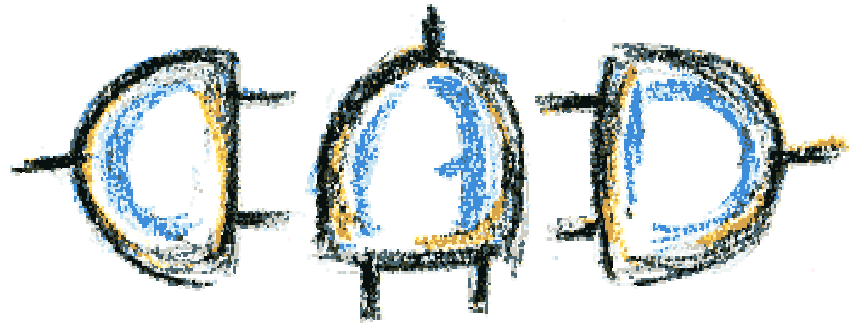


Single-event effects in SRAM-based FPGAs: effects and possible solutions

Matteo Sonza Reorda,
Luca Sterpone, Massimo Violante



www.cad.polito.it

Goal

- Discuss how Single Event Effects (SEEs) affect designs implemented through SRAM-based FPGAs and analyze hardening solutions
- Constraints:
 - Non rad-hard SRAM-based FPGAs are used
 - SEEs in the FPGA's configuration memory are considered, only
 - System level view (Cross section? What is that?)

Outline

- Introduction
- A system-level view of SEE effects
- Hardening approaches: introduction
- Hardening approaches: masking
- Hardening approaches: correction
- Conclusions

Outline

- Introduction
- A system-level view of SEE effects
- Hardening approaches: introduction
- Hardening approaches: masking
- Hardening approaches: correction
- Conclusions

Introduction

- SRAM-based FPGAs are particularly appealing:
 - Very high flexibility
 - High performance
 - High pin count
 - Low costs for low-production volumes compared to ASICs
 - Reduced turn-around time
- Good candidate for replacing antifuse FPGAs in critical applications

FPGA's architecture (I)

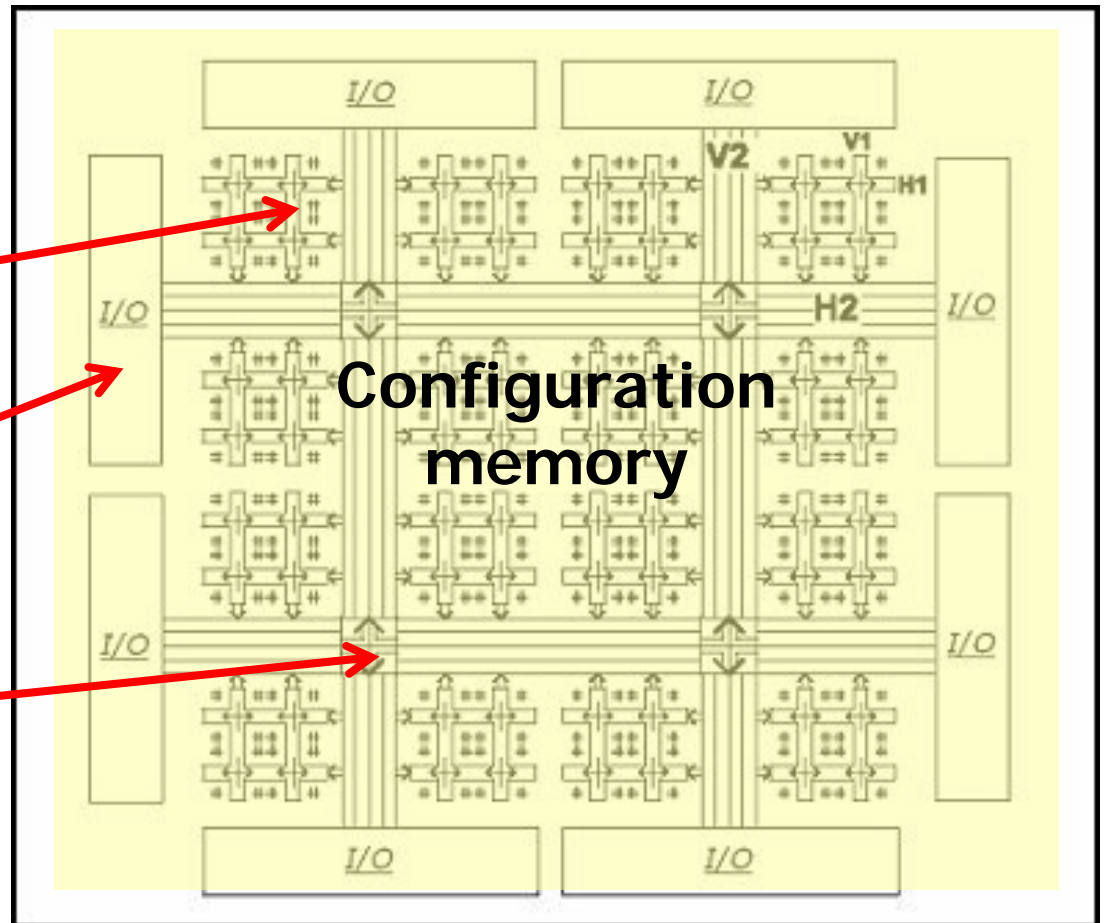
- Array of blocks
- Each block consists of an array of logic elements and routing channels
- Information about how the logic elements and routing channels work is stored in a SRAM-based configuration memory

FPGA's architecture (II)

Logic blocks:
LUTs, registers,
memories

I/O blocks

Routing resources



Major vendors

- Xilinx (Spartan, Xilinx)
- Altera (Cyclone, Stratix)
- Lattice (LatticeEC, LatticeECP)

SEE in SRAM-based FPGAs

- SRAM-based FPGAs embed:
 - User memory (registers, memory blocks, ...)
 - Configuration memory (LUTs, routing channels,...)
- SEEs modify either:
 - The user memory \Rightarrow information that the circuit elaborates
 - The configuration memory \Rightarrow information that defines how the circuit works!

User vs Configuration memory

- SEEs in the configuration memory are not negligible

Device	User memory [kbits]	Configuration memory [kbits]
XC2VP2	~200	~1,500
XC2PX20	~1,600	~8,000
XC2VP100	~8,000	~35,000

Failure rate due to SEEs

- FIT (1 failure in 10^9 hours) for some SRAM-based FPGAs (Xilinx and Altera)

Altitude [feet]	FIT
0	1,150
5,000	3,900
60,000	540,000

- Typical FIT rate for a highly reliable application:
10 to 20

Up to now we understood that...

- SRAM-based FPGAs are very sensitive to SEEs
- Both user memory and configuration memory must be hardened
- Techniques are needed to:
 - Understand how SEEs affect FPGA's resources
 - Make designs insensitive to SEEs

Outline

- Introduction
- A *system-level view of SEE effects*
- Hardening approaches: introduction
- Hardening approaches: masking
- Hardening approaches: correction
- Conclusions

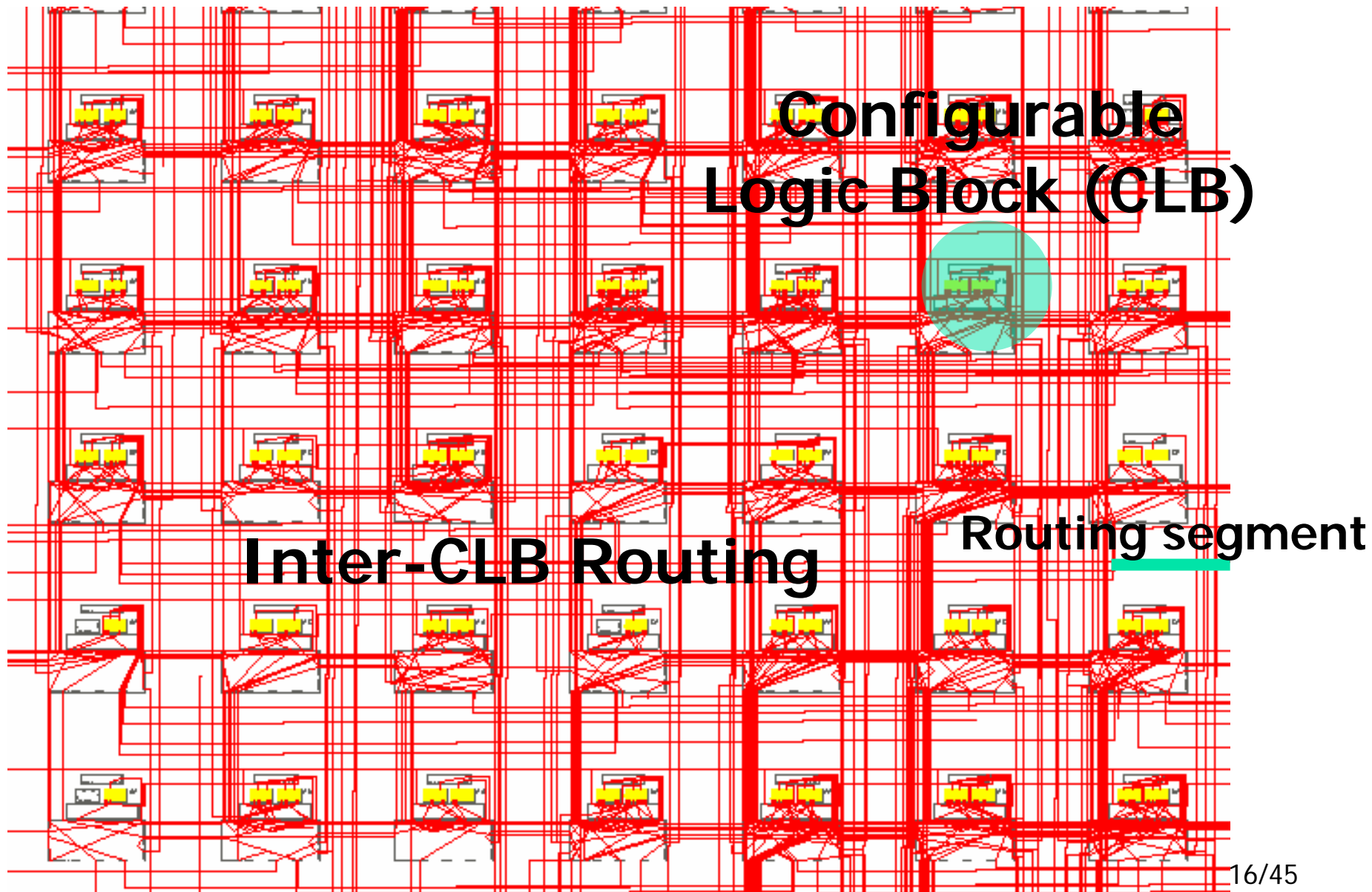
A system-level view of SEE effects

- SEEs in user memory
 - Close to SEEs in ASICs
 - **Easy to model**: bit-flip of a flip-flop or register
 - **Easy to predict**: simulation of the circuit is sufficient
 - Not addressed here
- SEEs in configuration memory
 - **Difficult to model**: the effect depends on what the affected memory cell controls
 - **Difficult to predict**: a detailed model of the FPGA is needed

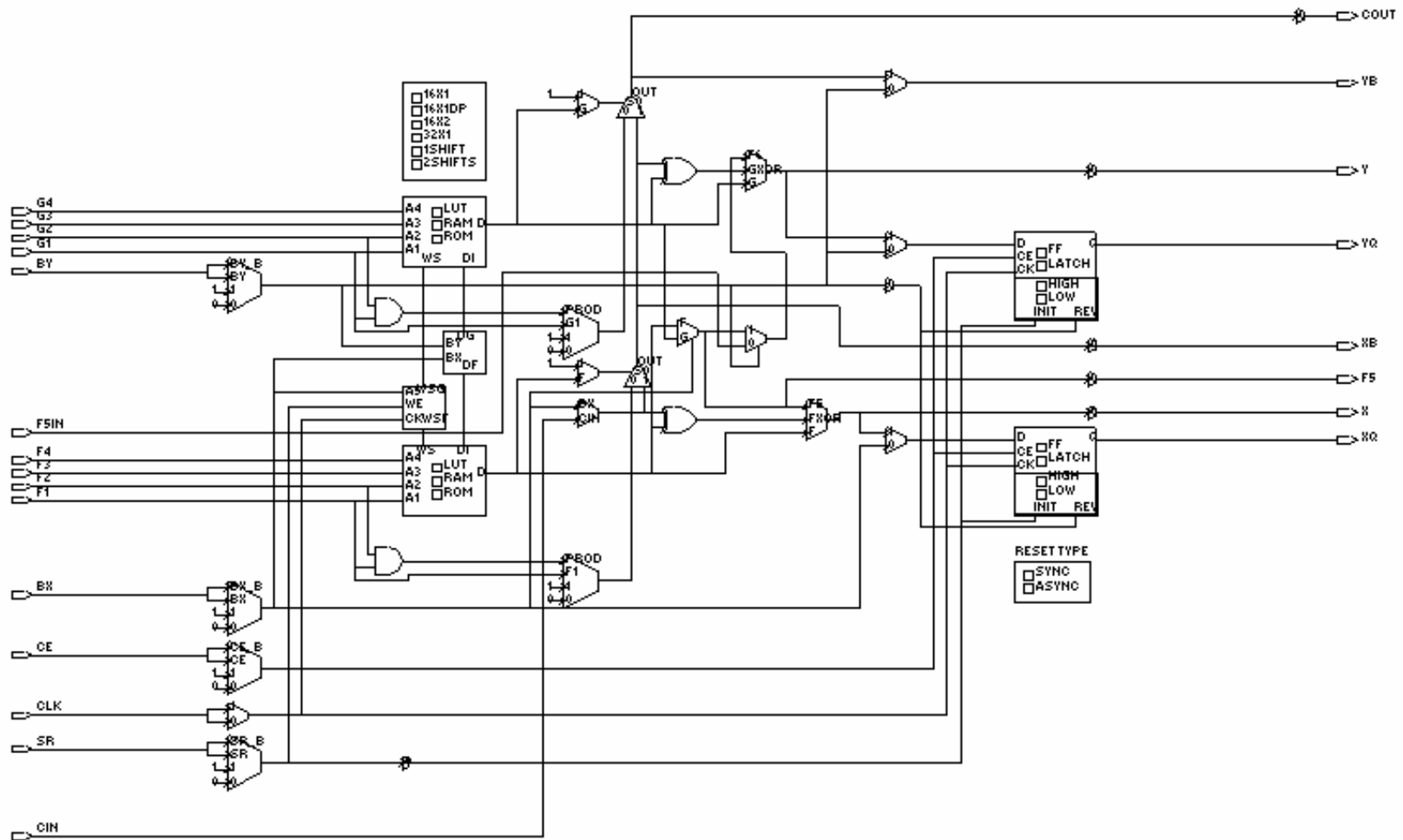
SEEs in configuration memory

- Effects on:
 - How logic functions are implemented
 - How FPGA's resources are initialized
 - How routing channels are used
- To model them it is mandatory to understand:
 - The resources available on the FPGAs
 - The mapping between configuration-memory's bits and FPGA' resources

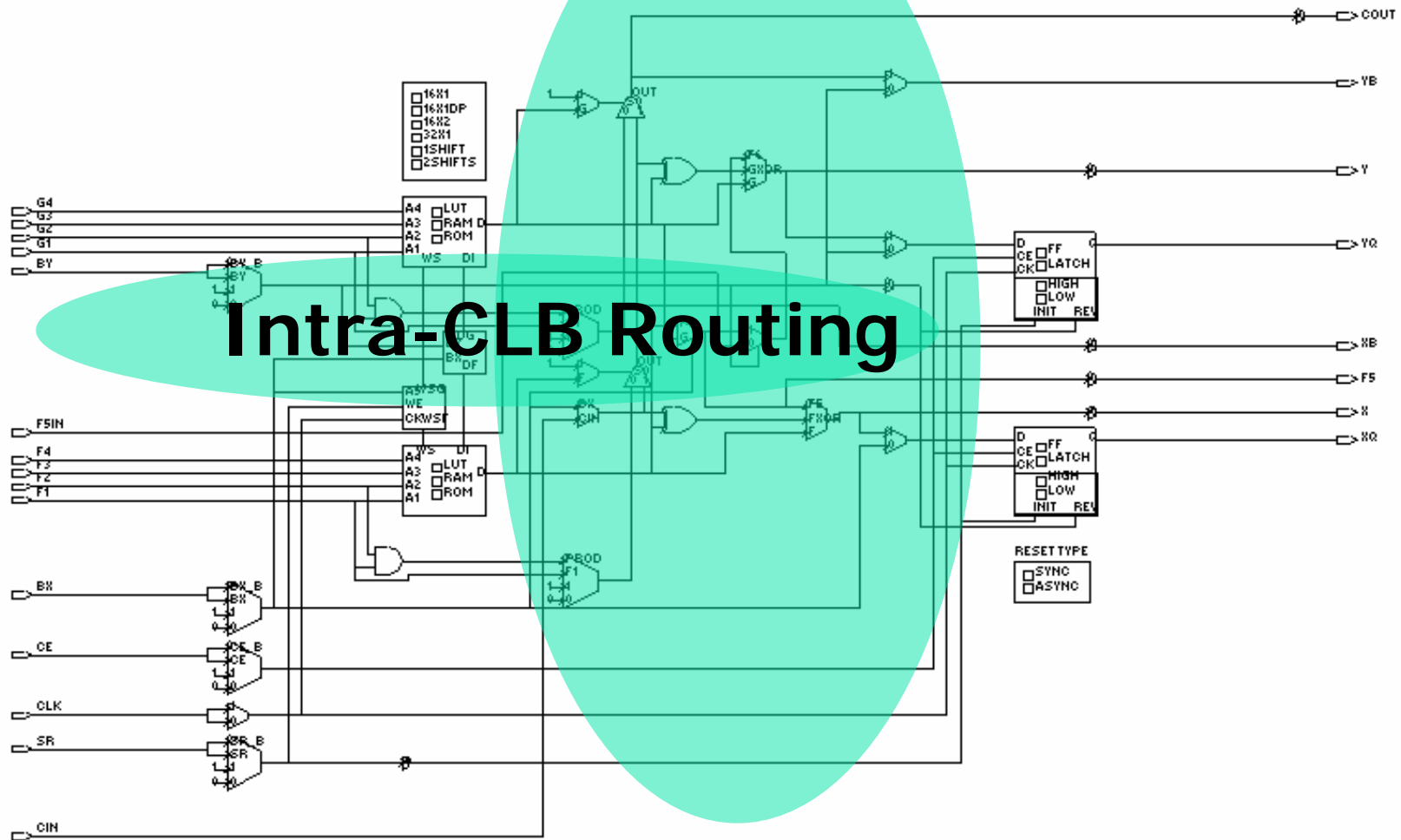
FPGA's architecture



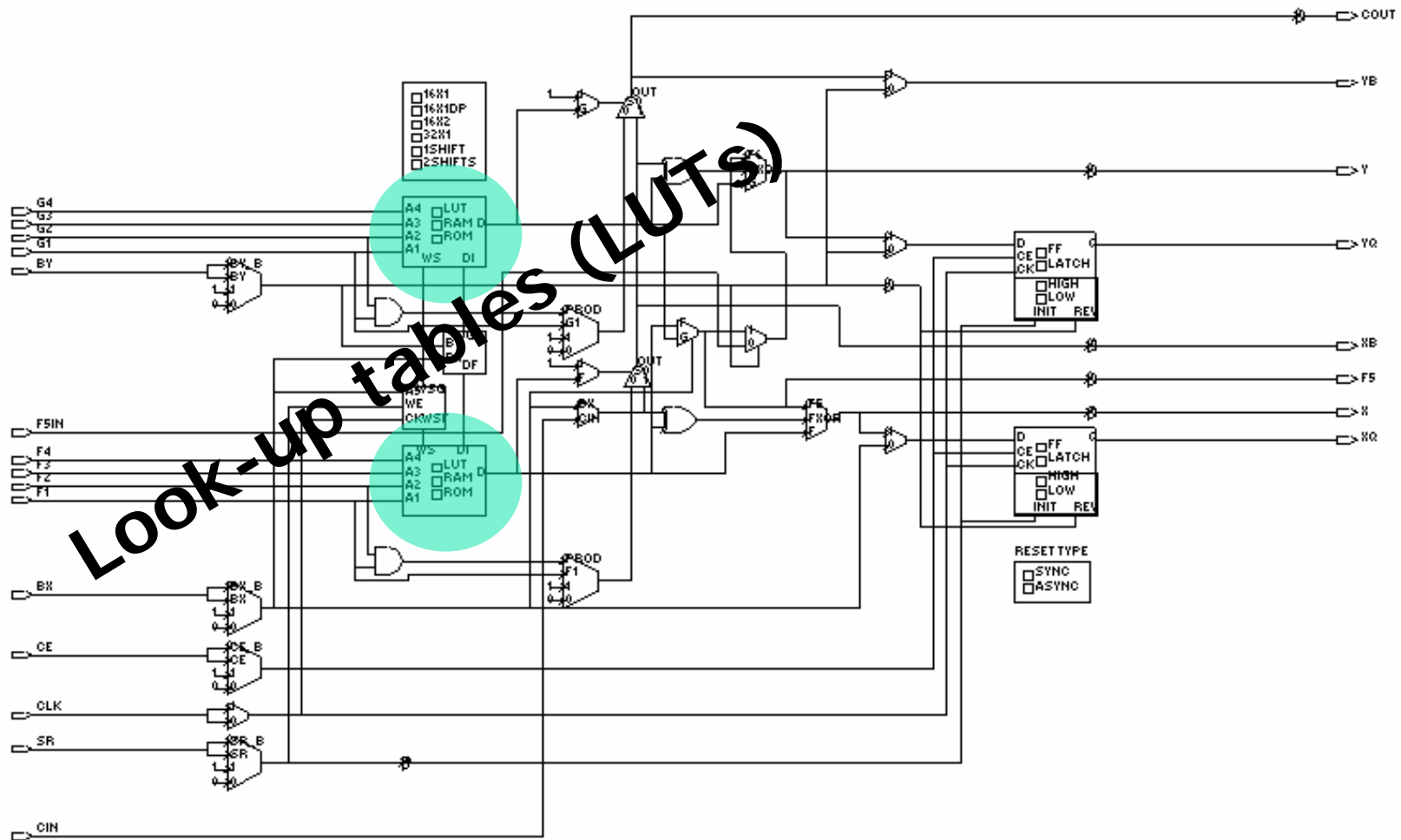
Configurable Logic Block (CLB)



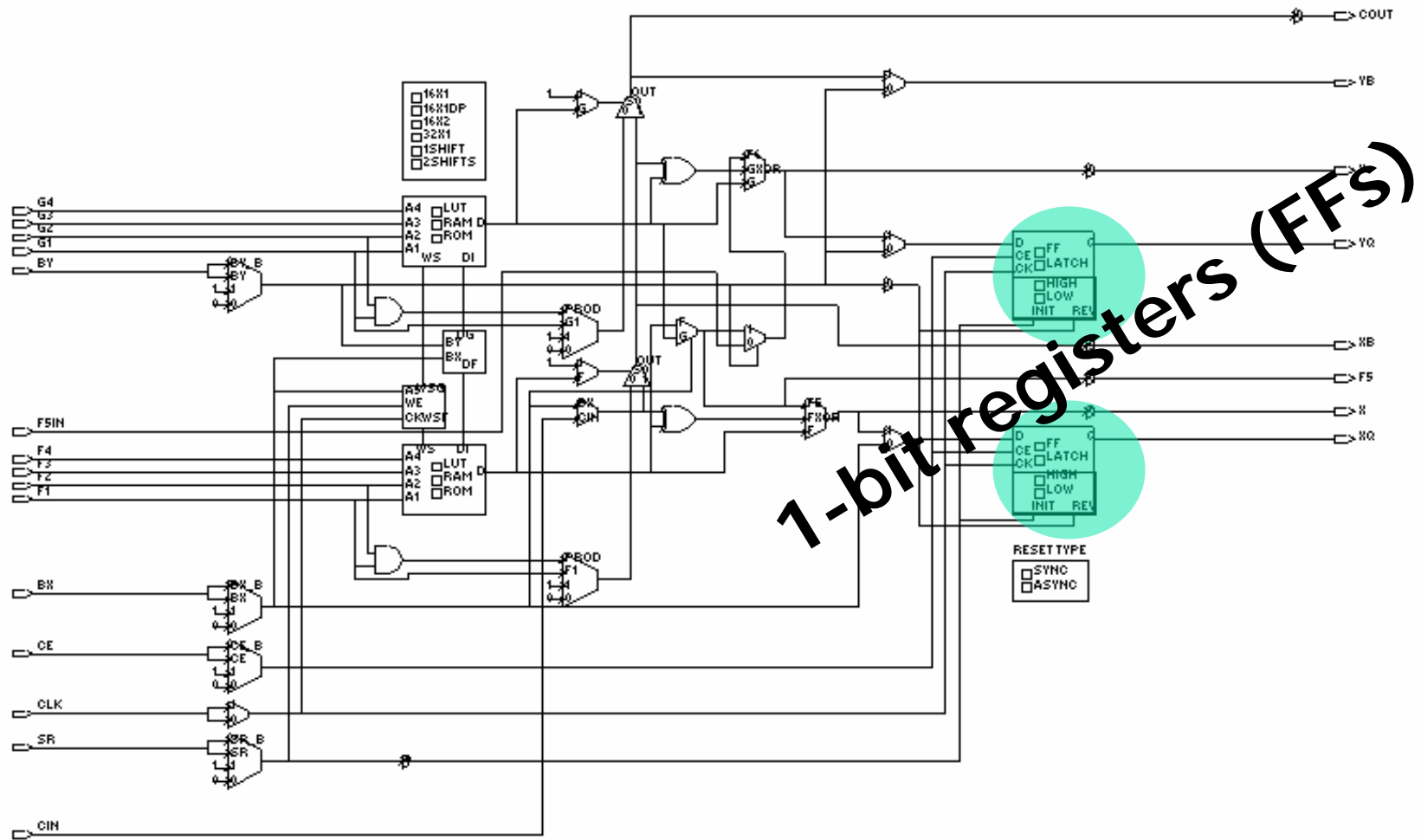
Configurable Logic Block (CLB)



Configurable Logic Block (CLB)



Configurable Logic Block (CLB)



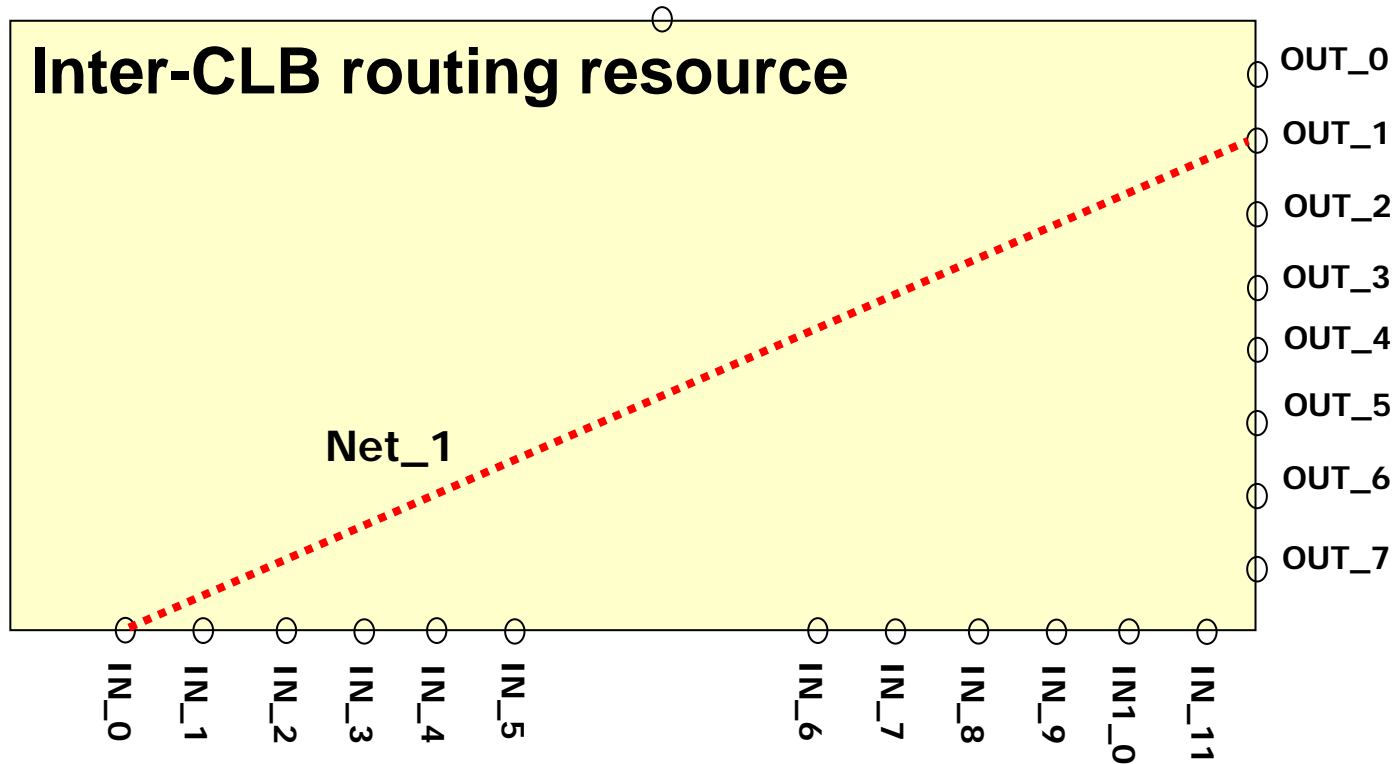
SEE in FPGA's resources

- SEEs affecting CLB resources result in:
 - **LUT defects:** modifications to the implemented logic function
 - **MUX defects:** modifications to the intra-CLB routing
 - **Initialization defects:** modifications to the initialization of the CLB internal components (e.g., reset's type)

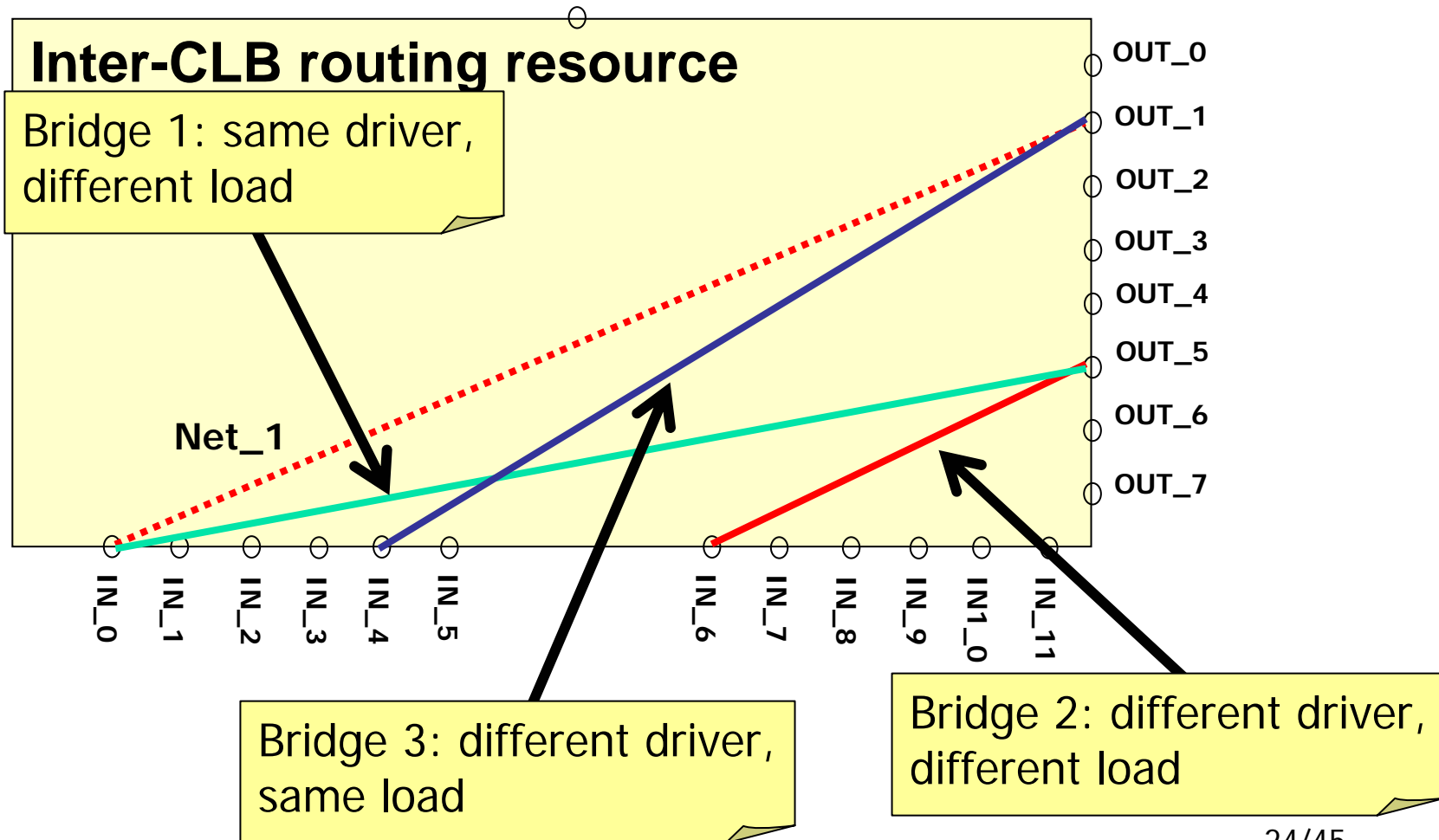
SEE in FPGA's resources (VI)

- SEEs affecting the configuration memory bits controlling inter-CLB routing:
 - **Open**: one enabled routing segment is disabled
 - **Bridge**: one enabled routing segment is disabled and a disabled one is enabled
 - **Short**: a routing segment is enabled that shorts together two already enabled routing segments

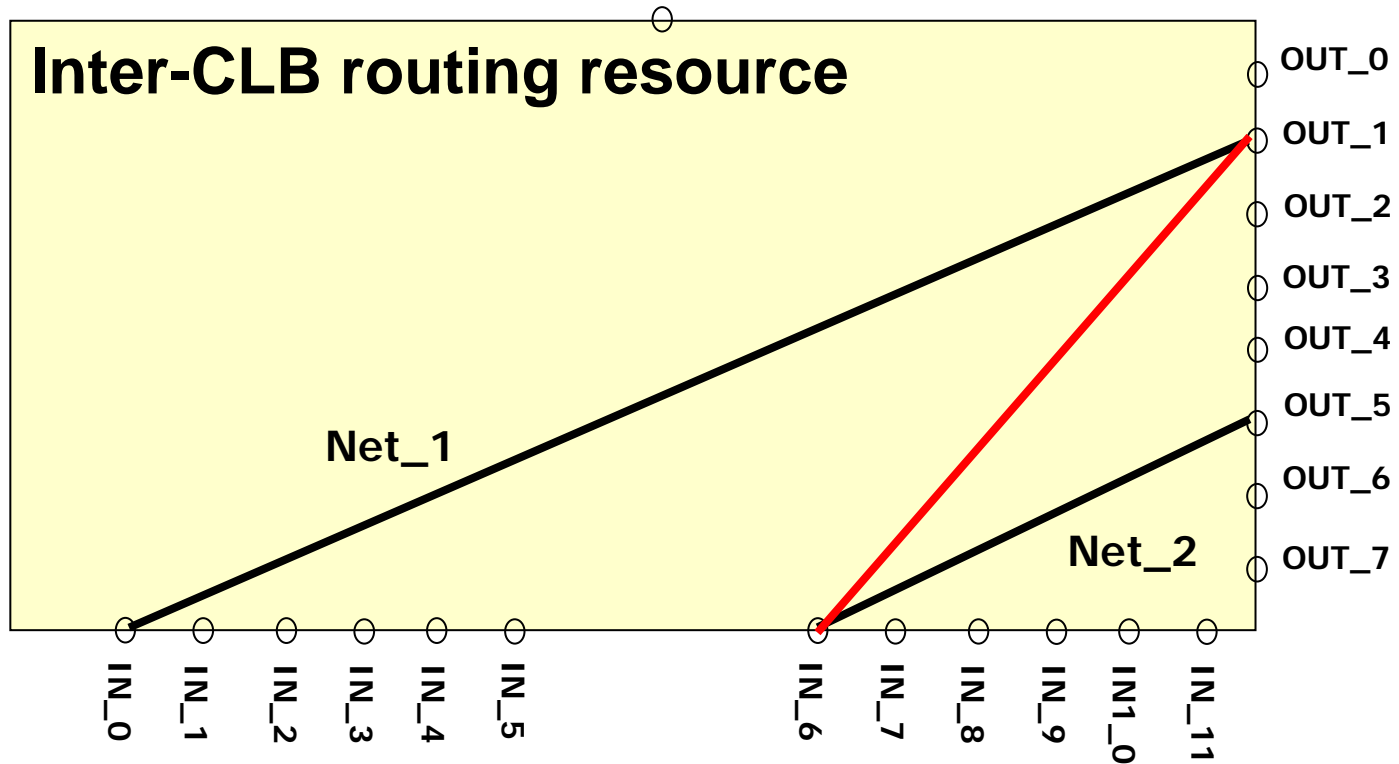
Open effect



Bridge effect



Short effect



Outline

- Introduction
- A system-level view of SEE effects
- Hardening approaches: introduction
- Hardening approaches: masking
- Hardening approaches: correction
- Conclusions

Hardening approaches

- Two needs:
 - **Masking**: to prevent the SEE's effects to propagate to the system's outputs
 - **Correction**: to remove the SEE's effects from the system
- Proposed solutions:
 - Modify the **circuit architecture** the system implements to achieve **masking**
 - Modify the **system architecture** to achieve **correction**

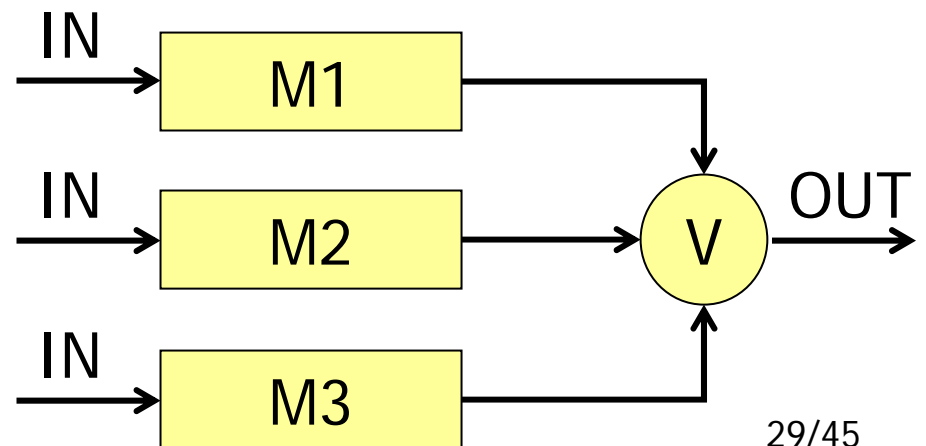
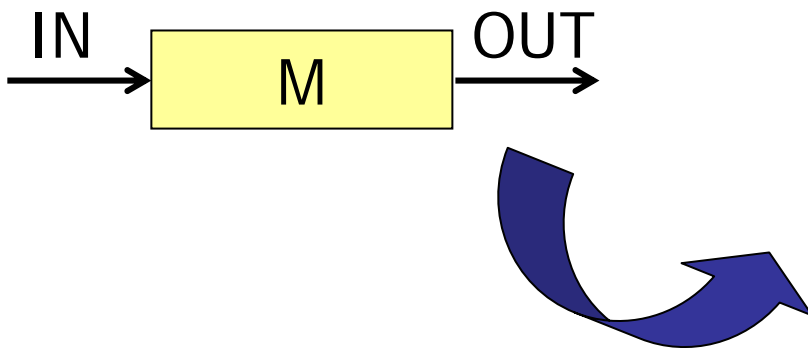
Outline

- Introduction
- A system-level view of SEE effects
- Hardening approaches: introduction
- Hardening approaches: masking
- Hardening approaches: correction
- Conclusions

Masking

- TMR approach:

- Triplicate any design element (logic, memories, interconnections, and inputs/outputs)
- Vote with majority voter (assumed as fault free)



Analysis of the TMR (I)

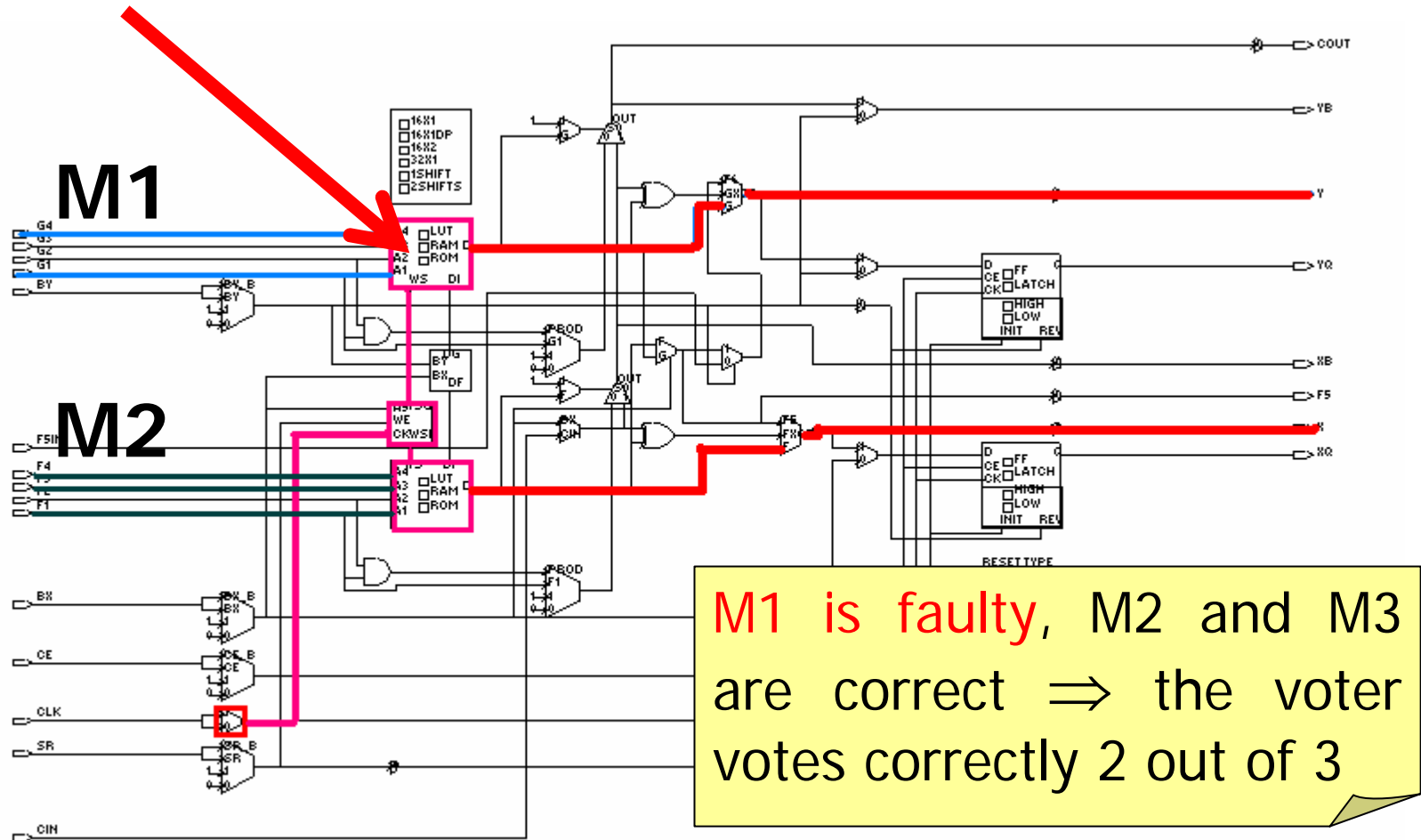
- Injection of SEE in the FPGA implementing TMR circuits
- SEE effects classified according to the affected resource:
 - CLB defects
 - Inter-CLB routing defects
- Two possible SEE effects:
 - **Critical**: it escapes the TMR
 - **Not critical**: the TMR masks it

Analysis of the TMR (II)

- CLB defects:
 - LUT defects are **not critical**:
 - Each function is implemented by 3 identical LUTs
 - One is faulty, but the other 2 continue to work
 - The voter decides correctly by voting 2 out of 3
 - MUX defects are **critical** if:
 - Same CLB implements 2 replicas M_i and M_j
 - Both M_i and M_j are faulty
 - Initialization defects are **critical** if:
 - Same CLB implements 2 replicas M_i and M_j
 - Both M_i and M_j are faulty

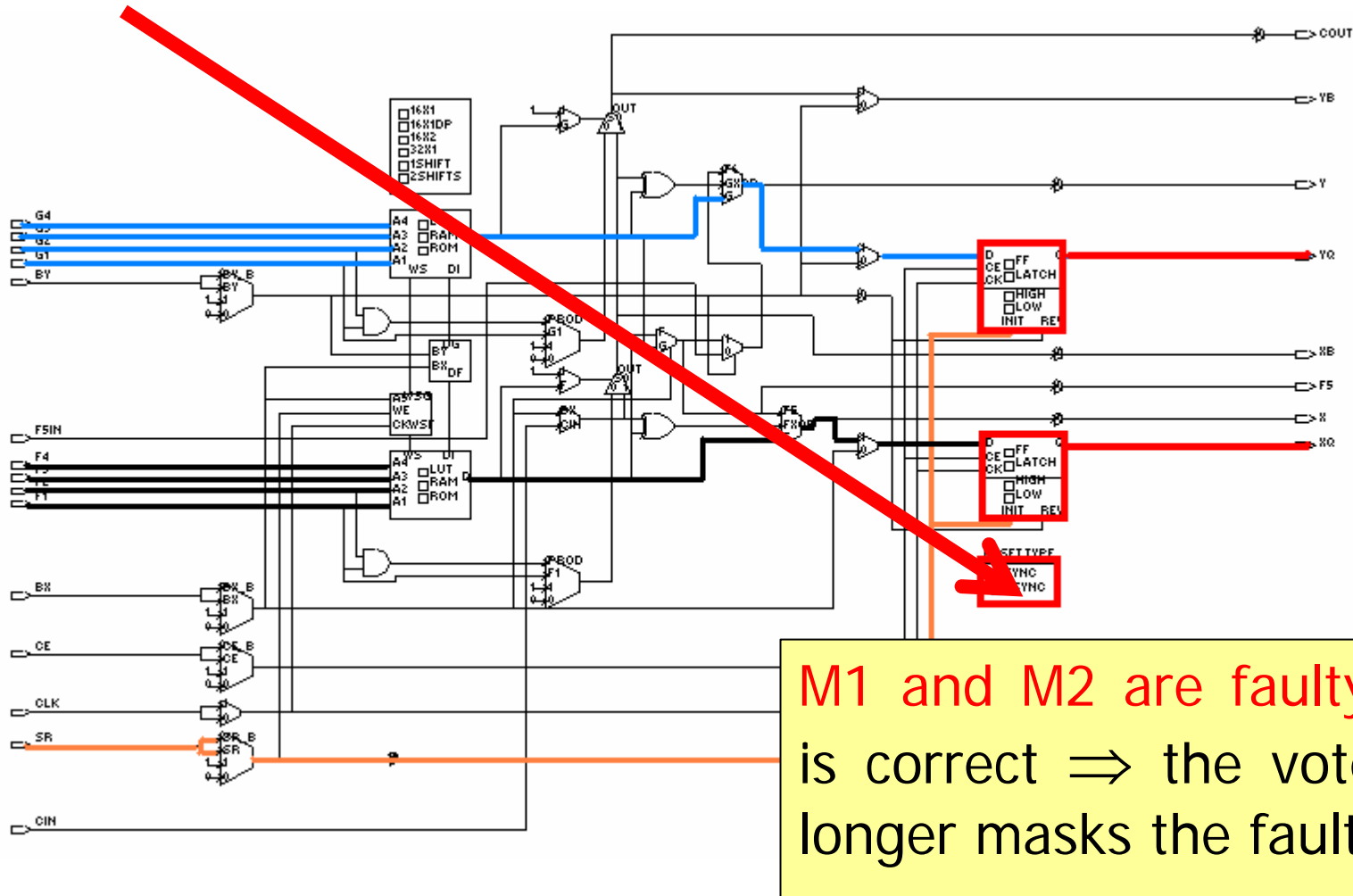
An example (II)

LUT defect



An example (IV)

The initialization defect changes reset from async to sync



Analysis of the TMR (III)

- Inter-CLB routing defects:
 - SEEs may be critical or not depending on how the design is routed
- Golden rule:
 - There is a multiple effect if different nets of different TMR replicas are routed by the same routing resource

A few figures

TMR	Injected faults [#]	Wrong Answers [#]
8-bit adder	15,000	1,352
16-bit adder	15,000	1,692
8-bit multiplier	15,000	1,977
Filter	15,000	1,981

Lessons learned

- Lesson 1:
 - Do not place M_i and M_j in the same CLB
 - Simple to implement: work on the placement constraints
- Lesson 2:
 - Avoid routing different nets of different TMR replicas by the same routing resource
 - Difficult to implement:
 - Ad-hoc developed router
 - Need a very good knowledge of FPGA's inter-CLB routing architecture

Improved TMR

Improved TMR	Injected faults [#]	Wrong Answers [#]	Reduction
8-bit adder	15,000	30	45x
16-bit adder	15,000	41	41x
8-bit multiplier	15,000	23	86x
Filter	15,000	44	45x

Overheads

Circuit	TMR		Improved TMR	
	Speed [MHz]	Area [# slice]	Speed [MHz]	Area [# slice]
8-bit adder	86	100	64	96
16-bit adder	85	103	62	105
8-bit multiplier	84	127	54	125
Filter	65	132	58	138

Up to now we understood that...

- TMR is very simple to implement and it provides hardening against some of the SEE's effects
- Some SEEs still escape TMR and even commercial implementations (e.g., XTMR by Xilinx) suffer from the same problem
- The **place** and **route** operations must be performed with dependability oriented tools (not yet available commercially)

Outline

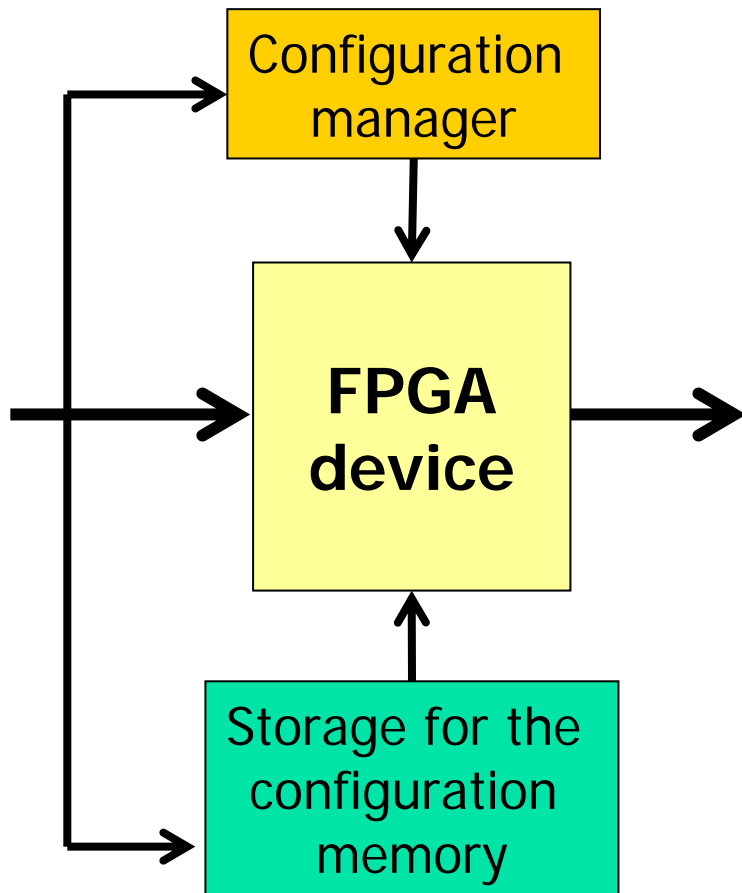
- Introduction
- A system-level view of SEE effects
- Hardening approaches: introduction
- Hardening approaches: masking
- Hardening approaches: correction
- Conclusions

Correction (I)

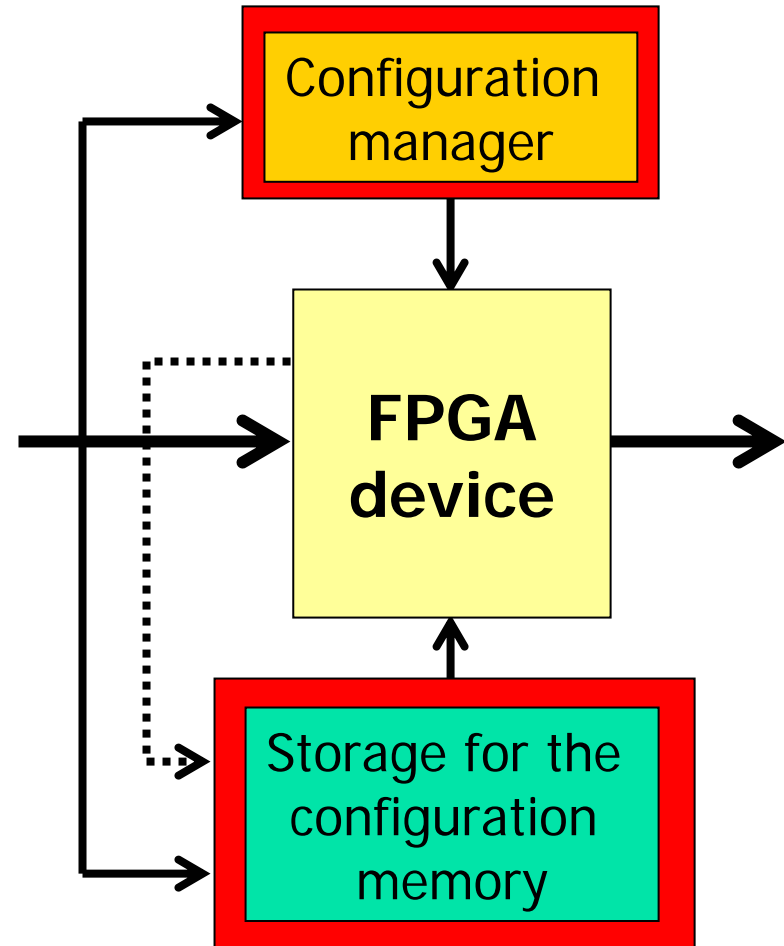
- Restore the correct configuration memory
- Scrubbing:
 - The whole configuration memory is periodically reloaded
- Partial Reconfiguration + Scrubbing:
 - The configuration memory is divided into separate segments
 - Each segment is read back one at a time and compared with a reference copy
 - If a mismatch is found, only the faulty segment is reloaded

Correction (II)

- Plain system



- System+Correction



Conclusions

- SRAM-based FPGAs may be used in critical applications provided that suitable **masking** and **correction** techniques are used
- Masking techniques are **not mature** enough (some faults still escape)
- **Design tools** may help in reducing escaped faults.