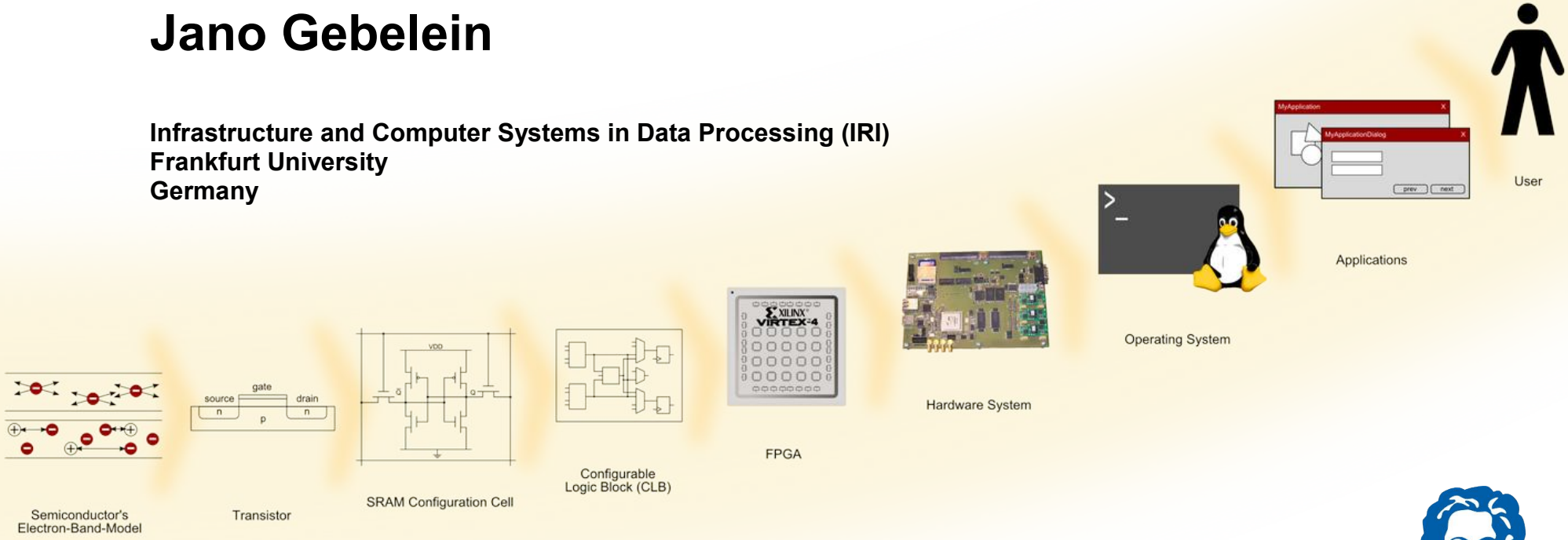
„Hamming FSM with Xilinx Blind Scrubbing - Trick or Treat“

Jano Gebelein

Infrastructure and Computer Systems in Data Processing (IRI)
Frankfurt University
Germany



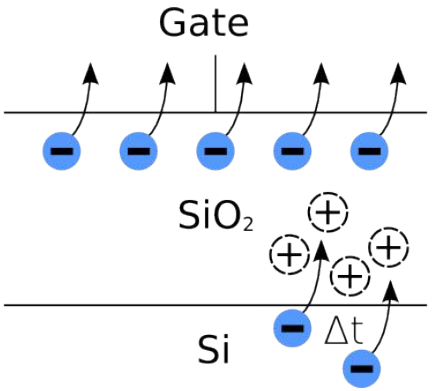
FPGA-CC
January 31st, 2012
Mannheim, Germany

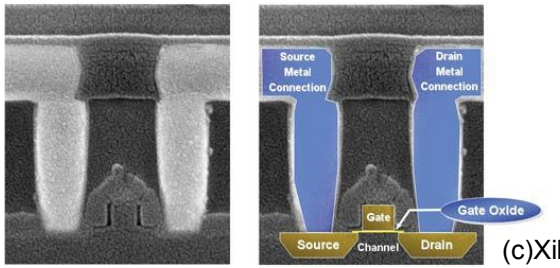
Outline

- **Quick Introduction to SRAM Radiation Effects**
- SEE-aware circuit design
- Hamming FSM Design
- Beamtest Results
- Xilinx Blind Scrubbing - Trick or Treat

SRAM Radiation Effects

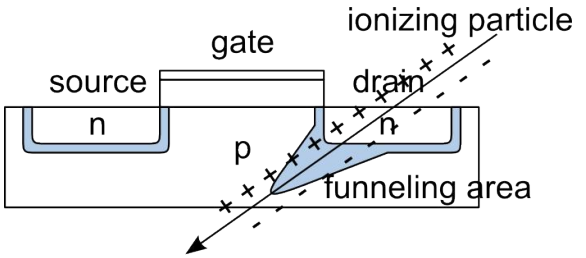
- well known and often shown
- Cause...
  - increased number of electron-hole pairs in doped silicon
  - recombination hindered by electric fields (powered transistors)
- ...and Effect
  - Cumulative Effects (Displacement, TID)
  - Single Event Effects
    - destructive errors (SEL, SEBO, SEGR)
    - non-destructive errors (SEU, SET)





(c)Xilinx

Figure 3a, 3b – Middle oxide thickness Virtex-4 transistor used in triple-oxide process and with highlighted portions of the transistors

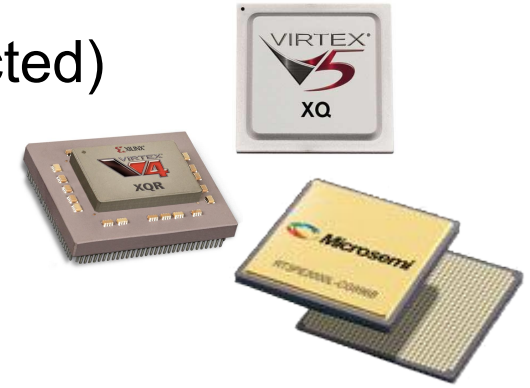


SRAM Radiation Effects ctd.

- Mitigation

MATERIAL

- Displacement, TID, SEL, SEBO, SEGR: extensive Materials, Shielding, Care
  - Xilinx Space-Grade XQV series (export restricted)
  - Microsemi Space-Grade RT series
  - TID: Reconfiguration/Scrubbing



LOGIC

- SEU, SET: Spatial and Temporal Sampling
  - CRC+ECC hardware circuits (Xilinx, Altera)
  - dynamic reconfiguration - Scrubbing (Xilinx, Altera)
  - Triple-Chip + Voter-Chip
  - DMR/TMR in design

Outline

- Quick Introduction to SRAM Radiation Effects
- **SEE-aware circuit design**
- Hamming FSM Design
- Beamtest Results
- Xilinx Blind Scrubbing - Trick or Treat

SEE-aware circuit design

- automated (TMR)
  - design time ✓, tool costs ✗, area consumption ✗ (up to 6x)
  - no designer expertise about FT required
  - tools available (→ ALL working on netlist):
    - Precision Rad-Tolerant Tool - Mentor Graphics
    - Partial TMR Tool (BLTmr) - Mike Wirthlin (BYU)
    - XTMR Tool - Xilinx
    - STAR Tool - Luca Sterpone (Politecnico di Torino)
- manually (→ working on RTL, avoiding TMR)
  - design time ✗, expertise required ✗, area consumption ✓
  - best optimization and fault tolerance results (designers know about their critical code patterns)

Fault-Tolerance Techniques

- Spacial Redundancy
  - synchronous data sampling at multiple routes
  - mitigates SEU
  - Dual/Triple Module Redundancy (DMR/TMR)
  - Error Detection and Correction Codes (EDAC) e.g. Hamming
- Temporal Redundancy
  - time shifted data sampling
  - mitigates SET
- combination of spacial and temporal redundancy
  - time shifted data sampling at multiple routes
  - mitigates SEU and SET

Fault-Tolerance Techniques ctd.

- Spacial Redundancy
  - multiple input data (TMR), single clock, majority voter
  - larger area-overhead than Temporal Redundancy
  - no different clock phases
  - no timing issues (just voter)
  - mitigates
    - SEU on data_x
  - susceptible to
    - SET on clk
    - DEU/MEU on data_x

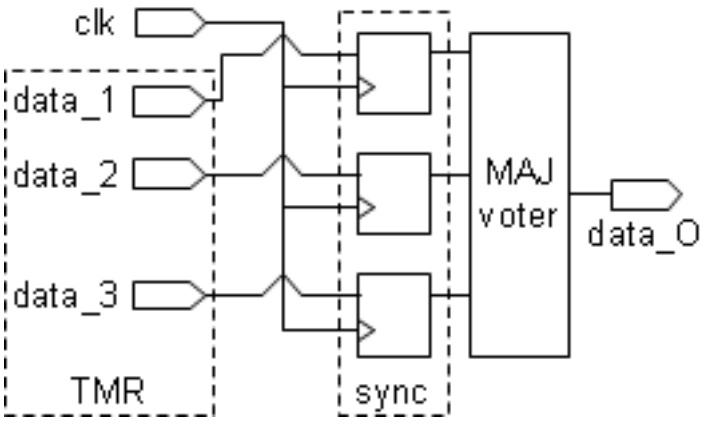
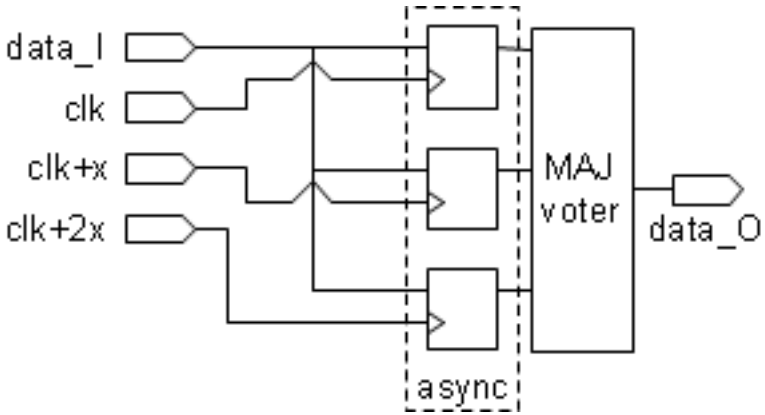
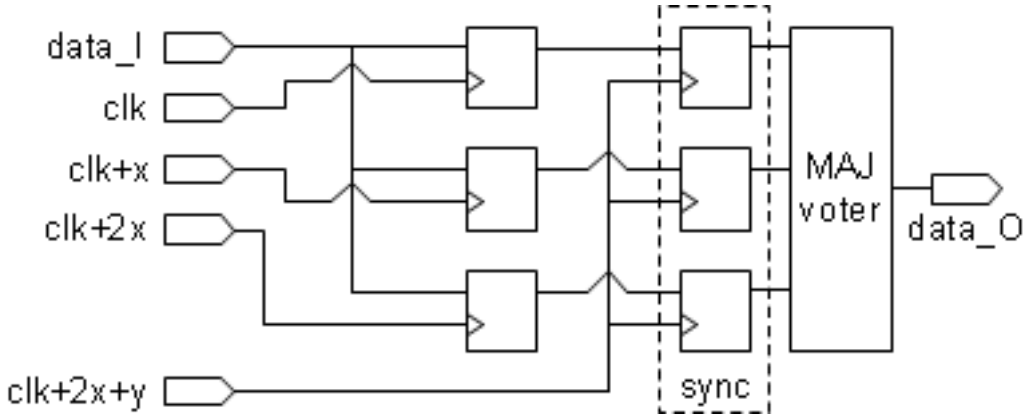


figure based on [Mav02] Mavis, „Single Event Transient Phenomena - Challenges and Solutions“, MRQW 2002

Fault-Tolerance Techniques ctd.

- Temporal Redundancy
  - single data, multiple delayed clocks, majority voter
  - increased execution time
  - less additional resources required
  - mitigates
    - SET on data_I
  - susceptible to
    - SEU/DEU/MEU on data_I

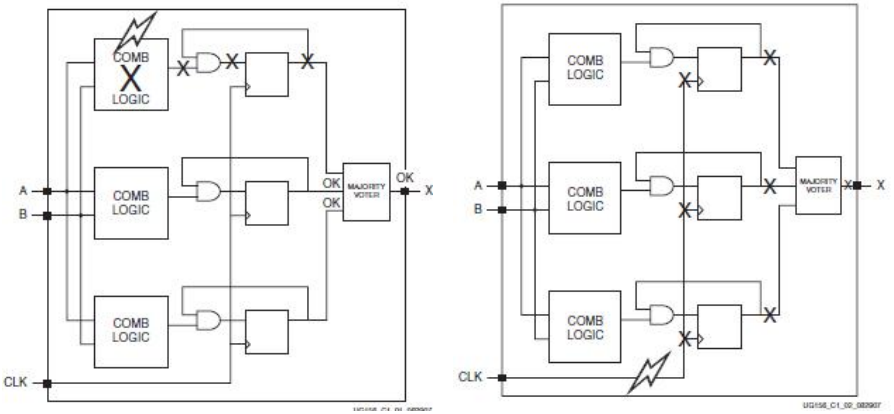




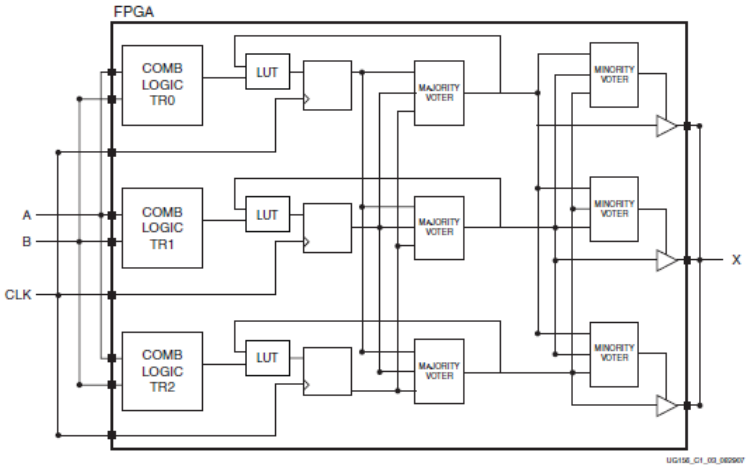
figures based on [Mav02] Mavis, „Single Event Transient Phenomena - Challenges and Solutions“, MRQW 2002

Xilinx TMR Tool (XTMR) [ug156]

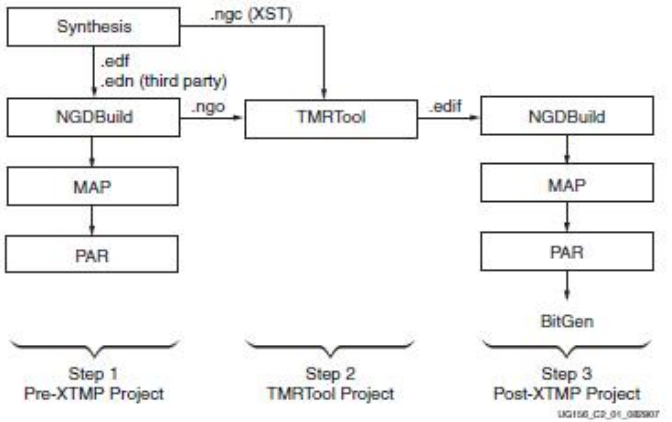
- conventional TMR:



- XTMR:
  - triplicated inputs, outputs, voters, combinational logic, routing
  - feedback logic



- works on EDIF, design flow:
- BRAM scrubbing with special macro [xapp962]
- but: forces ISE 9



Outline

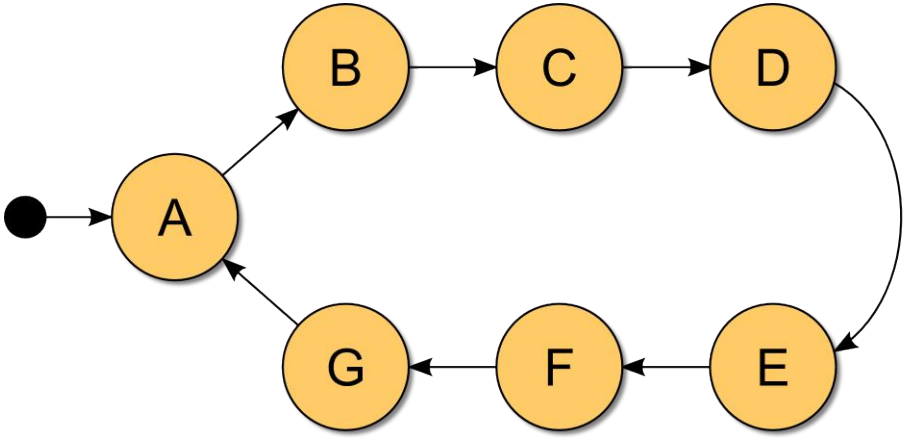
- Quick Introduction to SRAM Radiation Effects
- SEE-aware circuit design
- **Hamming FSM Design**
- Beamtest Results
- Xilinx Blind Scrubbing - Trick or Treat

Securing FSMs

- FSM = Finite State Machine
  - quintuple $(\Sigma, S, s_0, \delta, F)$
    - Σ - input alphabet (finite, non-empty set of symbols)
    - S - finite, non-empty set of states
    - s_0 - initial state, element of S
    - δ - state-transition function: $\delta: S \times \Sigma \rightarrow S$
    - F - set of final states (possibly empty, subset of S)

Securing FSMs ctd.

- example: ASCII 0 to 6 loop counter

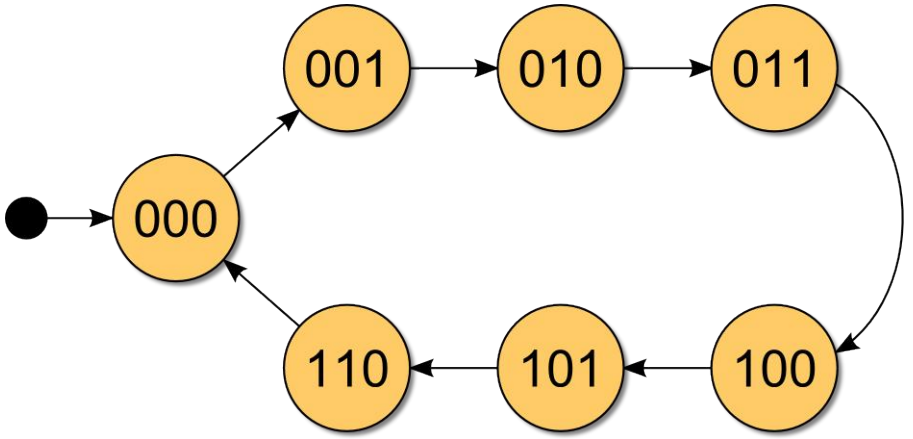



```
A ▶ B
B ▶ C
C ▶ D
D ▶ E
E ▶ F
F ▶ G
G ▶ A
```

- Σ: {'0','1'}
- S: {A,B,C,D,E,F,G}
- s_0: {A}
- δ: see above
- F: { }

Securing FSMs ctd.

- example: ASCII 0 to 6 loop counter



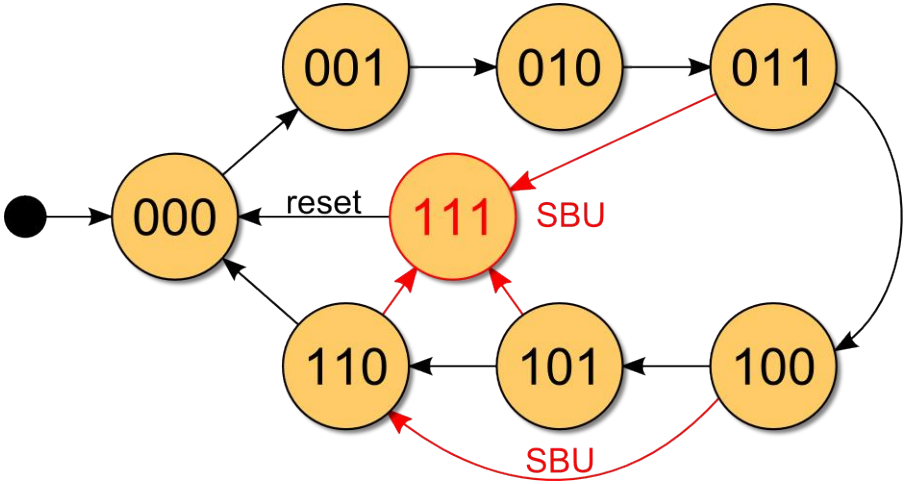

```
000 ▶ 001
001 ▶ 010
010 ▶ 011
011 ▶ 100
100 ▶ 101
101 ▶ 110
110 ▶ 000
others ▶ 000
```

- in VHDL:

```
case current_state is
     when "000" =>
          rs232_data_to_pc <= "00110000"; -- ASCII: 0
          next_state <= "001";
     when "001" =>
          rs232_data_to_pc <= "00110001"; -- ASCII: 1
          next_state <= "010";
[...]
     when "110" =>
          rs232_data_to_pc <= "00110110"; -- ASCII: 6
          next_state <= "000";
     when others =>
          rs232_data_to_pc <= "00110000"; -- ASCII: 0
          next_state <= "000";
end case;
```

Securing FSMs ctd.

- Problem: FSM SEE susceptibility

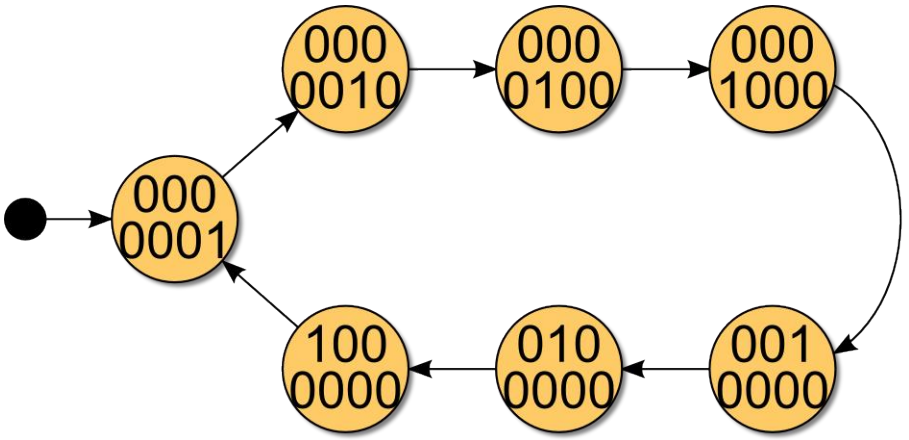



```
000 ▶ 001
001 ▶ 010
010 ▶ 011
011 ▶ 100
100 ▶ 101
101 ▶ 110
110 ▶ 000
others ▶ 000
```

- SBU/MBU:
  - illegal transitions ▶ faulty cycle and data
  - illegal states ▶ reset

Securing FSMs ctd.

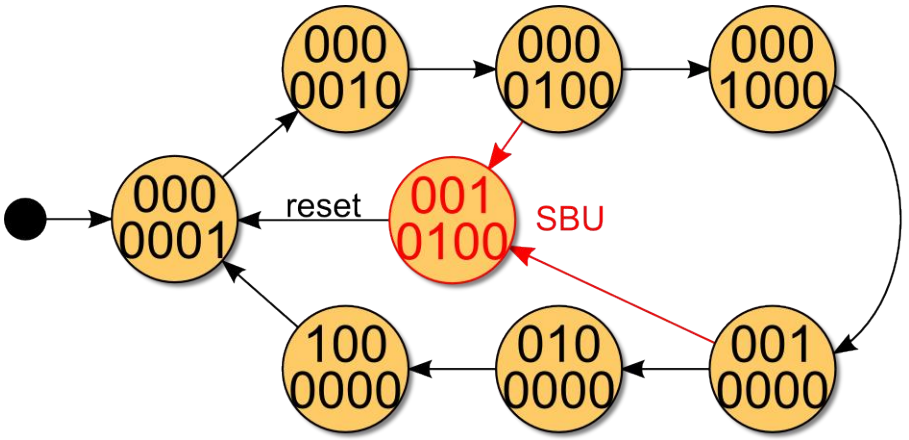
- avoiding illegal transitions: One-Hot FSM




```
0000001 ▶ 0000010
0000010 ▶ 0000100
0000100 ▶ 0001000
0001000 ▶ 0010000
0010000 ▶ 0100000
0100000 ▶ 1000000
1000000 ▶ 0000001
 others ▶ 0000001
```

Securing FSMs ctd.

- avoiding illegal transitions: One-Hot FSM




```
0000001 ▶ 0000010
0000010 ▶ 0000100
0000100 ▶ 0001000
0001000 ▶ 0010000
0010000 ▶ 0100000
0100000 ▶ 1000000
1000000 ▶ 0000001
 others ▶ 0000001
```

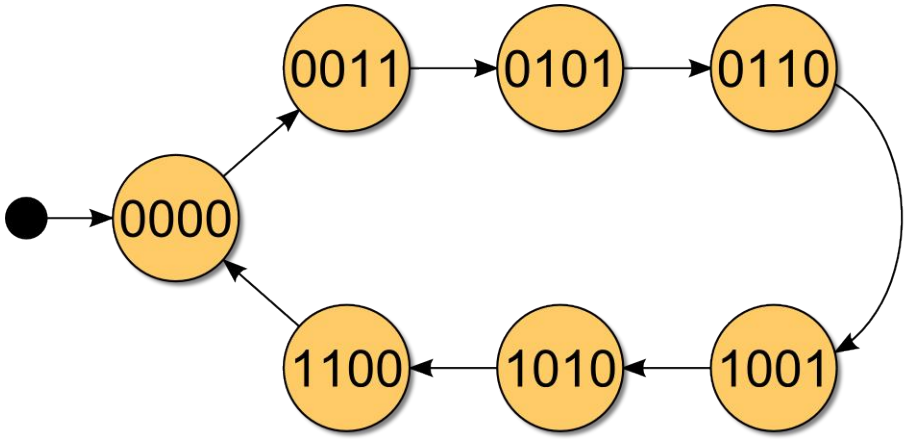
- SBU/MBU:
  - no illegal transitions
  - illegal states ▶ reset

Securing FSMs ctd.

- Hamming code
  - linear error-correcting code
  - data fault detection possible
  - data fault correction possible
  - Hamming(4,3) = 4 bit code words with 3 data bits available
  - Hamming Distance d = minimum number of differing bits
    - Example: '0011' and '0101' have d=2

Securing FSMs ctd.

- Hamming(4,3)

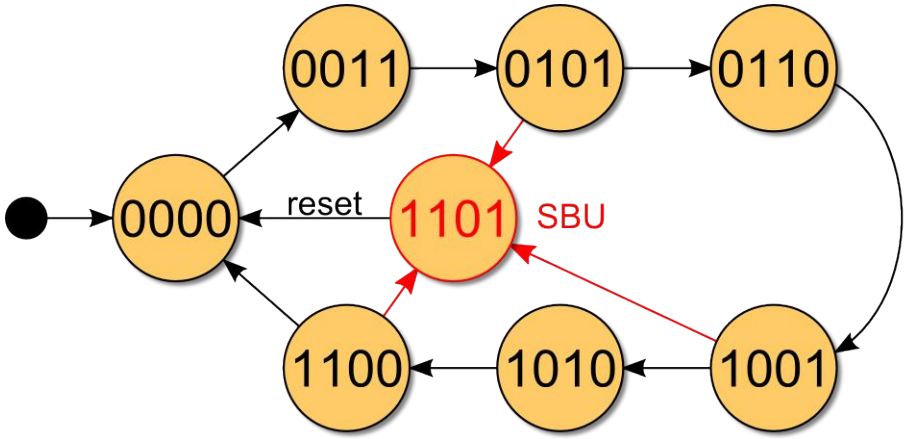



```
0000 ► 0011
0011 ► 0101
0101 ► 0110
0110 ► 1001
1001 ► 1010
1010 ► 1100
1100 ► 0000
others ► 0000
```

- Hamming Distance d=2 ► no faulty transitions (SBU)

Securing FSMs ctd.

- Hamming(4,3)

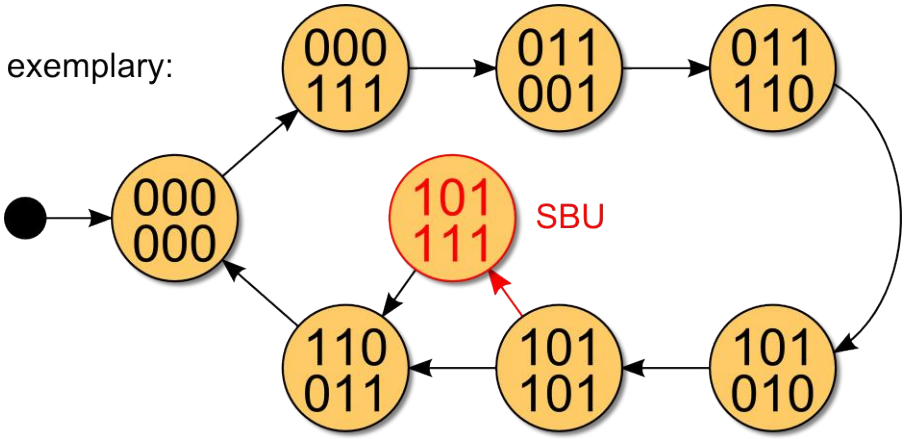



```
0000 ► 0011
0011 ► 0101
0101 ► 0110
0110 ► 1001
1001 ► 1010
1010 ► 1100
1100 ► 0000
others ► 0000
```

- Hamming Distance d=2 ► no faulty transitions (SBU)
- data fault detection ► reset (error state undecidable)
- no data fault correction

Securing FSMs ctd.

- Hamming(6,3)



- Hamming Distance d=3 ► no faulty transitions (SBU)
- SBU data fault
  - detection
  - correction

```
100000,010000,001000,000100,000010,000001,000000 ► 000111
100111,010111,001111,000011,000101,000110,000111 ► 011001
111001,001001,010001,011101,011011,011000,011001 ► 011110
111110,001110,010110,011010,011100,011111,011110 ► 101010
001010,111010,100010,101110,101000,101011,101010 ► 101101
001101,111101,100101,101001,101111,101100,101101 ► 110011
010011,100011,111011,110111,110001,110010,110011 ► 000000
                                          others ► 000000
```

Securing FSMs ctd.

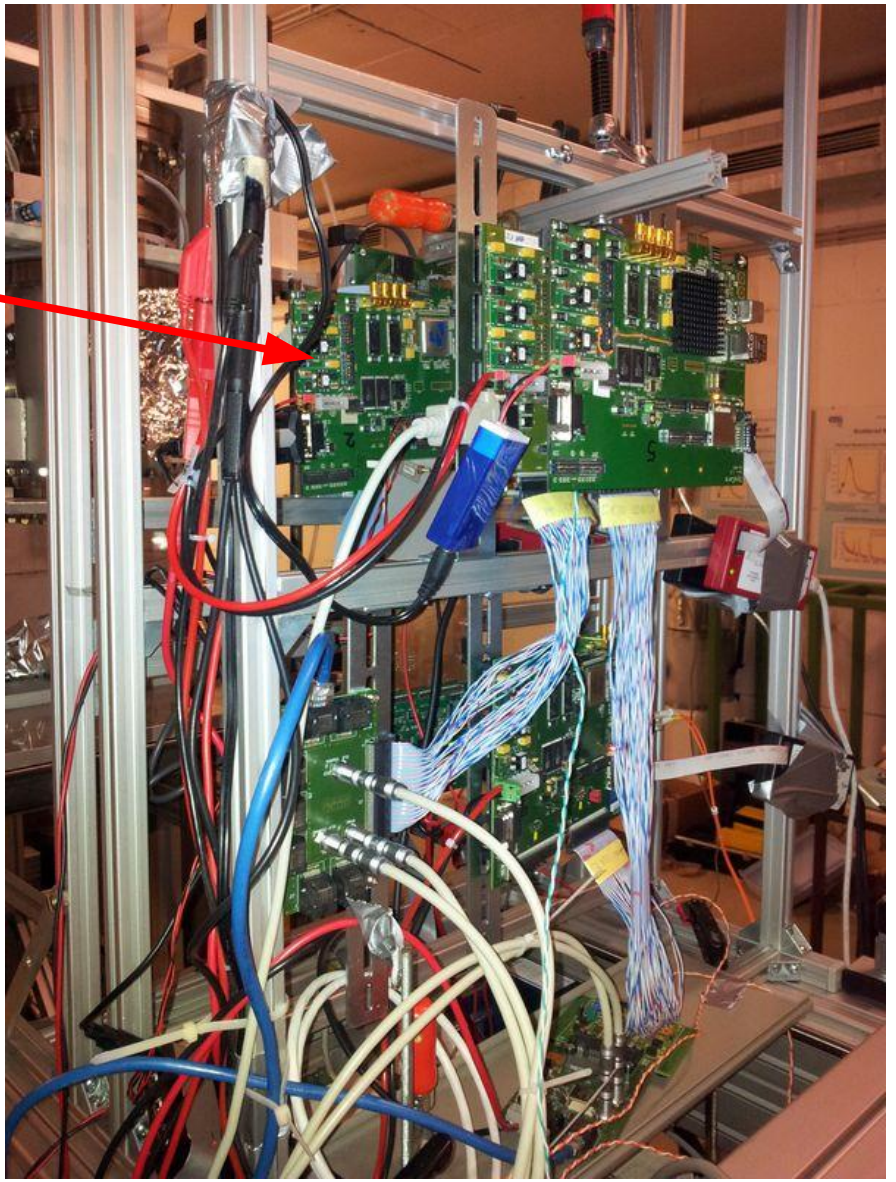
- Lessons Learned
  - tested: synthesis flip-flop requirements:
    - non-hardened: 64 ► SBU reset + wrong cycles and data
    - Hamming(4,3): 65 ► SBU reset
    - Hamming(6,3): 67 ► no SBU effects
    - (One-Hot: 68 ► SBU reset)
  - critical flip-flop requirements increase only with size of the input state bit-vector
  - of course, static LUTs explode because of the increased number of possible terms, but they are mitigated by configuration Scrubbing

Outline

- Quick Introduction to SRAM Radiation Effects
- SEE-aware circuit design
- Hamming FSM Design
- **Beamtest Results**
- Xilinx Blind Scrubbing - Trick or Treat

Beamtest: Setup

- SysCore ROC v1
  - Xilinx Virtex 4 FX20 FF672
  - Actel ProASIC 3 Scrubbing Controller
  - Blind Scrubbing
  - Flux monitoring by FPGA configuration readback:
    1) beam definition
    2) 1h readback
    3) design test
- beam: 2,4 GeV p+



Hamming FSM d=3 (without TMR)

```
constant s0 : std_logic_vector(14 downto 0) := "000000000000000";
constant s1 : std_logic_vector(14 downto 0) := "110100010000001";
constant s2 : std_logic_vector(14 downto 0) := "010100010000010";
case current_state is
     when s0 => next_state <= s1;
     when s1 => next_state <= s2;
     when s2 => next_state <= s0;
     when others => next_state <= s0;
end case;
```

Hamming FSM d=3 + dummy states (without TMR)

```
constant s0 : std_logic_vector(14 downto 0) := "000000000000000";
constant s0h0 : std_logic_vector(14 downto 0) := "100000000000000";
constant s0h1 : std_logic_vector(14 downto 0) := "010000000000000";
[...]
constant s1 : std_logic_vector(14 downto 0) := "110100010000001";
constant s1h0 : std_logic_vector(14 downto 0) := "010100010000001";
constant s1h1 : std_logic_vector(14 downto 0) := "100100010000001";
[...]
constant s2 : std_logic_vector(14 downto 0) := "010100010000010";
constant s2h0 : std_logic_vector(14 downto 0) := "110100010000010";
constant s2h1 : std_logic_vector(14 downto 0) := "000100010000010";
[...]
case current_state is
     when s0 | s0h0 | s0h1 | s0h2 | s0h3 | s0h4 | s0h5 | s0h6 | s0h7 | s0h8 | s0h9 | s0h10
          | s0h11 | s0h12 | s0h13 | s0h14 => next_state <= s1;
     when s1 | s1h0 | s1h1 | s1h2 | s1h3 | s1h4 | s1h5 | s1h6 | s1h7 | s1h8 | s1h9 | s1h10
          | s1h11 | s1h12 | s1h13 | s1h14 => next_state <= s2;
     when s2 | s2h0 | s2h1 | s2h2 | s2h3 | s2h4 | s2h5 | s2h6 | s2h7 | s2h8 | s2h9 | s2h10
          | s2h11 | s2h12 | s2h13 | s2h14 => next_state <= s0;
     when others => next_state <= s0;
end case;
rs232_data_to_pc <= current_state (exemplary; all hamming bits previously eliminated)
if rising_edge(clk100) then
     current_state <= next_state;
end if;
```

Beamtest: Test Designs

- all designs: Hamming FSM d=3, single DCM
- differences:
  - A: nearly TMR + FSM dummy transitions — chip usage: 84%
  - B: nearly TMR — chip usage: 47%
  - C: no additional fault tolerance — chip usage: 10%
  - D: TMR — chip usage: 47%
  - E: TMR + FSM dummy transitions — chip usage: 84%
- Hamming FSM d=3: 2048 states
- dummy transitions: 32768 theoretical transitions with d=1
- of course: TMR unused in FSM process

Beamtest: Results - Run Data

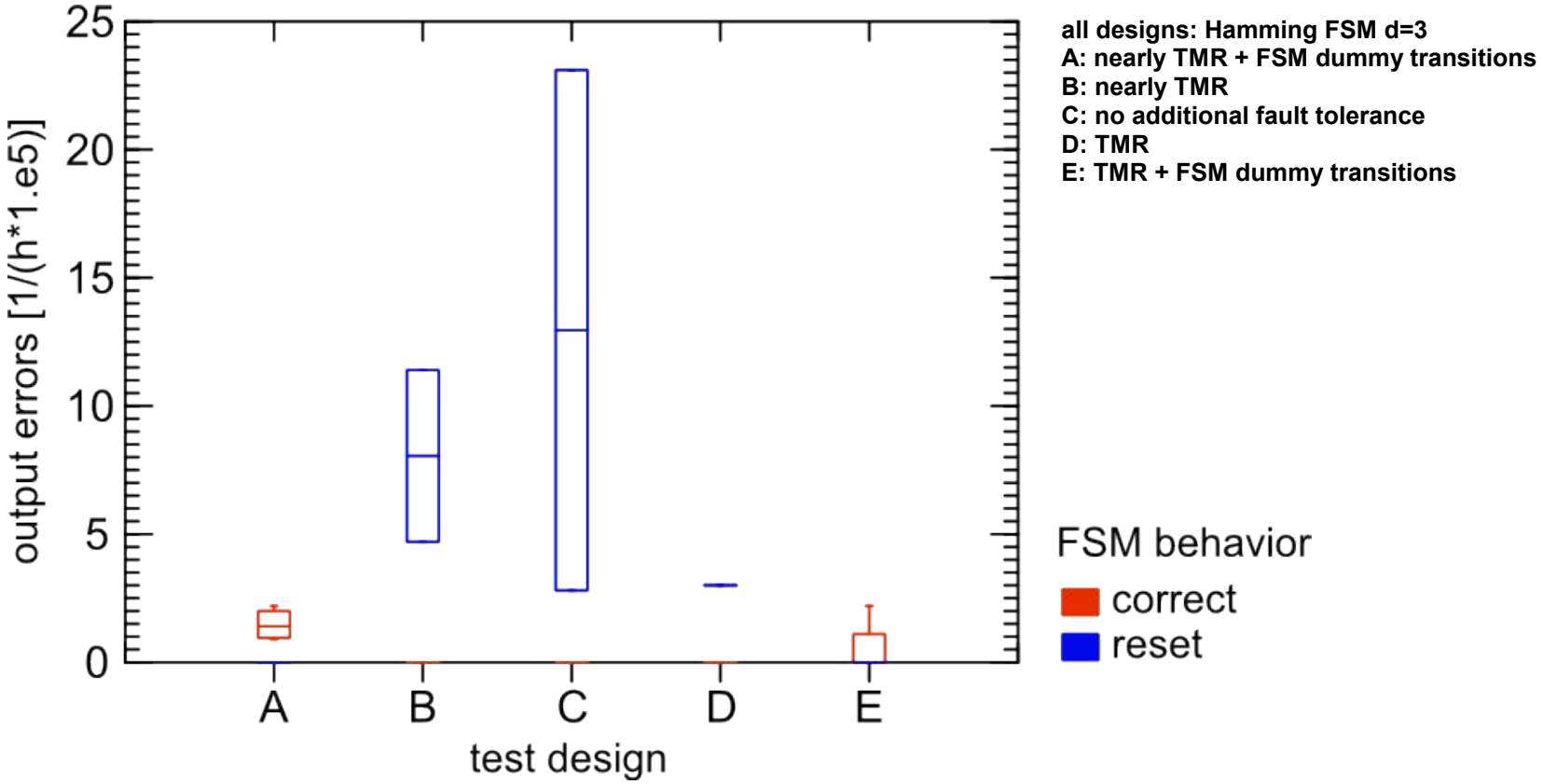
Design	Slices [%]	Flux [1/(s*cm²)]	Time [min]	Errors with FSM reset	Errors with no FSM reset	Errors [1/h] with FSM reset	Errors [1/h] with no FSM reset	Errors [1/(h*1.e5)] with FSM reset	Errors [1/(h*1.e5)] with no FSM reset
A	84	2,67E+05	60	0,0	4,0	0,0	4,8	0,0	1,8
A	84	2,67E+05	60	0,0	2,0	0,0	2,4	0,0	0,9
A	84	2,67E+05	53	0,0	2,0	0,0	2,7	0,0	1,0
A	84	2,67E+05	240	0,0	20,0	0,0	6,0	0,0	2,2
B	47	2,67E+05	21	5,0	0,0	30,4	0,0	11,4	0,0
B	47	1,46E+05	223	12,0	0,0	6,9	0,0	4,7	0,0
C	10	8,61E+03	302	1,0	0,0	2,0	0,0	23,1	0,0
C	10	4,82E+05	354	8,0	0,0	13,6	0,0	2,8	0,0
D	47	1,46E+05	147	5,0	0,0	4,3	0,0	3,0	0,0
E	84	1,72E+04	59	0,0	0,0	0,0	0,0	0,0	0,0
E	84	2,77E+04	95	0,0	0,0	0,0	0,0	0,0	0,0
E	84	2,37E+04	11	0,0	0,0	0,0	0,0	0,0	0,0
E	84	2,15E+05	61	0,0	4,0	0,0	4,7	0,0	2,2

normalized to:
- chip size (100%)
- time (1h)

normalized to:
- chip size (100%)
- time (1h)
- flux (100000p+)

Beamtest: Results - Boxplot

- data normalized to chip size (100%), time (1h), flux (1E5)



- Scrubbing + TMR + Hamming FSM perform only together
- SEFI rate reduced from 13 to 0.5 SEFI/h (1/26)

Outline

- Quick Introduction to SRAM Radiation Effects
- SEE-aware circuit design
- Hamming FSM Design
- Beamtest Results
- **Xilinx Blind Scrubbing - Trick or Treat**

Xilinx Blind Scrubbing - Trick or Treat

- Design C: Scrubbing repaired the broken FSM route immediately before next iteration:

```
FSM internal state:
x0427, x0428, x0429 routing upset - reset to x0001, x0002, x0003, ... continued without error

log:
20111127-130115: lost sync (pc expected: x0429, fpga sent: xFF00); fifo(0..5): xFF0000010002...;
20111127-130116: lost sync (pc expected: x0429, fpga sent: x0001); fifo(0..5): x000100020003...;
20111127-130116: sync at x0001
```

Xilinx Blind Scrubbing - Trick or Treat ctd.

- Design E: Scrubbing tried to repair a broken route between FSM (continues correctly) and RS232 multiple times without success ► multiple Scrubbing cycles required to repair some configuration registers:

```
FSM internal state:
x054D, x054E, x054F (10101001111) correct, x0550, x0551..x054E correct, x054F (10101001111) correct again, x0550,
x0551..x054E correct, ... multiple iterations before continued without error (finally repaired by Scrubbing)

RS232 state output:
x054D, x054E, x0547 (10101000111) wrong  , x0550, x0551..x054E correct, x0547 (10101000111) wrong again  , x0550,
x0551..x054E correct, ... multiple iterations before continued without error (finally repaired by Scrubbing)

log:
20111127-233742: lost sync (pc expected: x054F, fpga sent: x0547); fifo(0..5): x054705500551...;
20111127-233742: lost sync (pc expected: x054F, fpga sent: x0550); fifo(0..5): x055005510552...;
20111127-233742: sync at x0550
20111127-233742: lost sync (pc expected: x054F, fpga sent: x0547); fifo(0..5): x054705500551...;
20111127-233742: lost sync (pc expected: x054F, fpga sent: x0550); fifo(0..5): x055005510552...;
20111127-233742: sync at x0550
20111127-233743: lost sync (pc expected: x054F, fpga sent: x0547); fifo(0..5): x054705500551...;
20111127-233743: lost sync (pc expected: x054F, fpga sent: x0550); fifo(0..5): x055005510552...;
20111127-233743: sync at x0550
```

- solution: Selective Frame Scrubbing with Readback

Thank You!
Questions?