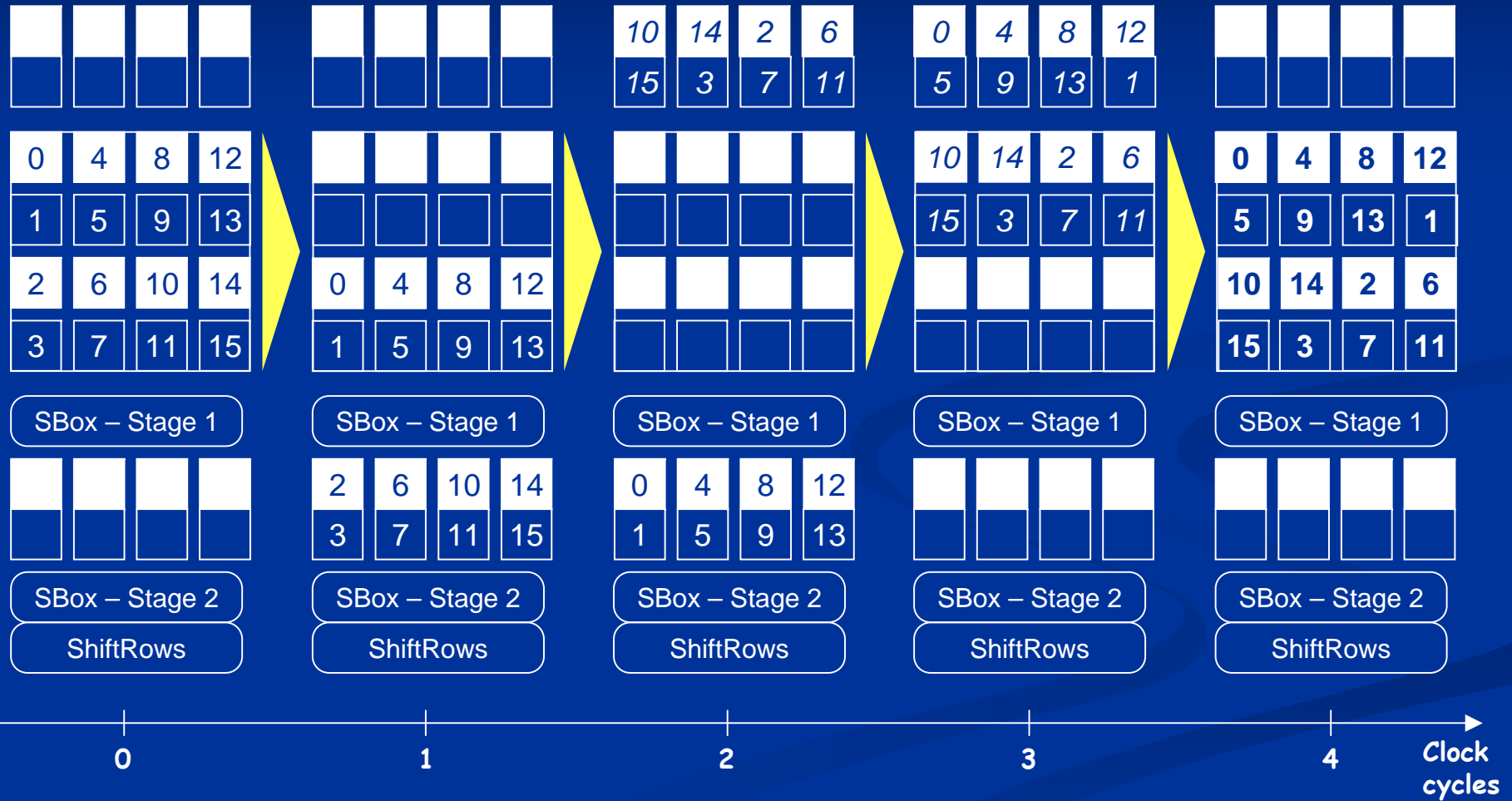
Data Alignment in AES

- The data alignment phase partitions the register space into two classes:
 - Registers triggered by ascending clock edge
 - Registers triggered by descending clock edge
- Alignment can be done:
 -  By columns: registers in the same columns share the clock alignment
 -  By rows: registers in the same rows share the clock alignment
 -  By checkers: elements of the partitions are interleaved both in columns and rows, like a chess board

Synchronization

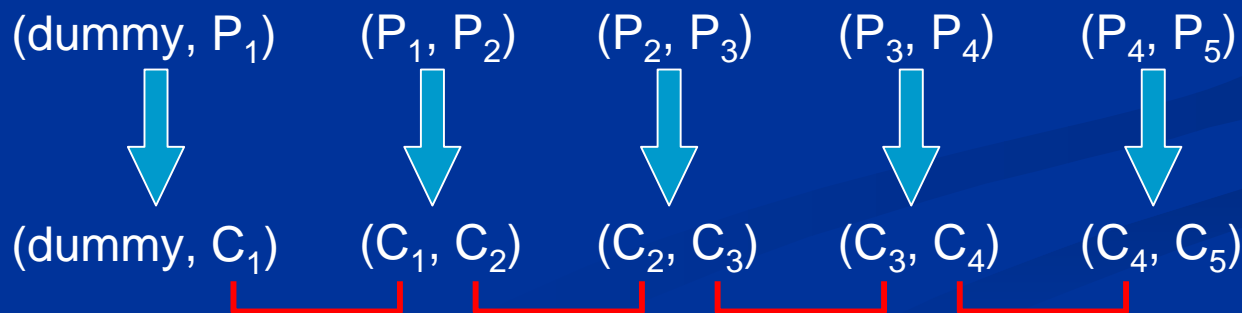
- DDR computation can be employed when we have scarce resources, high parallelism and no data dependency
 - In our design, SBoxes are the scarce resources
 - Row rotation is performed while moving data during non-linear substitution (collateral data-dependence)
 - **Row-wise DDR alignment is thus chosen**
- In AES, all operations are independent on each byte, but the *MixColumns* operation
 - MixColumns are not a scarce resource (each byte is computed locally), but values have to be stable (i.e., a latch is used)

Round Computation



Operation modes

- Single: the unit uses the DDR computation to improve its throughput and no check is performed on data
- Double: the unit uses the DDR computation to compute each round twice, checking for inconsistencies
- Interleaved: like the *Double* mode, but the first and second repetition process two different (consecutive) blocks in ECB mode, sharing the encryption key

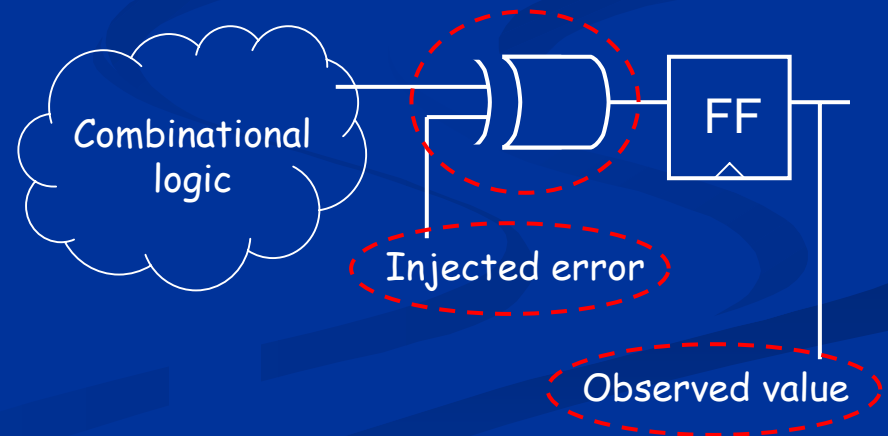
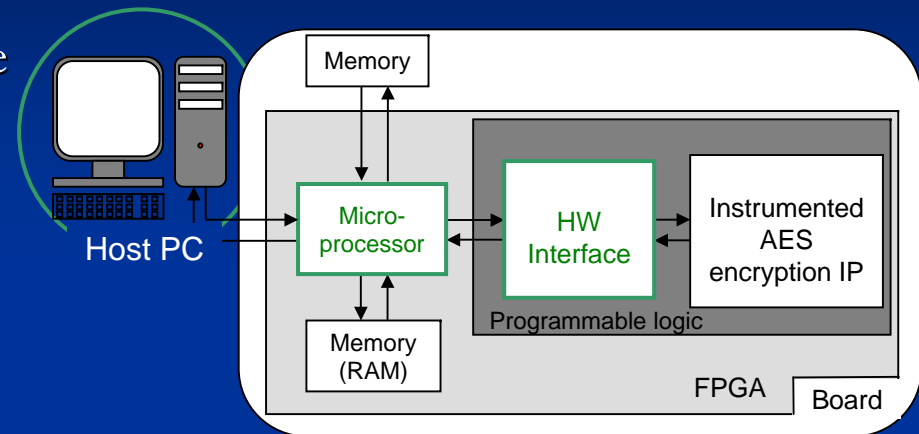


Cost Comparison

Architecture	Notes	Area Overhead	Throughput Reduction
Multiple Parity Bits Bertoni et al., TC '03	One parity bit per byte, expensive SBox protection	33%	3%
Inverse Process Karri et al., DAC '01	Dec after enc at block, round or operation level	19% - 38%	23% - 61%
Pipeline Recomputation Wu and Karri, DFT '01	Uses unused stages to redo computation in RC6	50%	18%
Single Parity Bit Karri et al., CHES '03	One parity bit per block Aimed at stuck-at faults	18% - 24%	NA
Non-linear Code Karpovsky et al., DSN '04	Non-linear scalable cubic network	77%	13%
DDR	Suitable for fast designs in slower systems	36%	15% - 55%

Fault Injection

- Fault injection was based on hardware emulation
- Injection software ran on the FPGA PowerPC
 - Reduced communication, thus faster execution of the campaign due to less wasted time
 - Load can be distributed at any level: hw logic, FPGA PPC, host
- Extra logic is added to the original AES description
 - For each targeted flip-flop, one XOR is inserted between the FF and the combination block at its input



Error Detection

Instrumented Target		Result Class [%]			
Location	Size (bits)	Silent	Undetected	False Pos	Detected
<u>Protected targets:</u>					
Linear layer	16*	66.10	0	0	33.90
SBox Output	16*	33.90	0	33.90	32.20
Inner SBox	24	1.88	0.06	50.72	47.34
<u>Non protected targets:</u>					
Misc ctrls	22	0.24	2.45	27.45	69.86
Key ctrls	3	17.68	53.27	2.91	26.15
Main FSM	19	0	16.30	1.87	81.63
Aux FSM	9	4.36	0.20	4.92	90.52
FSM Synchr	6	15.74	84.26	0	0

* Full search on single byte (8-bit target) gave the same results

Coverage Comparison

Architecture	Area Overhead	Throughput Reduction	Coverage Byte Errors in Datapath
Multiple Parity Bits Bertoni et al., TC '03	33%	3%	~67%
Inverse Process Karri et al., DAC '01	19% - 38%	23% - 61%	100%
Pipeline Recomputation Wu and Karri, DFT '01	50%	18%	~100%
Single Parity Bit Karri et al., CHES '03	18% - 24%	NA	~67%
Non-linear Code Karpovsky et al., DSN '04	77%	13%	~100%
DDR	36%	15% - 55%	~100%

Vulnerabilities

- DDR applies to data path only, control unit must be addressed with other protection means
 - Protection of the control unit is envisioned in a more recent version, exploiting selected duplication, transition verification, state validation
- Coverage of the data path is not 100% for multiple-bit errors
 - A small percentage (0.06%) of errors injected into the inner registers of SBoxes is not detected: this issue is currently under investigation
- Permanent fault may not be detected
 - They are outside the scope of this work, which is focused on transient faults (either natural or intentional)
- Tailored attacks are not detected
 - The attacker must be able to inject the same error value in the same location at very specific time slots: very difficult and unlikely with current attack capabilities

Conclusions

- The DDR approach is an alternative computation template to improve computation parallelism with scarce resources
- Like other solutions, more complex routing implies lower maximum frequency...
- ... but embedded in slower-clock systems it may double the throughput, or allow error detection by recomputation
- Coverage of short (one-cycle) multiple-bit errors in the data path is almost 100%
- Attacks are possible if the same error is injected twice at specific time slots, which is unlikely:
 - The attacker can finely control the injected error value
 - The second error value is equal to the first one by chance
 - Errors are due to permanent faults