Security in the Age of Nanocomputing (Part 2)

Matthew Tan Creti

Hacking Devices

The ESA estimates its total worldwide losses due to piracy at $3 billion annually [2]

One million unlocked iPhones could cost Apple $300 to $400 million in future revenue and profit [1]

Trends of Nanocomputing

- Nanotechnology is enabling ubiquitous embedded devices (e.g. media players, mobile ad-hoc networking, RFID, smart cards, sensor networks)
- “With this ‘embedded systems everywhere’ paradigm comes an ‘embedded security everywhere’ question” [3]

Side Channel Attacks

- Classical cryptography considers abstract computational adversaries, modeled as Turing machines
- Physical cryptography takes into account specific implementations
- Information that an adversary can use to crack a device include:
  - Power consumption
  - Electromagnetic radiation
  - Computation timing
  - Scan test access
- Fault injection attacks

Side Channels

- Ball Grid Array (BGA) replacement
- Timing based
- Electromagnetic
- Differential Power Analysis (DPA)

https://www.blackhat.com/presentations/bh-usa-07/De_Haas/Presentation/bh-usa-07-de_haas.pdf

Levels of Hackers

1. *Beginners* are just getting started or hack out of curiosity. They rarely put in great effort unless they have access to step by step information. A simple encoding scheme may be enough to deter them.
2. *Independent* class know where to find what they need and are willing to put time, effort, and money into their endeavor. This class should not be underestimated, strong encryption algorithms should be used.
3. *Business* class are trying to get a step ahead of their competition. If the cost outweigh the gains they may give up. Short of that, protecting IP from the business hacker is very difficult.
4. Government class is nearly impossible to protect against due to almost unlimited resources.

Designing Resistant Circuits against Malicious Faults Injection Using Asynchronous Logic

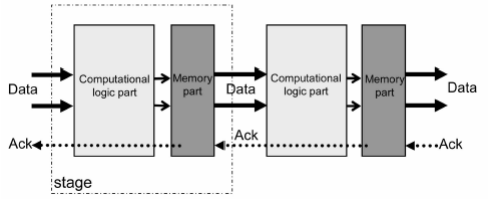
Yannick Monnet, Marc Renaudin, and Regis Leveugle. "**Designing Resistant Circuits against Malicious Faults Injection Using Asynchronous Logic**." IEEE Transactions on Computers, September 2006.

Yannick Monnet, Marc Renaudin, and Regis Leveugle. "**Asynchronous Circuits Sensitivity to Fault Injection**." IEEE International On-Line Testing Symposium 2004.

QDI Asynchronous Logic

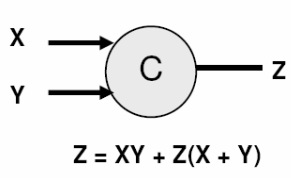
- Quasi Delay Insensitive (QUI)
- Higher speed
- Reduces electromagnetic interference
- Lower Power
- Overhead requires more than twice the area as equivalent synchronous circuit

Asynchronous Logic



- Composed of individual modules which communicate with each other by means of point-to-point communication channels
- A module becomes active when it senses the presence of incoming data
- After computation it send the result to the output channels
- Requires handshake protocol

Muller Gate



X	Y	Z
0	0	0
0	1	Z^{-1}
1	0	Z^{-1}
1	1	1

- A Muller gate outputs a 1 when both inputs are 1 and a 0 when both inputs are 0. Otherwise its output maintains its original state.

Muller Gate Sensitivity

Definition: An N-input Muller gate is said to be M-sensitive to zero (respectively, one) if, and only if, exactly M of its inputs as well as its output are equal to zero (respectively, one). In this case, if M faults are injected (or propagated) to these M inputs, the gate generates a rising (respectively, falling) transition.

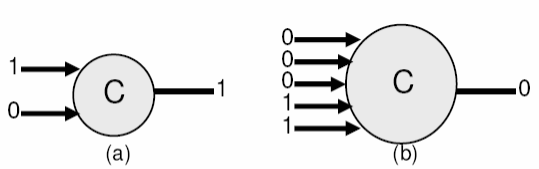


Figure 4. A "1-sensitive to 1" Muller gate (a) and a "3-sensitive to 0" Muller gate (b)

Muller Gate Use in QDI

- Muller gates are useful for synchronization of many inputs, for example to wait until all ready signals from preceding logic gates are set
- They are also able to store previous state
- For this reason Muller gates are used extensively in both the computational and memory parts of QDI logic
- **Definition:** The global circuit state is defined as the state of its Muller gates implemented in the memory parts. These gates hold data information at the behavioral level. Because they need an initial state, all Muller gates of memory parts have Set (MS) or Reset (MR) inputs.

4 Phase Handshake

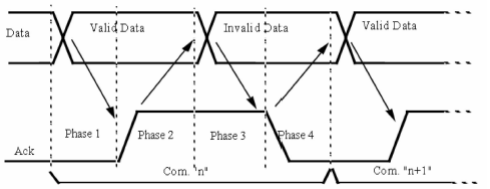


Table 1. Dual rail encoding of the three states required to communicate 1 bit

Channel data	A1	A0
0	0	1
1	1	0
Invalid	0	0
Unused	1	1

1. Valid data element is detected
2. Data element is acknowledged
3. Data element is reinitialized (return-to-zero)
4. Acknowledgment signal is reset

QDI Inherent Robustness against Attacks

- Isochronic forks add delay insensitivity
- Multirail encoding and isochronic forks make differential power analysis (DPA) more difficult

Fault Models

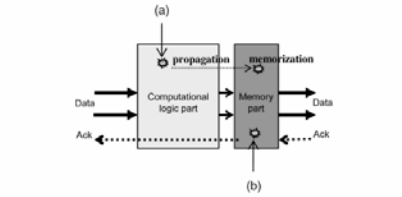


Fig. 4. (a) A transient fault injection in the computational part. (b) A memory bit flip in the memory part.

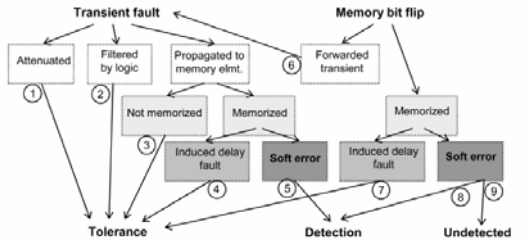
- *Delay Faults* modify the time needed for a transition to occur at a gate output. This fault does not affect the circuit logical function except on an isochronic branch (the only part of a QDI circuit assumed to have correct timing)
- *Transient faults* are usually cased by a current peak that toggle a logic level. They can propagate to memory.
- *Memory Bit Flips* are faults injected straight on the memory part of the chip

Behavioral Analysis of QDI Circuits

Transient fault caused by pulse

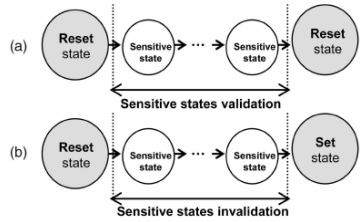
- Case 1: The pulse is attenuated until it disappears
- Case 2: The transient fault is logically masked in the computational part
- Case 3: It is propagated up to the memory part inputs, but not memorized. It is filtered by memory part Muller gates.
- Case 4: It is propagated up to a Muller gate input an is memorized. However, the cell was going to flip in the normal execution process anyway. So the fault cased a *premature firing* but did not violate the sequence of global state (i.e. the next state will still be correct)
- Case 5: The transient fault is propagated to a Muller gate and is memorized. The gate would not have flipped in normal execution. At this point we have a soft error which can cause the circuit to fail because global state has been corrupted.

Behavioral Analysis of QDI Circuits



- Case 6: The flipped output of the Muller gate behaves as a transient fault. This is because once a Muller gate flips it keeps that state until the next flip. This fault is propagated to the computational part of the next stage.
- Case 7: If the gate in case 6 was selected to flip in the current phase anyway then the fault is interpreted as a delay fault.
- Case 8: The bit flip is memorized and the gate was not selected to flip in this phase of the protocol. Similarly to case 5, this is a soft error, which could lead to a circuit failure. Some form of alarm could detect the wrong code.
- Case 9: Same as case 8, but the faulty state belongs to the set of legal states so it can not be detected.

Sensitive Validation/Invalidation

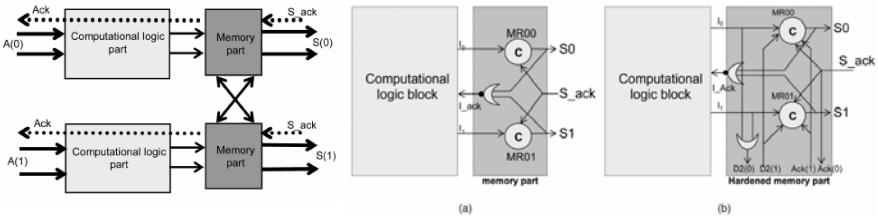


- An N input Muller gate is in one of the following states
  1. M-sensitive to zero 0<M<N
  2. M-sensitive to one 0<M<N
  3. Set
  4. Reset
- If the gate goes from reset to M-sensitive and back to reset then these states should be *validated* as fault-sensitive states. Because they could have caused a soft error.
- If the gate goes form rest to M-sensitive to set then the sensitivity of these states are invalidated because they can not produce a soft error. The worst they can do is cause a delay fault.

Simulator Tool

- The total time a gate spends in a sensitive states with respect to the total time the gate is monitored is a metric collected for each gate and each level of sensitivity
- The *N-sensitivity* of a circuit is the mean time spent in N-sensitive states by all monitored gates
- A circuit sensitivity map can be drawn to identify the most fault-sensitive blocks or gates
- The tool provides a quantitative comparison of the robustness of different architectures

Channel Hardening

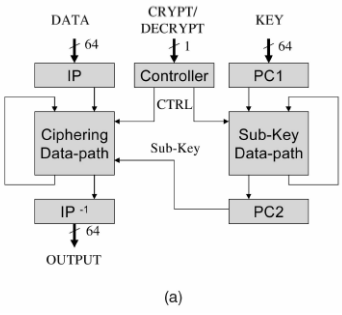


- We now look at an example of synchronizing two channels so that one channel cannot memorize data before data is ready for the other one.
- For a single fault, if the fault is correct then it will be masked, if it is not then the “11” code can be detected by an alarm
- This makes the assumption that the correct data has enough time to propagate to the memory block before the faulty data is acknowledged. When the assumption is wrong we get a soft error.
- This can be extended to synchronizing n-bit channels

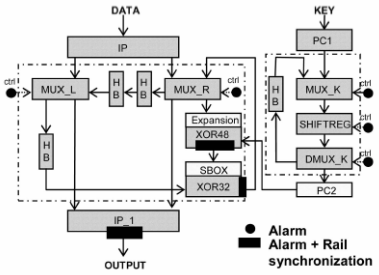
Multirail Weaknesses

- Multirail encoding helps to detect many soft errors, however there are still weaknesses
- *Data generation:* A soft error generates a valid code during the reset phase. The result is that an additional data element is inserted in the circuit that can not be detected.
- *Data vanishing:* A soft error removes a data element.
- *Data modification:* A data element is turned into another valid data element. (e.g. “01” to “10”)
- In loop structures it may be the case that the number of data elements is controlled. In this case the circuit will deadlock.

Hardened DES Architecture



(a)



(b)

Results for a Single Block

Sensitivity Analysis of the Muller Gates of the Reference Block and the Hardened Block

Reference block (MULLER2)			
state	total time (ps)	Occ	ports
Valid '1 to 0'	32008	9	A
Invalid '1 to 0'	6272	1	A
Invalid '1 to 0'	25120	8	B
Invalid '1 to 1'	20490	9	B
Reset	51330	16	
Set	17780	9	

Hardened block (MULLER4)			
state	total time (ps)	Occ	ports
Valid '1 to 0'	23330	7	D1
Valid '2 to 0'	20850	9	D2,D1
Valid '2 to 0'	3440	7	Ack2,D1
Valid '3 to 0'	2440	6	Ack1,D2,D1
Valid '3 to 0'	30	1	Ack2,D2,D1
Valid '3 to 0'	7550	7	Ack2,Ack1,D1
Invalid '1 to 0'	7690	9	D2
Invalid '1 to 1'	8370	9	D2
Invalid '2 to 0'	6030	9	D2,D1
Invalid '2 to 1'	50	1	Ack1, D2
Invalid '2 to 1'	3160	8	D2, D1
Invalid '3 to 0'	2950	9	Ack1,D2,D1
Invalid '3 to 1'	160	1	Ack2,D2,D1
Invalid '3 to 1'	4190	8	Ack1,D2,D1
Reset	53820	16	
Set	28340	9	

- For one of the active blocks it was 1-sensitive 21% of the time for unhardened and 12% of the time for hardened
- For blocks that are mostly inactive it went from 99% to 1%

Laser Fault Injection Results

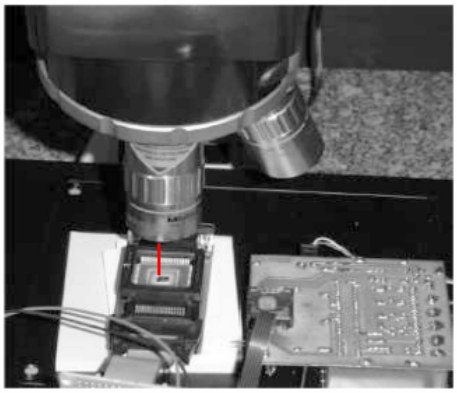


Fig. 14. Gemplus laser platform [21].

Fault Tolerance Results

SBOX	Number of shots	Number of errors
Reference	5600	955 (17%)
Hardened	5600	377 (7%)

SBOX	Detected errors	Undetected errors
Reference	642 (67%)	313 (33%)
Hardened	377 (100%)	0 (0%)

References

1. Connie Guglielmo, “Apple Users Unlocked 1 Million IPhones, Analyst Says,” bloomberg.com, January 2008.
2. Eric Bangeman, “Console mod-chippers busted in nationwide raids,” arstechnica.com, August 2007.
3. Jean-Jacques Quisquater and François Xavier Standaert, “Physically Secure Cryptographic Computations: From Micro to Nano Electronic Devices,” Workshop on Dependable and Secure Nanocomputing 2007.
4. Jeremy Lee, Mohammad Tehranipoor, Chintan Patel, Jim Plusquellic. “Securing Designs against Scan-Based Side-Channel Attacks,” IEEE Transactions on Dependable and Secure Computing, October 2007.