Single Event Upsets (SEUs) That Cause Unanticipated Multiple Bit Upsets (MBUs) in Complex Designs and the Effects of Logic Masking

Presented by: Melanie Berg, MEI Technologies in support of NASA/GSFC Melanie.D.Berg@NASA.gov

Ken LaBel, Jonathan Pellish: NASA Goddard Space Flight Center

Hak Kim Christina Siedlick: MEI Technologies in support of NASA Goddard Space Flight Center

To be presented by Melanie Berg at the Single-Event Effects Symposium and Military and Aerospace Programmable Logic Devices (SEE-MAPLD), La Jolla, CA, April 9-12, 2013, and published on http://nepp.nasa.gov/.

Acknowledgements

- *Some of this work has been sponsored by the NASA Electronic Parts and Packaging (NEPP) Program and the Defense Threat Reduction Agency (DTRA)*
- *Thanks is given to the NASA Goddard Radiation Effects and Analysis Group (REAG) for their technical assistance and support. REAG is led by Kenneth LaBel and Jonathan Pellish*
- *Thanks is given to Microsemi for supplying test devices*

Contact Information:

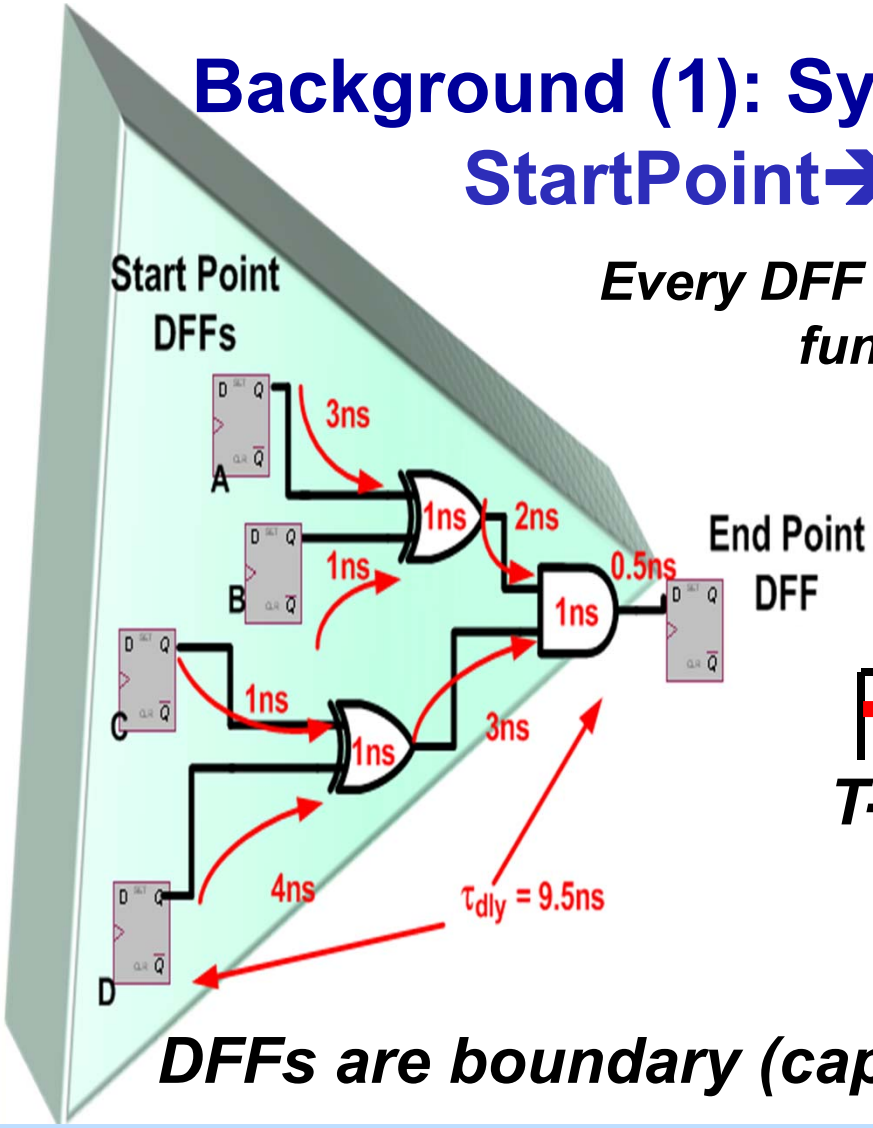
Melanie Berg: NASA Goddard REAG FPGA Principal Investigator:

Melanie.D.Berg@NASA.GOV

Abstract

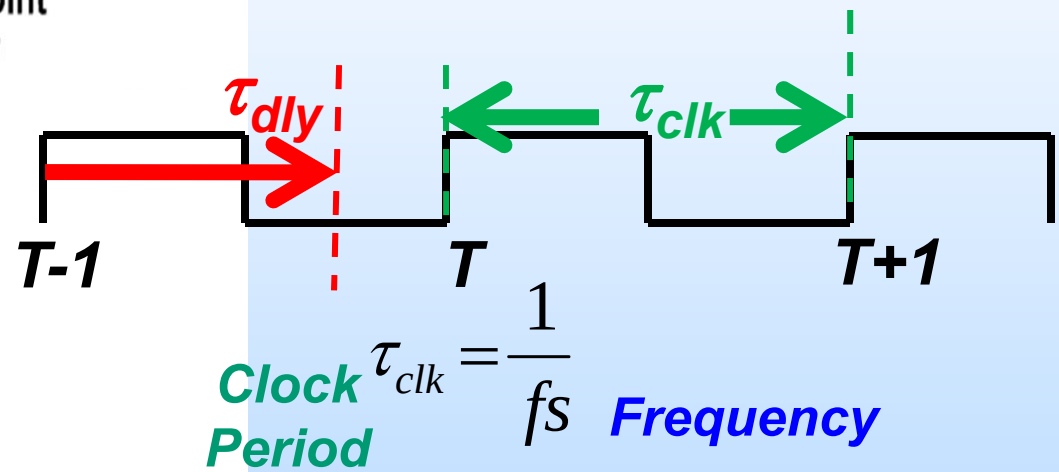
- We investigate how a multiple bit upset (MBU) can occur in a synchronous design data-path due to a single event upset (SEU) occurring in one flip-flop (DFF)
- The resultant MBU is referred to as a domino-effect
- In this presentation we use synchronous counters as the design under analysis. However, the domino-effect applies to any synchronous design with fan-out.
- The NASA Goddard REAG FPGA SEU model is referenced to better understand the proliferation of SEUs throughout a synchronous data-path.
- Emphasis is given to the model's logic masking (P_{logic}) component via combinatorial logic (CL).

Background (1): Synchronous Data Paths: StartPoint→CL →EndPoint



Every DFF can be analyzed as an EndPoint with a function that determines its state:

$$EndDFF(T) = f(StartDFFs(T-1), CL)$$



$$\tau_{clk} = \frac{1}{fs}$$

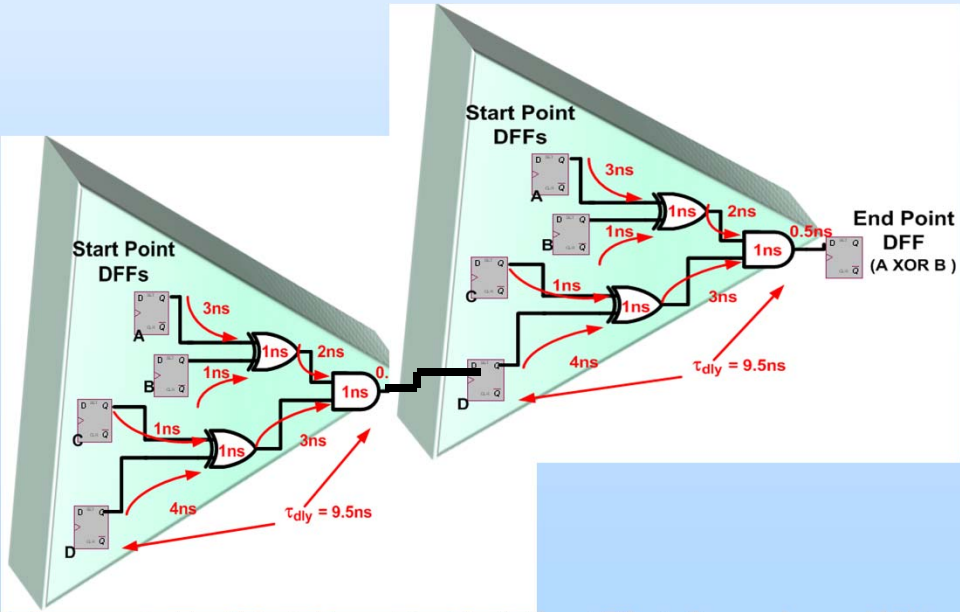
DFFs are boundary (capture and storage) points.

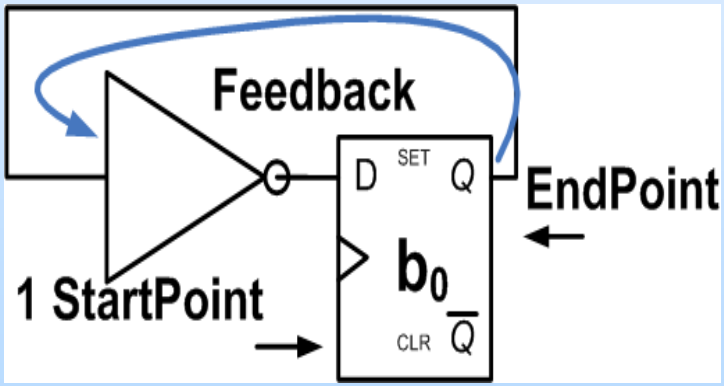
CL create function and delay (τ_{dly}) from StartPoints to EndPoints

τ_{dly} is calculated using Static Timing Analysis design tools

Background (2) : DFF Analysis from the Perspective of StartPoints and EndPoints

- All DFFs are analyzed as both StartPoints and EndPoints
- DFF is only analyzed as an EndPoint at the instance of a clock edge – i.e.; during capture
- DFF is analyzed as a StartPoint at all other moments; i.e., in between clock edges

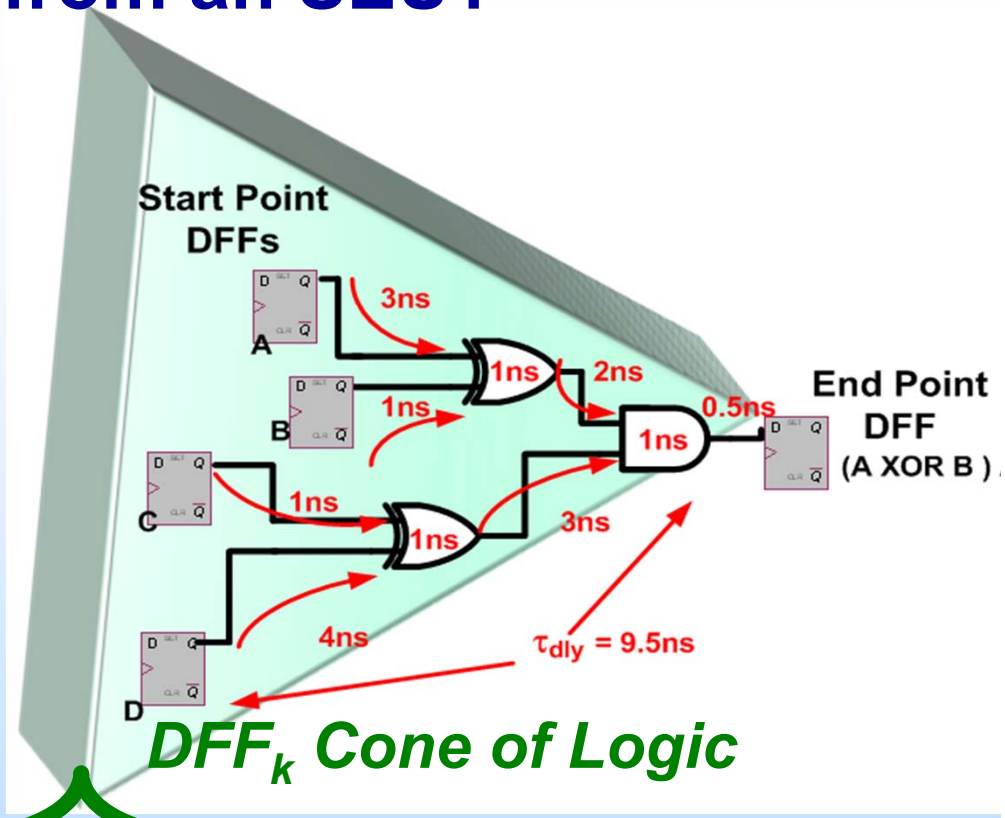




EndPoint = b_0

Background (3) : How can a DFF Contain an Incorrect State from an SEU?

Every DFF has a unique cone of logic



DFF_k Cone of Logic

$\forall_{DFF}$

EndPoint DFF SEUs + StartPoint DFF SEUs + CL SETs

DFF upsets that occur at the clock edge

DFF upsets that occur between clock edges and are captured by EndPoints

Single Event Transients captured by EndPoints

Background(4) : The NASA REAG FPGA Model

System Error	Configuration Error	Functional logic Error	Single Event functional Interrupt Error

$$P(fs)_{error} \propto P_{configuration} + P(fs)_{functionalLogic} + P(fs)_{SEFI}$$

EndPoint

StartPoints

Combinatorial Logic

$$\sum_{k=1}^{\#EndPoint\ DFFs} P_{logic(k)} * \left(\alpha P(fs)_{DFFSEU(k)} + \sum_{j=1}^{\#StartPoint\ DFFs} \left(\beta P_{DFFSEU(J)} (1 - \tau_{dly(j)} fs)) * P_{logic(j)} \right) + \sum_{i=1}^{\#CL} (P_{gen_{(i)}} * P_{prop_{(i)}} * P_{logic_{(i)}} * \tau_{width_{(i)}} fs) \right)$$

Probabilities:

- $P_{logic(k)}$ ➔ ***EndPoint Logic Masking***
- $P_{logic(j)}$ ➔ ***StartPoint Logic Masking***
- $P_{logic(i)}$ ➔ ***Combinatorial Logic Masking***

- $P(fs)_{DFFSEU}$ ➔ ***DFF Upset***
- $P_{prop(i)}$ ➔ ***Electrical Masking***
- $P_{gen(i)}$ ➔ ***SET generation***

τ_{width} = ***average width of SET***

Background(5) : EndPoint Next State Capture of StartPoint SEUs

- **For designs with non-mitigated DFFs the dominant upsets are StartPoints (i.e., SETs and EndPoint upsets are insignificant)… Hence we concentrate on the term:**

Each sum-term is the next state of one EndPoint

$$\sum_{k=1}^{\#EndPoint\ DFFs} P_{logic(k)} * \left(\alpha P(fs)_{DFFSEU(k)} + \sum_{j=1}^{\#StartPoint\ DFFs} \left(\beta P_{DFFSEU(J)} (1 - \tau_{dly(j)} fs)) * P_{logic(j)} \right) + \sum_{i=1}^{\#CL} \left(P_{gen_{(i)}} * P_{prop_{(i)}} * P_{logic_{(i)}} * \tau_{width_{(i)}} fs \right) \right)$$

We differentiate between:

- ***$\beta P(fs)_{DFFSEU(j)}$ = the probability a StartPoint DFF is upset***
- ***$\sum \beta P(fs)_{DFFSEU(j)}(1-\tau_{dly} fs)*P_{logic(j)}$ = the probability that the EndPoint will be affected by the StartPoint upset by the next clock cycle***

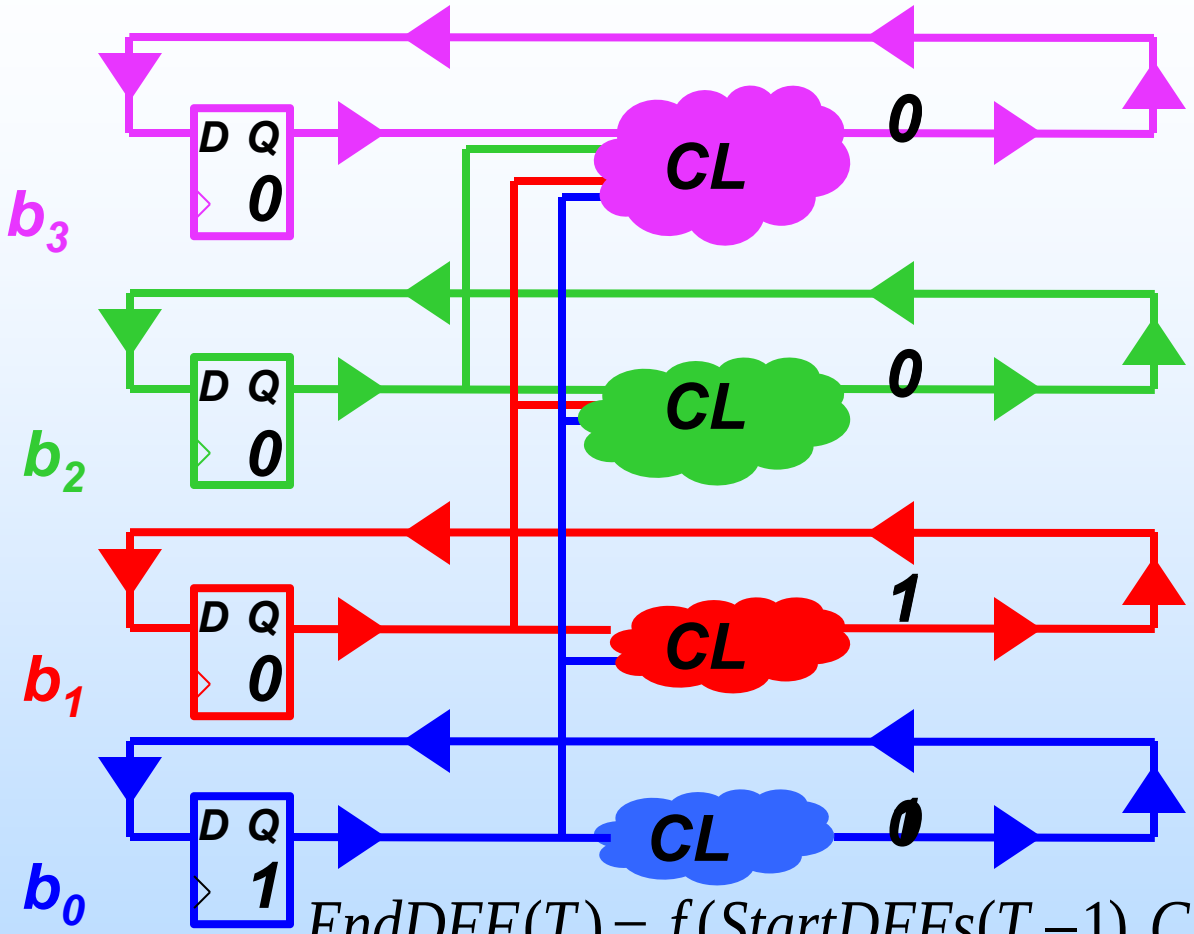
Synchronous Counter Definition: B=B+1

- Synchronous counters are circuits that increment at each clock edge if their logic is enabled during the clock-edge event.
- Each bit of a counter is stored in a DFF. An N-bit counter has N DFFs that are numbered b_0 to b_{N-1};
- b_0 is the least significant bit (LSB)
- b_{N-1} is the most significant bit (MSB)

4-Bit Counter: States 0..15

b3	b2	b1	b0	State
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Current State and Next State Definition



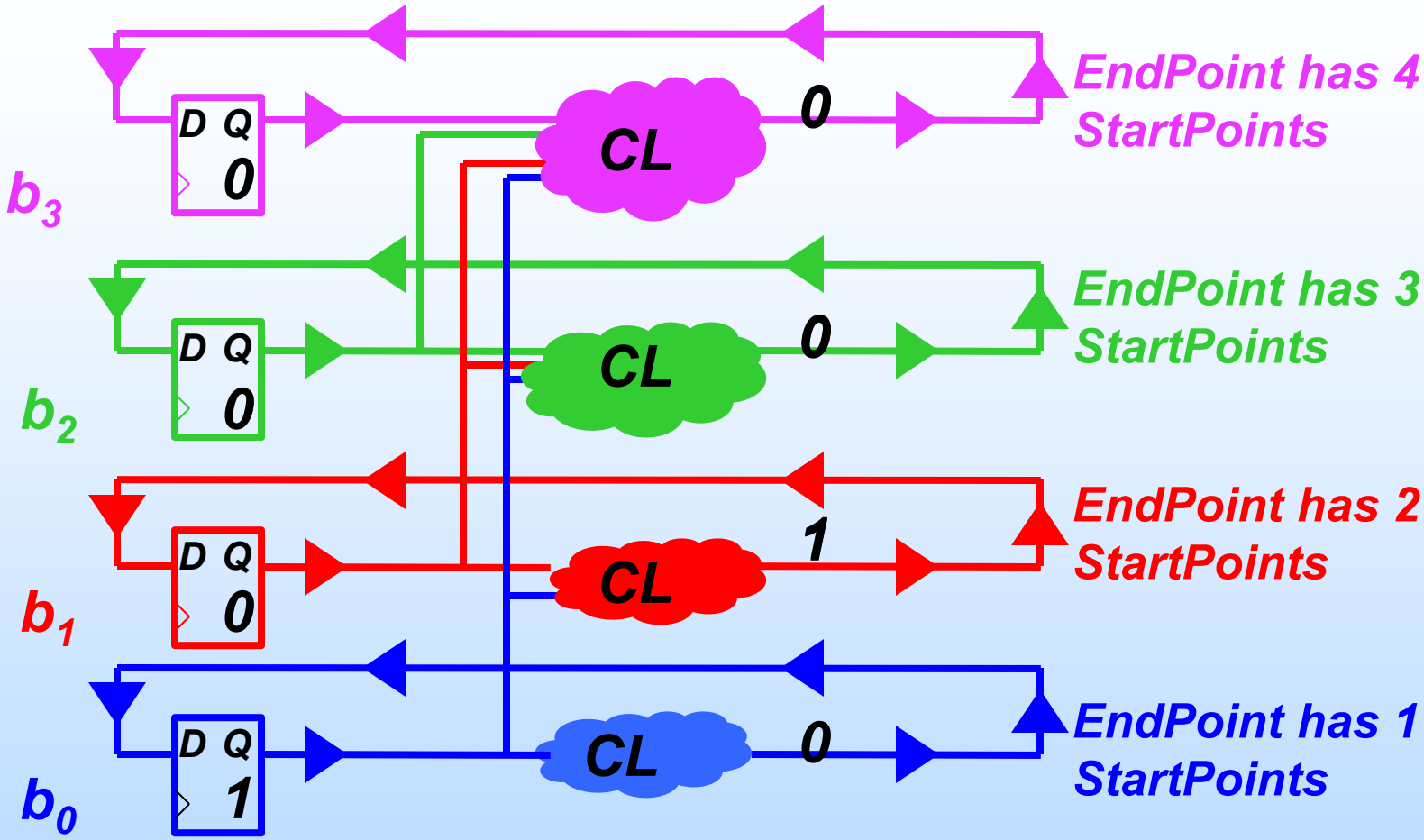
$$EndDFF(T) = f(StartDFFs(T-1), CL)$$

At the instance of the next clock edge, the next state gets latched:

The DFFs become EndPoints at that instance only

DFF are StartPoints at all other instances in time

Current State and Next State Definition

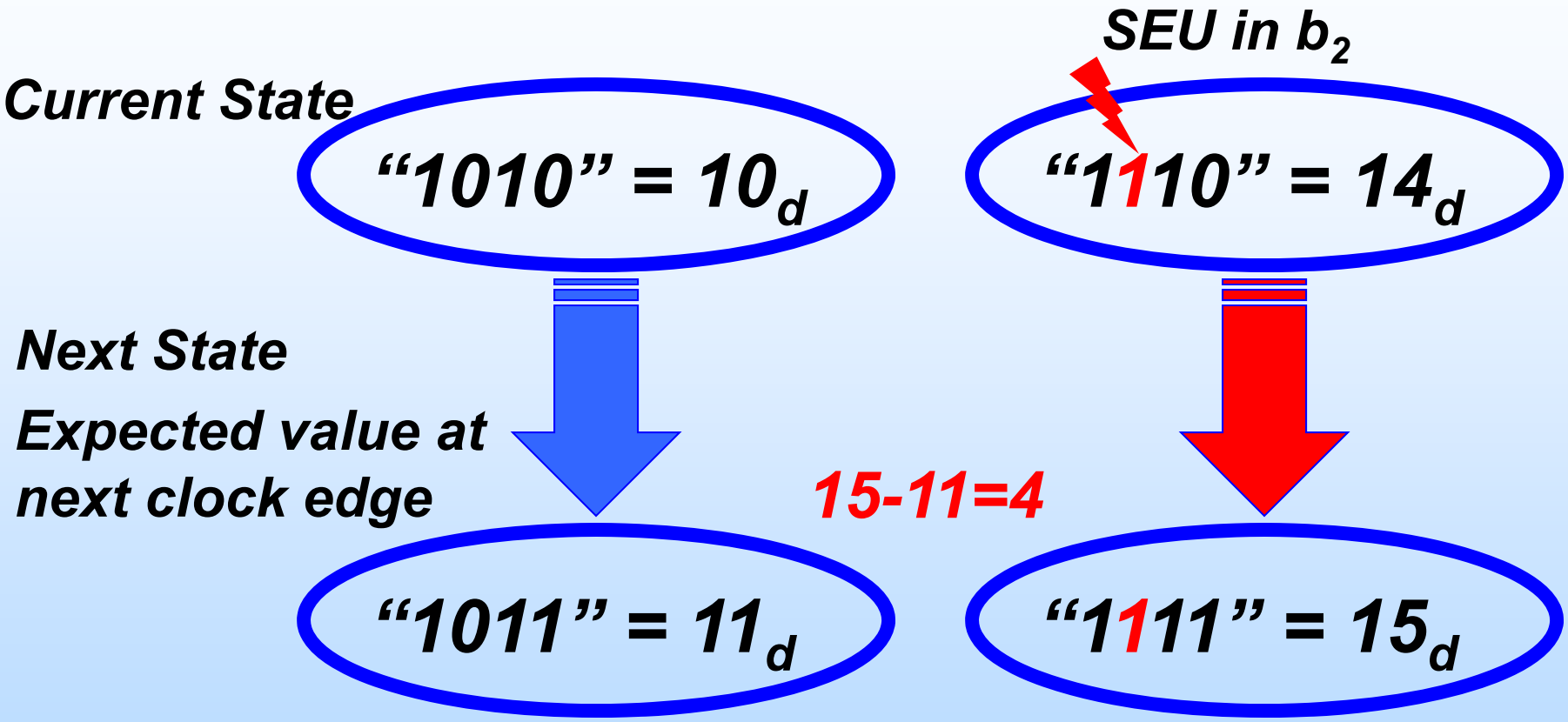


Each bit fans-out to all of its upper order bits;

This is the premise of the Domino-Effect

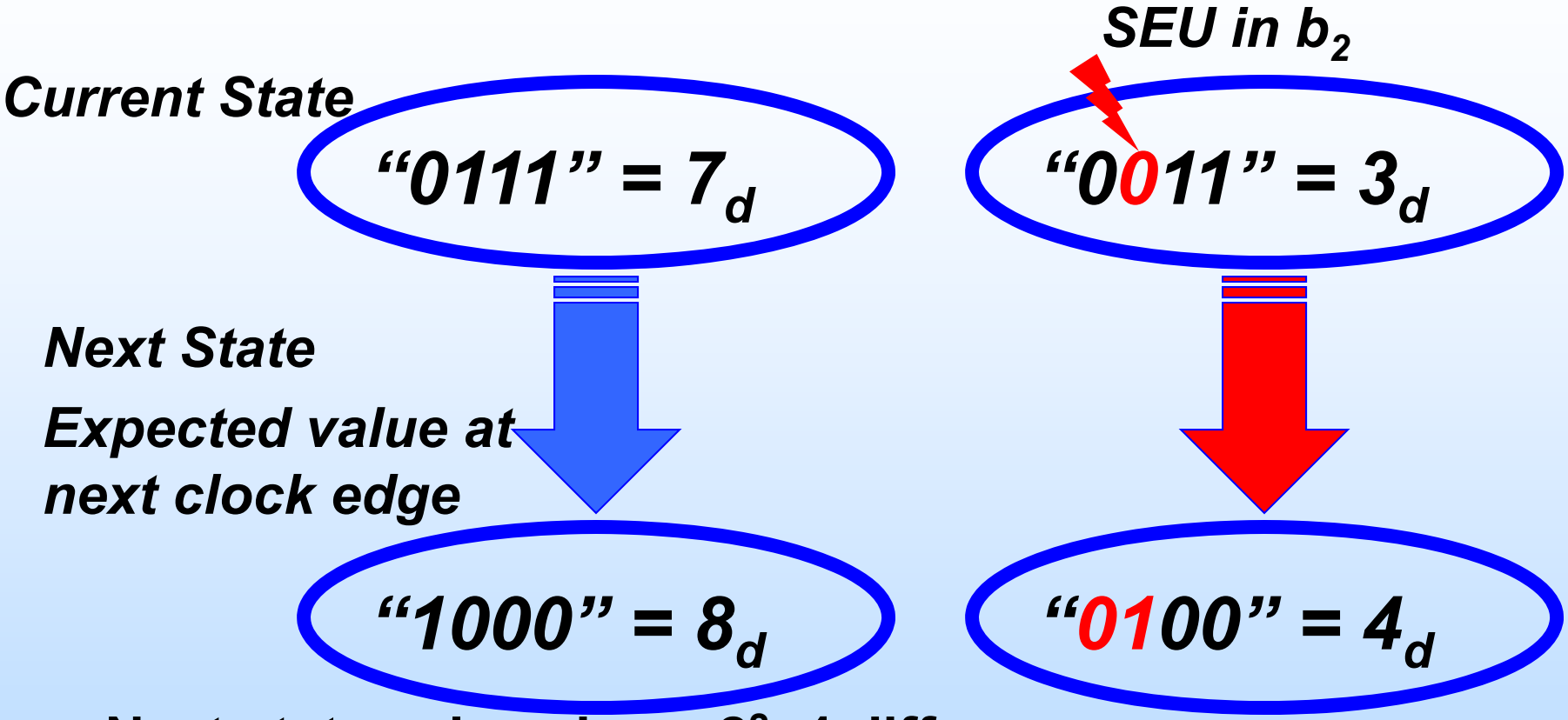
However, logic masking may block the Domino-Effect

Current State and Expected Next State : Single bit SEU Occurring between Clock Edges– Single bit SEU in Next State



- If upset is in b_k, difference in next state values is 2^k
- Difference in expected next and error is $2^2 = 4$
- Upset in b_2 causes next state to have an upset in b_2

Current State and Expected Next State: The Domino Effect



- Next state values have 2^2=4 difference
- However an SEU in b_2 also caused b_3 to be upset.
- The number of EndPoints that will be incorrect will depend on the current state, τ_{dly}, and P_{logic}.

CL between StartPoints and Endoint Define Logic Masking (P_{logic})

$$EndDFF(T) = f(StartDFFs(T-1), CL)$$

$$\sum \beta P(fs)_{DFFSEU(j)}(1-\tau_{dly} fs) * P_{logic(j)}$$

- P_{logic} =1 :
  - No logic masking
  - Examples: Inverter, buffer, XOR
- 0<P_{logic} <1
  - Variable logic masking
  - Examples: AND, OR, MUX
- P_{logic} =0: full logic masking
  - Complete logic masking 100% of the time
  - Example: Triple Modular Redundancy (TMR) with voter

Once we figure out the equation for each EndPoint, we can determine the impact of P_{logic} and potential Domino-Effects

EndPoint b_0 Circuitry and Data-Paths

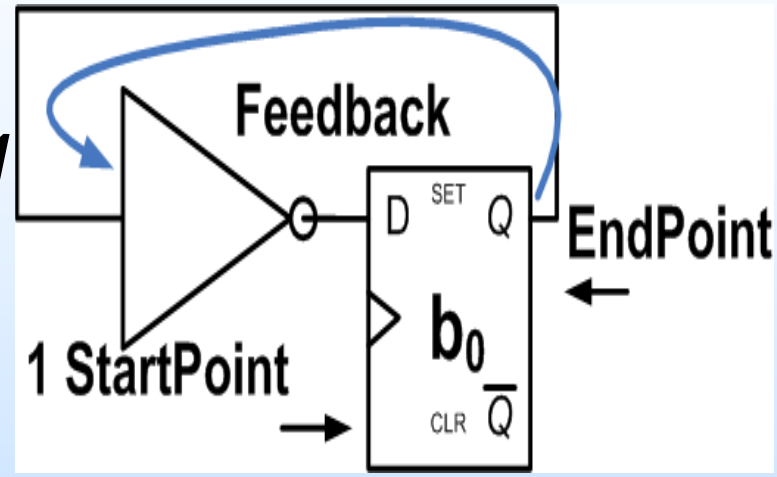
b_0= INV b_0

4-Bit Counter: States 0..15

b3	b2	b1	b0	State
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

$$EndDFF(T) = f(StartDFFs(T-1), CL)$$

P_{logic} = 1



EndPoint = b_0

- **b_0: next state = inverse of current state**
- **Only one data path: b_0 → INV → b_0**

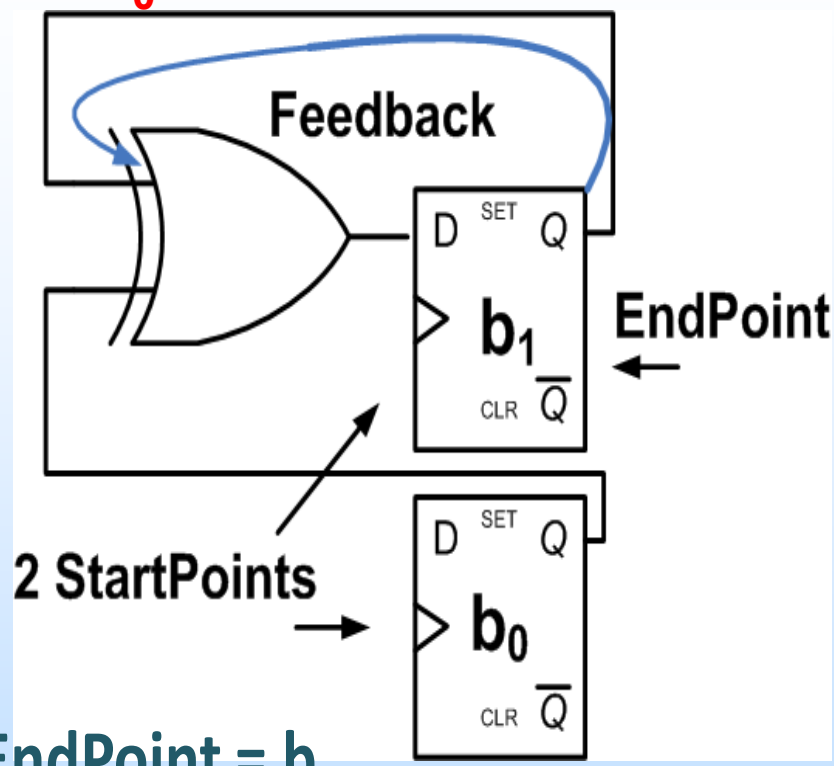
EndPoint b_1 Circuitry and Data-Paths:

4-Bit Counter: States 0..15

b3	b2	b1	b0	State
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

$b_1 = b_1 \text{ XOR } b_0$

$P_{logic} = 1$



EndPoint = b_1

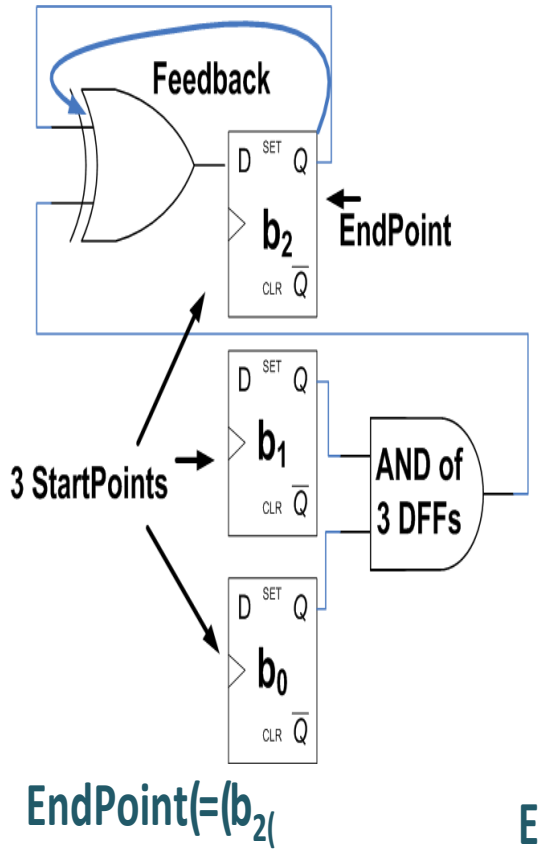
- b_1: next state = XOR of current state for b_0 and b_1
- Two data paths:

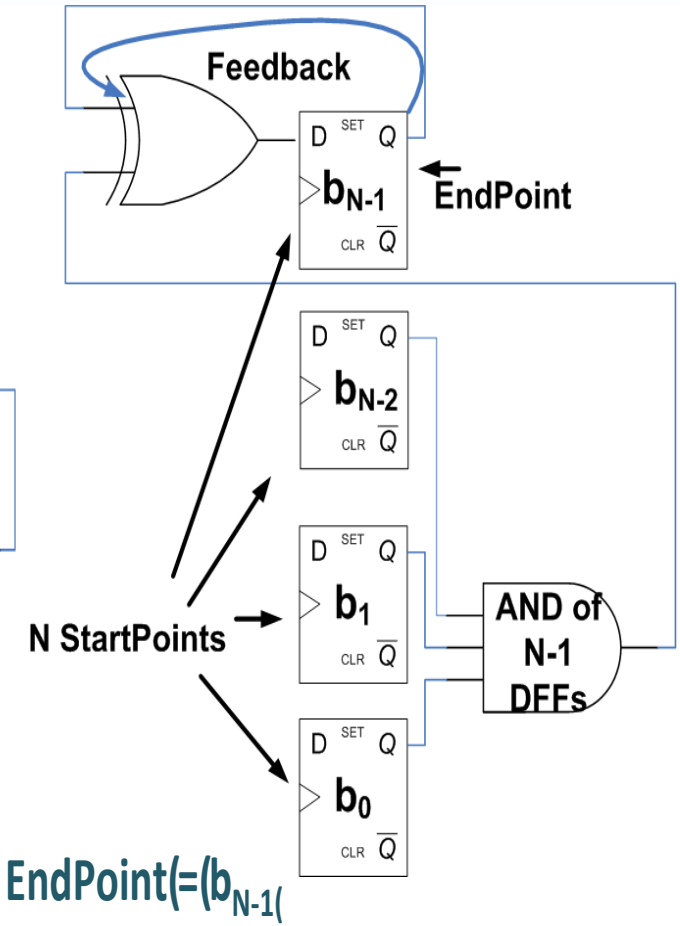
$b_0 \rightarrow XOR \rightarrow b_1$ and $b_1 \rightarrow XOR \rightarrow b_1$

EndPoints b_k Circuitry and Data-Paths

4-Bit Counter: States 0..15

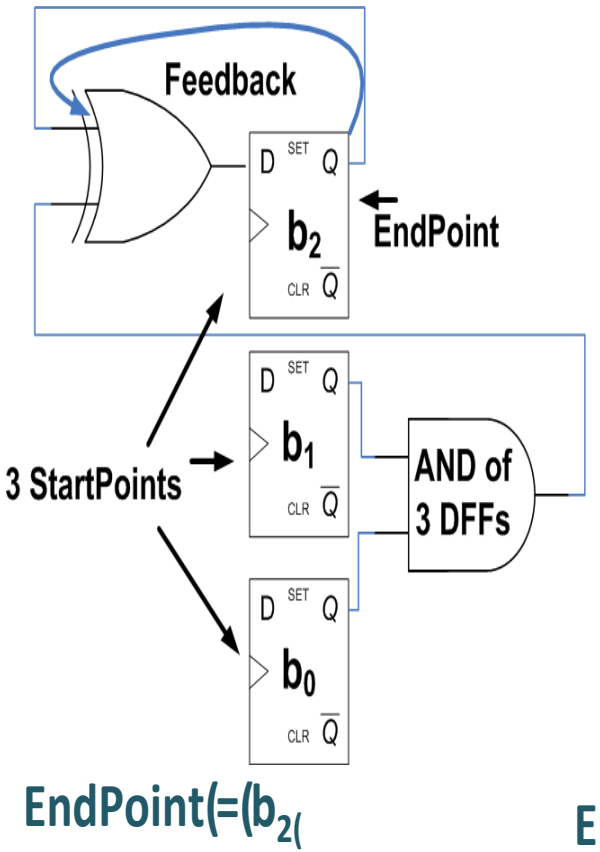
b3	b2	b1	b0	State
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

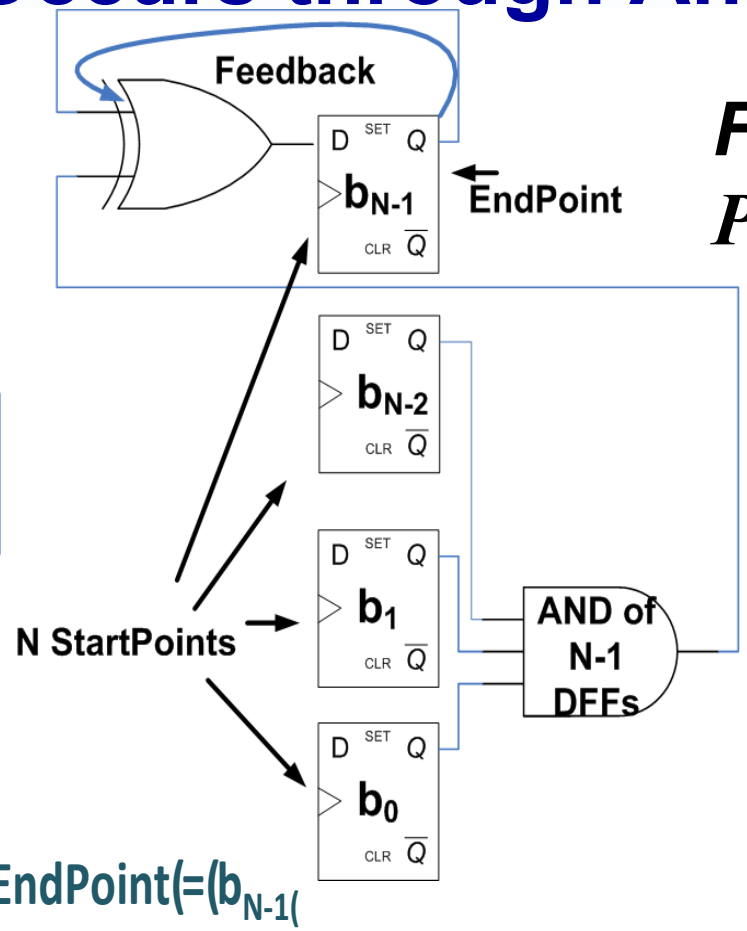




- **b_k next state = inverse of current state if all lower bits are equal to a logic '1'**

b_k Circuitry and Data-Paths: Domino-Effect Occurs through And-Path





Feedback Path:

$$P_{logic} = 1$$

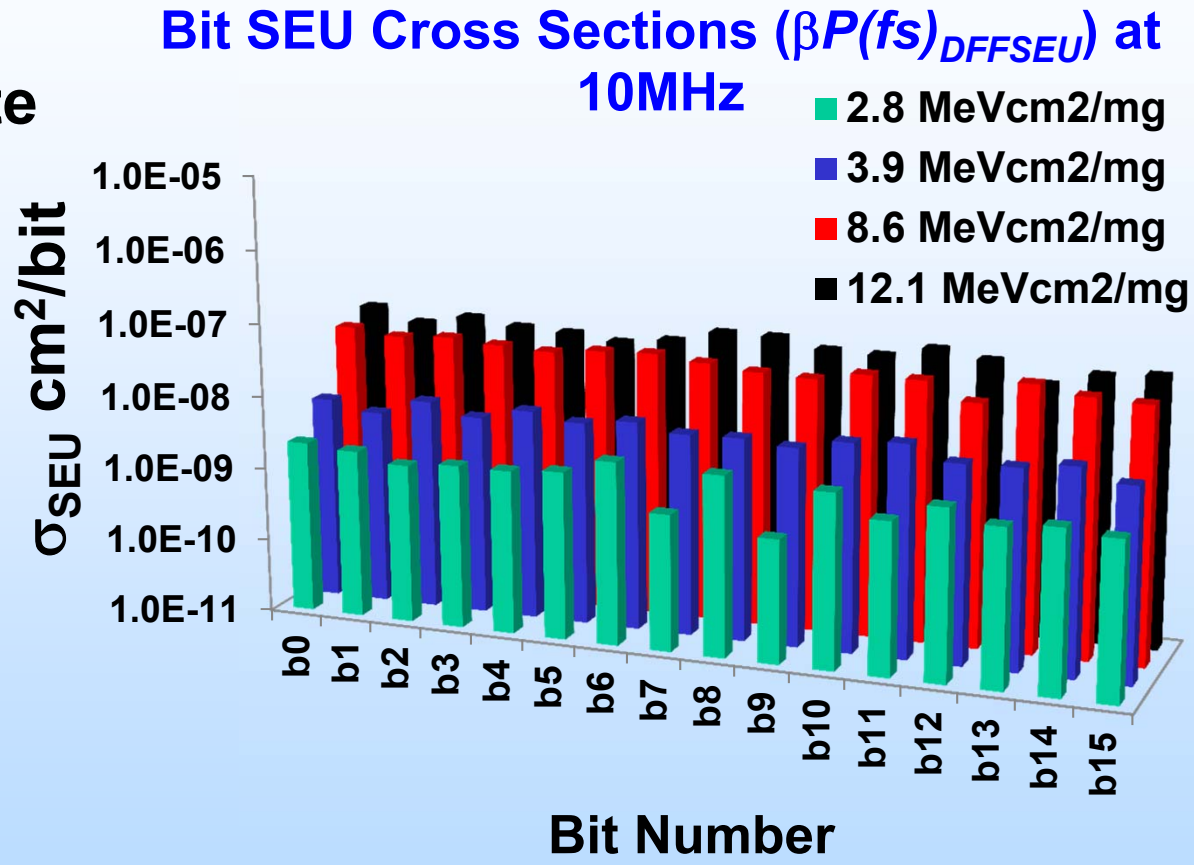
AND Path:

$$P_{logic} = 2^{-k}$$

- $b_k \rightarrow$ XOR $\rightarrow b_k$ feedback data: 1 path
- $b_j \rightarrow$ AND $\rightarrow$ XOR $\rightarrow b_k$ k paths: $0 \leq j \leq (k-1)$

σ_{SEU}s for StartPoint DFFs obtained at Texas A&M Cyclotron Facility

- Data does not represent Next State domino effects
- Alternatively, they represent σ_{SEU}'s of StartPoint DFF upsets = $\beta P(f_s)_{DFFSEU(j)}$
- Results show that each bit has a statistically similar probability of upset.



Example: Probability the Next State of b_2 will not Match its Expected Value

$$\sum_{j=1}^{\#StartPoint\ DFFs} \left(\beta P_{DFFSEU(J)}(1-\tau_{dly(j)} fs)) * P_{logic(j)} \right); \quad for\ \tau_{dly(j)} \ll \tau_{clk} \quad \sum_{j=1}^{\#StartPoint\ DFFs} \left(\beta P_{DFFSEU(J)} * P_{logic(j)} \right);$$

Feedback:

$P_{logic} = 1$

And-Path:

$P_{logic} = 2^{-2}=0.25$

And-Path:

$P_{logic} = 2^{-2}=0.25$

$$\beta P(f_s)_{DFFSEU(b2)} + 0.25\beta P(f_s)_{DFFSEU(b1)} + 0.25\beta P(f_s)_{DFFSEU(b0)}$$

From the σ_{SEU} graph at LET = 12 MeV-cm^2/mg

$$\beta P(f_s)_{DFFSEU(b2)} \approx \beta P(f_s)_{DFFSEU(b1)} \approx \beta P(f_s)_{DFFSEU(b0)} \approx 4.0 \times 10^{-7};$$

σ_{SEU} for the next state of b_2 will not match its expected value: $\approx 4.2 \times 10^{-7}$

- **Same analysis is performed for each bit analyzed as an EndPoint**

Conclusion

- By using a counter as the design under analysis, we show that a domino-effect can cause an SEU to turn into an MBU within a synchronous circuit.
- However, the domino-effect is reduced by data-path logic masking for higher order counter bits.
- We illustrate a method for calculating data-path logic masking and applying the calculations to radiation data.
- Counter circuits were used as an example:
  - The same type of analysis can be used for more than counters
  - Domino effects can occur in any circuit that has fanout and that has P_{logic} >0