
Dependability Analysis: a New Application for Run-Time Reconfiguration

Régis Leveugle*, Lőrinc Antoni*+, Béla Fehér+

*** TIMA Laboratory**

Institut National Polytechnique de Grenoble (France)

+ Dep. Of Measurement and Information Systems

Budapest University of Technology and Economics (Hungary)

Motivations

- **Increasing use of fault injection approaches to:**
 - ◆ **Validate dependability characteristics** (post-design activity)
 - ◆ **Analyze faulty behaviors** (design activity)

- **Recent use of hardware emulation systems (FPGA-based)**
 - ◆ **Modifications in (synthesizable) VHDL models or at gate level**
 - ◆ **Less time consuming than simulations (experiment runs)**
 - ◆ **Possible "in-system" emulation**

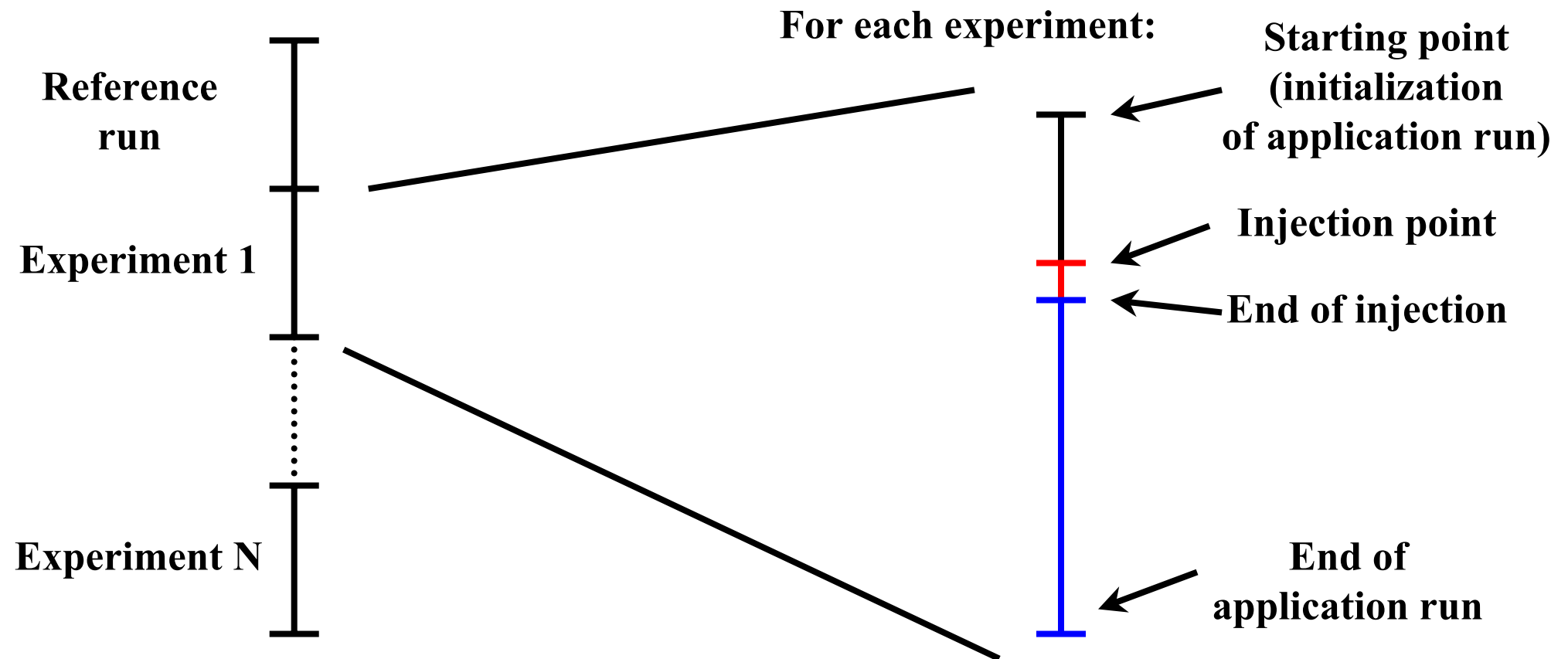
- **Focus of this work**
 - ◆ **Explore a new approach based on the reconfiguration capabilities**
 - ◆ **Fault injection of SEU-like faults in hardware prototypes (bit-flips)**
 - ◆ **Identify the main parameters for a reduction of the global length of the fault injection experiments**

Outline

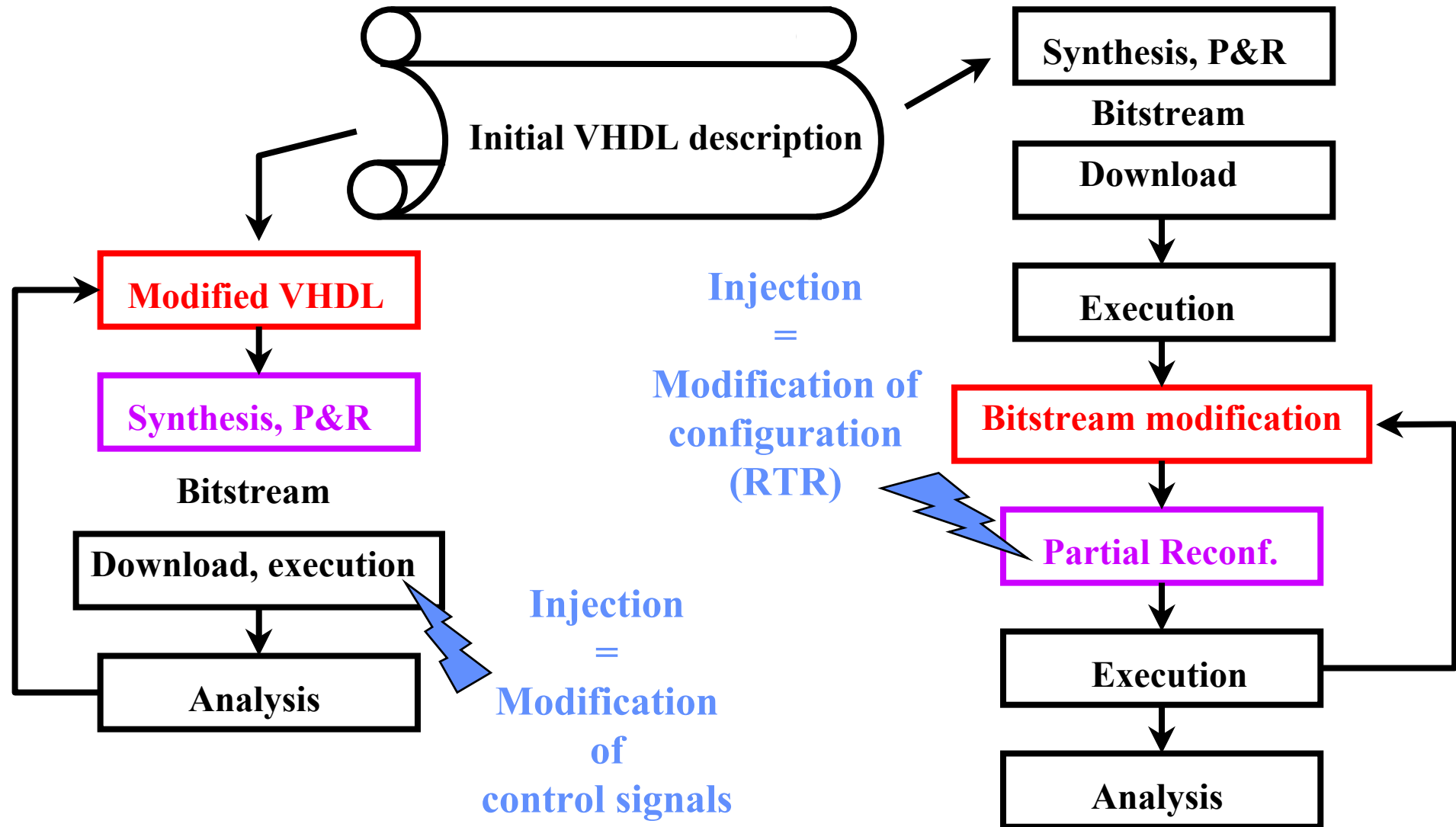
- ❑ **Fault injection and hardware prototyping**
- ❑ **Device-level reconfiguration for SET and SEU injection**
- ❑ **Implementation (Xilinx Virtex FPGAs)**
- ❑ **Discussion of results**
 - ◆ **Experiments with a development board**
 - ◆ **Device-based analysis**
- ❑ **Conclusion and perspectives**

A fault injection campaign

Injection campaign using simulation or emulation (**functional** analysis):



Alternative flows for fault injection in FPGAs



Main characteristics

- **"Classical" approach using FPGA-based prototypes:**
 - ◆ **Instrumentation of the initial circuit description**
 - **Control signals, observation outputs, extra hardware**
 - ◆ **Hardware limitations => several modified descriptions (synthesis, P&R)**
 - **RTR-based approach:**
 - ◆ **No modification of the initial description**
 - **Only one synthesis and P&R (no need for sub-campaigns)**
 - **Reduced hardware complexity of the prototype (smaller FPGA)**
 - **Better maximal emulation frequency**
 - ◆ **Use of partial reconfiguration capabilities of the device (e.g. Virtex, AT6000)**
 - ◆ **Use of read-back capabilities for internal signal monitoring**
 - ◆ **Direct (local) modification of the bitstream**
 - ◆ **One run-time reconfiguration per fault injection (or removal)**
-

Length of a fault injection campaign

□ Preparation times

- ◆ (P1) Modification of the initial circuit description
- ◆ (P2) Synthesis, P&R (one or several runs)
- ◆ (P3) Initial configuration(s) of the emulator (CTR)

□ Run times

- ◆ (R1) Application run (number of patterns per experiment * clock period * number of experiments)
- ◆ (R2) Communication with the host computer
- ◆ (R3) Run-time Reconfigurations of the prototype

RTR can be efficient if R3 (and potentially the loss in R2) is less than the gains on P1, P2, P3, R1

Two types of functional faults

- **Permanent or transient faults in the combinatorial parts**
 - => can be injected through structural reconfigurations (two reconfigurations for transient faults, or SETs)**

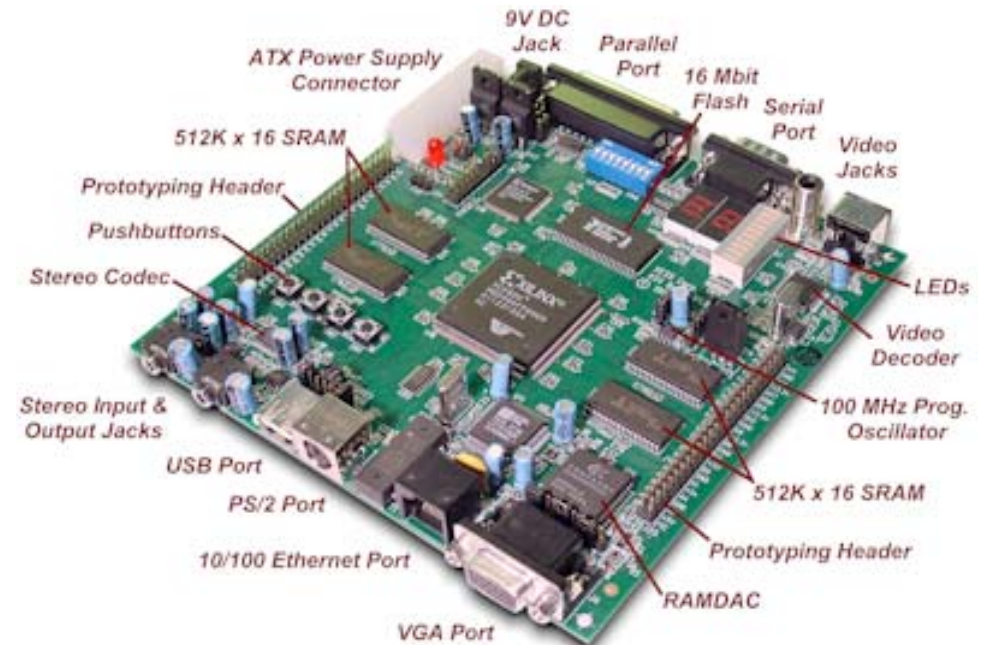
- **Transient faults in flip-flops (asynchronous bit-flips, or SEUs)**
 - ◆ **Bit-flip: depends on the execution context (current value in the flip-flop)**
 - ◆ **Direct modification in the flip-flop without activation of the clock signal (that may be a gated clock)**

=> modification of behavior

Experimental environment

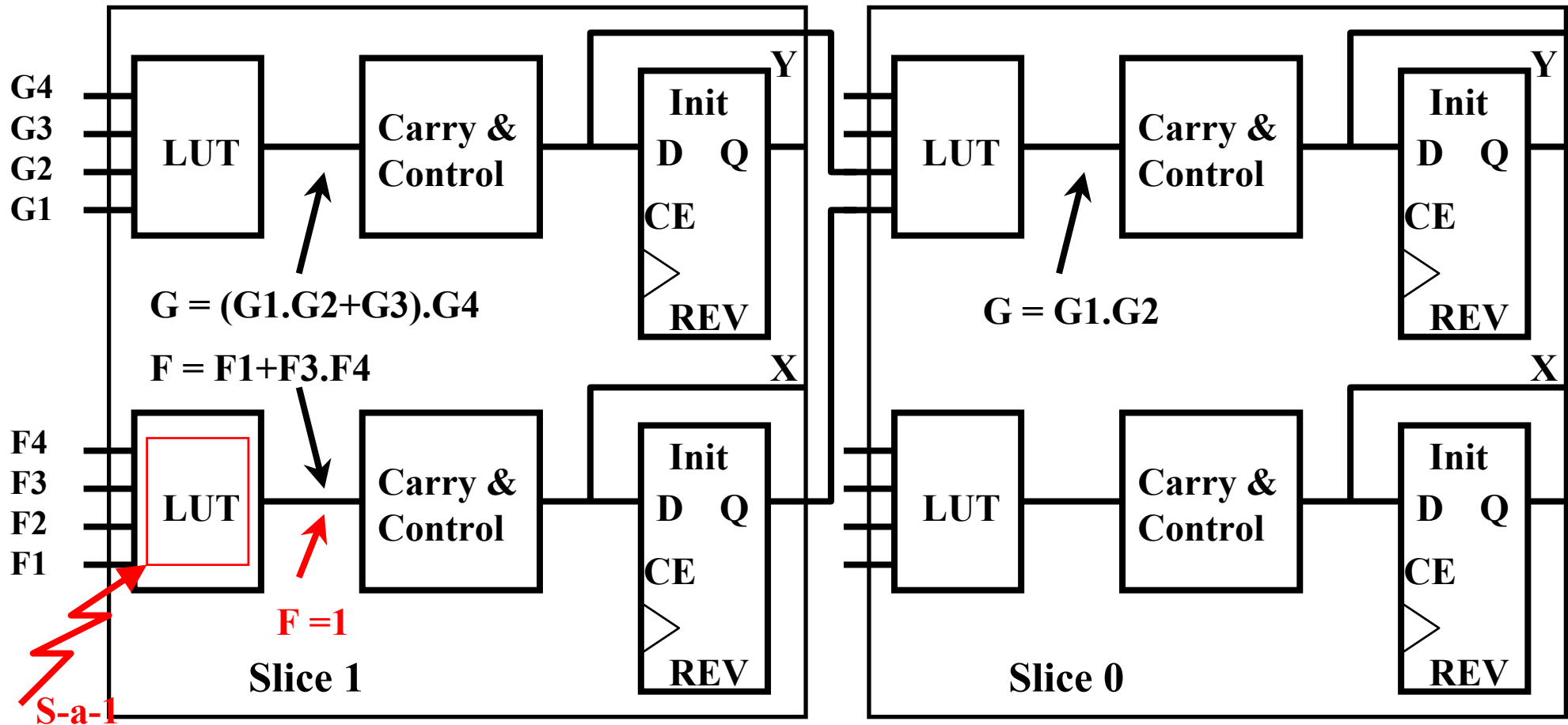
- **Xilinx Virtex XCV50 device**
 - ◆ 16x24 CLBs
 - ◆ Partial configuration capabilities

- **XESS XSV board**
 - ◆ XCV50 device
 - ◆ Parallel port connected to a PC



Application example in combinatorial parts

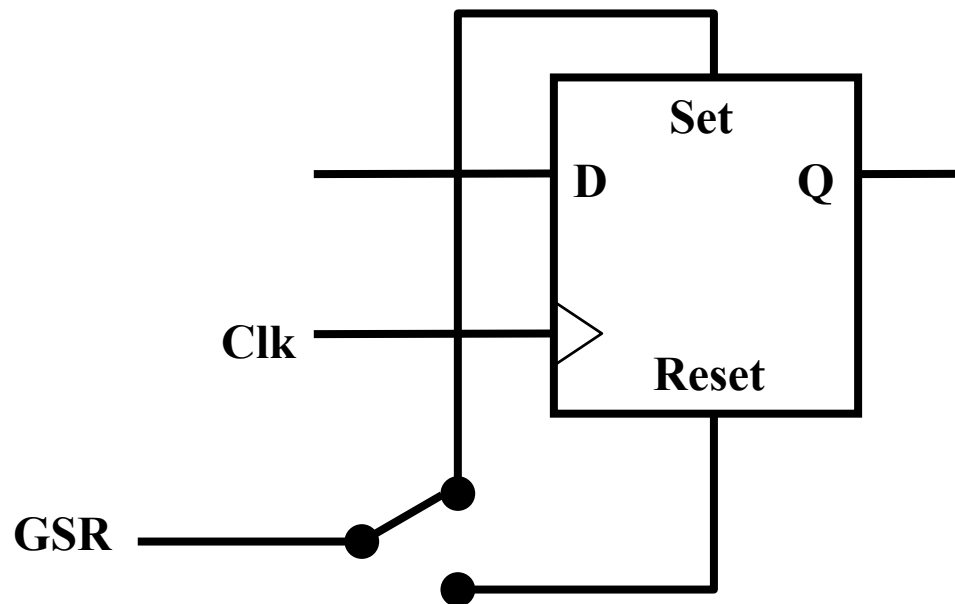
- Example of a stuck-at or SET injection in Virtex CLBs



SEUs: constraints of Virtex architecture

□ Asynchronous injection

- ◆ Asynchronous Set/Reset inputs on the CLB flip-flops
- ◆ Only a global control signal GSR
- ◆ Configuration of the signal functionality for each flip-flop (set/reset switches)



SEUs: injection on Virtex architecture

□ Basic injection steps

◆ **Initialization: full configuration of the device**

◆ **Store the states of set/reset switches**

◆ **For each injection experiment:**

— **start execution of the application**

— **stop the execution at injection cycle**

— **read the states of FFs**

— **change the set/reset switches where needed
(so that only selected FF contents are modified)**

— **pulse the GSR line**

— **re-change the modified switches**

— **continue execution of the application until end or next injection**

**Efficiency
depends on:**

**Read-back
efficiency**

**Reconfiguration
time**

Practical implementation

- **Based on JBits 2.8**

- ◆ **JAVA-based tool set available from Xilinx**
- ◆ **Application programming interface (API) for bitstream modification and device read-back or reconfiguration**

- **Complete approach automated and feasibility demonstrated on the development board (with reduced performances)**

- ◆ **Measurements**

- **Very slow reconfiguration process: 3.5 s per injection**
- **Mainly due to limitations of the board design (e.g. configuration clock at only 4 kHz)**
- **Other limitation: 50 Kbps on the parallel port**

- ◆ **Need of a specific board, designed for**

- **Accelerated reconfiguration and read-back of the device**
- **Accelerated communication with the host computer**



Device-based performance analysis (SETs)

- **XCV50 architecture and characteristics**
 - ◆ Configuration in parallel (8-bits) or serial mode
 - ◆ Configuration frequency up to 60 MHz
 - ◆ Read-back of flip-flops in each of the 24 columns: four 384-bit frames
 - ◆ Configuration of switches in each of the 24 columns: four 384-bit frames

- **Injection (or removal) of one SET or stuck-at**
 - ◆ 8 bits to modify in a LUT
 - ◆ 8 frames to reconfigure (due to the organization of the frames
=> limitation of the Virtex architecture for such an application)

 - ◆ Length depending on the size of the device
(less than 1 ms per injection – 0.816 ms for a XCV2000E)

Device-based performance analysis (SEUs)

□ Injection of one SEU

- ◆ Number of frames to read/write dependent
 - On the position of the functional flip-flops (i.e., **on the placement and routing**), not only on their number in the implemented circuit
 - On the number of switches to commute (i.e., on the initial configuration and on the correct behavior)
- ◆ Serial configuration at 1 MHz, no optimization of FF positions
=> 100 ms on an average for one fault injection
- ◆ Maximal capabilities => less than 1 ms per injection

Conclusion and perspectives

- ❑ **A new approach has been proposed for injection of SETs or SEUs into hardware prototypes**

- ❑ **Feasibility demonstrated**

- ❑ **Main parameters to optimize:**
 - ◆ **Configuration time (parallel configuration, maximal frequency)**
 - ◆ **Read-back time**
 - ◆ **Placement and routing algorithms (minimized number of frames)**
 - ◆ **Communication speed with the host computer**
 - ◆ **... and internal architecture to minimize the reconfiguration data ...**

- ❑ **Further work: development of an efficient board for fault injection using RTR**