# Embedding Asynchronous FIFO Memory Blocks in Xilinx Virtex Series FPGAs Targeted for Critical Space System Applications

**Melanie Berg**

MEI Technologies- NASA/GSFC Radiation Effects and Analysis Group

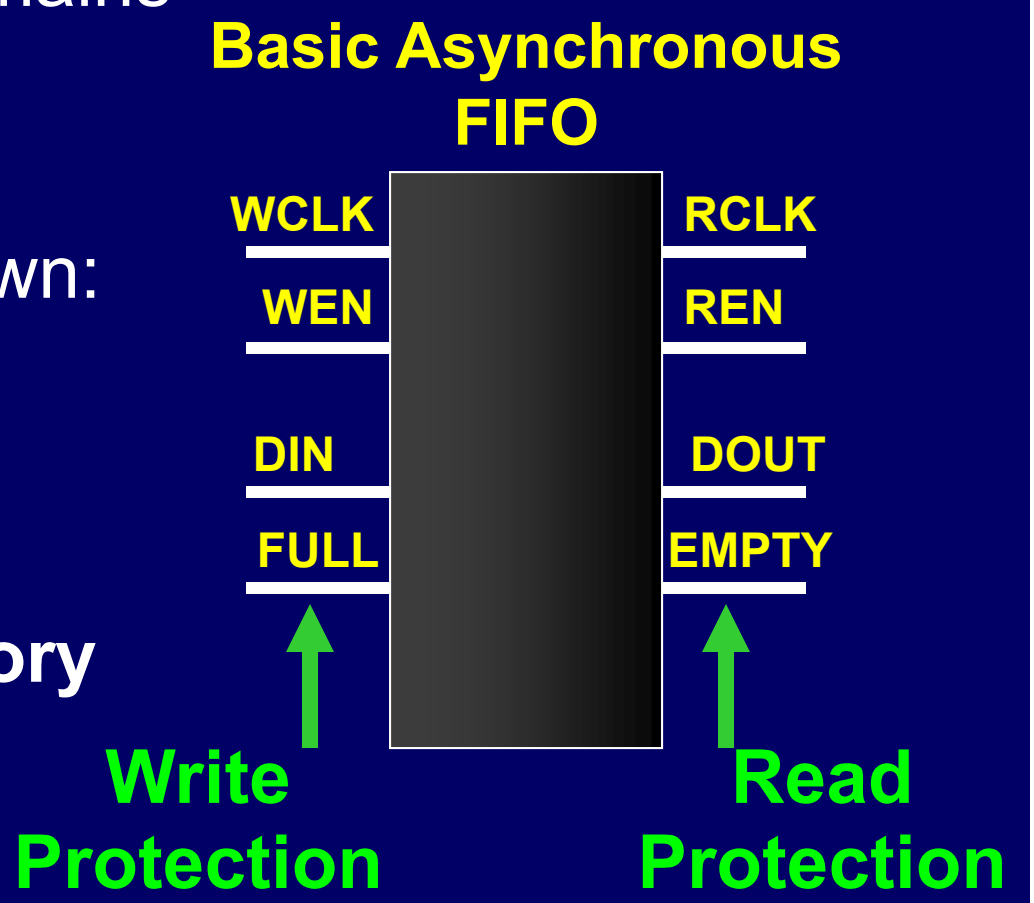
Presented by Melanie Berg at the Military and Aerospace Programmable Logic Devices (MAPLD) Conference, Greenbelt MD, August 31 - Sept. 3, 2009, and posted on http://nepp.nasa.gov.

# Agenda:

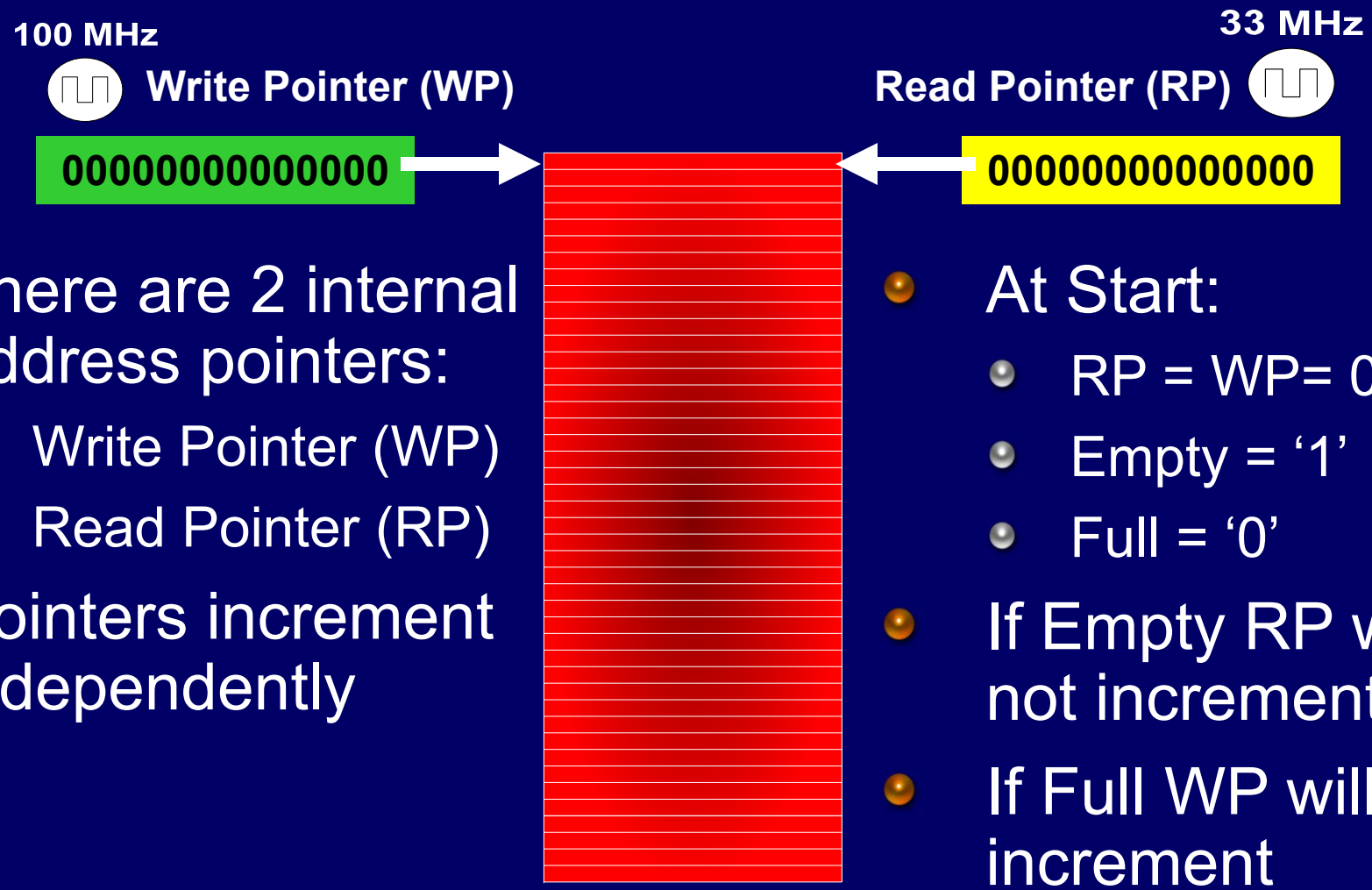
- Asynchronous FIFO (AFIFO) Basics: Operation and Architecture
- Single Event Upsets (SEUs) and Xilinx FPGA Devices
- Mitigating Radiation Effects is AFIFOs with Triple Modular Redundancy (TMR):
  - Intellectual Property (IP) Cores
  - Customized AFIFO implementation
- Conclusion

# AFIFO Basics

- Used to pass data between two separate clock domains
  - Write Domain
  - Read Domain
- Each domain has its own:
  - **Clock**
  - **Enable**
  - **Address Pointer**
- **Domains share memory space**



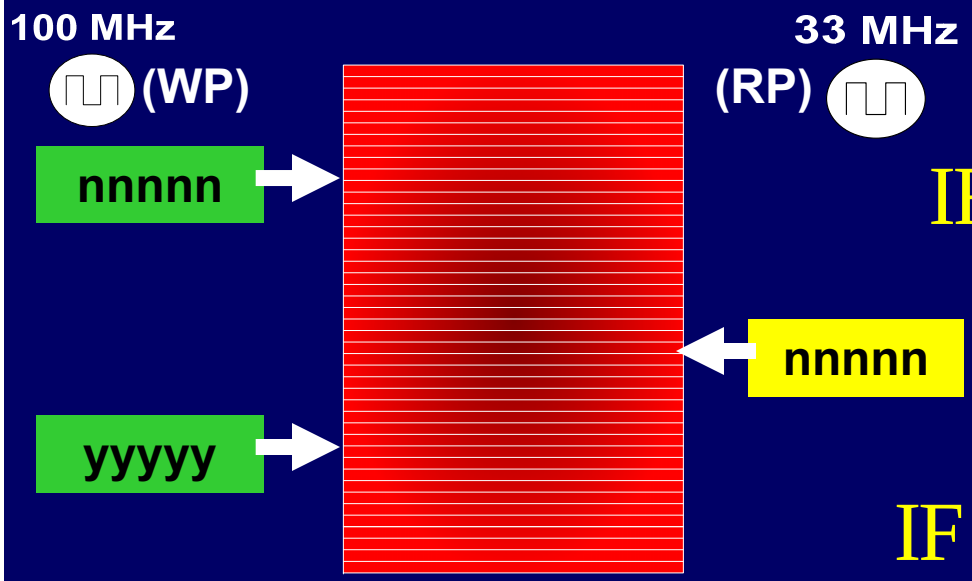
# AFIFO Internal Operation: Address Pointers




- There are 2 internal address pointers:
  - Write Pointer (WP)
  - Read Pointer (RP)
- Pointers increment independently

- At Start:
  - RP = WP= 0
  - Empty = ‘1’
  - Full = ‘0’
- If Empty RP will not increment
- If Full WP will not increment

# AFIFO Operation: FULL Detection

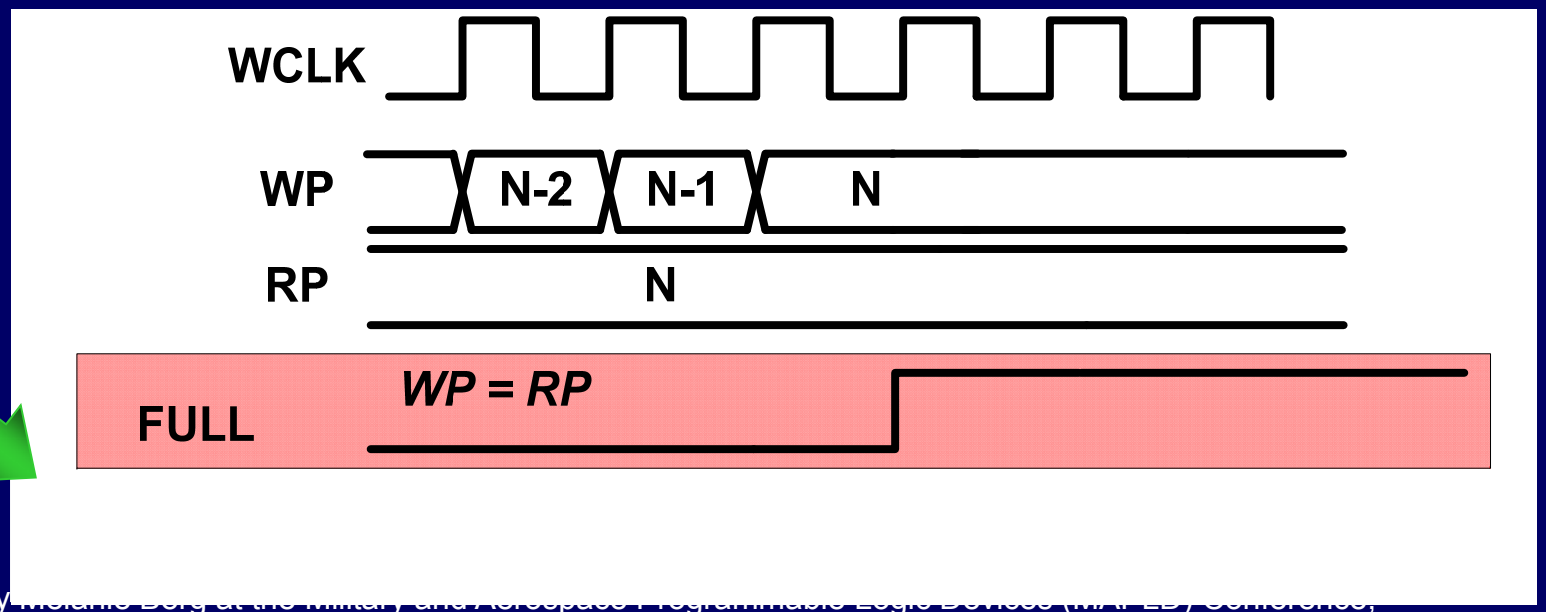



***Traditional:***

$$\text{IF } (WP = RP) \rightarrow FULL = 1$$

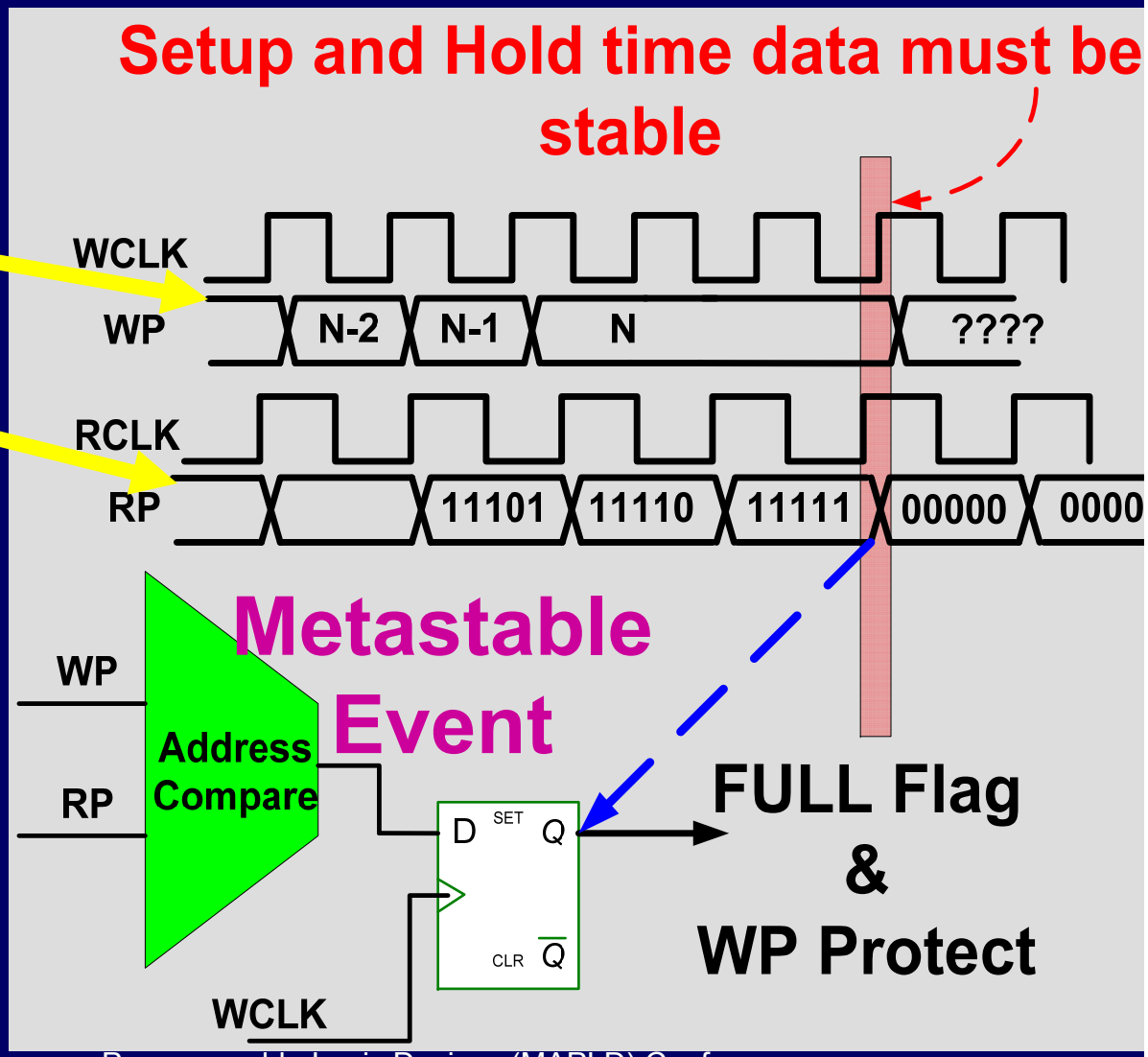
***Look Ahead:***

$$\text{IF } (WP + 1 = RP) \rightarrow FULL = 1$$



# WCLK Example: Comparing WP and RP Across Asynchronous Clock Domains

- WP changes @ $WCLK_{edge}+(\tau_{clk\rightarrow q})$
- RP changes @ $RCLK_{edge}+(\tau_{clk\rightarrow q})$
- Compare: RP Crosses into WCLK
- One DFF for every pointer bit



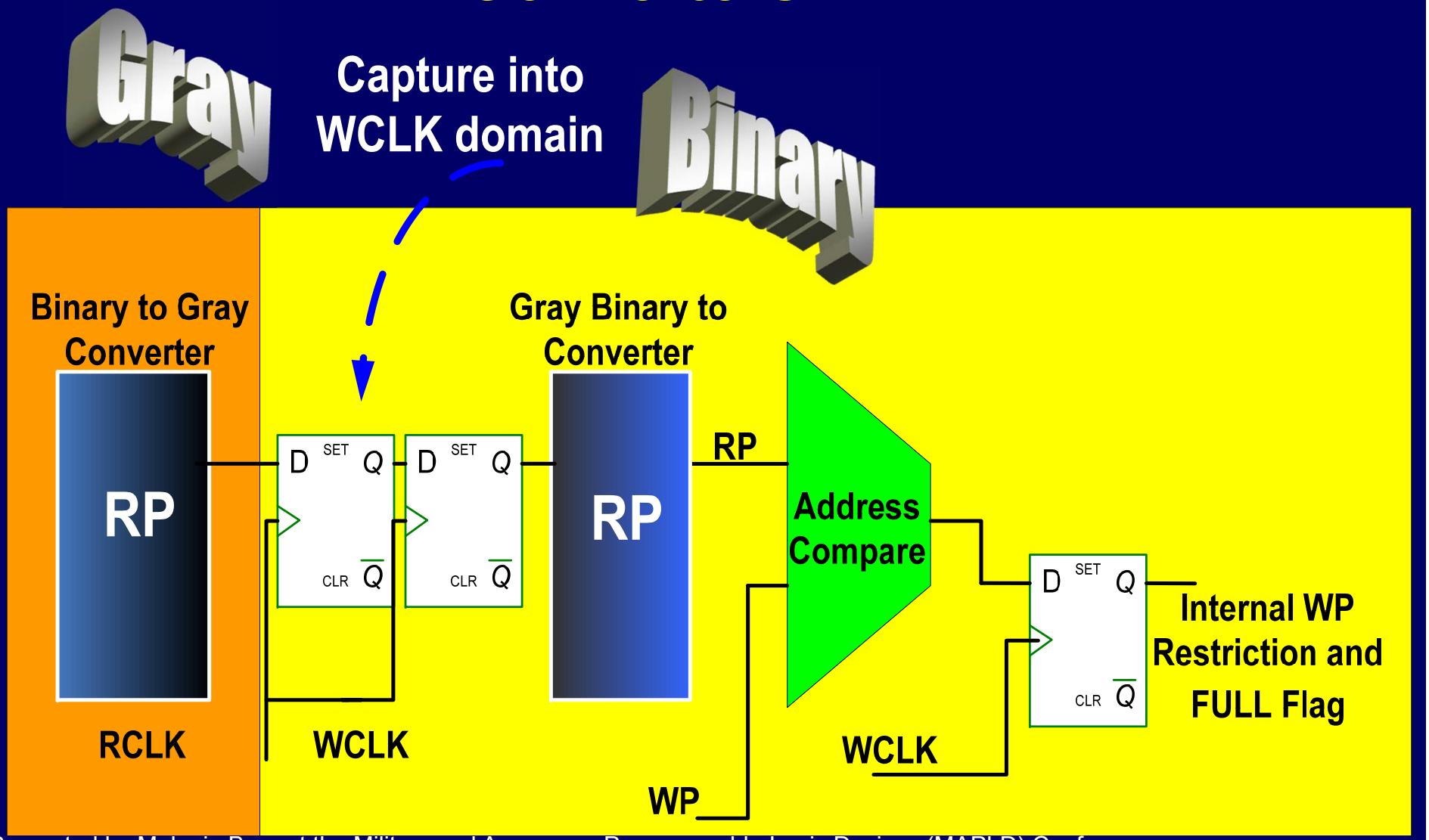
# Reducing the Number of Metastable bits in Compare Path

- **Problem:** With binary encoding too many address bits can change simultaneously during an increment: **11111→ 00000**

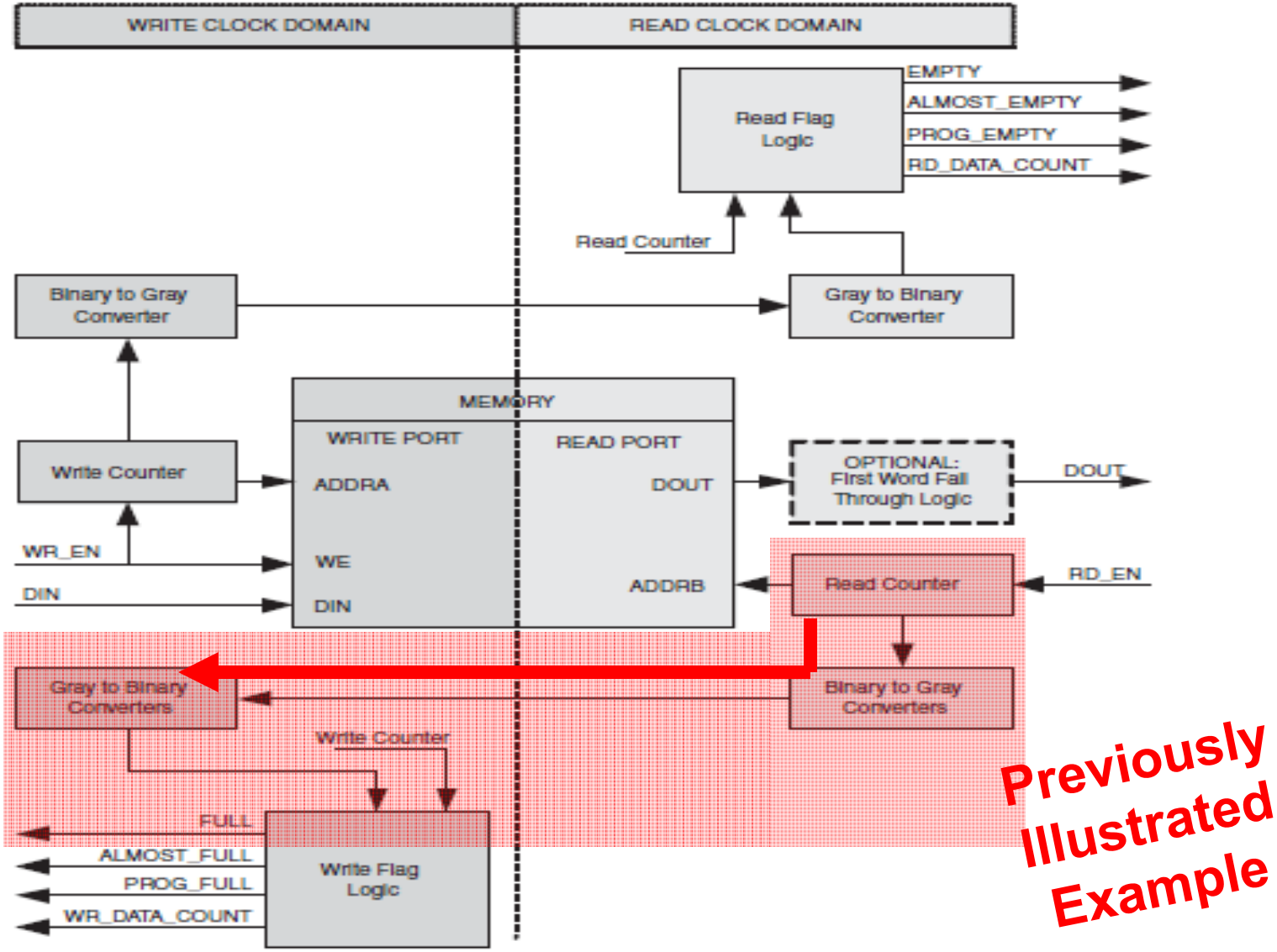
- **Resolution:** Use gray encoding
  - Only used to cross domains
  - Only one bit changes per clock cycle
  - Comparison must still be performed in binary

| Binary | Gray |
|---|---|
| 000 | 000 |
| 001 | 001 |
| 010 | 011 |
| 011 | 010 |
| 100 | 110 |
| 101 | 111 |
| 110 | 101 |
| 111 | 100 |

# Scheme: Reducing the Number of Metastable bits in Compare Path using Converters

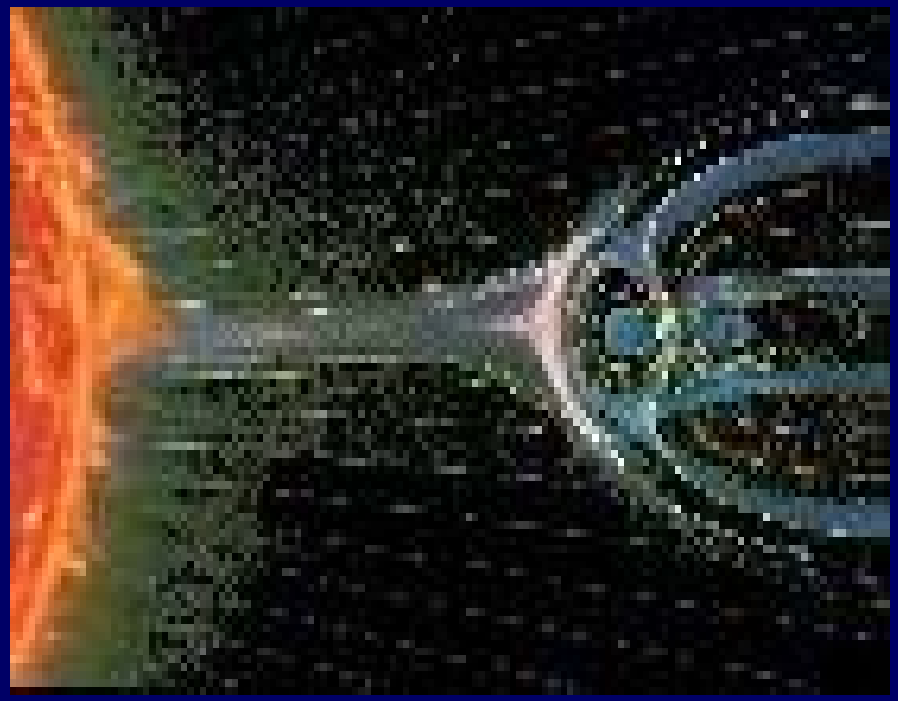


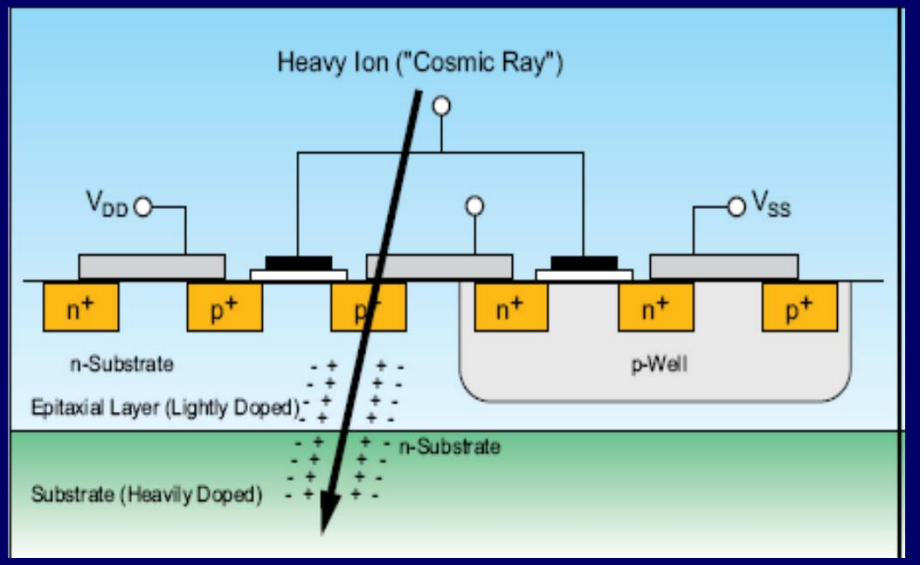
# Xilinx AFIFO IP Core Block Diagram




**Previously Illustrated Example**

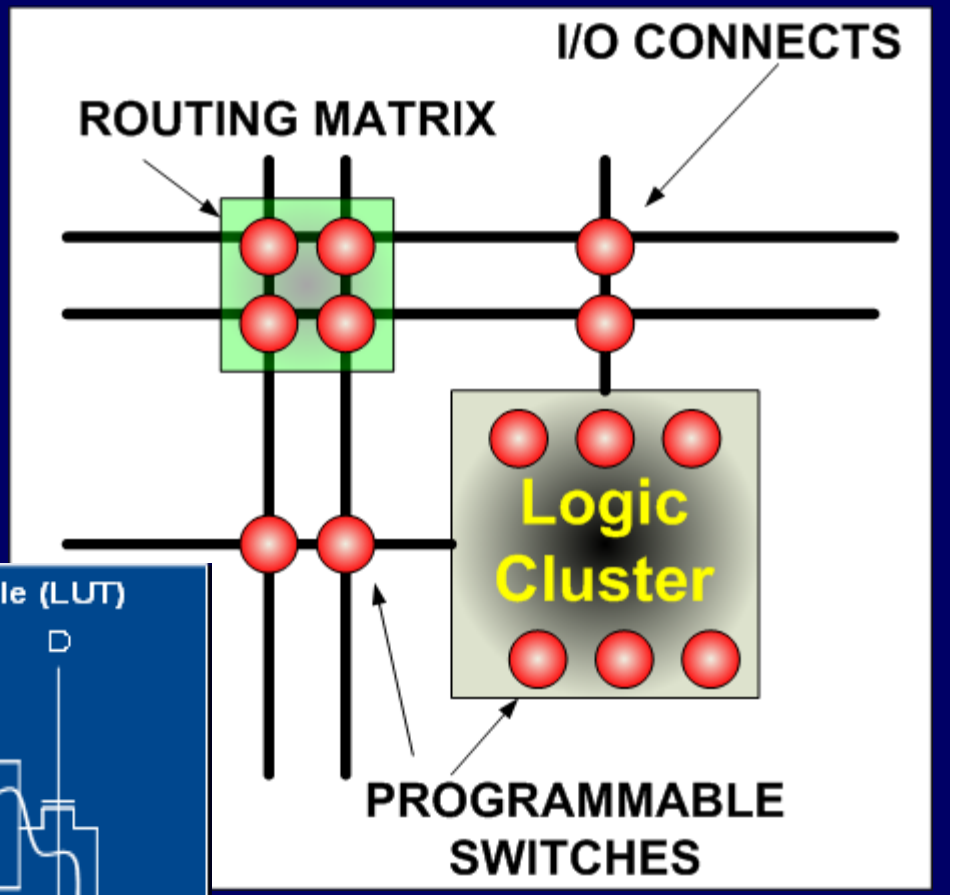
# Single Event Upsets (SEUs) and Xilinx FPGA Devices

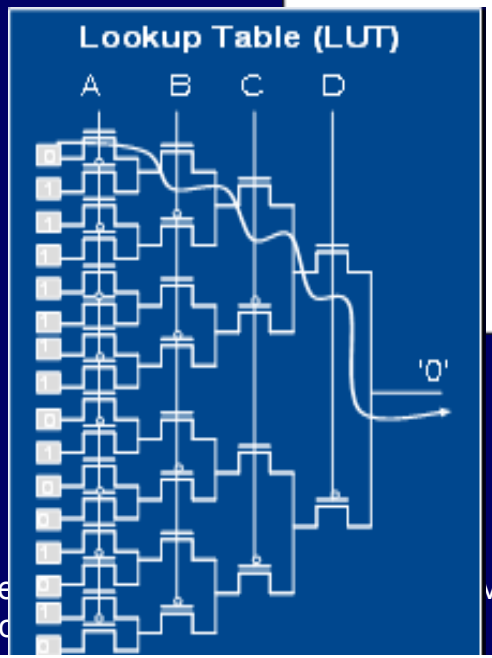




# Xilinx Architecture Basics

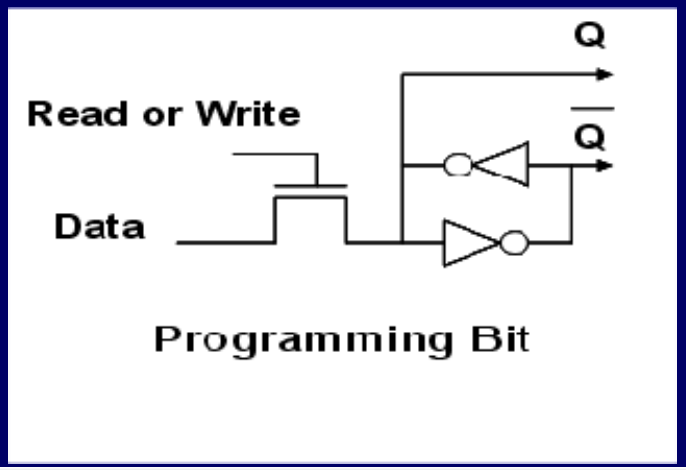
- Configuration
  - programmable switches
  - static during operation
- Logic:
  - Switches during operation
  - Performs expected functionality

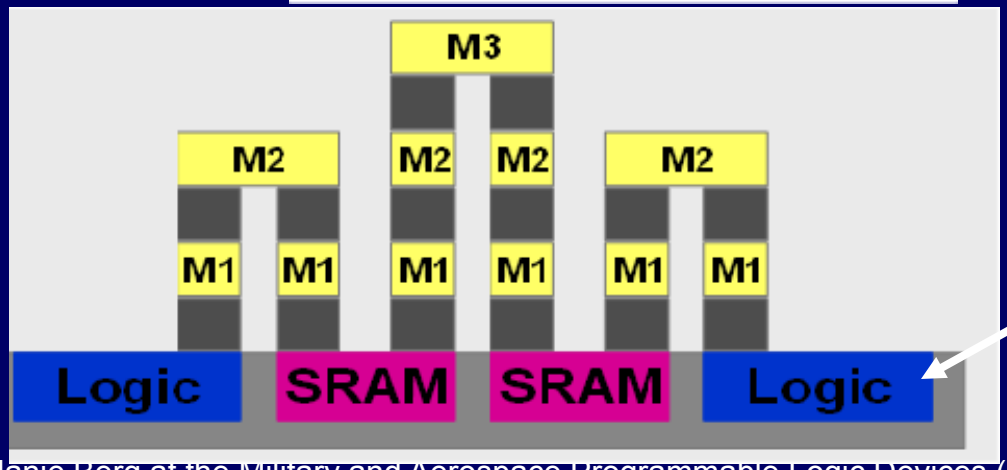


# Programmable Switch Implementation and (SEU) Susceptibility

## SRAM (Reprogrammable Configuration)




Sensitive Volume

# Xilinx XQR4VSX55: Radiation Test Data

*Xilinx Consortium:* ***VIRTEX-4VQ STATIC SEU CHARACTERIZATION SUMMARY: April/2008***

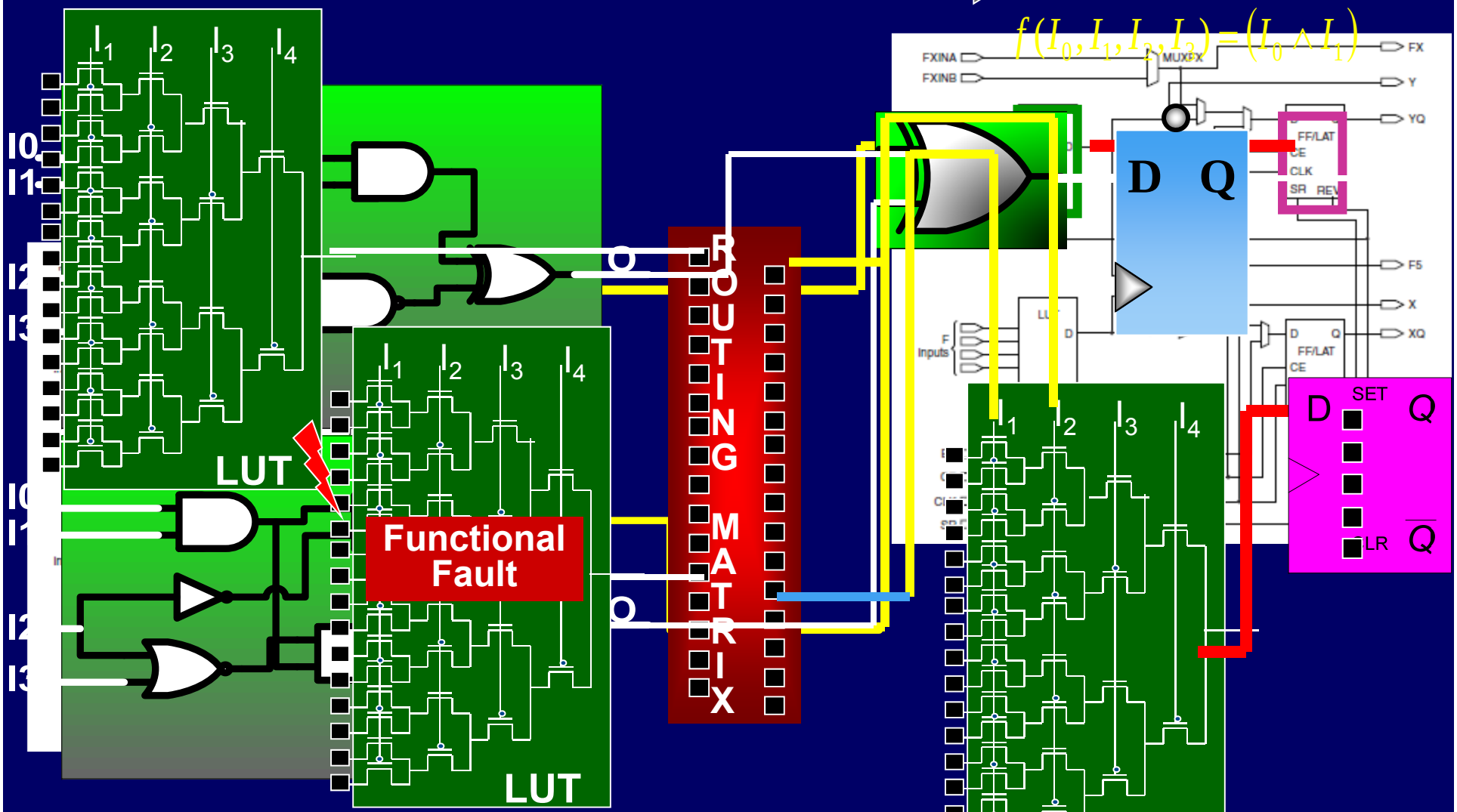
| | Probability | Error Rate | LEO $\frac{Upsets}{device-day}$ | GEO $\frac{Upsets}{device-day}$ |
|---|---|---|---|---|
| **Configuration Memory: XQR4VSX55** | $P_{configuration}$ | $\frac{dE_{configuration}}{dt}$ | **7.43** | **4.2** |
| **Combined SEFIs per device** | $P_{SEFI}$ | $\frac{dE_{SEFI}}{dt}$ | **$7.5 \times 10^{-5}$** | **$2.7 \times 10^{-5}$** |

- For non-mitigated designs the most significant upset factor is: $P_{Configuration}$

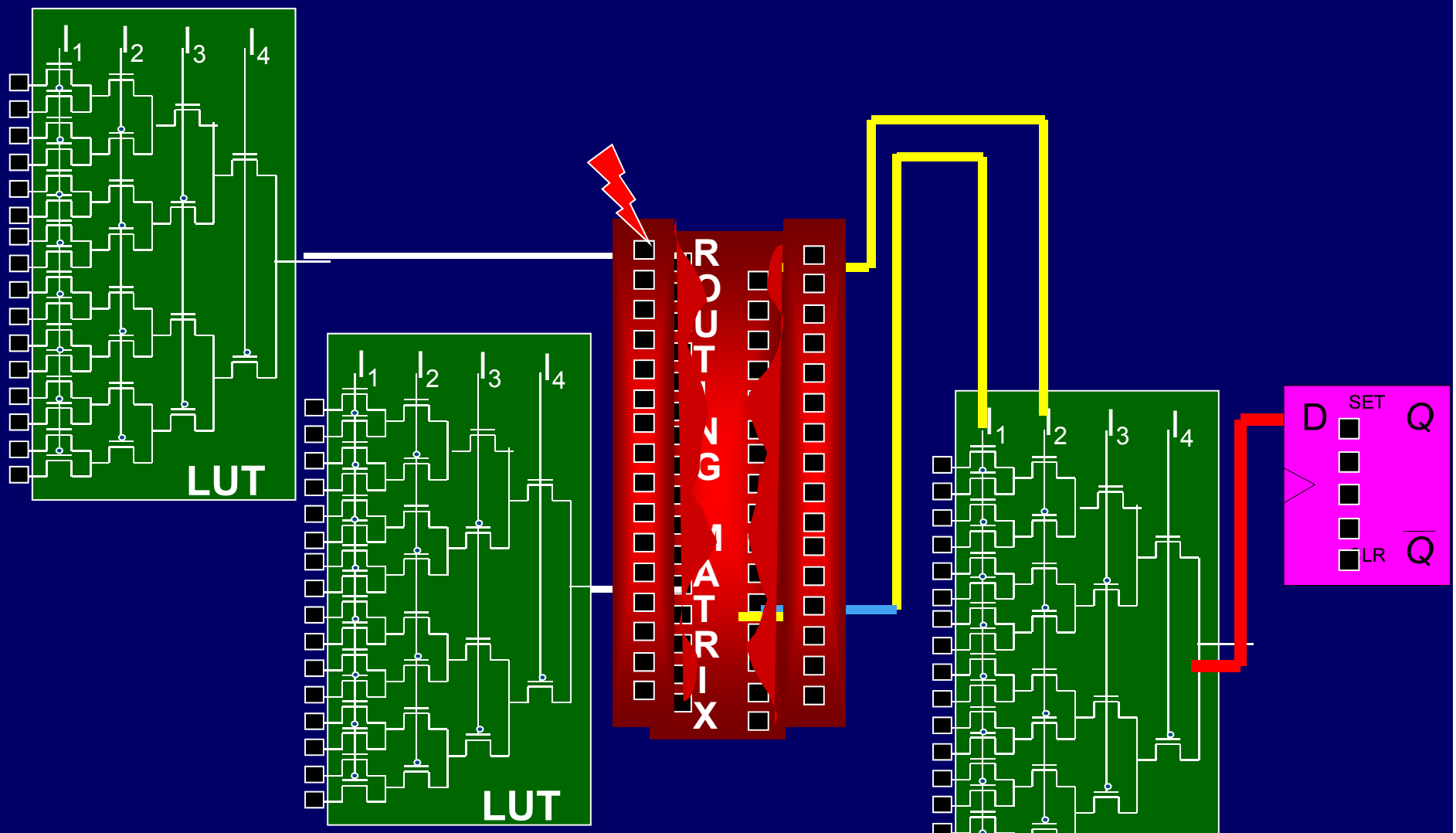
M Berg, Trading ASIC and FPGA Considerations for System Insertion; IEEE Nuclear Science Radiation Effects Conference 2009

# Configuration Upsets: LUTs and DFFs

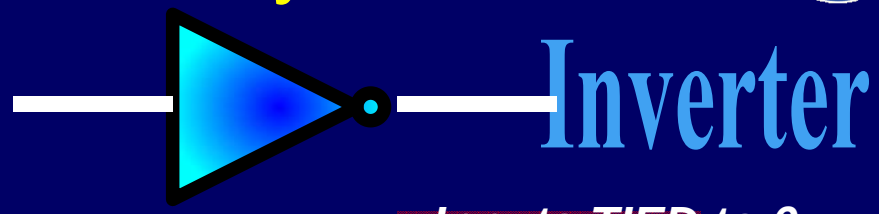
$$f(I_0, I_1, I_2, I_3) = (I_0 \wedge I_1) \vee I_3(\wedge(I_2 \wedge I_1))$$



# Routing Configuration Upsets



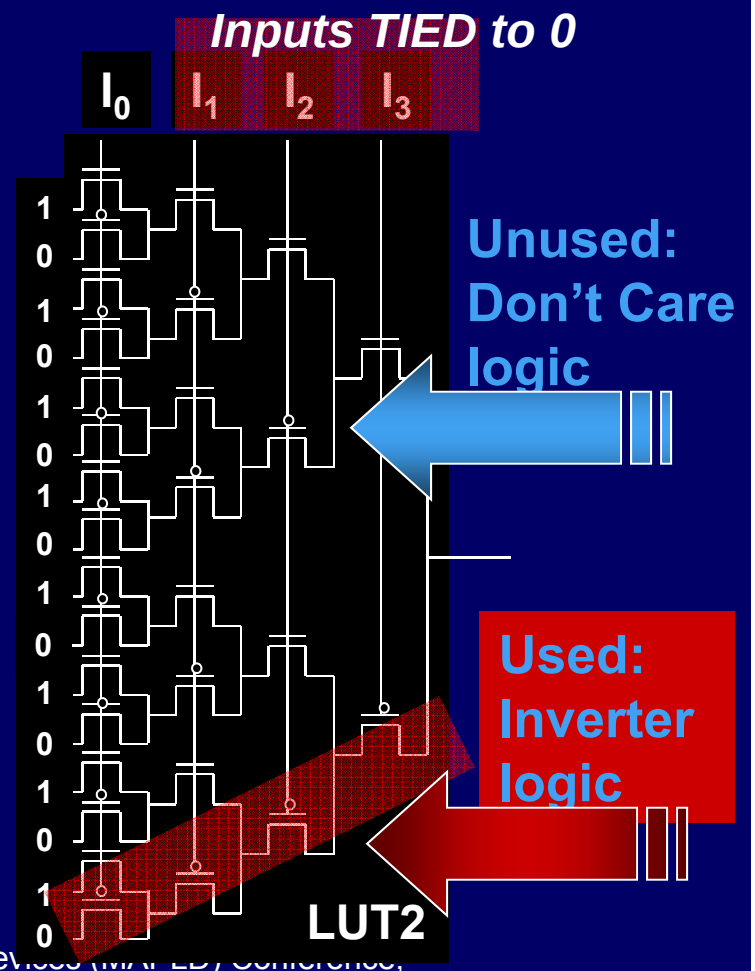
# Good News: Percentage of Used Configuration is Relatively Low




- Most of the existing configuration bits are routing bits
- Many routing bits are unused
- Many of the LUT bits are unused
- Validated via NASA Goddard Virtex Family Fault Injector

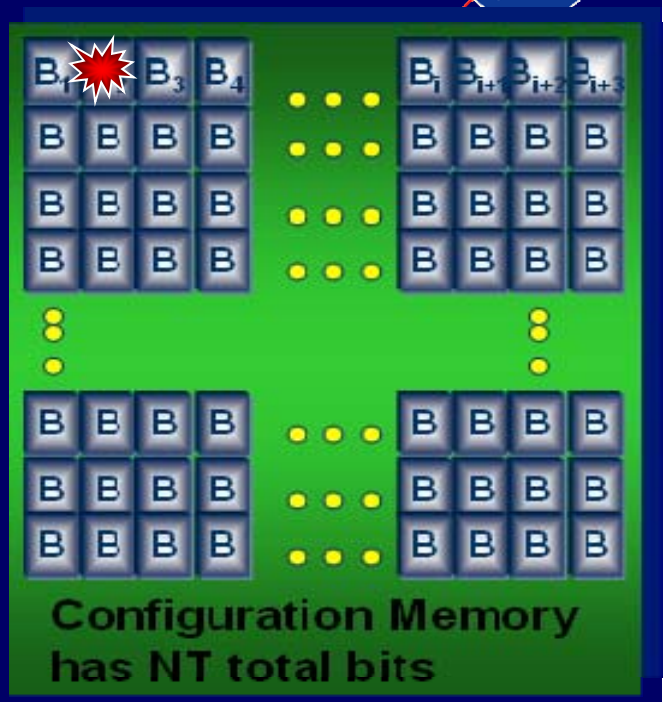
$$\%ConfigMemory < 15\% \left[\frac{device}{design}\right]$$

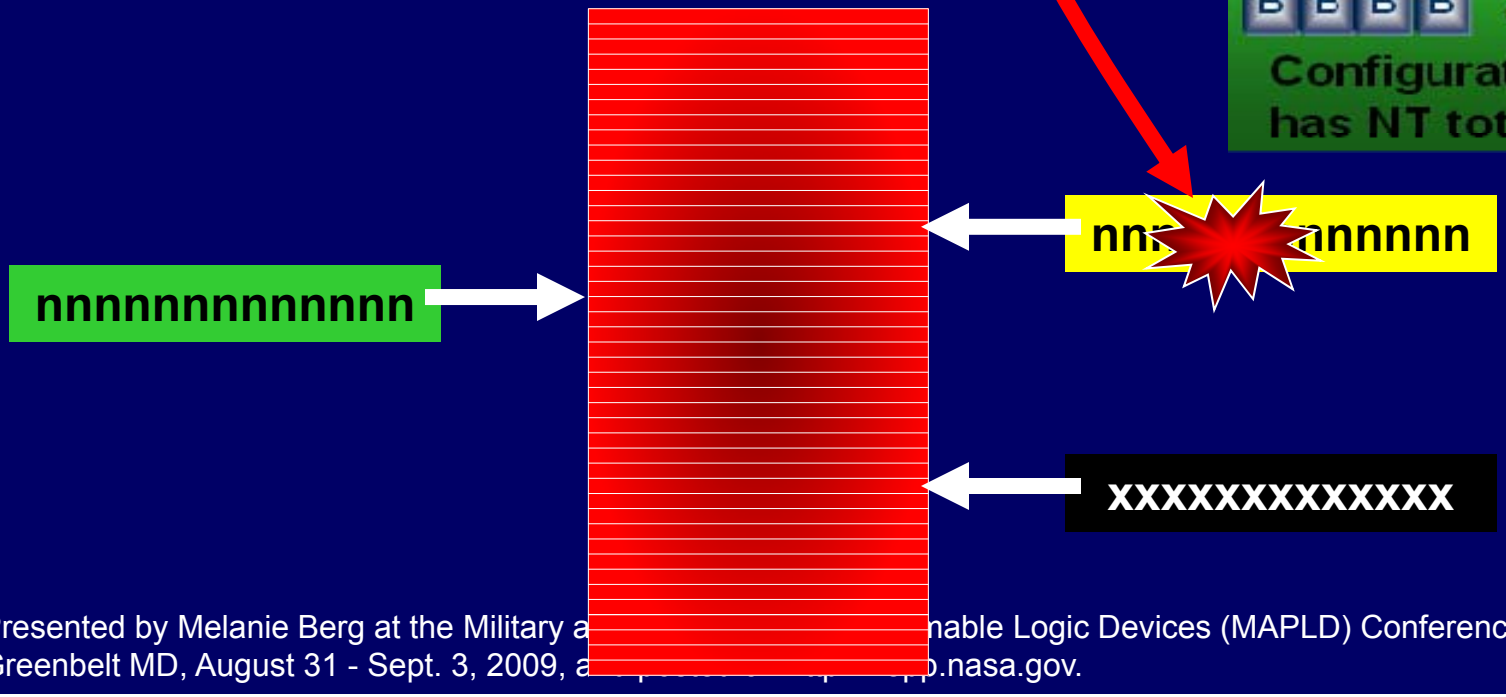
*Source: M.Berg & C. Perez SEE Symposium 2009*



# AFIFO Error Signatures

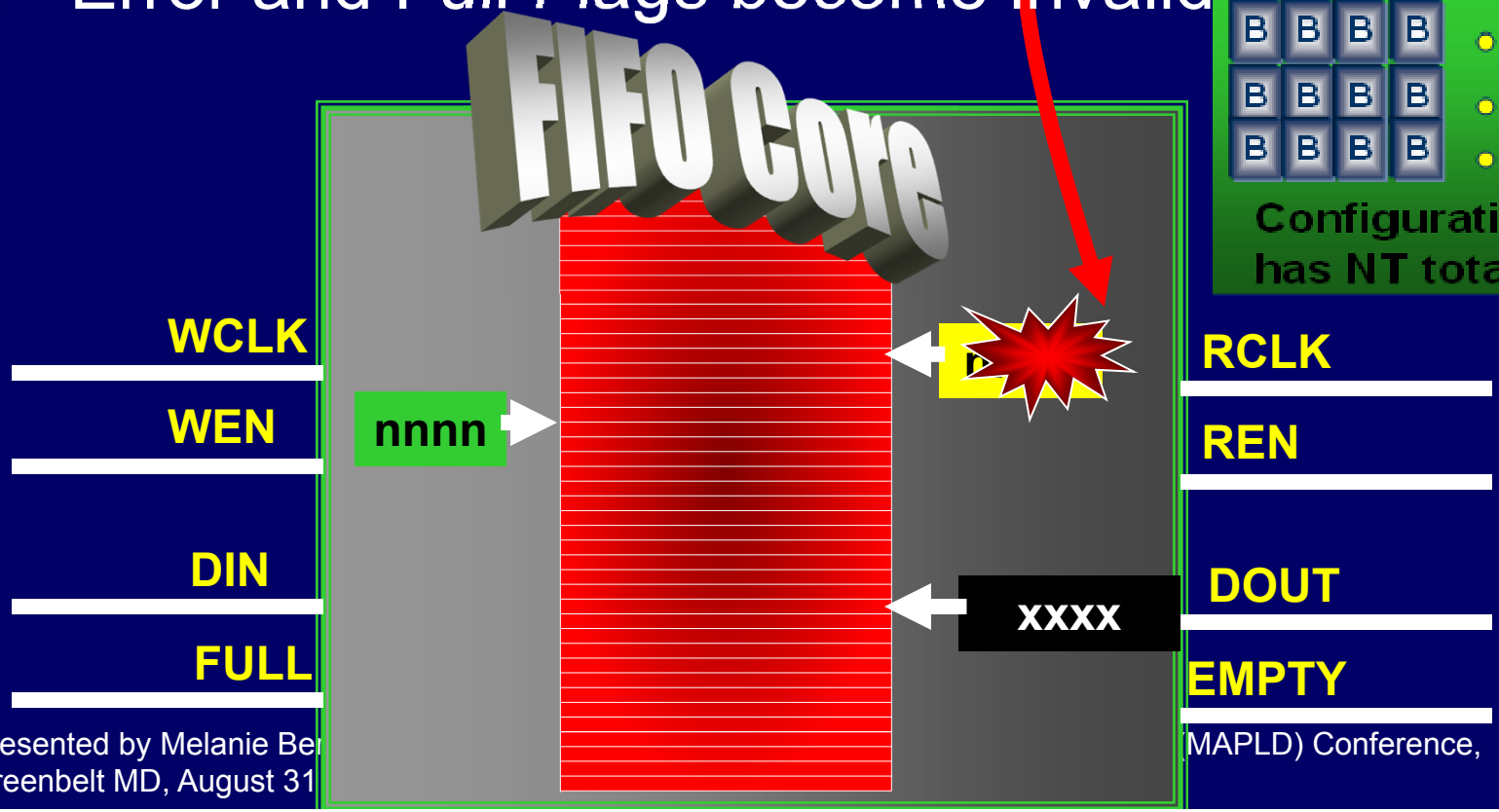
- Will Read Invalid locations
- Can write over unread data
- **Fixing the configuration bit will not fix the address pointer**
- Most importantly, a large amount of data can be lost

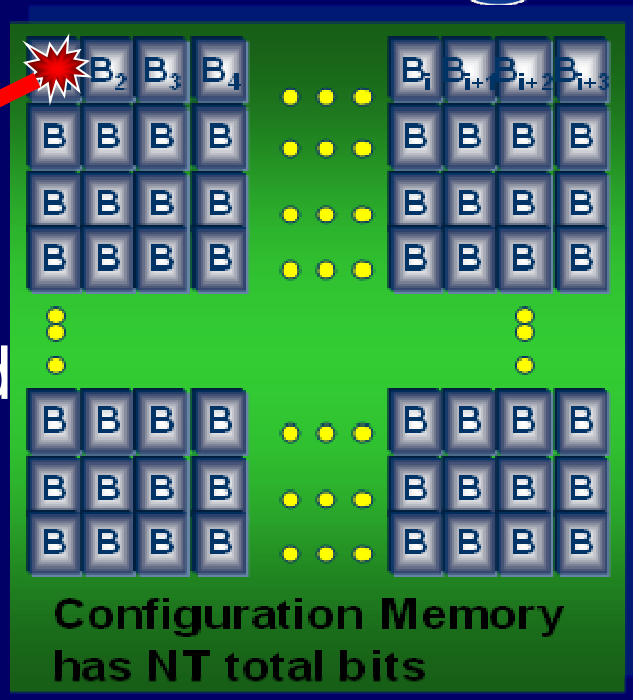


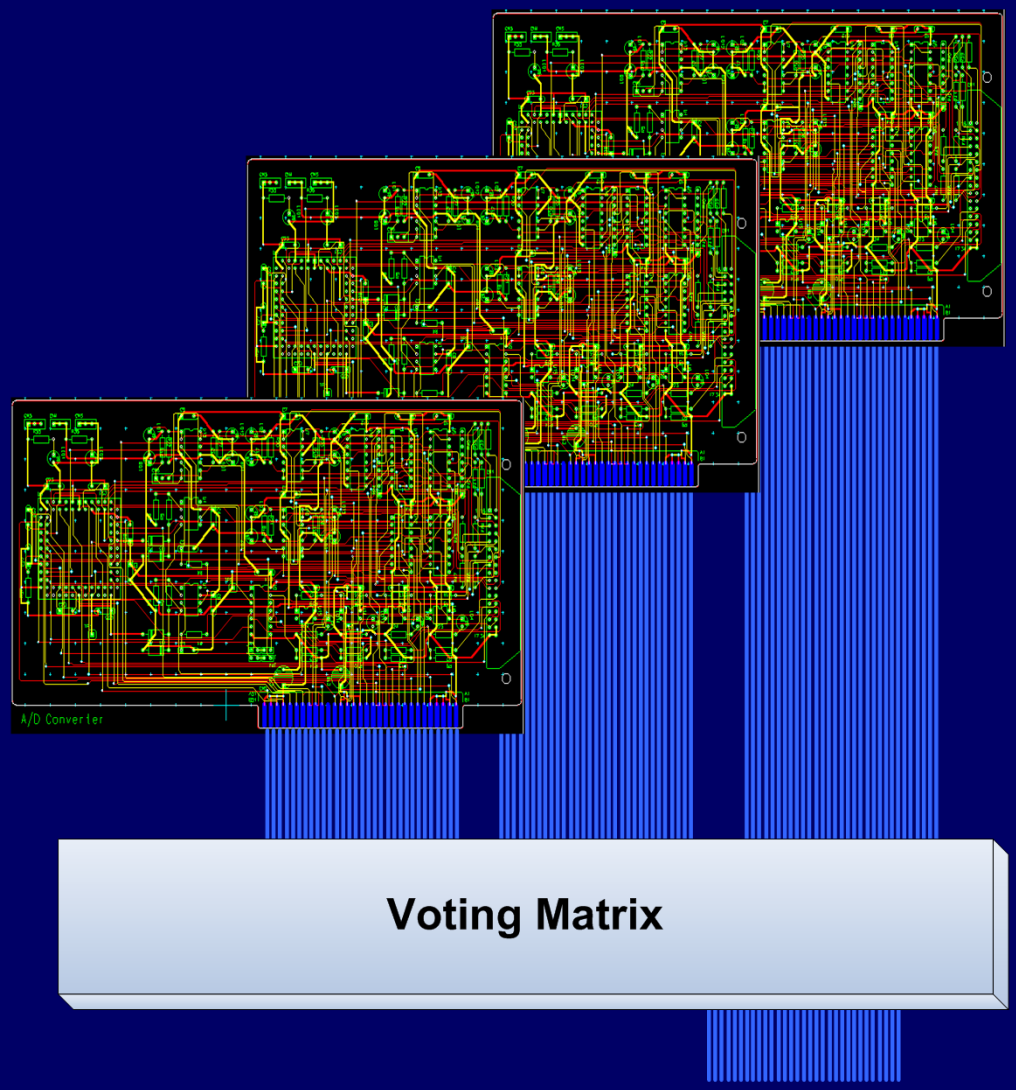



# AFIFO IP Core Error Signatures

- Address Pointers are internal to FIFO and cannot be fixed
- Pointer Corruption (and data loss) is not detectable
- Error and Full Flags become invalid



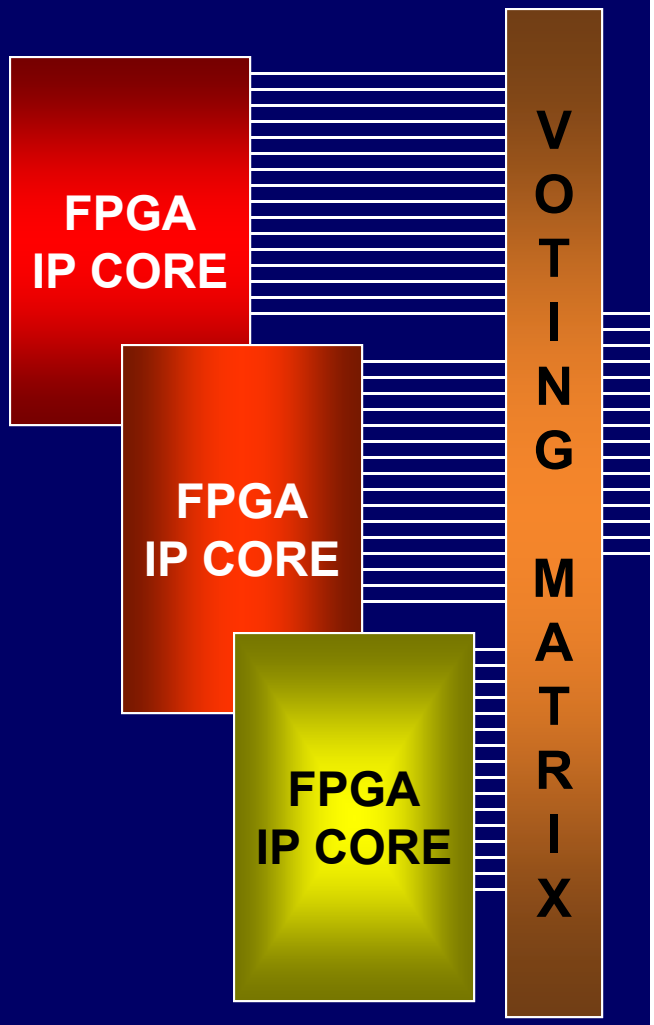






# Mitigating Radiation Effects in AFIFOs with Triple Modular Redundancy (TMR)

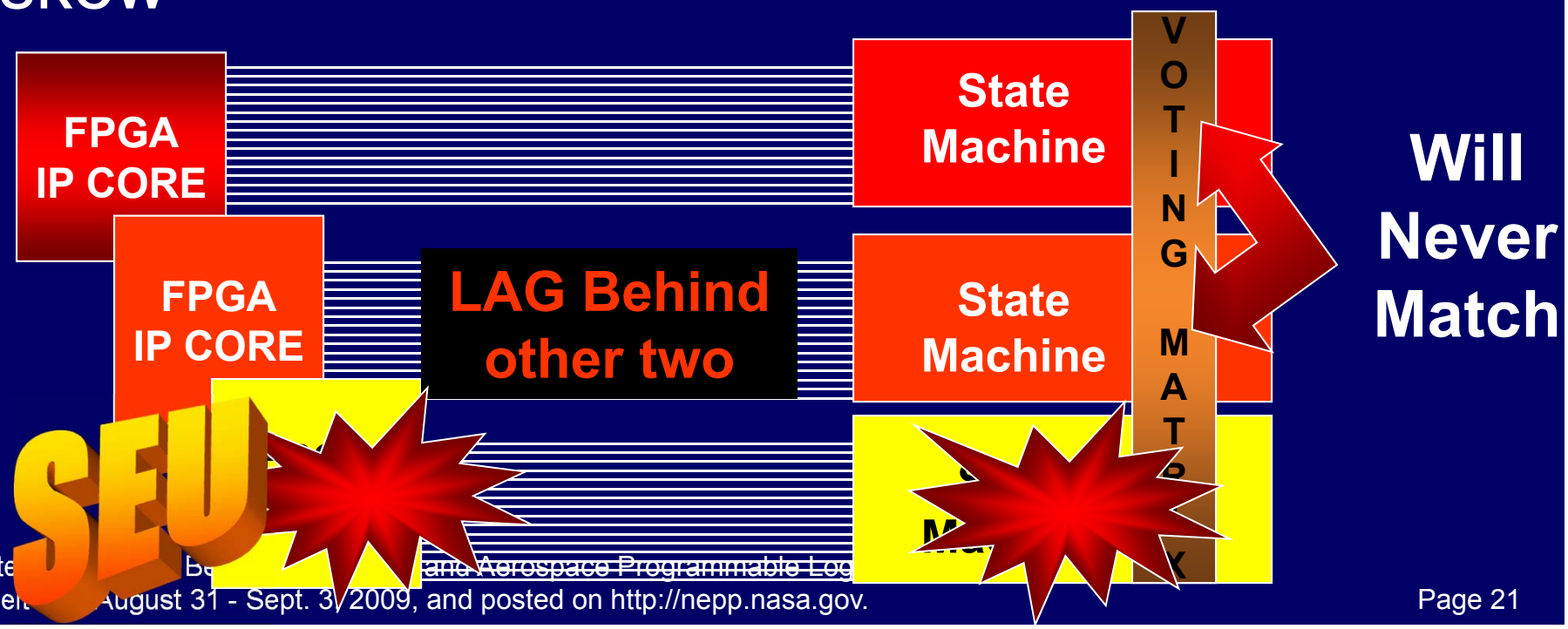
# TMR and IP Core Insertion: Mitigation can only be performed at IP I/O

- **Inability to correct read and internal address pointers**
- More than one core will go bad over time
- Scrubbing will not fix logic state of IP Core (stuck fault)
- Where do you place voters?
  - Directly after outputs
  - Only on the next DFF with Feedback



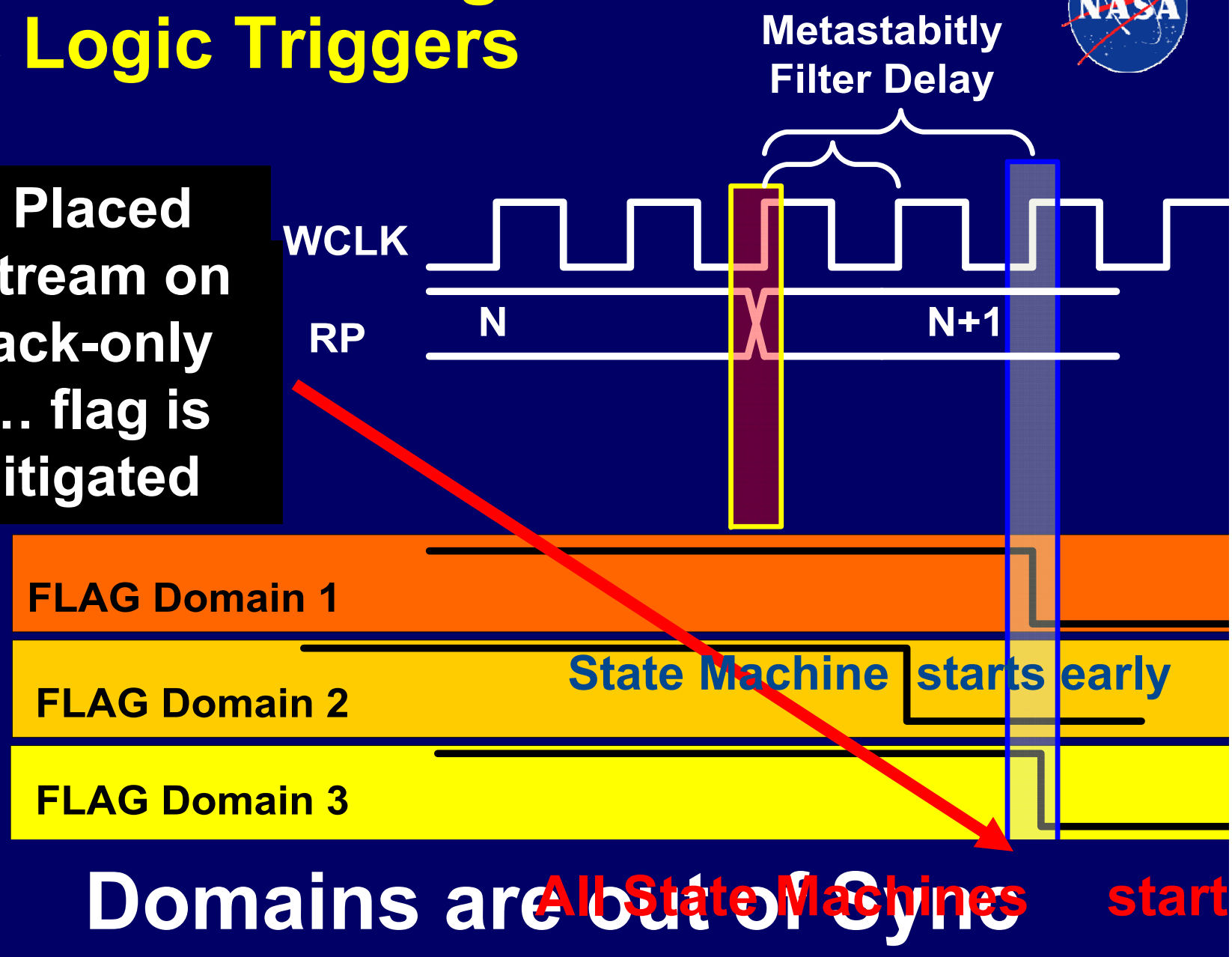
# Asynchronous issues with DFF feedback-only voter insertion:

- Downstream Logic can get triggered in different clock cycles (skewed redundant chains)
- SEU event will not get corrected if chains have skew



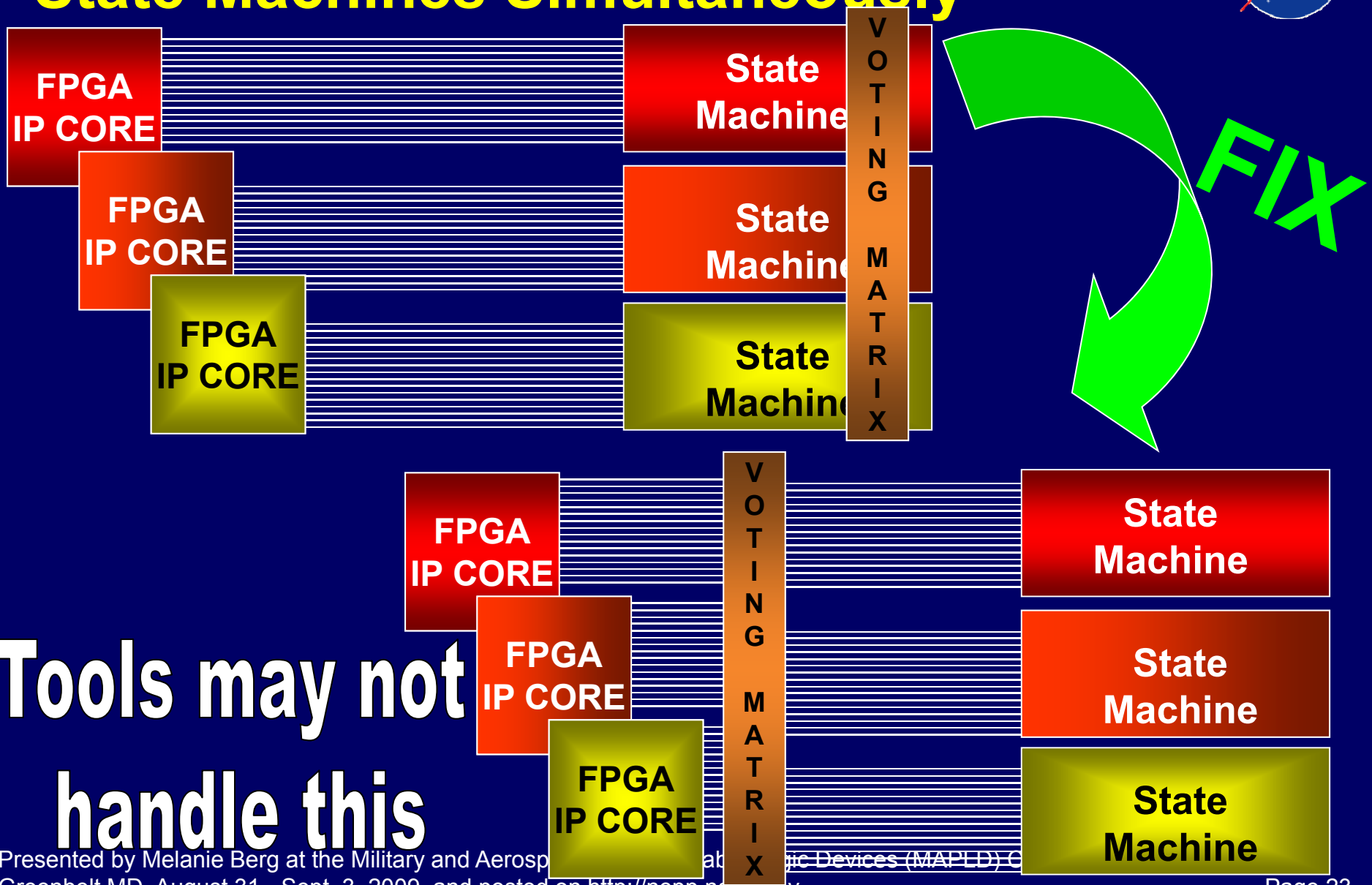
# Asynchronous Flags and Out-of-Sync Logic Triggers

Voter Placed downstream on feedback-only DFFs… flag is not mitigated




Domains are out of Sync

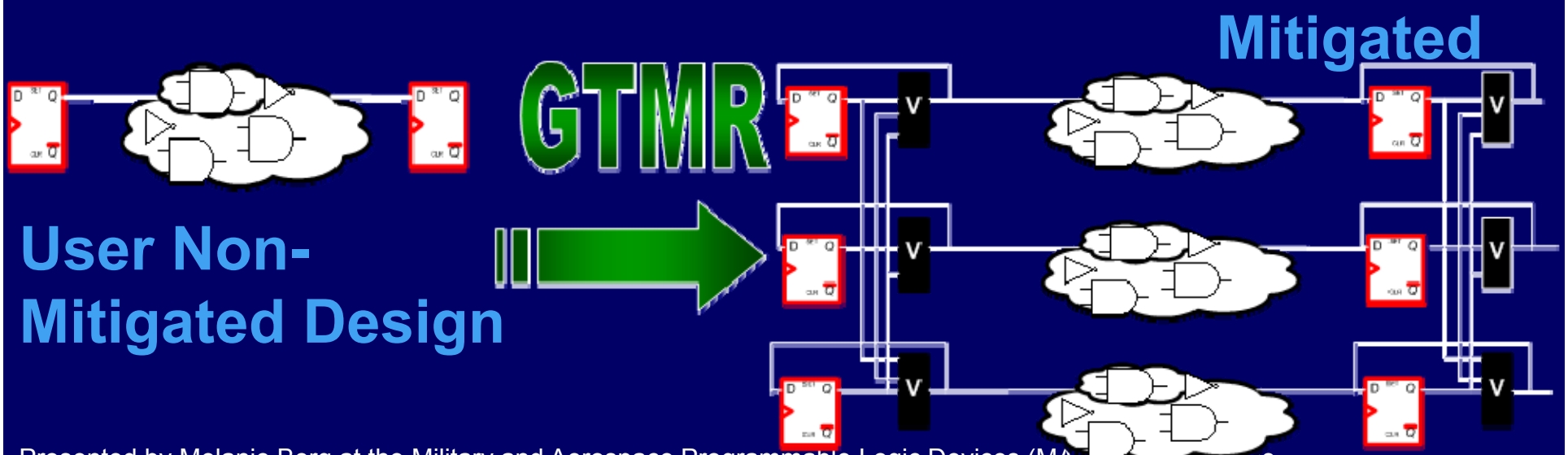
# Moving Voter Directly after IP I/O Starts State Machines Simultaneously




**Tools may not handle this**

# User AFIFO Custom Design and Global TMR (GTMR)

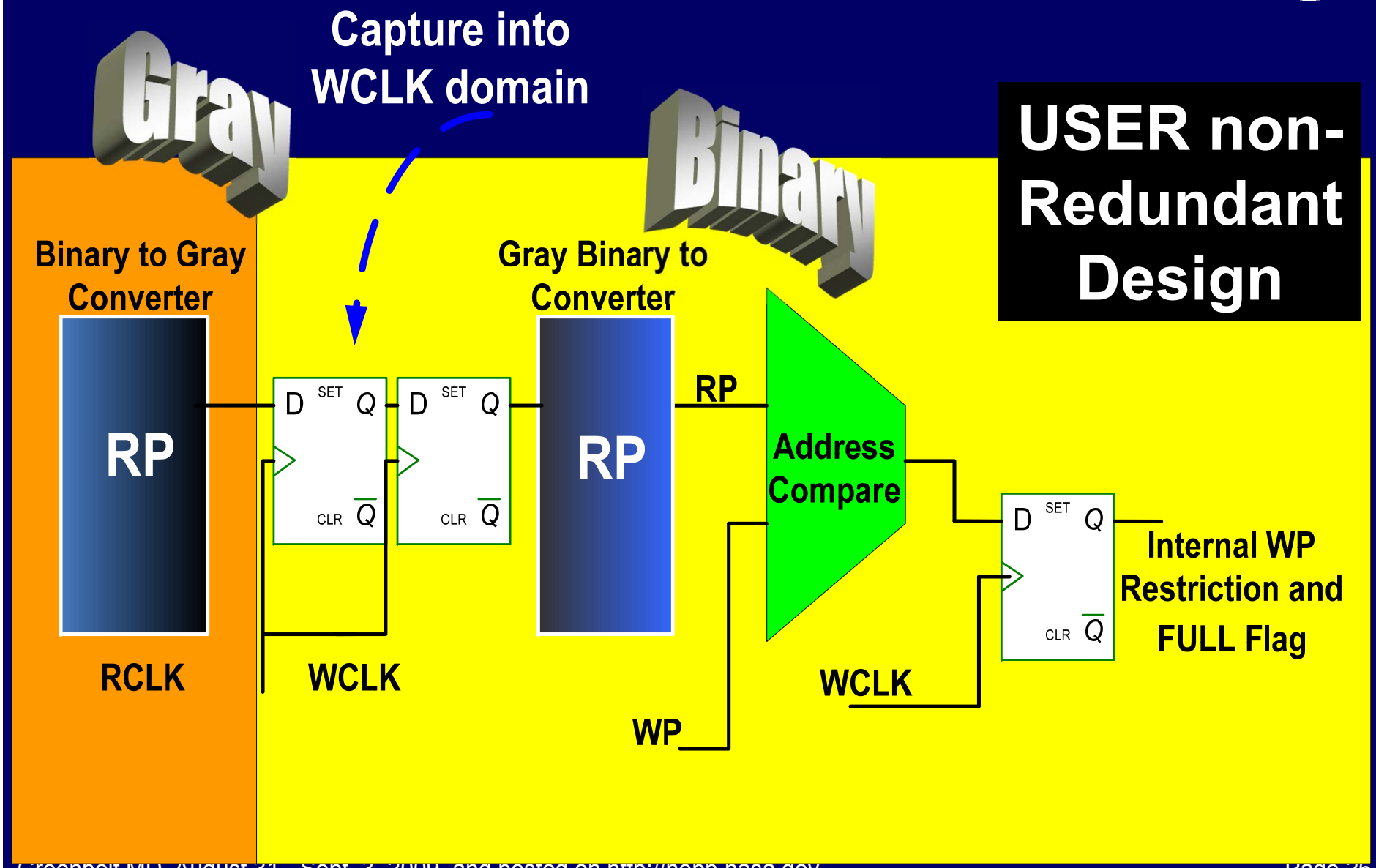
- Triplicate everything in circuit: Clocks, I/O, logic, resets…
- Place Voter after every DFF (or only DFFs with feedback)
- Level of triplication can vary (e.g. no clock redundancy, I/O redundancy, or only voters in feedback paths)



# Global TMR (GTMR)… Continued

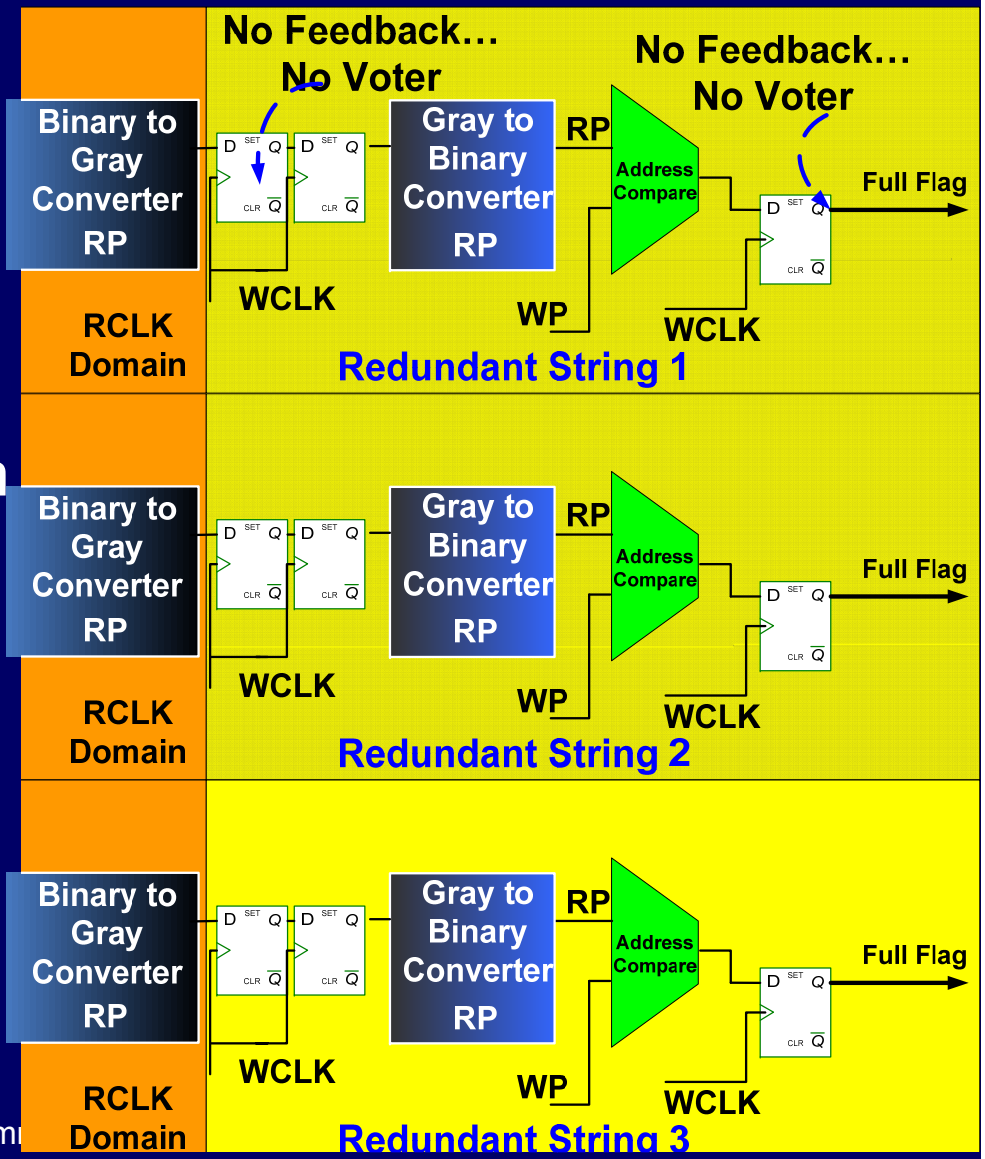
- **Redundancy and Mitigation created via a tool:**
  - Tool creates redundancy and inserts voters post-synthesis (at gate level)
  - Can be difficult to validate tool TMR insertion (did anything break or is everything mitigated as expected?)
- **Commercially Available tools:**
  - Xilnx … XTMR (currently available for V4 and below)
  - Mentor Graphics (soon to be available)
    - **Will have a formal checker to validate RTL matches tool**
- **Redundancy and Mitigation created via RTL:**
  - RTL implementation Best for design reviews
  - Difficult to implement and may be impractical

# GTMR Example: Non Redundant Flag Generation Logic Path… We Start Here!

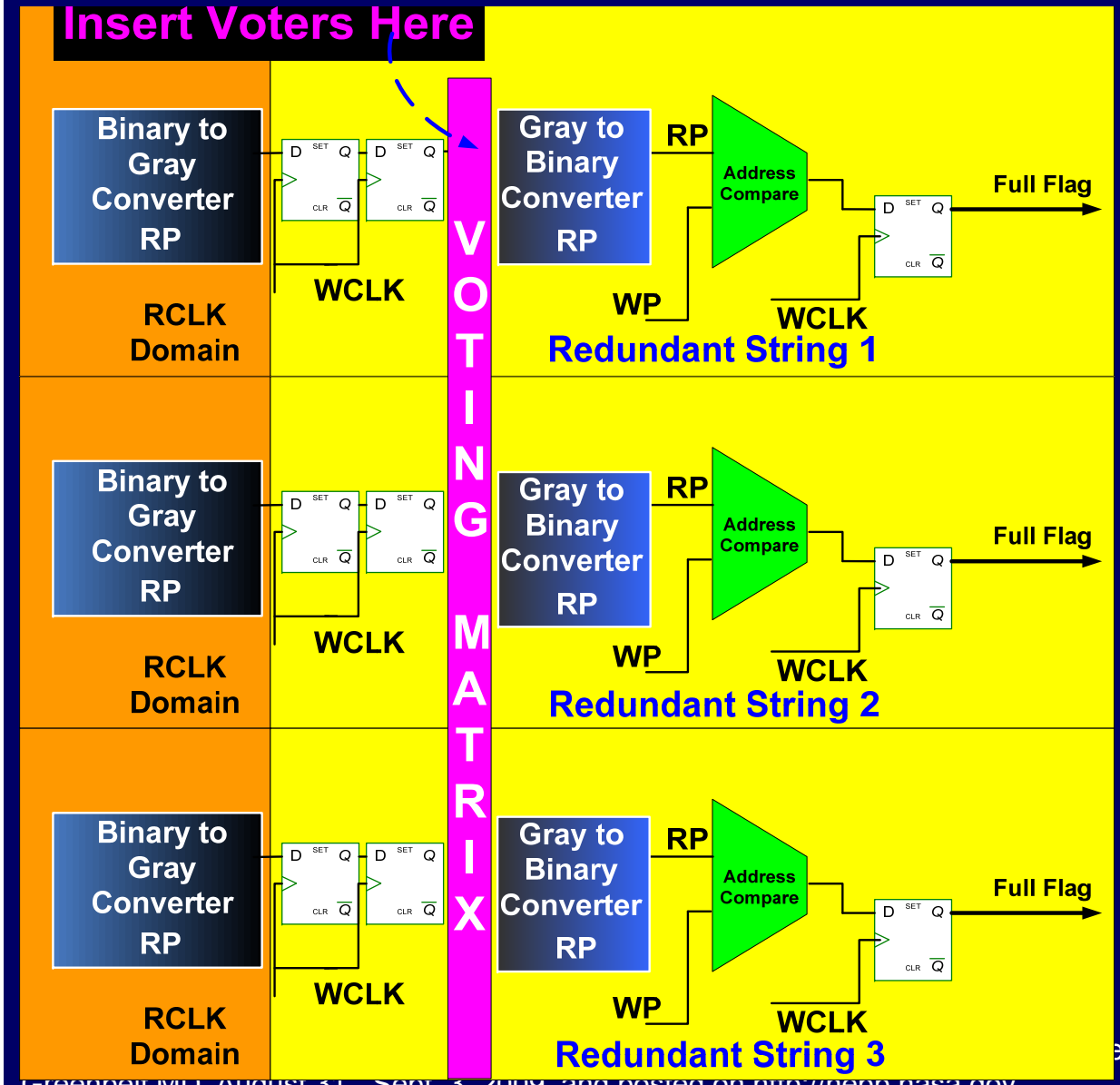


# User AFIFO Design Implementation and Efficient GTMR Mitigation

- Voters only placed after DFFs with feedback
- No voter – no correction
- **Problem**:
  - Redundant flag paths can be one cycle out of sync
  - Will not have correction capability for flag triggered logic



# GTMR User Implementation Solution #1: Insert Voters after Metastability Filter

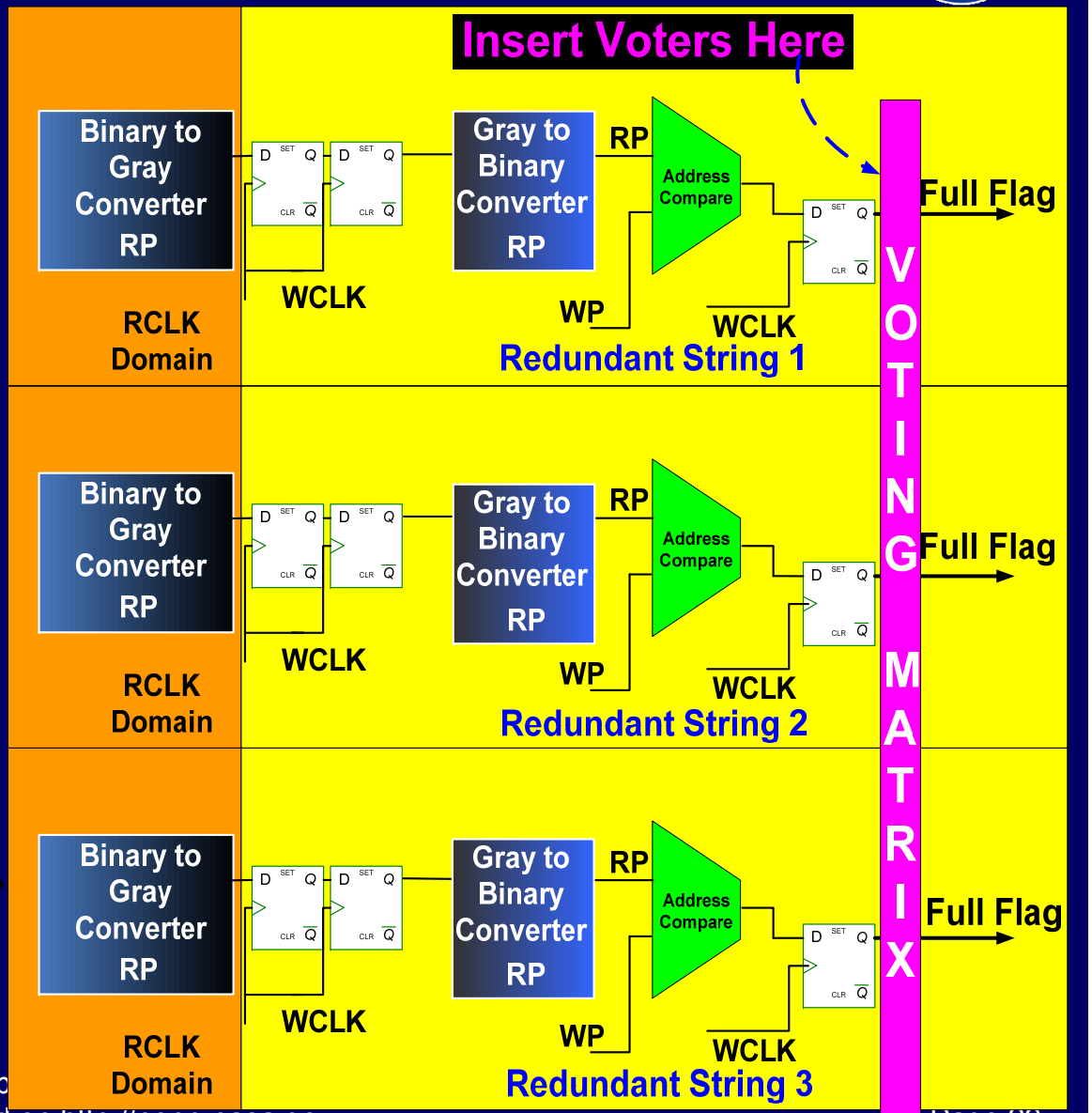



**Voters Will Correct out of sync Redundant String**

# GTMR User Implementation Solution #2: Insert Voters after Flag DFFs

- Voters can be inserted by hand or….
- Voters can be instantiated by making the flag DFF have feedback (use of enable)

**Voters Will Correct Out of sync Redundant String**



# Advantages of Custom Designed AFIFOs with GTMR

- Addresses can be corrected with voters
  - Addresses will be synchronized for all three clock domains
  - FIFOs will always write to the same location
  - ***Clock skew must be controlled***
- Availability of operation is prolonged due to internal correction capability
- Design reviews are enhanced because actual AFIFO implementation can be evaluated in the RTL (VHDL or Verilog).

# Conclusion

- AFIFOs pass data from one clock domain to another
- AFIFO's SEU susceptibility:
  - Configuration can have several upsets per day (Xilinx)
  - Large amounts of data can be lost
- AFIFO IP Cores have no internal correction and can be significantly susceptible to SEUs
- Implementing a customized GTMR AFIFO design reduces SEU susceptibility and availability of operation
- Asynchronous concerns must be addressed:
  - Most tools will not handle these situations
  - Design reviewers must validate domain crossings