



Single Event Upsets in SRAM FPGAs

Michael Caffrey¹

Paul Graham¹

Eric Johnson^{1,2}

Michael Wirthlin²

Nathan Rollins²

Carl Carmichael³

¹Los Alamos National Laboratory

²Brigham Young University

³Xilinx Inc.

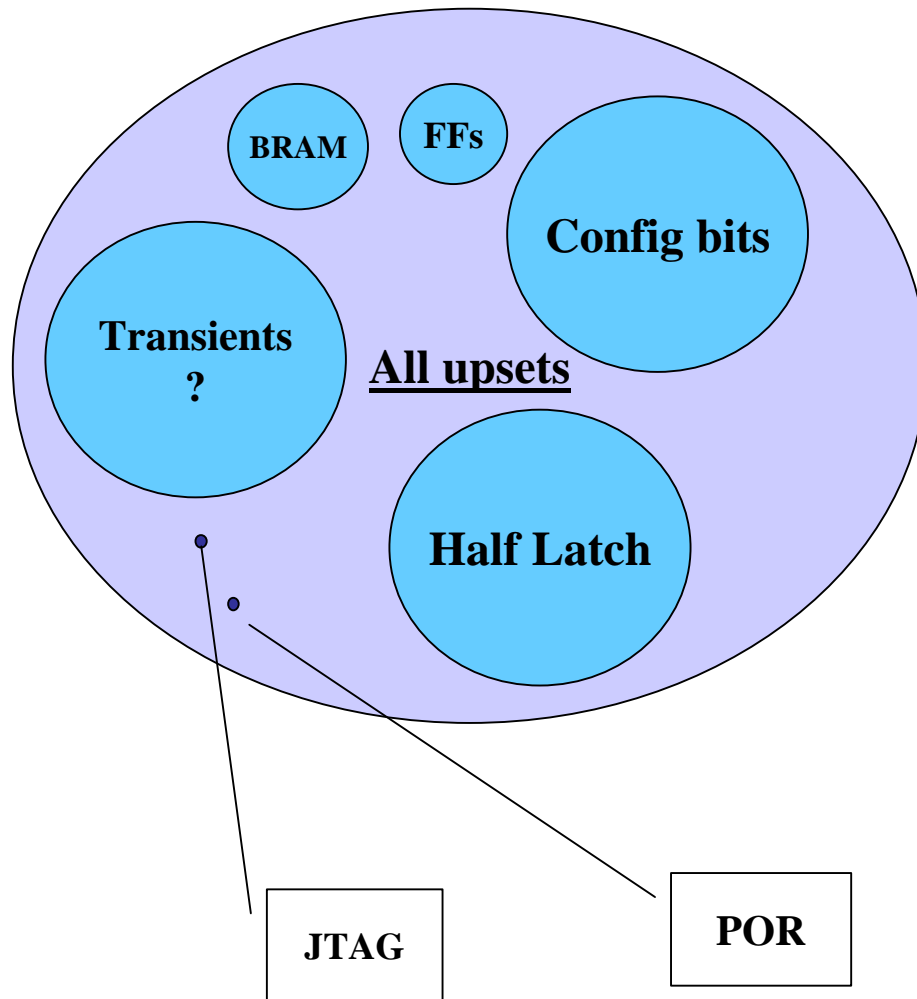
LA-UR-02-5365



Outline

- SEU Categories
 - Flip-Flops
 - Block RAM
 - Functional Interrupts
 - Transients/Unknowns
 - Configuration Bitstream
 - Half-Latches
- System Design Considerations
- Future Work

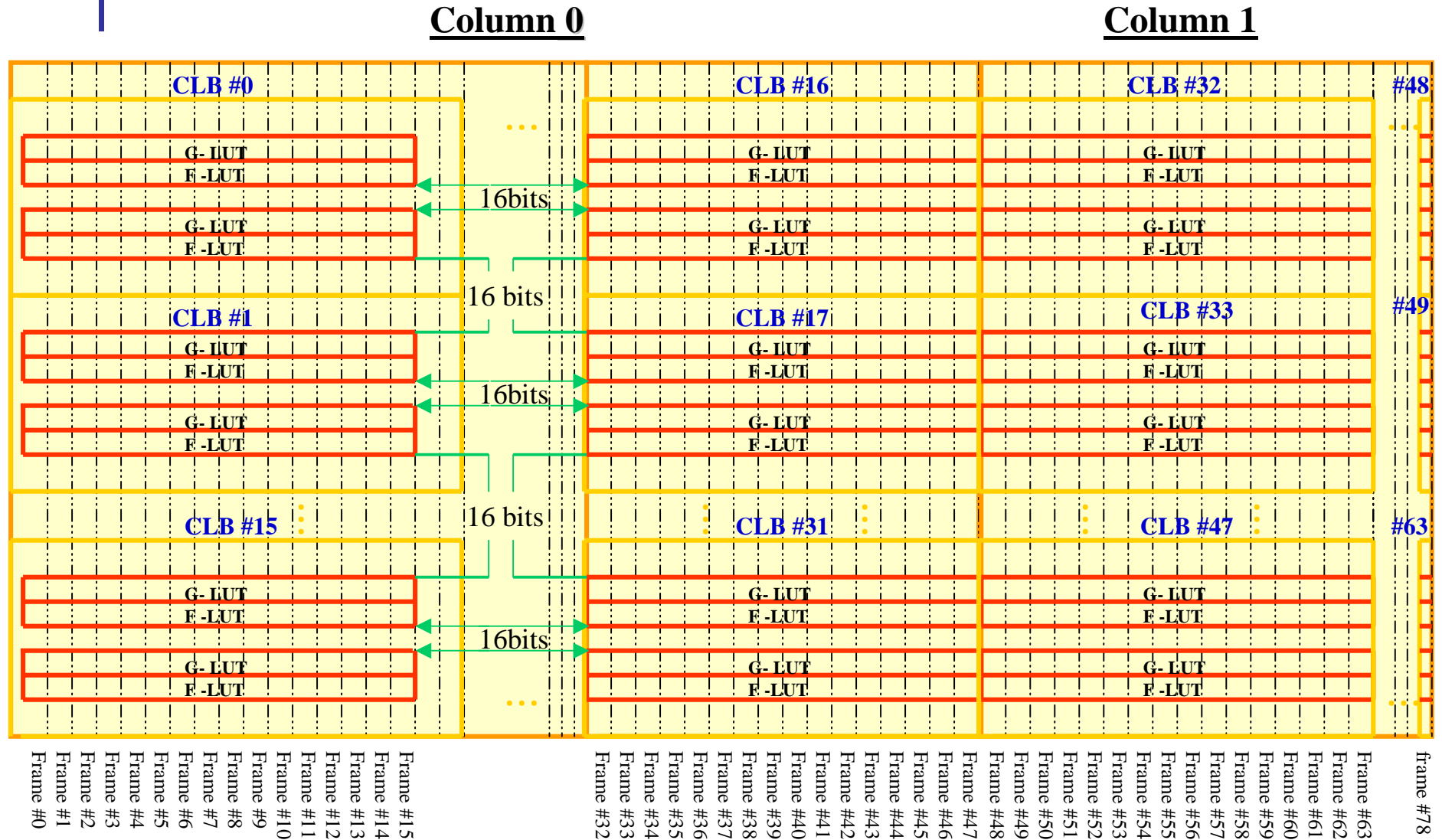
SEU Categories



- Configuration bitstream: detect and repair
- Block SelectRAM: error control coding
- User registers: minimal redundancy
- Half-latch structures: removal software
- JTAG TAP controller: external circuitry
- Power-On Reset in configuration controller: detect, reconfigure

- LUT
- CLB's
- Column's

Xilinx Virtex Bitstream Architecture



Relative Static Xsection of the categories (XCV1K)

- User flip-flops
- User distributed SelectRAM (LUT) memory
- User Block SelectRam memory
- Configuration memory
- Hidden xSection (1 E-5 cm^2) (**Reset, JTAG, Configuration Pins, etc.**)

User Flip-Flops	26,112	0.4%
User LUT RAM	393,216	6.4%
User BlockRAM Bits	131,072	2.1%
Configuration Bits	5,603,456	91.0%
Hidden registers	< (?)	< .0021%

Most of the state in configuration bits!

Flip Flop {CLB & IOB} Upsets

- Flip Flop upsets will occur approximately 1/200 the rate of bitstream upsets
- Must be handled in logic (redundancy, CRCs etc)
- IOB FFs
 - No single bit error can turn an input to an output (See BYUs paper in this years MAPLD)
 - A single error can turn a tristate into an OUTPUT! Consider the consequences to your system design.
 - SelectMap configuration pins are sensitive, but with and extremely small Xsection (this can prevent readback or partial reconfiguration)

BlockRAM

- BlockRAM is NOT accessible to readback under normal operating conditions
- Traditional techniques such as parity or Error Correcting Coding (ECC) can be used to detect/correct SEUs

Functional Interrupts -JTAG

- The JTAG TAP controller must be sensitive, but has not been observed, implying very small Xsection (Katz et al)
 - The Test Clock on the JTAG interface should have a free-running clock which does NOT connect to any other FPGA IO pin (possible contention would prevent the clock from cycling the TAP state machine, preventing recovery)
 - TMS should be pulled up, allowing the JTAG TAP controller to recover in 5 cycles

Functional Interrupts –POR & SelectMap

- The configuration FSM can reset and clear the device (Xsection < 1E-5 cm²)
- Configurable Selectmap pins (CSn, Wen, D0-7, BUSYn, etc) configuration can be upset which will interfere with partial configuration and/or readback (~15 bits in the bitstream, tiny Xsection)
 - We look for extraordinary number of errors in the readback stream that suggests an interface error, if found reconfigure the device

Transients?

- In accelerator tests previously reported (Fuller, mapld 2000) 45% of errors detected were not due to bitstream errors.
 - Those tests did not use test designs with half-latches removed
 - The designs were 'self' testing
 - No clock rate sensitivity was found
- An upcoming accelerator test will take another look at the existence of transients or some other, unaccounted for, Xsection
- If another Xsection exists, redundancy is likely the best approach for mitigation

SEU Management: Configuration Bitstream

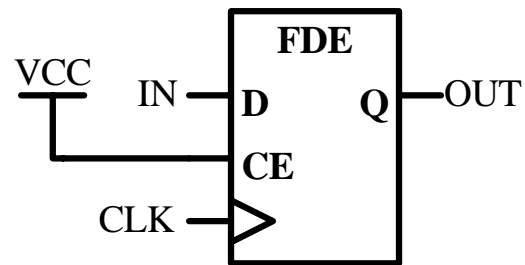
- Approach 1:
 - Triple Module Redundancy
 - Configuration bitstream scrubbing
- Approach 2:
 - Bitstream readback
 - CRC check for detection
 - Partial configuration for repair
 - Minimal redundancy
- Differences:
 - Approach 1:
 - Tolerates SEUs, expensive in area, power and I/O (3x)
 - Approach 2:
 - Allows SEUs to cause errors in output
 - Can track SEU rates
 - Less expensive in device resource

The “Half-Latch” Problem

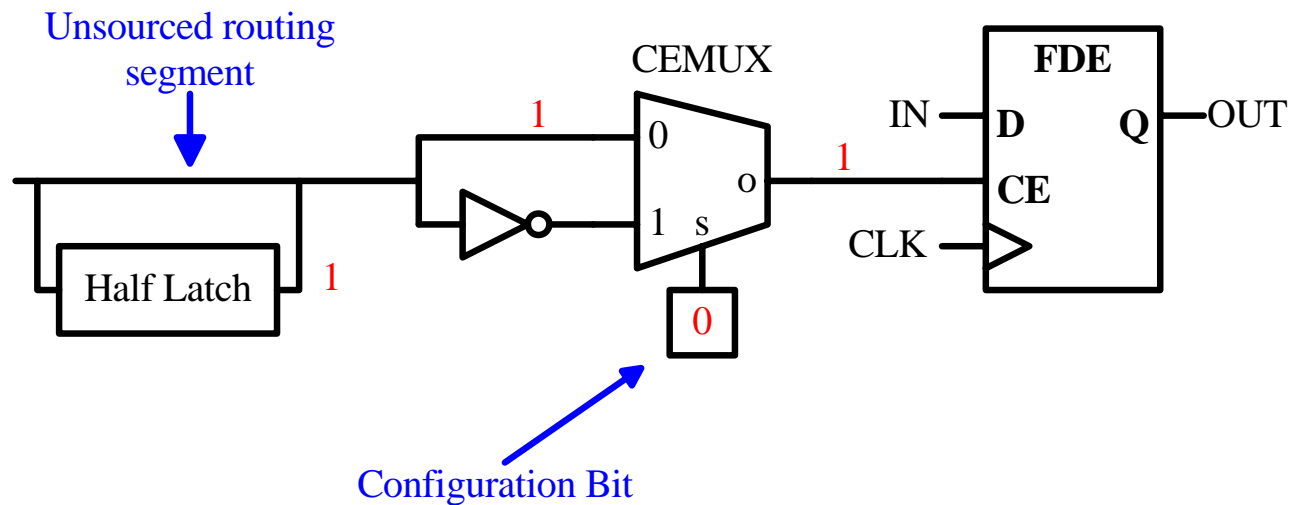
- Half-latches
 - “Hidden” state in the Virtex architecture
 - Operate like level-sensitive state elements with weakly driven outputs
 - Used by Xilinx tools to provide “free” logic constants
 - Sensitive to SEUs
- When upset, bitstream is still O.K. but the design may not work properly without full reconfiguration.

Half-Latch Example (Part 1)

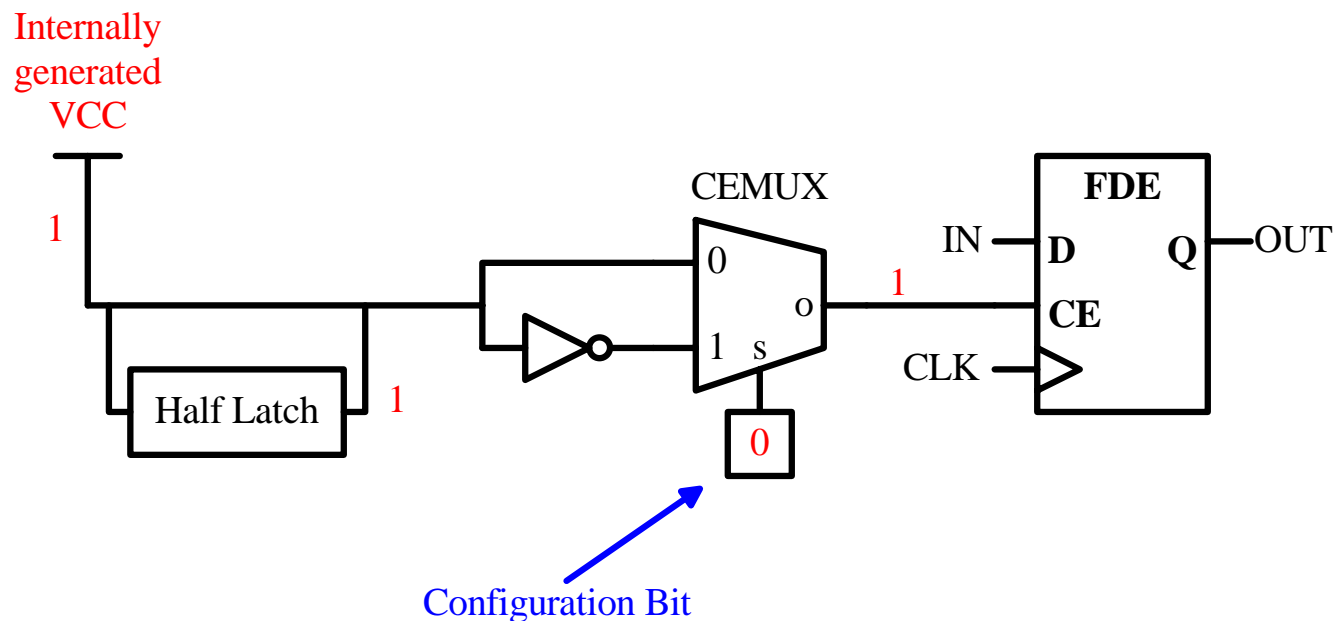
Intended
Circuit



Implementation
(half latch holds
the "1" value)

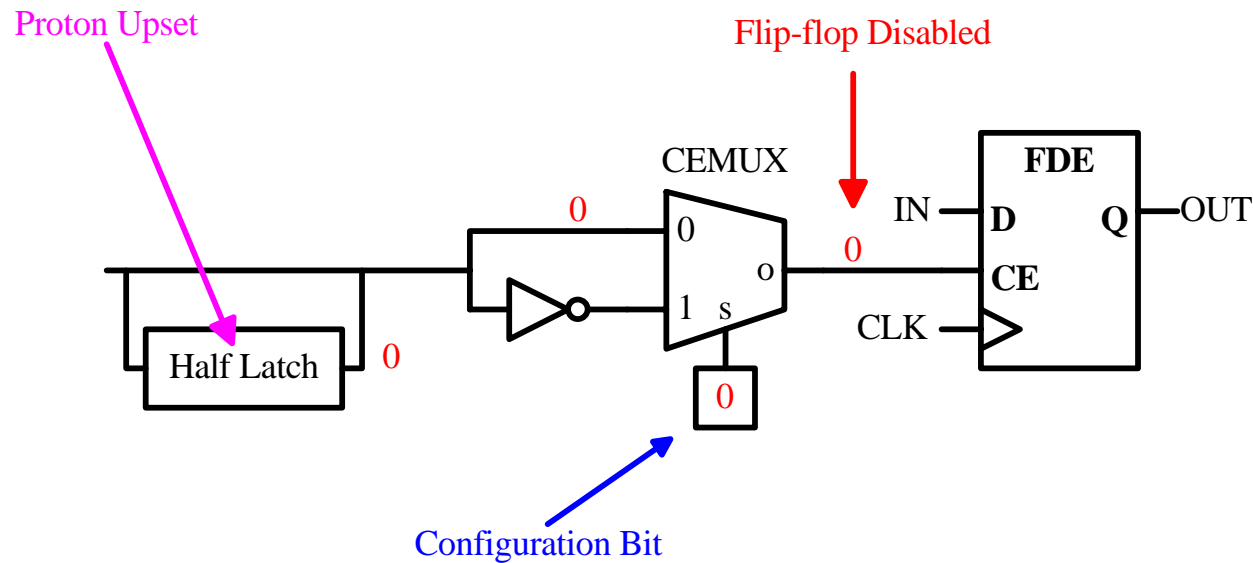


Half-Latch Example (Part 2)



During a full reconfiguration of the FPGA, the half latches are initialized to logic "1".

Half-Latch Example (Part 3)



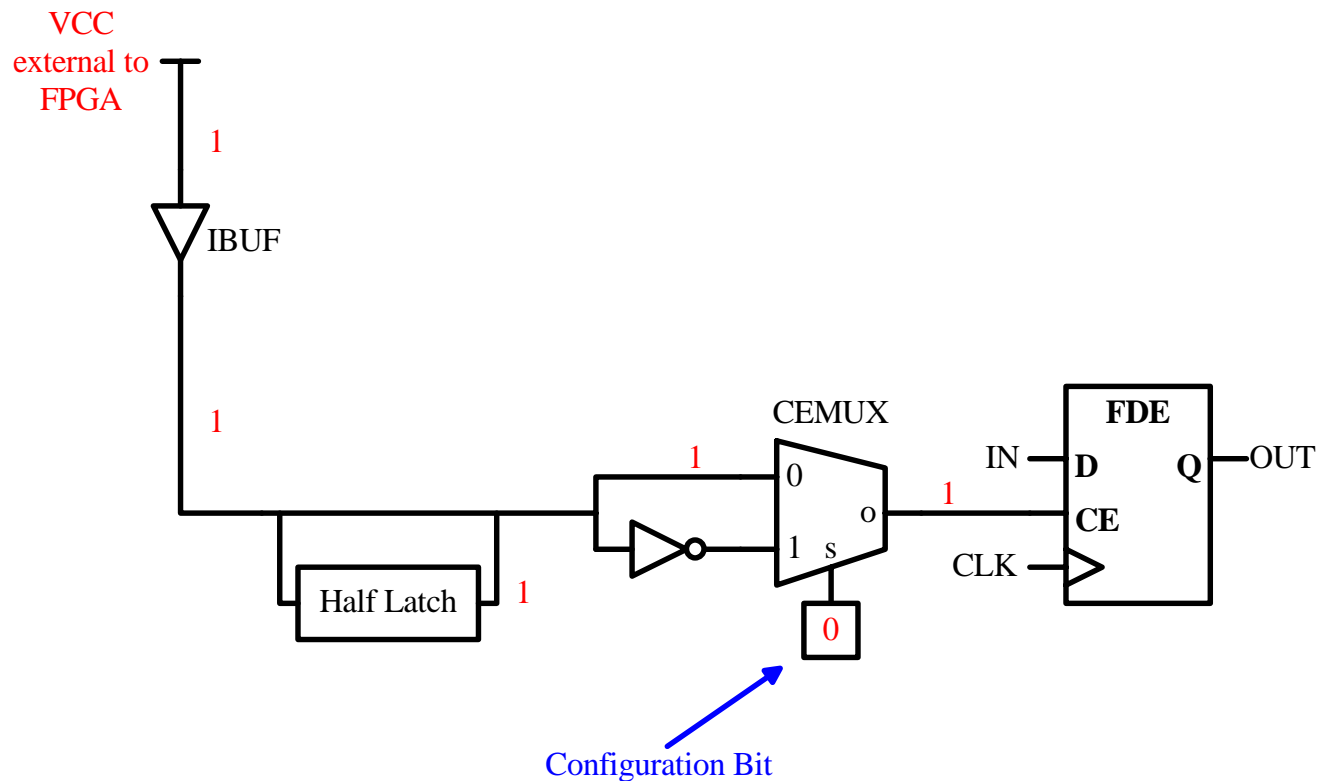
If the half latch gets upset in this case, the flip-flop becomes disabled until a full-reconfiguration of the FPGA or another upset occurs. This problem is not visible through configuration readback and cannot be practically repaired through partial configuration.

Half-Latch Mitigation Concept

- Remove the reliance on half latches for generating constants
- Use internally or externally generated constant logic values whose configuration can be observed through configuration readback and repaired through partial configuration
 - Internally generated techniques are likely to allow for more redundancy at a lower cost (unused LUTs and flip-flops are used instead of scarce IOB resources).
- Possible (but painful) to handle through design entry

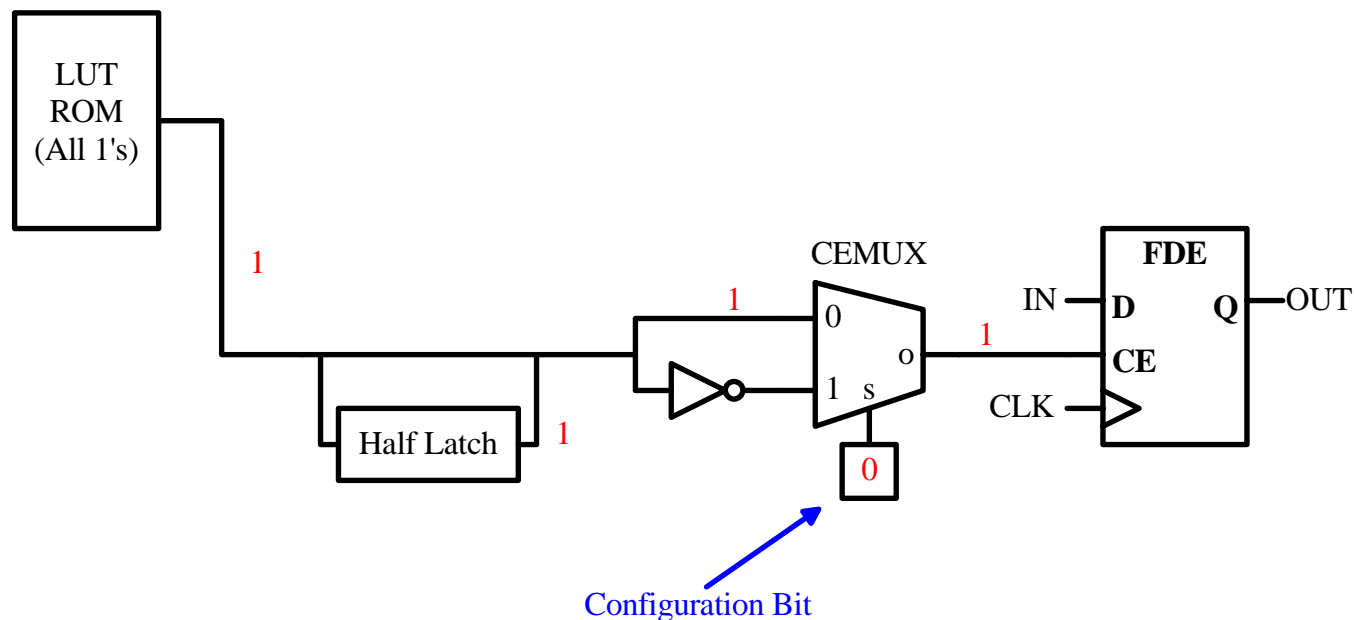
Result: half latches still exist, but their outputs are overcome by stronger drivers

Half-Latch Mitigation: External Logic Value



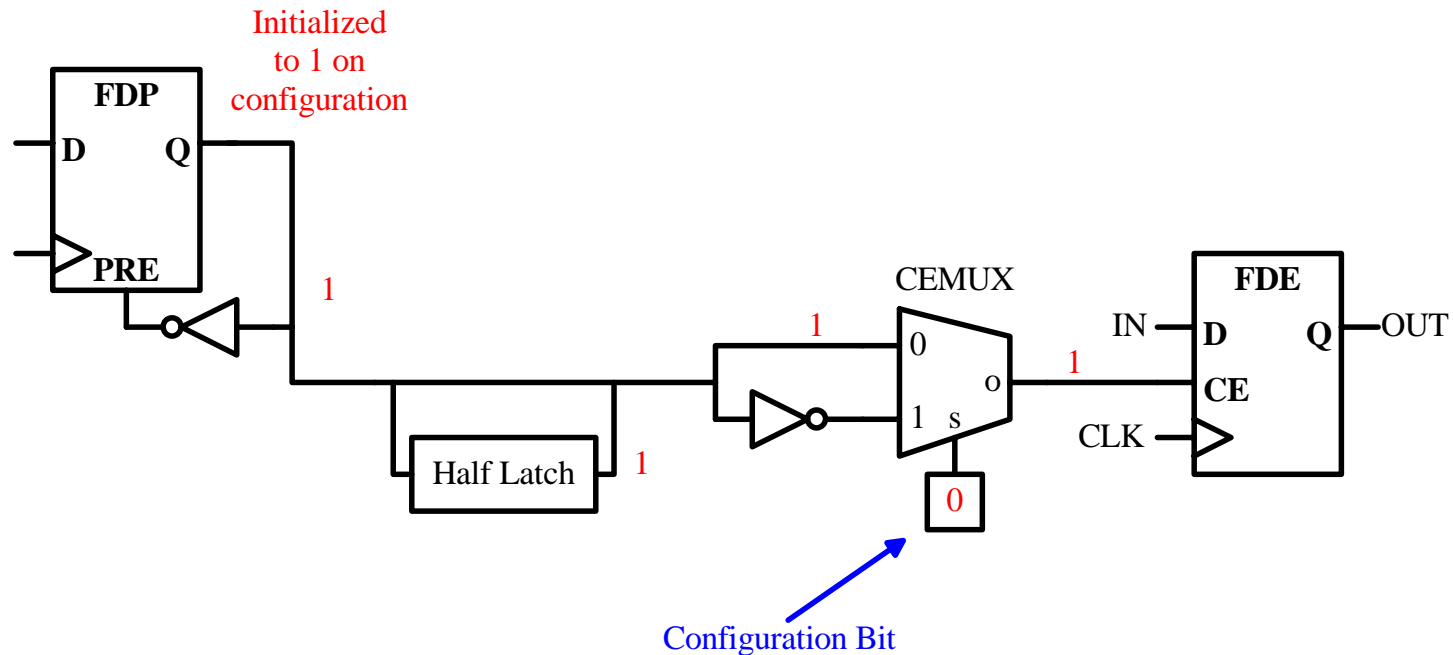
A "1" is provided by a source external to the chip through an IOB.

Half-Latch Mitigation: LUT ROM



A LUT ROM can be used to generate a logic "1". When filled with 1's, the ROM itself is resistant to half-latches on its inputs (all addresses produce a "1").

Half-Latch Mitigation: Flip-Flop with Feedback



A flip-flop with appropriate feedback can be made to be resistant to SEUs in its state as shown above (PRE is an asynchronous set to "1").

Virtex Resources Sensitive to Half-Latches

Structure	Inputs
GCLK	CEMUX
DLL	RSTMUX
Block SelectRAM	WEAMUX, ENAMUX, RSTAMUX, WEBMUX, ENBMUX, RSTBMUX
IOB	SRMUX, TRIMUX, TCEMUX, OMUX, OCEMUX, ICEMUX
SLICE	BXMUX, BYMUX, CEMUX, SRMUX, F1-F4, G1-G4
STARTUP_VIRTEX	GWEMUX, GTSMUX, GSRMUX
CAPTURE_VIRTEX	CAPMUX

LANL's RadDRC: Critical Half-Latch Removal Tool

- Analyzes low-level, placed and routed design in XDL format
- Prints report about critical half-latches and LUT RAMs
- Can produce FPGA Editor script for removing critical half-latches
 - Logic "1" through IOB
 - Route to all half-latch driven inputs
 - Modify input muxes depending on if a "1" or "0" is needed

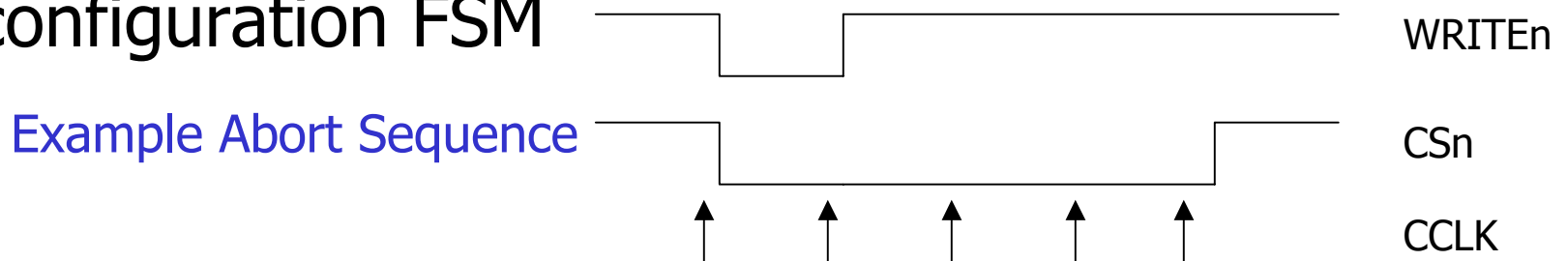
The designer can ignore the half latch issue during design entry, allowing for simpler designs.

System Design Considerations

- SelectRAM (LUTRAMs) are not compatible with readback, do not enable readback if using these primitives (for example, the SRL16)
- When performing readback, do NOT abort before reading all the data requested in the command sequence
 - The internal FSM can be confused (however, a full frame partial configure will reset it)
- When partially configuring, always use '0's for PAD frame and data values, pad data can show up in readback due to the pipeline registers and may result in false upset detection

Further Design Considerations

- Always issue the 'Abort sequence' before commanding the FPGA
- Follow with dummy word and sync word before command sequence to synchronize the configuration FSM



EXAMPLE READBACK COMMAND SEQUENCE

```
x"FF",x"FF",x"FF",x"FF", -- dummy word
x"AA",x"99",x"55",x"66", -- sync word
x"30",x"00",x"20",x"01", -- write to FAR
x"00",x"00",x"00",x"00", -- data: start frame address
x"30",x"00",x"80",x"01", -- write to CMD
x"00",x"00",x"00",x"04", -- data: RCFG
x"28",x"00",x"60",x"00", -- read from FDRO
x"48",x"02",x"D8",x"0D"  -- data: # of data words
```

Bitstream SEU Simulation

- Simulate configuration memory SEUs by *inserting* errors into the configuration bitstream
 - Toggle configuration bit within the bitstream
 - Probe the circuit to identify circuit failure
 - Repair configuration bitstream
 - Repeat for *all* or *random* configuration bits
- Advantages of the SEU Simulator
 - Inexpensive alternative to accelerator tests
 - Speed (fast reconfiguration/readback)
 - Control over FPGA clock
 - Access to FPGA state
 - Ability to apply test-vectors

Future Work

- In-system silicon test in space (test for hard faults)
- Tools for automating the introduction of redundancy into portions of designs for SEU mitigation
 - Validated with simulator
- Virtex II Pro analysis
- Half-latch (HL) removal enhancement
 - Use distributed nets for VCC & GND
- Proton tests
 - Validate SEU simulator
 - Validate HL removal tools
 - Take another look at the total Xsection (Transients?)