ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΔΠΜΣ Space Technologies, Applications and seRvices (STAR)

M806 Space Data Systems

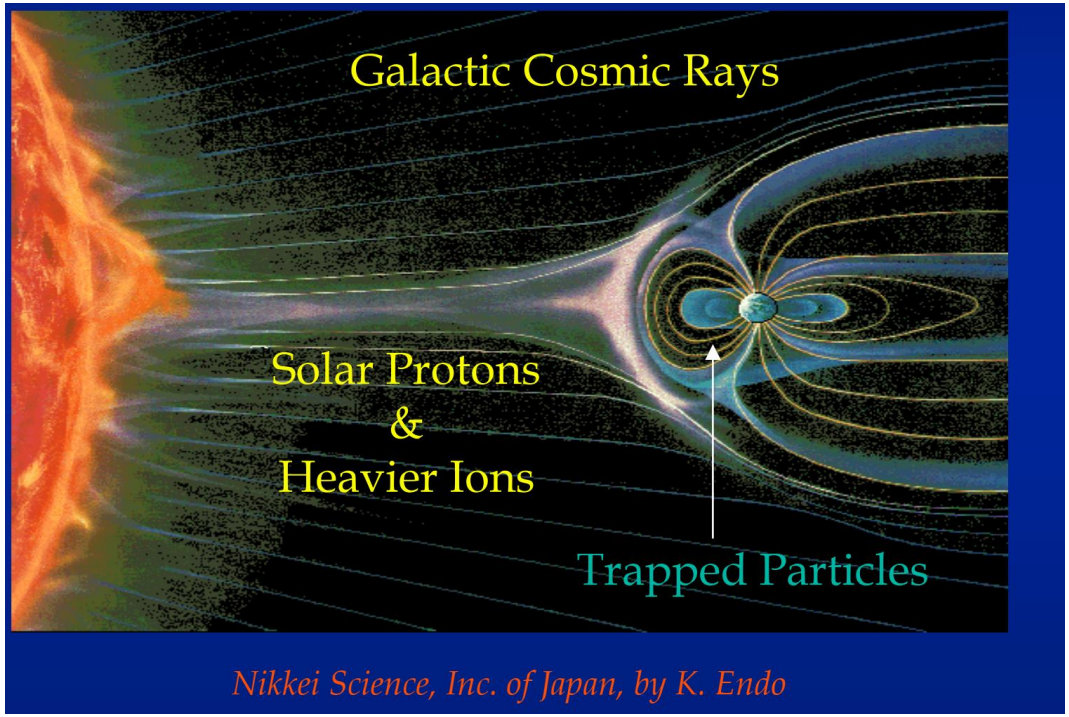
ΕΠΙΠΤΩΣΕΙΣ ΤΗΣ ΑΚΤΙΝΟΒΟΛΙΑΣ ΚΑΙ ΤΕΧΝΙΚΕΣ ΜΕΤΡΙΑΣΜΟΥ

Ακαδημαϊκό Έτος 2023-2024

Νεκτάριος Κρανίτης

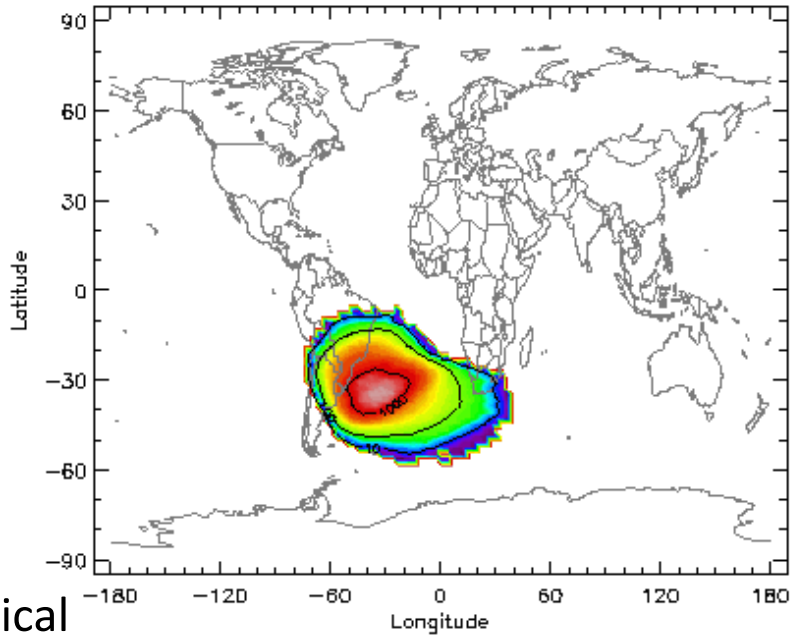
Space radiation environment

- Space radiation environment comprises a large range of energetic particles
- Energies range from several keV up to GeV and beyond (TeV for GCR)
- Three main sources contributing to a radiation environment:
  - Trapped particles
  - Solar Energetic Particles (SEPs)
  - Galactic and extra-galactic Gosmic rays (GCR)



Space radiation environment

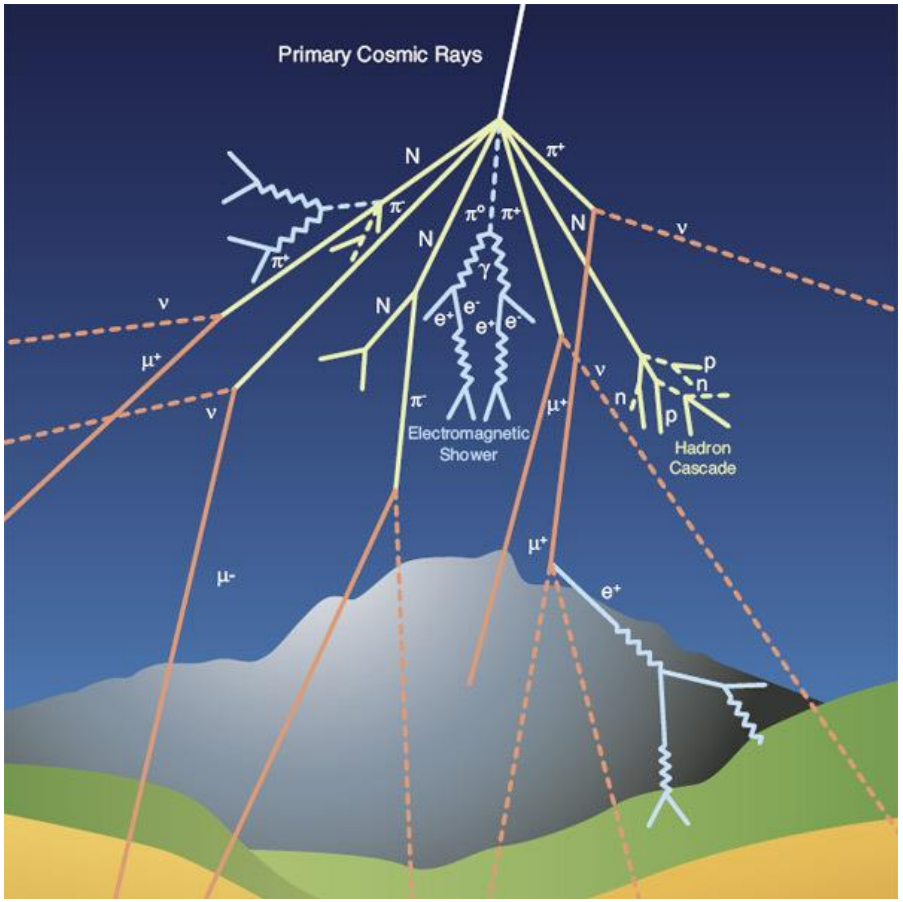
- **Trapped particles**:
  - Energetic electrons and ions are magnetically trapped in **Van Allen radiation belts**
  - Extend from 100 km to 65,000 km
  - Consist mainly of:
    - electrons up to a few MeV and
    - protons of up to several hundred MeV
  - Earth's magnetic field: not symmetrical
    - Leading to local distortions
    - South Atlantic Anomaly: <u>Spacecrafts passing this area are exposed to an increased level of radiation</u>



World map at 500 km altitude of the trapped proton (>10 MeV)

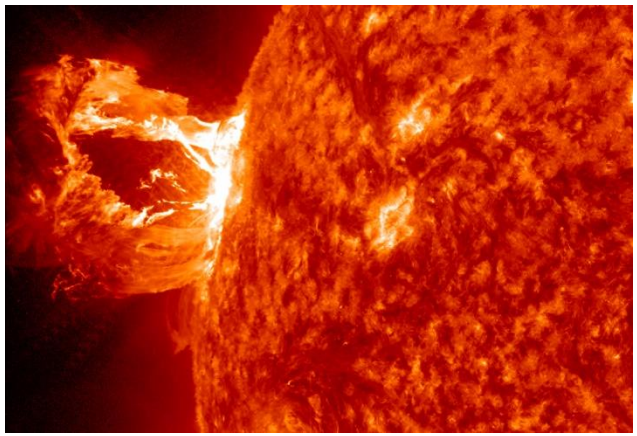
Space radiation environment

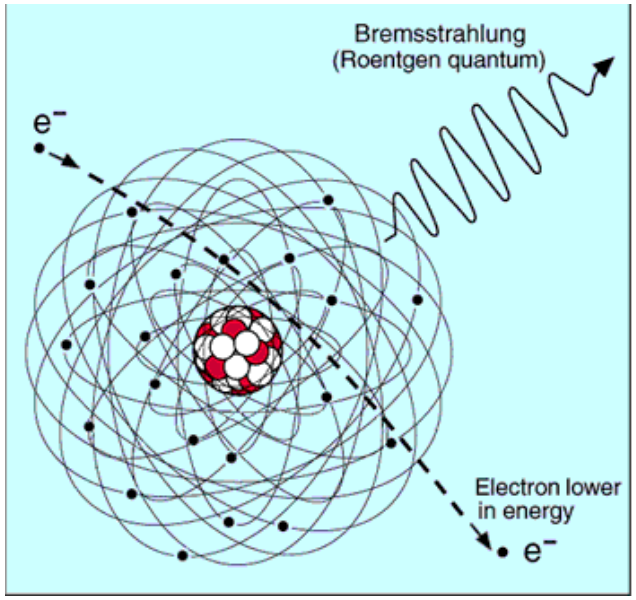
- **Galactic cosmic rays (GSR)**:
  - High-energy charged particles
  - Enter our solar system from outside
    - our own Milky Way galaxy or
    - from distant galaxies
  - Move at nearly the speed of light
  - Composed of protons, electrons, and fully ionized nuclei
  - TeV (300,000,000 TeV max. detected)



Space radiation environment

- **Solar energetic particles** during solar flares
- Solar flares: High-energy particles encountered in interplanetary space and close to Earth
  - In short bursts associated with other solar activity
  - Duration: a few hours up to several days
  - Consist of: protons, electrons, and heavy ions
  - Energy range: a few tens of keV to GeV and beyond
  - Secondary radiation is generated by interaction of energetic particles with materials
    - Bremsstrahlung: a high-energy **electromagnetic radiation** caused by deceleration of a charged particle in materials





Radiation Effects in ICs

- Radiation effects in ICs can be separated into:
  - **Cumulative effects**
    - Leading to progressive degradation of the FPGA characteristics
  - **Single Event Effects (SEEs)**
    - Including different types of events, destructive or not, induced by a single particle

Radiation Effects in ICs

Table 4-2: Summary of radiation effects parameters, units and examples

Effect	Parameter	Typical units	Examples	Particles
Total ionising dose (TID)	Ionising dose in material	grays (material) (Gy(material)) or rad(material) 1 Gy = 100 rad	Threshold voltage shift and leakage currents in CMOS, linear bipolar (note dose-rate sensitivity)	Electrons, protons, bremsstrahlung
Displacement damage	Displacement damage equivalent dose (total non-ionising dose) Equivalent fluence of 10 MeV protons or 1 MeV electrons	MeV/g cm^{-2}	All photonics, e.g. CCD transfer efficiency, optocoupler transfer ratio Reduction in solar cell efficiency	Protons, electrons, neutrons, ions
Single event effects from direct ionisation	Events per unit fluence from linear energy transfer (LET) spectra & cross-section versus LET	cm^2 versus $MeV \cdot cm^2/mg$	Memories, microprocessors. Soft errors, latch-up, burn-out, gate rupture, transients in op-amps, comparators.	Ions Z>1
Single event effects from nuclear reactions	Events per unit fluence from energy spectra & cross-section versus particle energy	cm^2 versus MeV	As above	Protons, neutrons, ions
Payload-specific radiation effects	Energy-loss spectra, charge-deposition spectra charging	counts s^{-1} MeV^{-1}	False count rates in detectors, false images in CCDs Gravity proof-masses	Protons, electrons, neutrons, ions, induced radioactivity (α, $\beta\pm$, γ)

Credit: ECSS-E-HB-20-40A

Cumulative effects

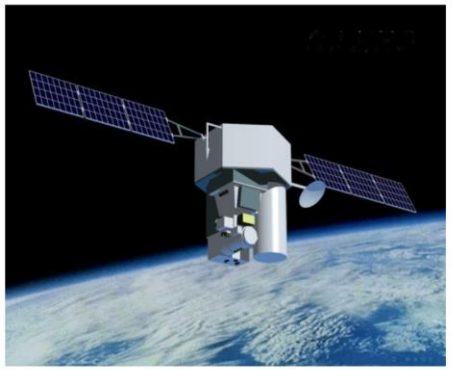
- Cumulative effects: Exposure to space radiation produces relatively stable, long-term changes in IC's characteristics that can result in parametric degradation and ultimately in functional failure
- **Total Ionizing Dose (TID):** The most common cumulative effect
  - Induced by ionization of ICs is caused by electrons, protons, and bremsstrahlung
  - Leads to a degradation due to increasing leakage currents and other effects
  - Accumulation such effects is referred to as TID expressed in Gray (Gy) or rad (100 rad = 1 Gy), with 1 Gy = 1 J/kg
  - Depending on spacecraft mission and orbit, and the device shielding, the received TID typically ranges from few krad(Si) to several 100's krad
  - Low-Earth-Orbit (LEO) spacecrafts accumulate < than 5 krad/year
  - Geostationary (GEO) platforms up to 15 krad/year

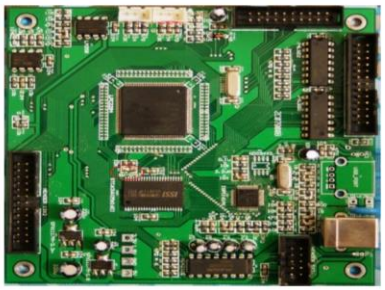
Table 1: **Radiation Characteristics**

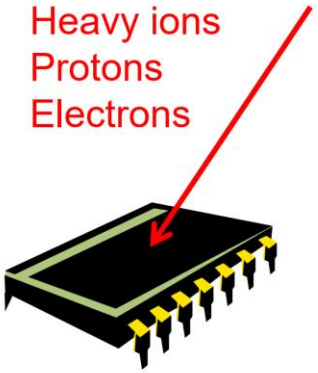
Symbol	Description	Min	Typ	Max	Units
TID	Total Ionizing Dose (GEO)	-	100	120	Krad (Si)
SEL	Single-Event Latch-Up Immunity(1)	-	80	-	MeV-cm^2/mg
SEU_{CRAM}	Single-Event Upset in Configuration RAM (GEO)(2)(3)	-	9.5e-9	-	Upset/bit/day
SEU_{BRAM}	Single-Event Upset in Block RAM (GEO)(2)(3)	-	2.3e-8	-	Upset/bit/day

Single Event Effects (SEEs)

- The charge deposited by a single ionizing particle can produce a wide range of effects
- Non-destructive (can be recovered, also called **soft-errors**)
  - Single-Event Upset (SEU)
  - Single-Event Transient (SET)
  - Single-Event Functional Interrupt (SEFI)
- Destructive and can lead to permanent damage
  - Single Event Latch-up (SEL)







Non-Destructive SEEs (Soft errors)

- **Single event transient (SET):** momentary voltage/current disturbance that may propagate through circuitry and eventually manifests as SEU once it reaches a latch or other memory elements
- **Single Event Upset (SEU):** Changes the state of a bistable element
  - Triggered by heavy ions and protons and results from ionization by a single energetic particle or the nuclear reaction products of an energetic proton
  - Ionization induces a current pulse in a p-n junction whose charge may exceed the critical charge that is required to change the logic state of the element
  - Result: The value of a memory bit can be flipped
  - SEU is most common effect for SRAM-based FPGAs, as affects configuration memory AND memory cells that are used as part of the user logic (flip-flops, embedded RAM).
  - **Multiple Cell Upset (MCU)** is the change of state of two or more logic cells induced by a single particle strike. The corrupted cells are usually, but not always, physically adjacent
  - **Multiple Bit Upset (MBU)** is a particular case of MCU when the corrupted cells are in the same word. MBU cannot be corrected by a simple (single-bit) error correction code
- **Single event functional interrupt (SEFI):** Soft error that causes the component to reset, lock-up, or otherwise malfunction
  - Two main types of SEFI depending on actions required to restore functionality: reset by software or by power cycling
  - Stored data can or cannot be lost

Destructive SEEs

- **Single Event Latchup (SEL):** Destructive SEE that can trigger parasitic thyristor structures (PNPN or NPNP) in a device
  - When occurs, a high current flows and if the power supply is maintained, the device can be destroyed by thermal effect
  - SEL signature is a self-sustainable current flowing in the low impedance path of the triggered parasitic thyristor structure whose gain increases with temperature
  - The only way to remove SEL is to power-reset the circuit
  - SEL are not to be mistaken with temporary current spikes resulting from SET induced logic conflicts or SEFI

Table 1: **Radiation Characteristics**

Symbol	Description	Min	Typ	Max	Units
TID	Total Ionizing Dose (GEO)	-	100	120	Krad (Si)
SEL	Single-Event Latch-Up Immunity(1)	-	80	-	MeV-cm^2/mg
SEU_{CRAM}	Single-Event Upset in Configuration RAM (GEO)(2)(3)	-	9.5e-9	-	Upset/bit/day
SEU_{BRAM}	Single-Event Upset in Block RAM (GEO)(2)(3)	-	2.3e-8	-	Upset/bit/day

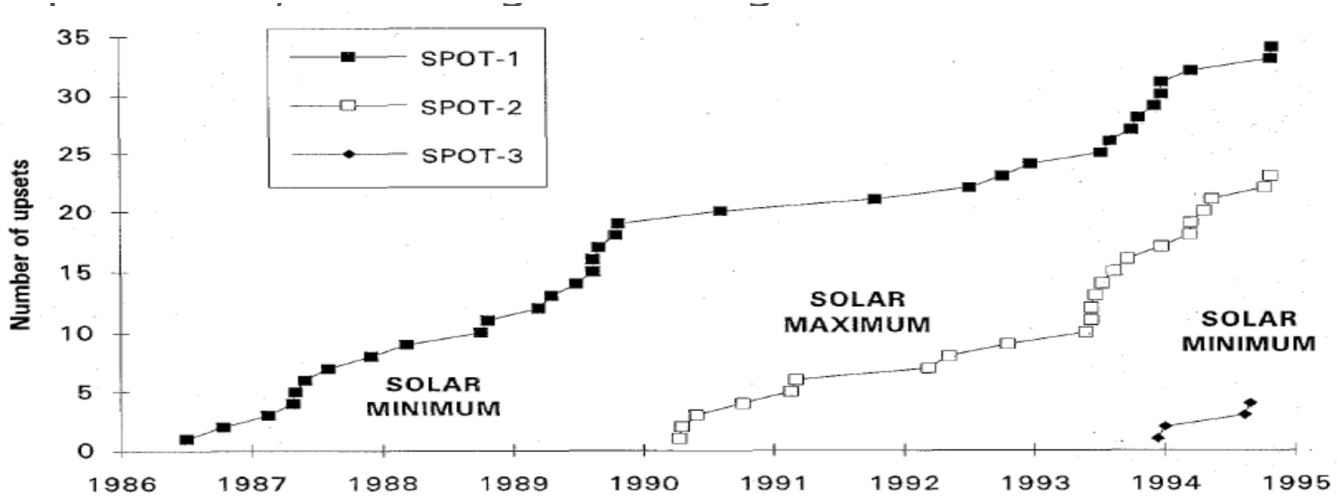
SEEs List

- Non-Exhaustive list, more in ECSS E-ST-10-12C

Single Event Upset (SEU)	corruption of the information stored in a memory element	Memories, latches in logic devices
Multiple Bit Upset (MBU)	several memory elements corrupted by a single strike	Memories, latches in logic devices
Single Event Functional Interrupt (SEFI)	corruption of a data path leading to loss of normal operation	Complex devices with built-in state machine/control sections
Single Hard Error (SHE)	unalterable change of state in a memory element	Memories, latches in logic devices
Single Event Transient (SET)	Impulse response of certain amplitude and duration	Analog and Mixed Signal circuits, Photonics
Single Event Disturb (SED)	Momentary corruption of the information stored in a bit	combinational logic, latches in logic devices
Single Event Latchup (SEL)	high-current conditions	CMOS, BiCMOS devices
Single Event Snapback (SESB)	high-current conditions	N-channel MOSFET, SOI devices
Single Event Burnout (SEB)	Destructive burnout due to high-current conditions	BJT, N-channel Power MOSFET
Single Event Gate Rupture (SEGR)	Rupture of gate dielectric due to high electrical field conditions	Power MOSFETs, Non-volatile NMOS structures, VLSIs, linear devices …

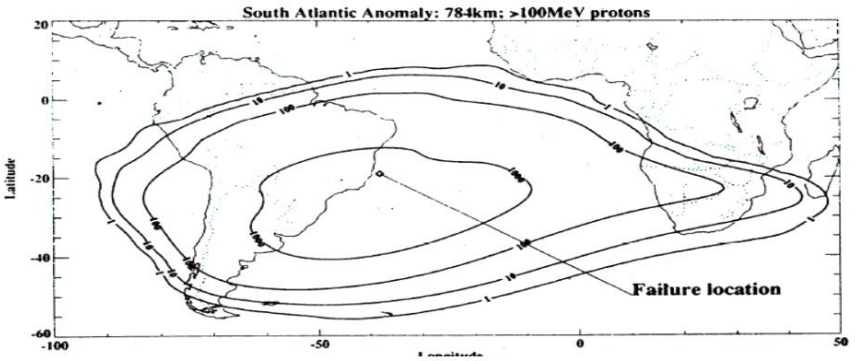
SEEs Historical facts

- **1975: [Binder]** first reported "single event effect" SEE anomalies
  - unexpected triggering in bipolar digital circuits due to cosmic rays
- **1978 – 1985:** SEUs in Pioneer 12 (Venus probe), in a 1024 bit PMOS shift register
- **1985-1995:** SEU example in the OBC of Spot1-2-3
  - Half of these SEUs lead to operational problems, including switching the satellite to safe mode

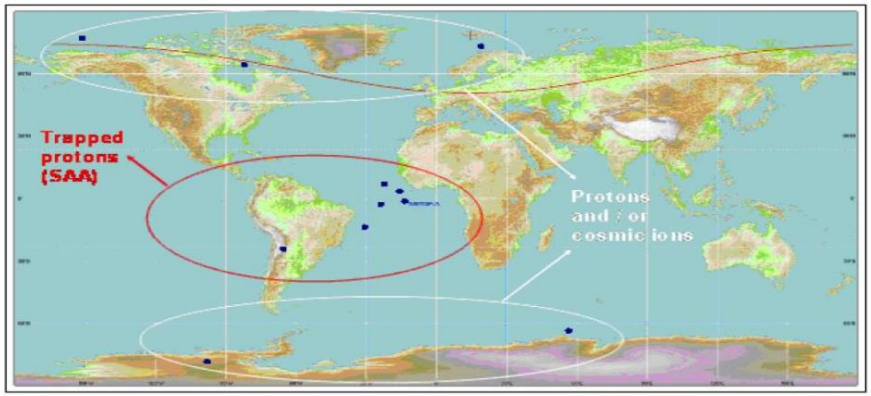


[R. Ecoffet, TNS 1995-2013]

Examples of SEE anomalies



PRARE on ERS-1. A 64-kbit CMOS SRAM failed due to SEL. PRARE only lasted operational for 5 days



IAISI on METOP. About half the SEUs are located in the SAA. Those upsets may be related to the emerging issue of high-Z recoils within the radiation hardened device (HX6228 memory).

[R. Ecoffet TNS 2012]

ISS. SETs on optocoupler 6N134 are held responsible of unexpected reset on a reset circuit that was designed only to be used for ground testing.

Credit: ESA

SEEs as function of technology

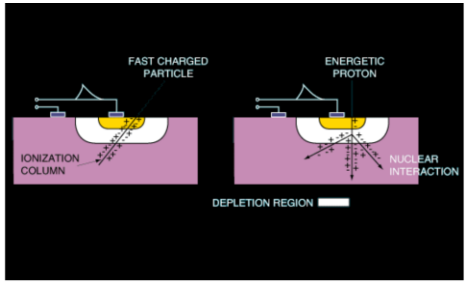
Table 4-1: Summary of single event effects (SEE) as a function of component technology and family

Technology	Family	Function	SEL, SESB	SEGR, SEDR	SEB	SEU	MCU/MBU	SEHE	SEFI	SET
			destructive SEE			Non-destructive SEE				
Power MOS				X	X					
CMOS, BiCMOS and SOI	Digital	SRAM	X			X	X	X		
		DRAM	X			X	X	X	X	
		FPGA	X			X	X	X	X	X
		Flash EEPROM	X			X		X	X	
		µP / µcontroller	X			X	X		X	X
	Mixed signal	ADC	X	X	X	X			X	X
		DAC	X	X	X	X			X	X
	Linear		X	X	X					X
Bipolar	Digital			X	X	X				X
	Linear			X	X	X				X

Credit: ECSS-E-HB-20-40A

SEEs Basic Mechanisms

- Energy Deposition
  - As an ionizing particle moves through a material, it loses energy due to interactions with atoms and molecules in material
  - Interactions cause ionization and excitation of atoms, leading to energy deposition along the particle's path



- SEE effects can be produced:
  - By direct ionization
    - E.g. heavy ions and low energy protons (for deep sub)
    - The heavy ion ionising effect is usually expressed by the **linear energy transfer (LET)**
    - LET (in MeV*cm^2/mg): ionising energy transferred along the ion path, normalized by the material volumetric mass density
    - Higher LET deposits more energy in a smaller volume of the semiconductor material
  - By indirect ionization (secondary particles)
    - From nuclear reactions or elastic collisions, as typically produced by protons
      - Most protons pass through the device with little effect
      - A few protons (~10^{-5}) cause nuclear reactions
    - In this case particle (e.g. proton) energy is usually expressed in MeV

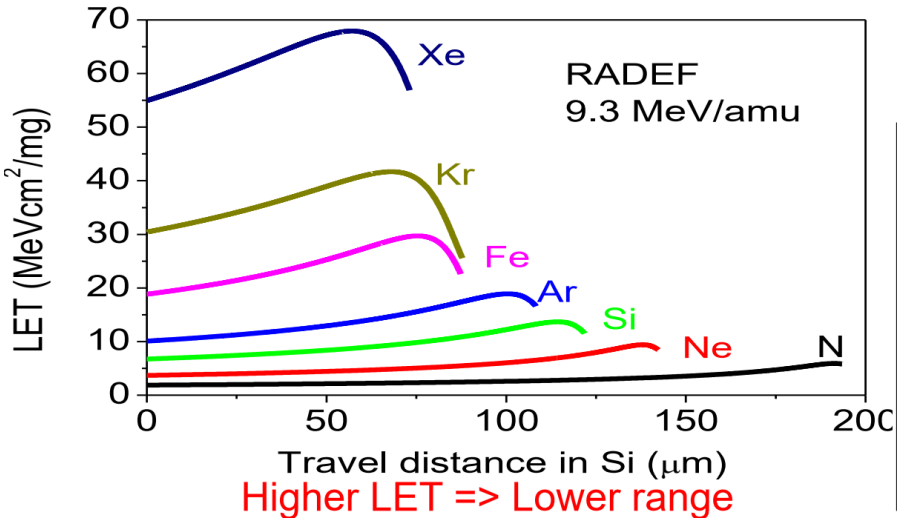
Cross-Section (σ)

- The sensitivity of an IC device to SEE is often expressed by the **cross-section σ** as a function of the ion LET or proton energy
- **σ**: the ratio of the number of observed single events by the particle fluence (particles per cm^2) received by the component under test

$$\sigma = \text{number of events/fluence}$$

- **Cross-section: probability an impinging particle provokes SEE**
- Cross-section unit: cm^2 for all types of circuits
  - For memory upsets it can be normalized by the number of bits
    - i.e. expressed in cm^2/bit
- When SEE are induced by direct ionization from ions, the cross-section is representative of a sensitive area
- For indirect ionization from high energy protons, the SEE cross-section also includes the probability of nuclear interaction, which renders its interpretation more complex

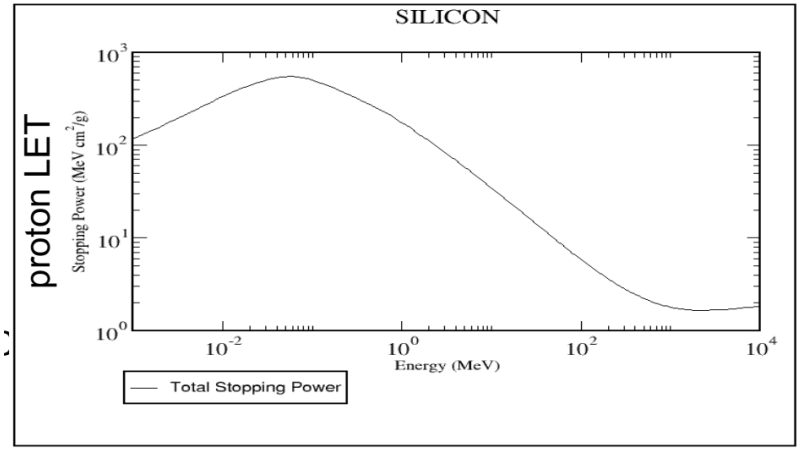
Linear Energy Transfer (LET)



Higher LET => Lower range

- LET is related to the penetration depth of the particle in the material
  - Higher LET particles may not penetrate as deeply as lower LET particles, but they cause more localized damage
  - Lower LET particles can travel further, potentially affecting deeper layers of a device, but with a lower likelihood of causing immediate SEEs

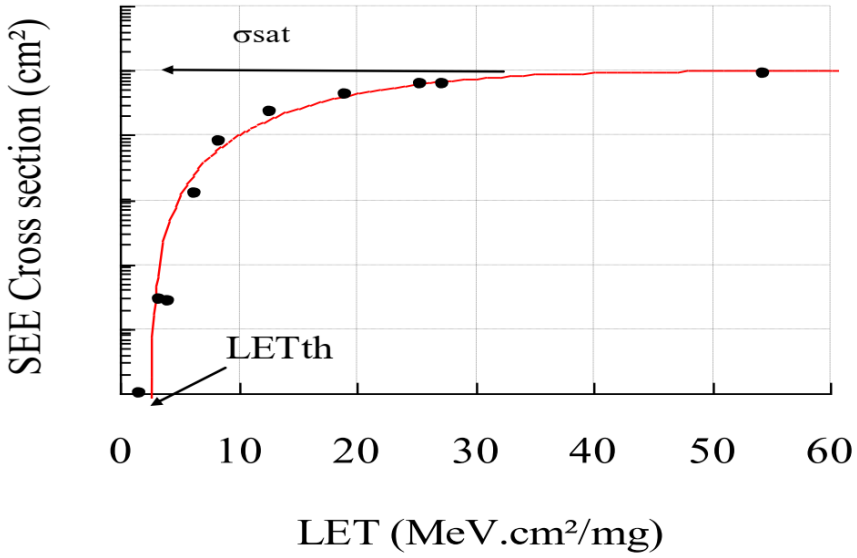
Higher Energy => Lower LET



- Lower energy particles have higher LET
  - Because they interact more frequently with atoms, depositing more energy per unit distance
  - Beyond a certain energy threshold, the LET decreases with increasing particle energy
- High-energy particles travel faster and interact less frequently with the medium, resulting in lower energy deposition per unit distance
  - Eg very high-energy protons and heavy ions will have a lower LET compared to their lower-energy counterparts

Credit: ESA

Cross-Section (σ) *vs LET*



The SEE cross section measures the probability for a SEE to occur

$$[cm^2] \longrightarrow \sigma = \frac{N_{events}}{Fluence} \longleftarrow [N_{particules}/cm^2]$$

Fit with Weibull (integral form)

$$\sigma = \sigma_{sat}\left(1 - \exp\left(\frac{LET - LET_{th}}{W}\right)^{S}\right)$$

W and S are fitting parameters

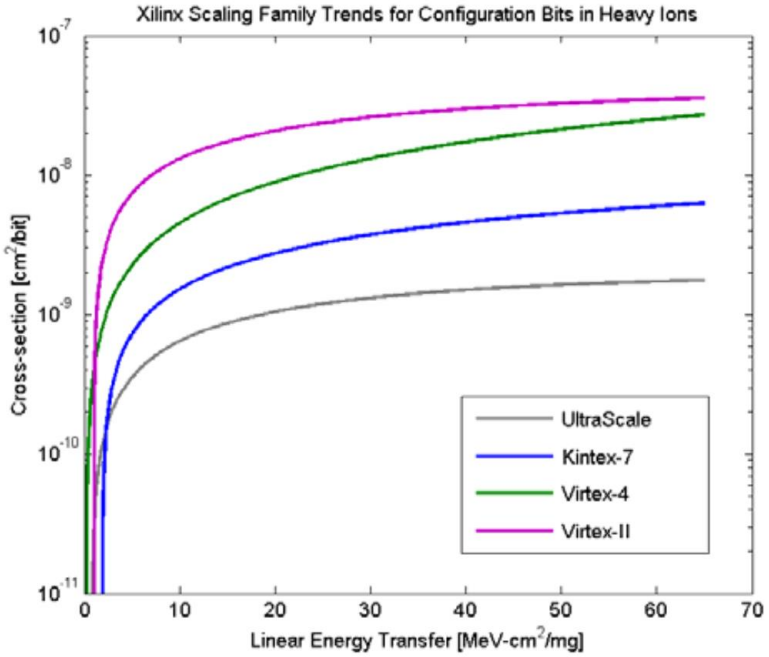
SEE cross-section is a crucial input for in-orbit SEE rate prediction.

Credit: ESA

Example: Xilinx FPGAs Configuration Memory

Single-Event Characterization of the 20 nm Xilinx Kintex UltraScale Field-Programmable Gate Array under Heavy Ion Irradiation

David S. Lee, *Graduate Student Member, IEEE,* Gregory R. Allen, *Member, IEEE,* Gary Swift, *Member, IEEE,* Matthew Cannon, *Student Member, IEEE,* Michael Wirthlin, *Senior Member, IEEE,* Jeffrey S. George, *Member, IEEE,* Rokutaro Koga, *Life Member, IEEE,* and Kangsen Huey



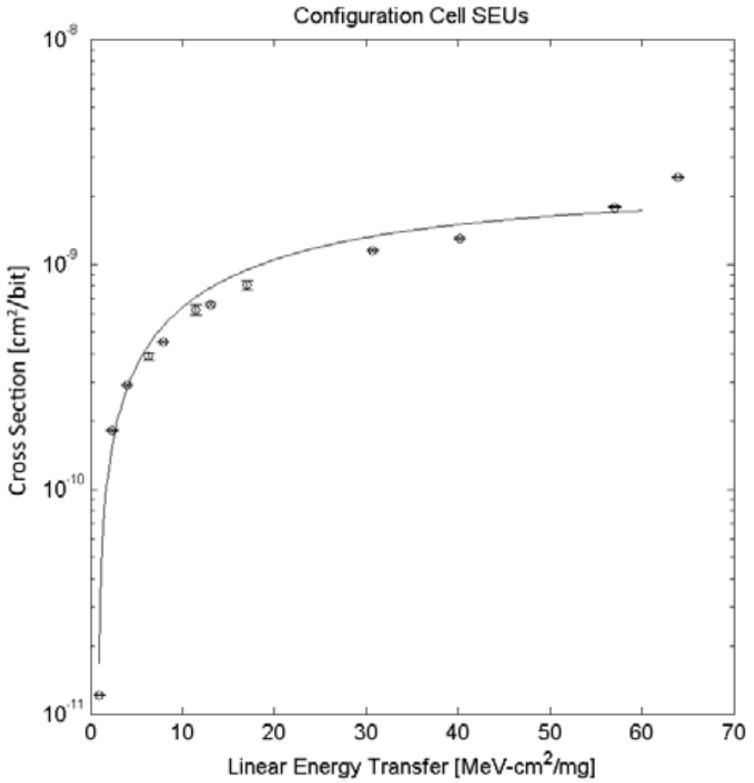


Fig. 4. Weibull curve for configuration memory cell upsets. L_{th}=0.8 MeV-cm^2/mg], σ_{sat}=2.0e-9 [cm^2/bit], W=27.0 [MeV-cm^2/mg], S=0.88. Error bars are small and, in some cases, smaller than the marker symbol.

Table 1: **Radiation Characteristics**

Symbol	Description	Min	Typ	Max	Units
TID	Total Ionizing Dose (GEO)	-	100	120	Krad (Si)
SEL	Single-Event Latch-Up Immunity(1)	-	80	-	MeV-cm^2/mg
SEU_{CRAM}	Single-Event Upset in Configuration RAM (GEO)(2)(3)	-	9.5e-9	-	Upset/bit/day
SEU_{BRAM}	Single-Event Upset in Block RAM (GEO)(2)(3)	-	2.3e-8	-	Upset/bit/day

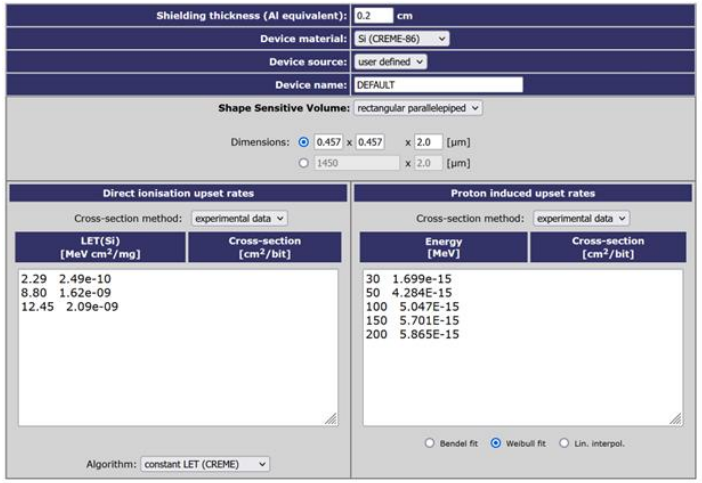
Example: SEU rate calculation with SPENVIS

The Space Environment Information System

- SPENVIS: SPace ENVironment Information System
  - WWW interface to models of space environment and its effects including galactic cosmic rays, solar energetic particles, natural radiation belts, plasmas, gases, meteoroids and debris
- Radiation environment modeling
  ->LET spectrum calculation
  - Inputs: spacecraft orbit details, radiation source and effects model
- SEU rate prediction
  - Inputs: Weibull parameters(S, W, Lth, σ_{sat}) or experimental data required for the specific device, shielding thickness, shape sensitive volume

Overview input
Particle spectra : AP-8 MAX trapped protons SAPPHIRE peak flux solar particles (H - U) ISO 15390 GCR particles (H - U) **Spacecraftshielding thickness (Al equivalent) :** 0.15 cm **Nr of devices:** 1 Remark: the minimum ion energy for the LET spectrum is set at 0.1 MeV/n

Device name	Heavy ion method	Proton method
#01: DEFAULT (user defined) Mat.: Si RPP: 0.46 x 0.46 x 2.00 (μm^3)	Using fit to measured cross sections SEU algorithm: CREME	Param. fit: S = 0.38 E_0 = 30.00 MeV W = 6.52 MeV σ_{lim} = 5.92E-15 cm^2/bit Using fit to measured cross sections



Segment averaged and total SEU rates							
		Mission total			Mission segment 1		
Device	Effect	(bit^{-1})	(bit^{-1} s^{-1})	(bit^{-1} day^{-1})	(bit^{-1})	(bit^{-1} s^{-1})	(bit^{-1} day^{-1})
DEFAULT	Direct ionization	2.1980E-02	2.3233E-10	2.0073E-05	2.1980E-02	2.3233E-10	2.0073E-05
	Proton induced ionization	5.4369E-03	5.7467E-11	4.9652E-06	5.4369E-03	5.7467E-11	4.9652E-06
	Total	2.7417E-02	2.8980E-10	2.5039E-05	2.7417E-02	2.8980E-10	2.5039E-05

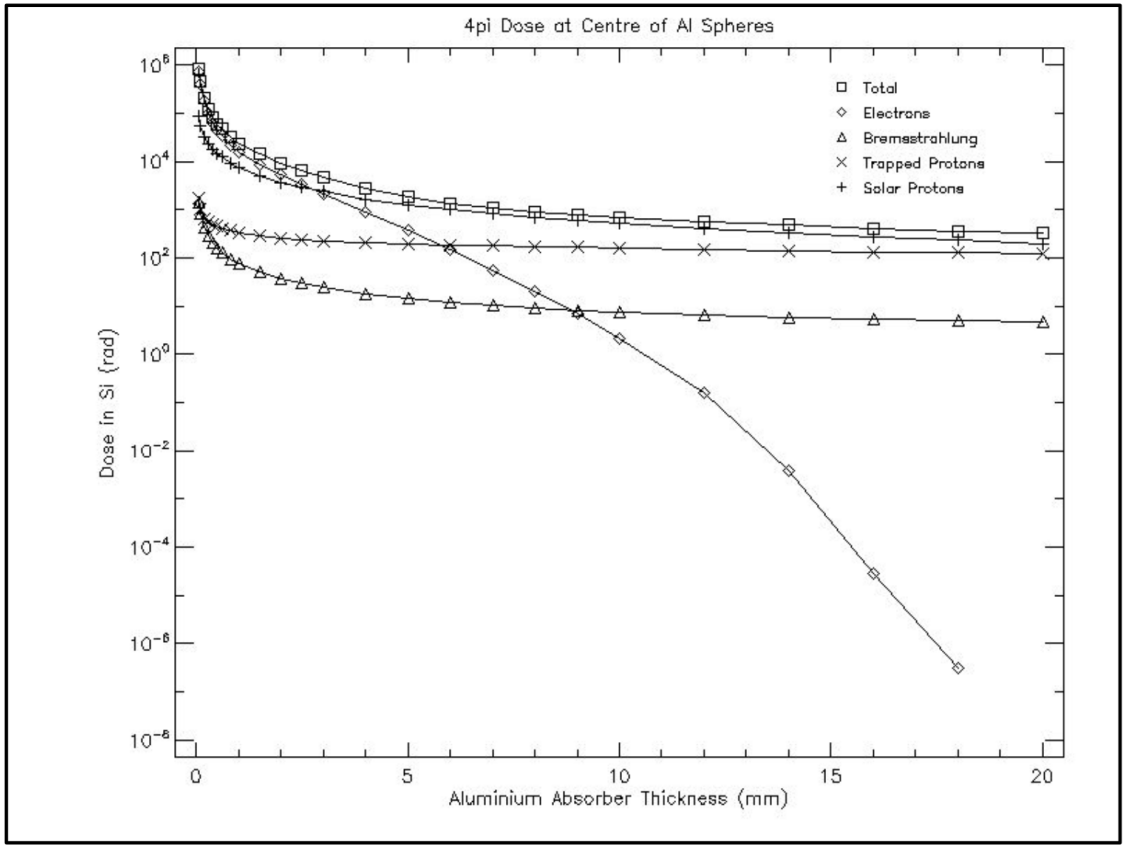
Example: TID calculation with SPENVIS

The Space Environment Information System

Nominal 3 years Mission

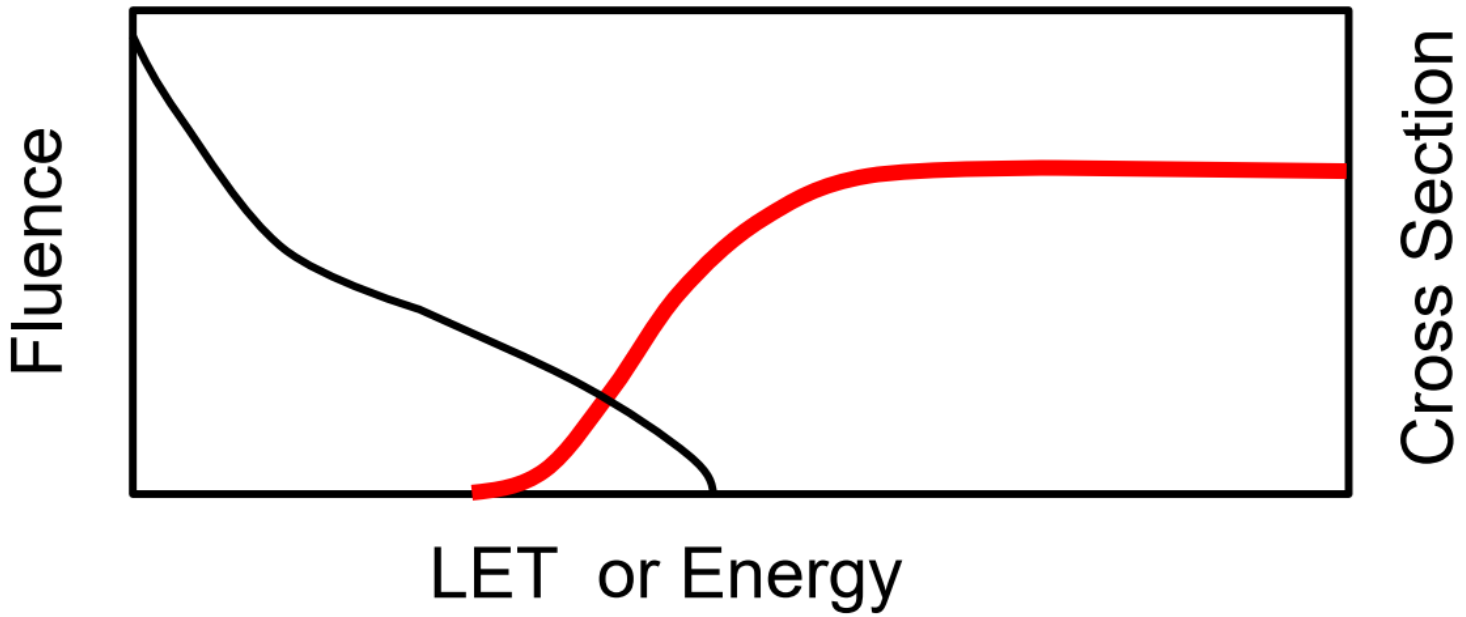
For 1.5 mm of Al shielding and Solar maximum (worst case scenario): TID = 13.39 kRad

We will compare the Total Ionising Dose (kRad) with the Radiation Tolerance (kRad) of each Component.



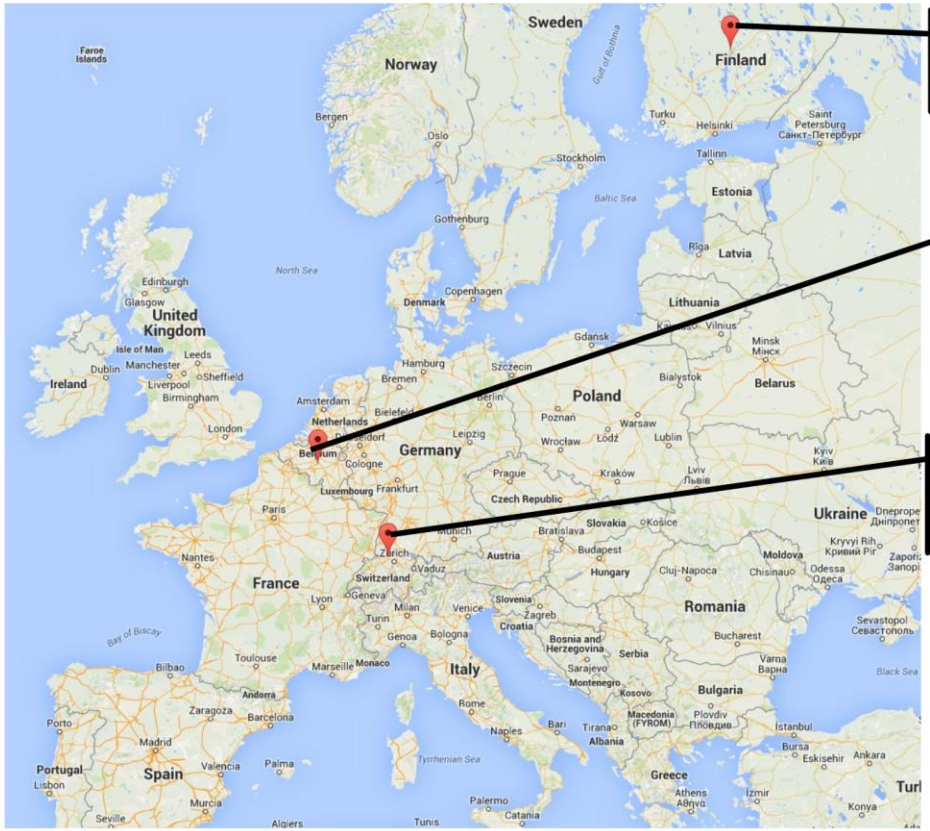
Why SEE Testing? SEE rate prediction

- Need to know:
  - Space Environment: Integral flux as a function of LET or energy
  - **Cross-section vs. ion LET or proton energy**



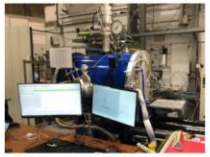
Test facilities (supported by ESA)

- Heavy ions and protons and electrons

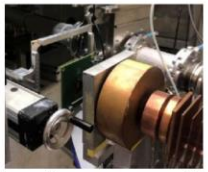


RADEF, JYFL Heavy ions, protons, electrons
Jyväskylä, Finland

UCL Heavy ions, protons
Louvain-la-Neuve
Belgium



PSI Protons, electrons
Villigen Switzerland



TEC-QEC has been collaborating with these facilities for more than 25 years. PSI, UCL, since 1990-1992. RADEF since 2004 beam in 2007-2008

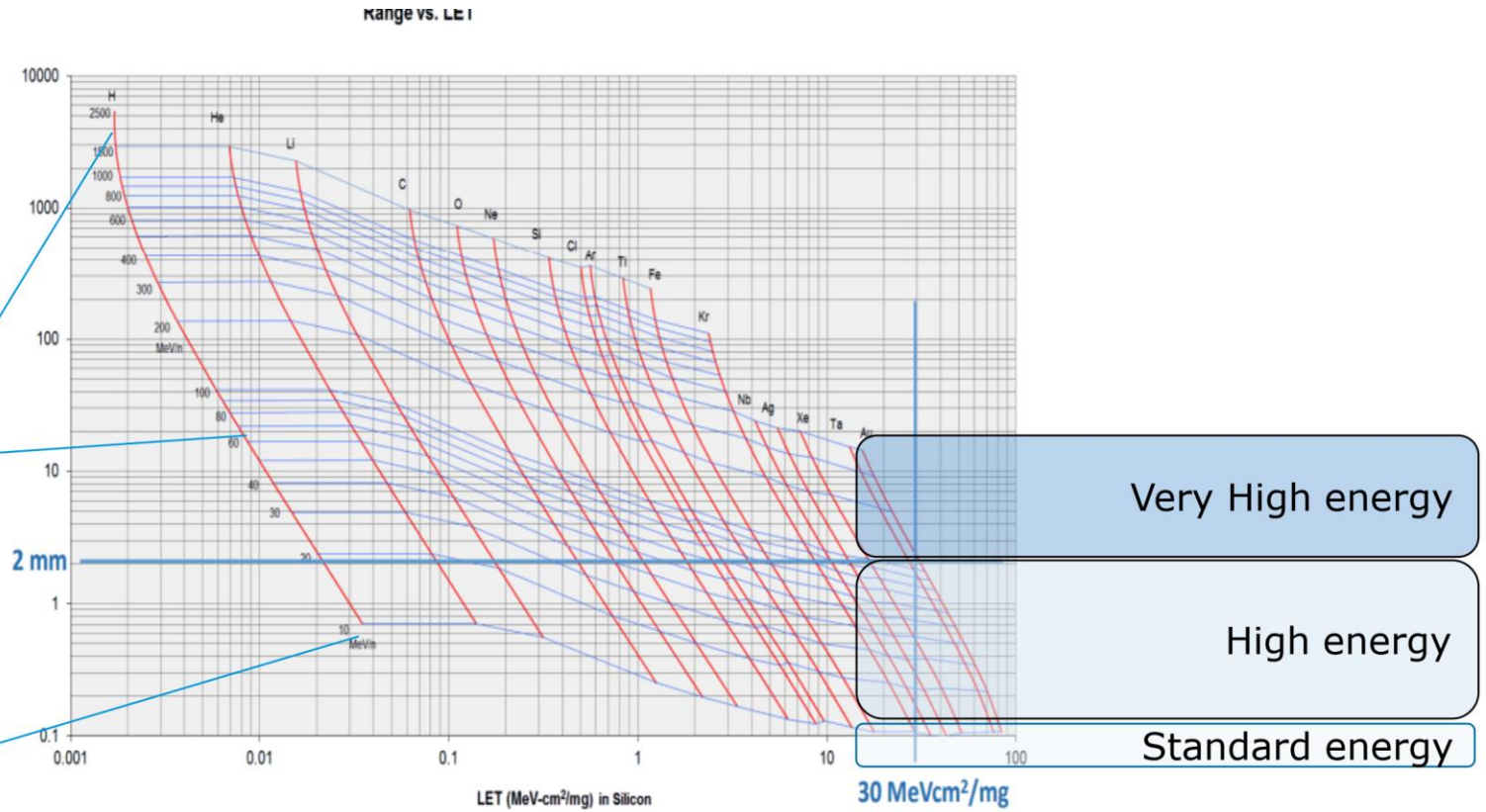
Aiming at continuous improvement of the quality of the beam, dosimetry and testing infrastructure

Stable flux and energy levels, high particle selectivity, accurate dosimetry, electrical/optical interfaces for cabling

Credit: ESA

Test facilities: LET & range

Facilities	Energy (MeV/nucleon)	Range of heavy species (Xe) in silicon
CERN CHARM	6-160 GeV/nucleon	meters
GSI SIS18	50 MeV/n to 1-1.5 GeV/n	2.4 mm to 7.8 cm
GANIL G4	27 to 60MeV/n	50 µm to 685 µm
KVI CART	30 MeV/n	333 µm
RADEF	22 MeV/n , 16.3 MeV/n, 9.3 MeV/n	255 µm 155 µm 92 µm
UCL HIF	8-10 MeV/n	73 µm



Credit: ESA

High-energy test facilities

- European space industry: critical competitive <u>disadvantage</u> due to lack of radiation testing opportunities of High Energy Ion beams
  - Currently, only facilities in USA offer High Energy Ion beams

Europe

Facilities	Energy (MeV/nucleon)	Availability per year
GANIL G4 (Caen, France)	27 to 60 MeV/n	1-2 weeks
GSI SIS18 (Darmstadt, Germany)	50 MeV/n to 1-1.5 GeV/n	Less than 1 week Only scientific experiments

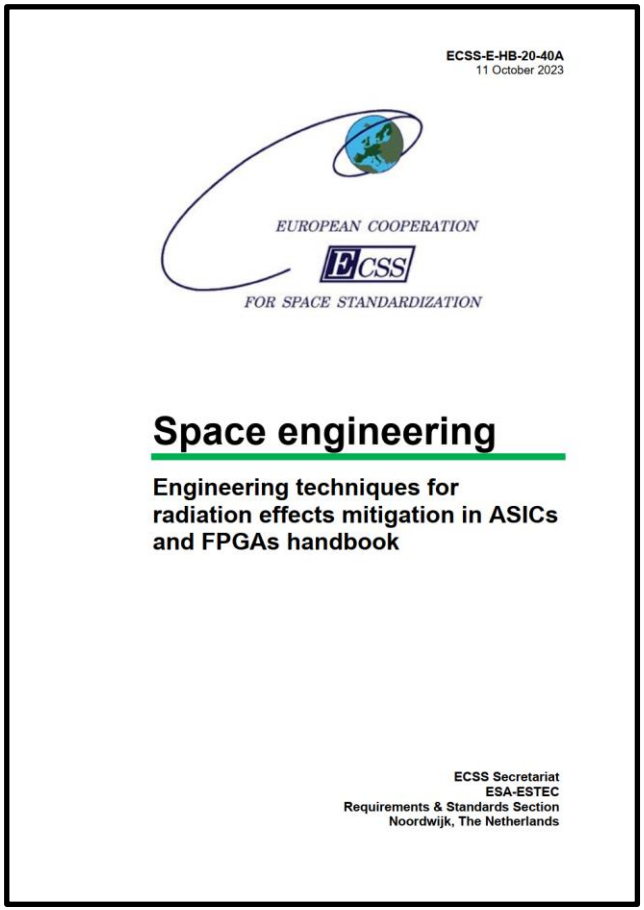
USA

Facilities	Energy (MeV/nucleon)	Availability per year
TAMU (College Station, TX, USA)	15 MeV/n 25 MeV/n 40 MeV/n	About 20-25 weeks
NSRL (Brookhaven, USA)	1500-217 MeV/n (light to heavy ions)	~20 weeks NASA funded or scientific proposals

Credit: ESA

ECSS Space Product Assurance

- ECSS-E-HB-20-40A provides a compilation of different techniques that can be used to mitigate the adverse effects of radiation in Ics with almost exclusive attention to ASICs and FPGAs
- The target users of this handbook are developers and users of ICs which are meant to be used in a radiation environment



FPGA Technologies used in Space

- **Antifuse FPGAs** use electrical structures, called antifuse, performing the opposite function to a fuse
  - Antifuse starts with a high resistance and is designed to permanently create an electrically conductive path (typically when the current through the antifuse exceeds a certain level)
  - Drawback: the configuration is not reversible
  - However, this is an advantage in terms of radiation tolerance since the configuration layer is immune to radiation induced bit-flips
- **SRAM-based or Flash-based** memory cells: reconfigurable
  - Can be more or less sensitive to radiation depending on technology
  - Bit-flips in configuration memory can impact user logic
    - In such case, even an application reset does not allow recovering
  - Such a permanent mutation can thus have critical consequences and an FPGA reconfiguration is necessary to recover the nominal configuration

FPGA Characteristics and Vendors

Configuration memory nature	Antifuse	Flash	SRAM
Characteristics	• Electrically programmable switch which forms a low resistance path between two metal layers	• Electrically programmable transistors which hold the configuration that controls a pass transistor or multiplexer connected to predefined metal layers	• The state of a static latch controls a transistor or multiplexer connected to predefined metal layers
	• Configuration is NON volatile	• Configuration is NON volatile	• Configuration is volatile
	• One-time programmable	• Re-configurable	• Re-configurable
Representative manufacturers	Cobham (former Aeroflex) Microsemi	Microsemi	Xilinx Microchip Atmel

Mitigation techniques for digital systems

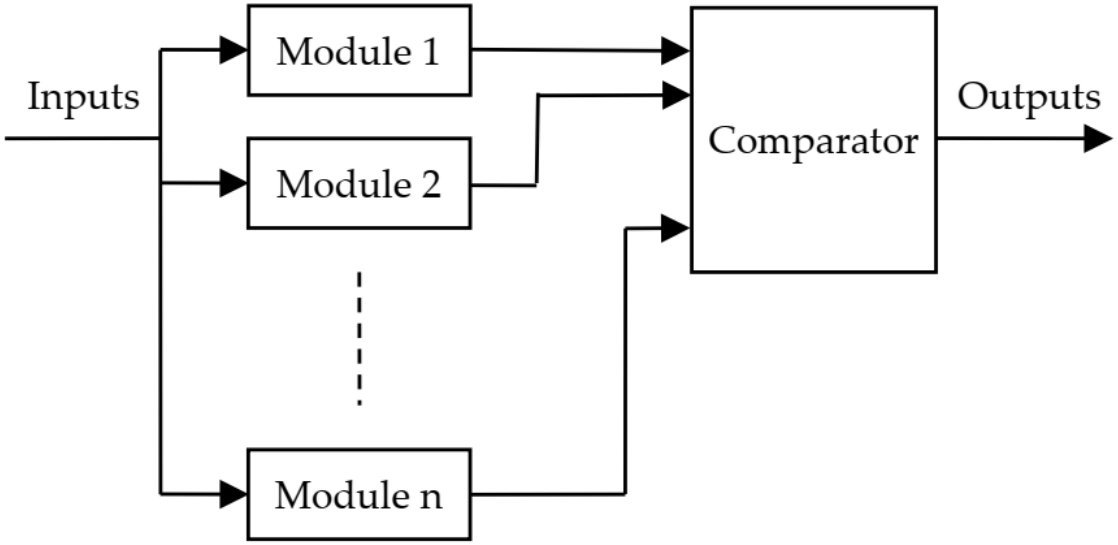
- Several mitigation techniques out of one or more of these 5 groups can be applied:
- **Spatial redundancy**: resources are replicated to process the same task in parallel
  - A downstream comparison or voting circuitry is in charge of error detection and eventually error correction, depending on the number of implemented replicas
  - Depending on the selected architecture, the hardened system can handle a more or less wide scope of errors (e.g. SET and SEU).
- **Temporal redundancy**: signals are sampled (or full functions executed) at different instants and a voting/comparison circuit allows rejecting SETs and SEUs
- **Memory cell hardening**: memory cells often represent a large percentage of the total silicon area occupied by a digital circuit
  - Designers should take special precaution to ensure radiation robustness meets the mission criteria
  - Suitable solution: Replacement of memory cells (e.g. flip-flops, registers or latches) by RadHard ones
- **Memory block hardening**: to prevent radiation induced errors in more than one bit of a "data block" residing in a memory cell array we can implement mitigation techniques at memory block level, avoiding that bits belonging to a same "data block" are stored physically too close to each other, therefore can be altered by the same radiation event
- **Information redundancy**: error-detecting codes and error-correcting codes are able to protect integrity of data blocks that reside in memory cell arrays from radiation effects
- Permanent errors due to TID cannot be mitigated with such techniques

Mitigation techniques and radiation effects they address

Mitigation techniques	Radiation effects	
	SET	SEU
Spatial redundancy	X	X
Duplex architectures	X	
Triple Modular Redundancy architectures	X	X
Basic TMR	X	X
Full TMR	X	X
Temporal redundancy	X	X
Triple Temporal Redundancy combined with spatial redundancy	X	X
Minimal level sensitive latch	X	X
Fail-safe, deadlock-free finite state machines		X
Selective use of logic cells, clock and reset lines hardening	X	X

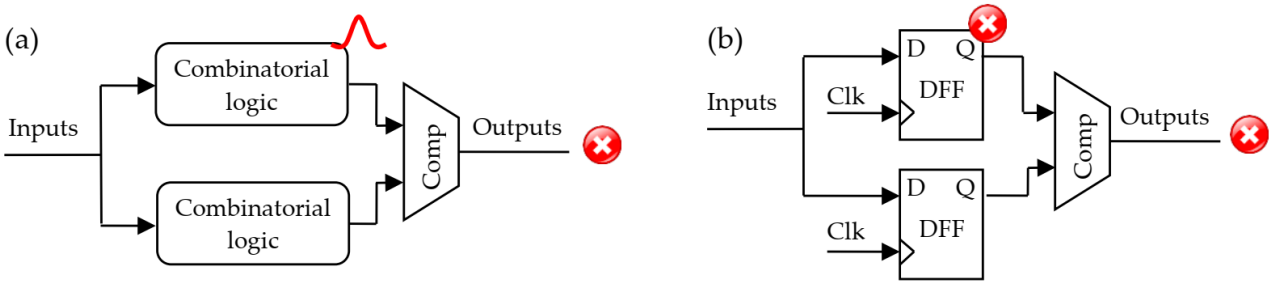
Spatial redundancy

- Spatial redundancy solutions can be classified into two categories depending on whether they can provide:
  - Error detection only: this is the case for duplex architectures, also called **Dual Modular Redundancy (DMR)**
  - Error detection and correction: as it is the case for architecture having:
    - three, called **Triple Modular Redundancy (TMR)**, or
    - more replicas, called **N-Modular Redundancy (N-MR)**

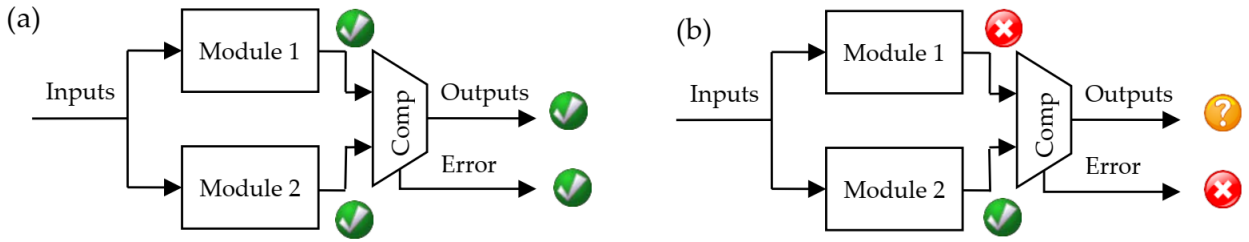


Duplicate with Compare (DWC)

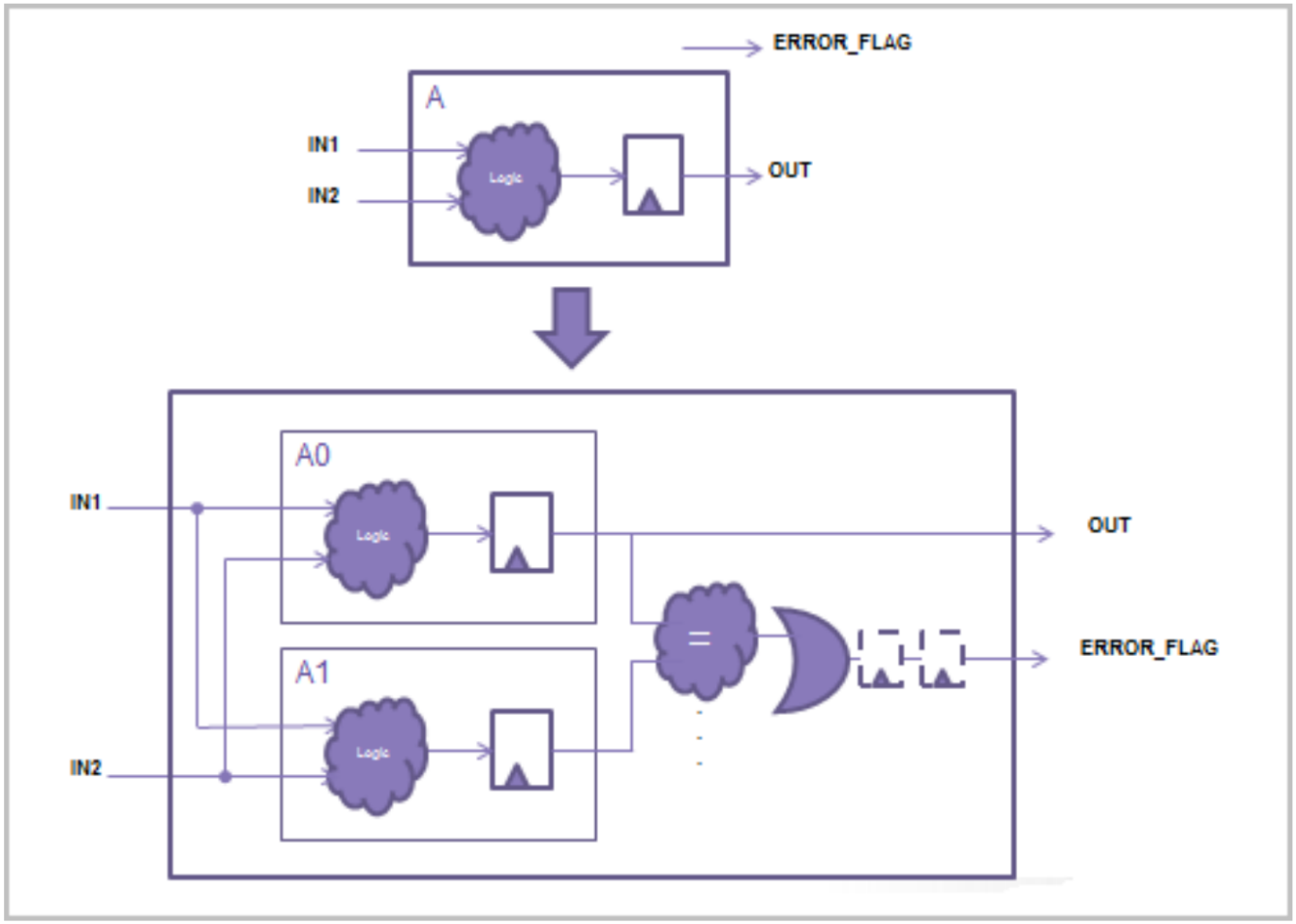
- DWC architecture uses two replicas of a processing unit and compares outputs to detect potential differences provoked by SEEs and then either flag the difference or prevent a wrong value from propagating (by going into high impedance mode)
  - Applicable for both combinatorial and sequential logic and provide respectively SET (a) and SEU (b) detection



- Fail-stop architecture: <u>Can detect faults but not recover them</u>
  - When both results are identical (but not necessarily correct), comparator assumes that both are correct
  - When they differ, comparator detects an error but is not capable to determine the non-faulty one
    - In this case two recovery mechanisms can be applied: either to skip this value and move on the next one, or to process the data again in order to obtain the correct value
    - This choice depends on the critical risk to the application

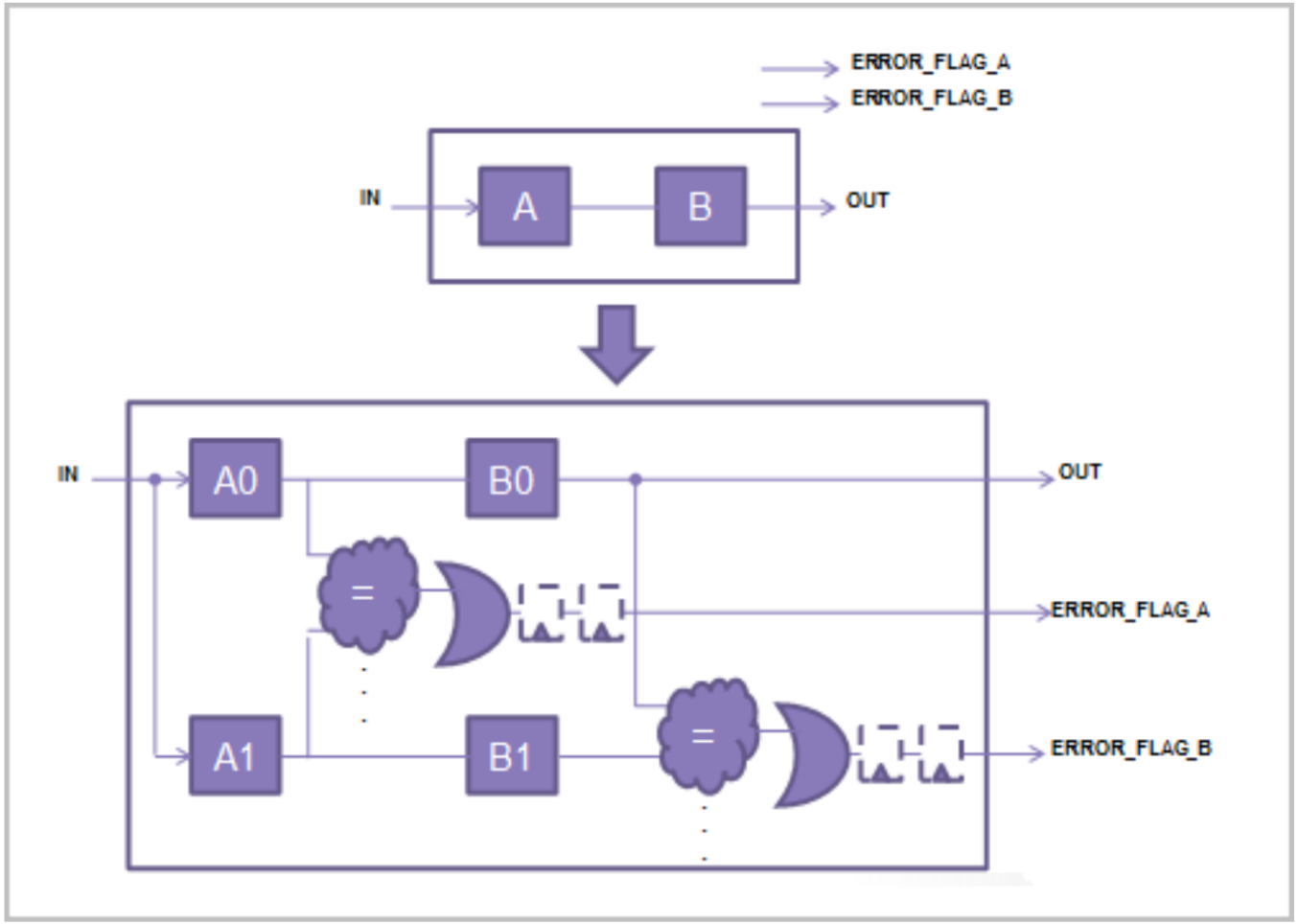


Example1: DWC with Single Module



Credit: Synopsys

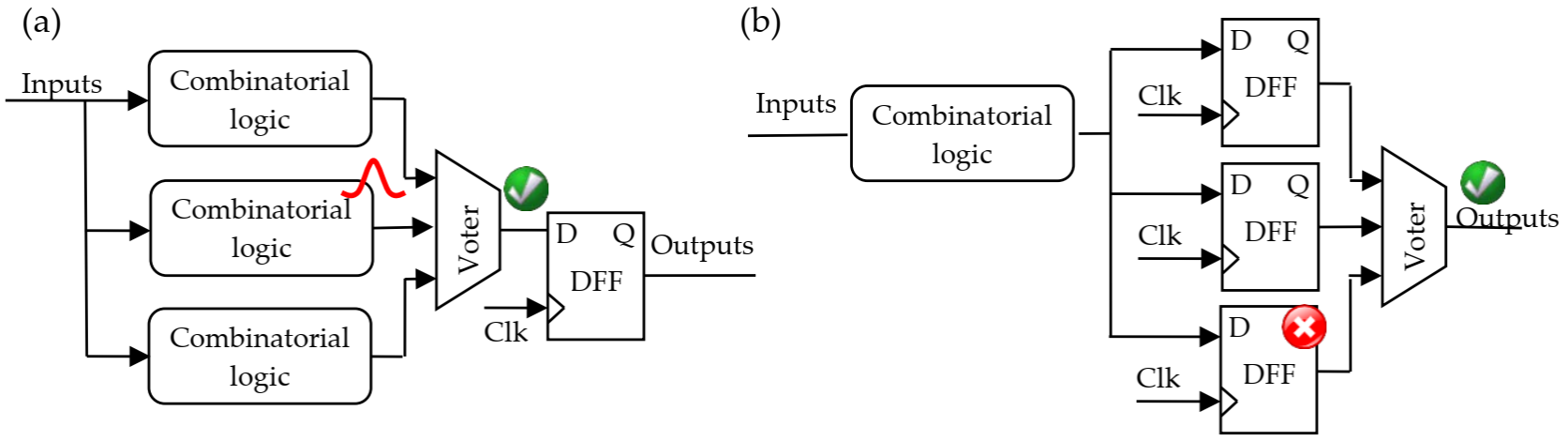
Example2: DWC with Multiple Modules



Credit: Synopsys

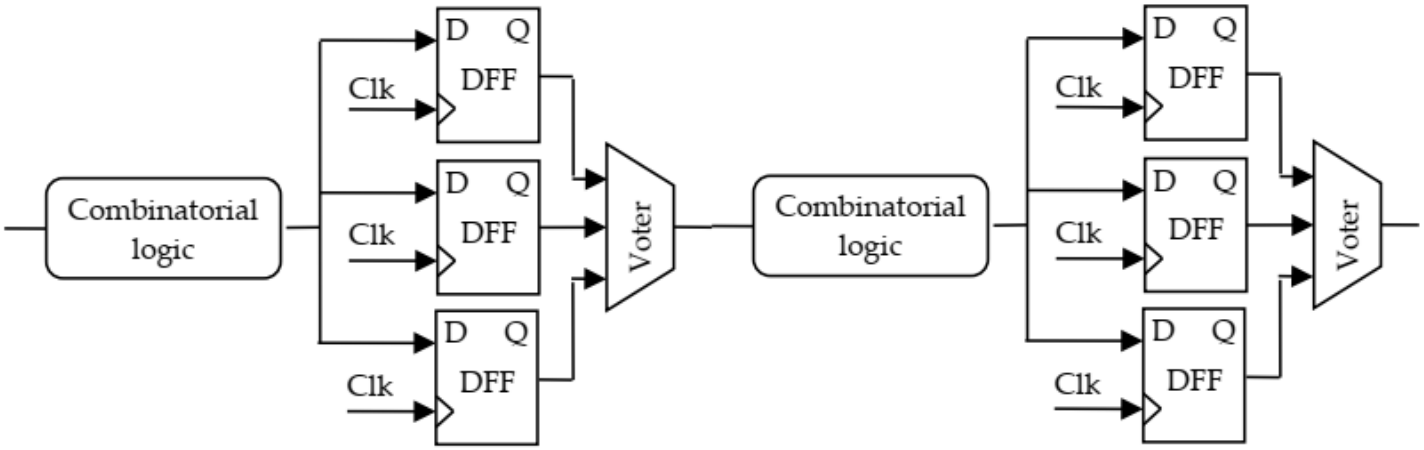
Triple Modular Redundancy (TMR)

- Triple Modular Redundancy (TMR) architecture implements three identical flip flops processing the same task and whose outputs are compared by a majority voter
- The main advantage of TMR is its capability to detect and correct single event transients (SETs) (a) and upsets (SEU) (b)
- The following FPGA CAD vendors provide automatic insertion of TMR:
  - Xilinx X-TMR tool
  - Synopsys Synplify Premier
  - Mentor Graphics Precision Hi-Rel

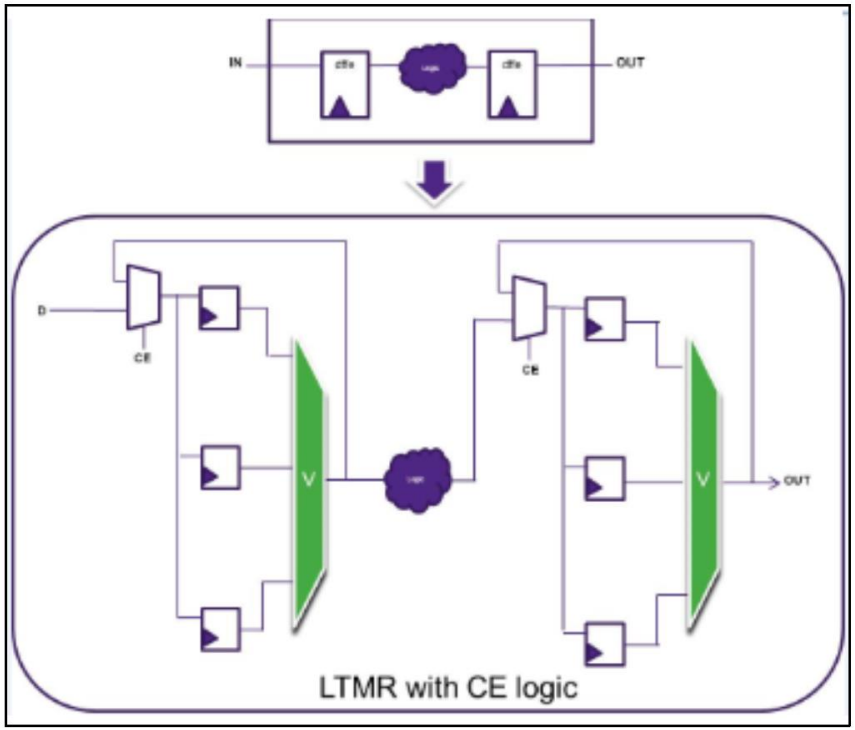


Local TMR

- Also called Register-based TMR
- Local TMR consists in triplicating only flip-flops and voting their outputs
  - Some FPGA vendors offer FPGAs with flip-flops that have already been hardened at transistor level so that the FPGA user can consider these flip-flops as “locally-TMRed-FFs” e.g. Microsemi RTAX-S/SL
  - In other cases where FPGAs that do not embed a local hardening scheme for their flip-flops (e.g. commercial grade FPGAs), local TMR can be applied by the FPGA user in the HDL description of the design
- Can be used for low-speed designs and thus with low probability of capturing SETs in the flip-flops



Example: Local TMR



To set a syn_radhardlevel value for all the registers of a module, do the following:

– Set the value in the source file. The following sets all registers of module_b to tmr:

VHDL	Verilog

`library synplify;`
`use synplify.attributes.all;`
`attribute syn_radhardlevel of`
`  behav: architecture is "tmr";`
	`module module_b (a, b, sub,`
	`clk, rst) /*synthesis`
	`syn_radhardlevel="tmr"*/;`

To set a syn_radhardlevel value on a per-register basis, do the following:

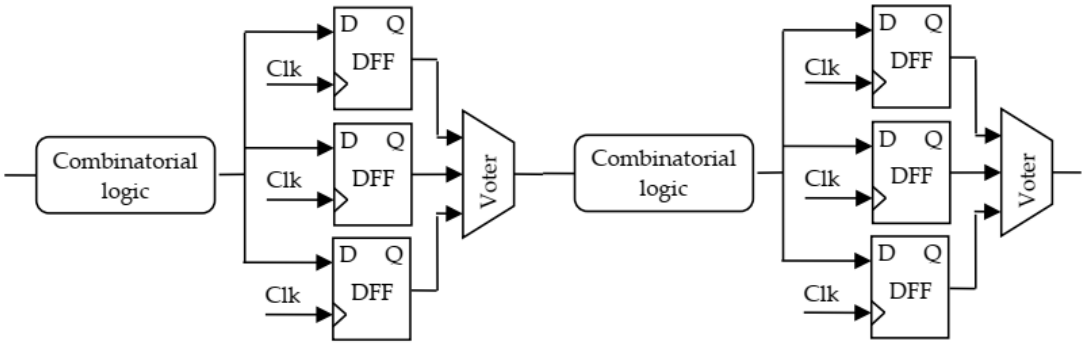
– Set the value on the register in the source file for the module. For example, to set the value of register bl_int to tmr, enter the following in the module source file:

VHDL	Verilog

`library synplify;`
`use synplify.attributes.all;`
`attribute syn_radhardlevel of`
`  bl_int: signal is "tmr"`
	`reg [15:0] a1_int, b1_int`
	`/* synthesis syn_radhardlevel =`
	`"tmr" */;`

Credit: Synopsys

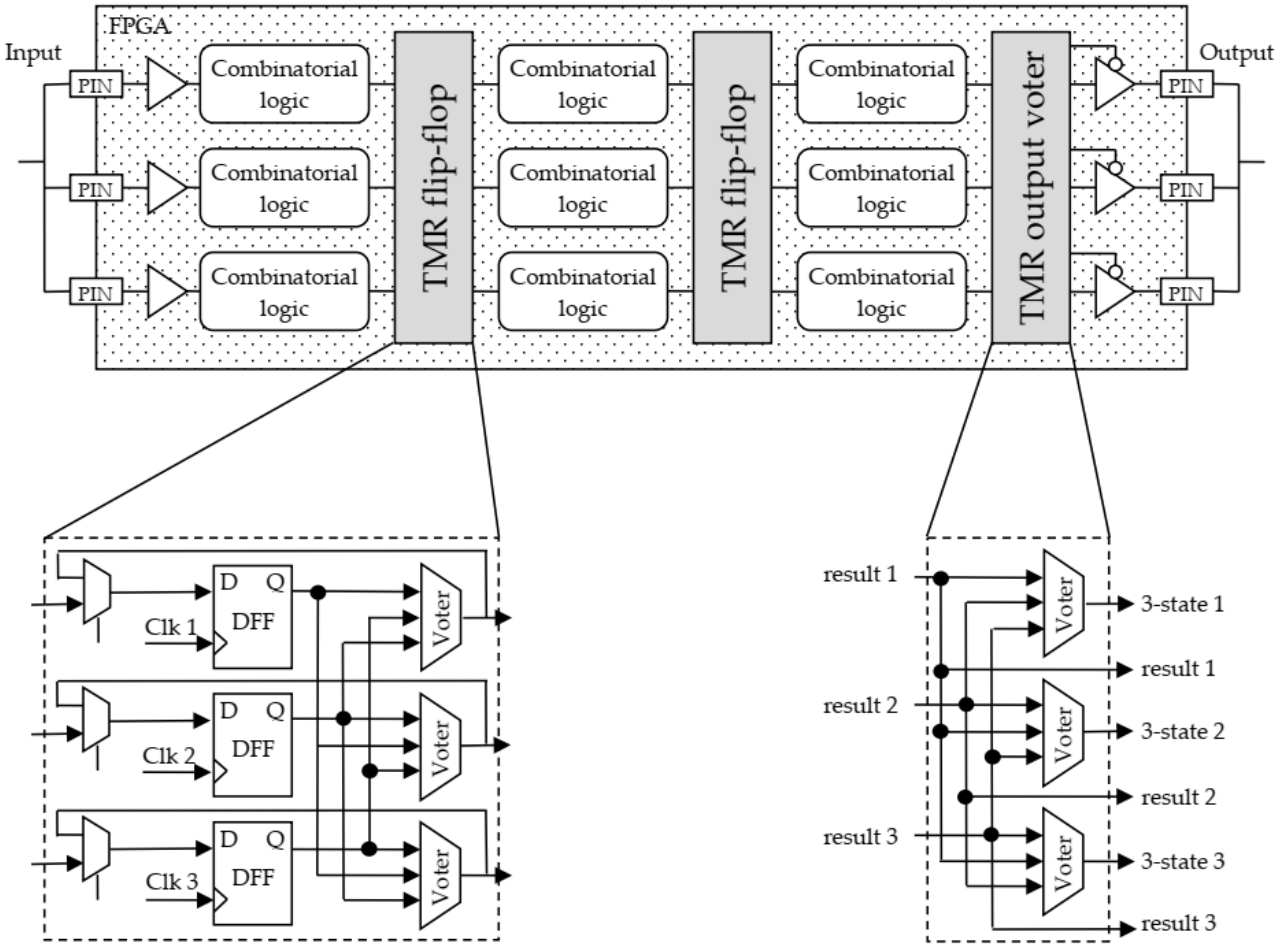
Local TMR



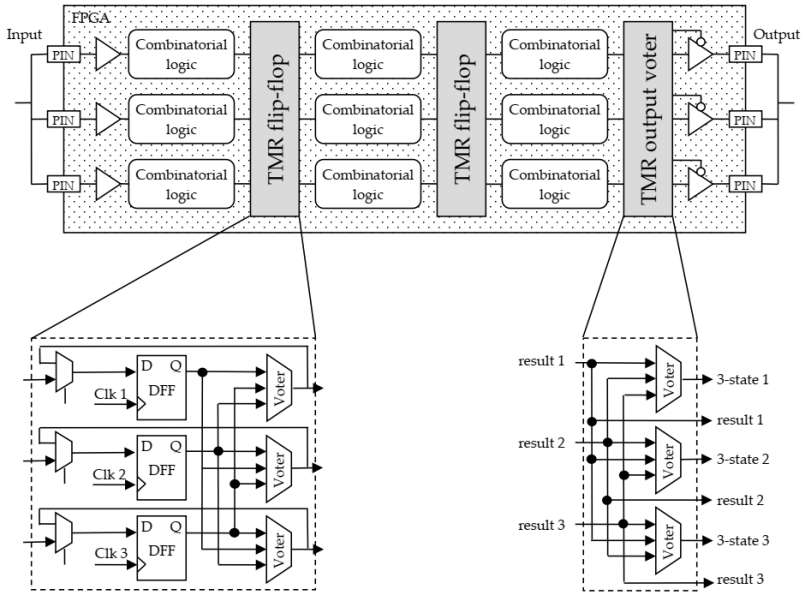
- Benefits
  - Protects against SEUs in the flip-flops (FF).
  - Area penalty is limited to registers as combinatorial logic is not replicated
- Weaknesses
  - Area overhead is 3 times more flip-flops plus the additional voting logic per triplet
  - Timing overhead
  - A SET occurring in the combinatorial logic propagates to the FFs and if concurrent with sampling clock pulse, the error is latched and the voter has three identical, but false, results and consequently it does not detect the error. A solution is “global TMR”
  - Does not protect against a Multiple Bit Upset (MBU) that affects flip-flops of the same TMR triplet
  - Does not protect against upsets in “configuration logic” when the FPGA uses SRAM or EEPROM configuration memory cells, which cannot be hardened by the user by applying local TMR

Global TMR

- Global TMR triplicates all the resources of an application, including clock tree and IOBs
- It can be applied at the RTL level (HDL design) either by the user or through the use of dedicated tools such as Xilinx X-TMR tool or Mentor Precision Rad-Tolerant which are both able to automatically apply Global TMR to the user's design

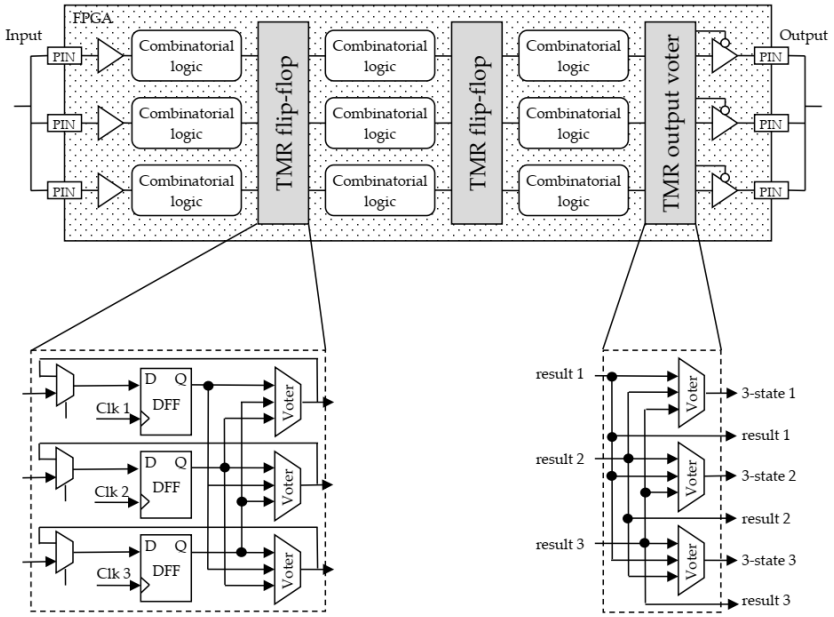


Global TMR



- Entire processing chain is triplicated from the input pins to the output pins
- FFs are hardened using 3 redundant FFs, 3 voters and feedback paths for fault recovery
- Final stage, called TMR output voter, controls the enable input of a tri-state buffer
  - This buffer is used in high-impedance mode whenever a faulty result is encountered, hence avoiding the output of erroneous result
- The only sensitive part of the architecture is its output voter
  - However, the three outputs being connected together operate like an “analogue voter”: two correct results force the output value to the correct logical level
  - Other voting techniques can also be implemented with the redundant outputs at board level, thus completing the mitigation strategy with “system level” (off-chip) measures

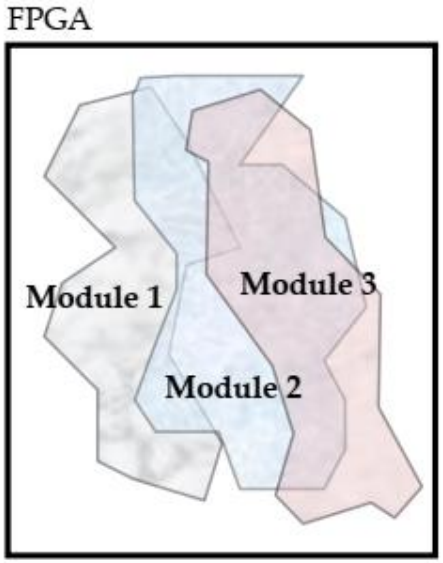
Global TMR



- Benefits
  - Protects the whole design from SET in combinatorial logic and SEU in registers
  - Moreover, helps to mask but not to correct upsets in the configuration memory
- Weaknesses
  - Has increased area overhead
  - Timing overhead
  - Requires clock skew management
  - Validation of global TMR is not easy

Global TMR

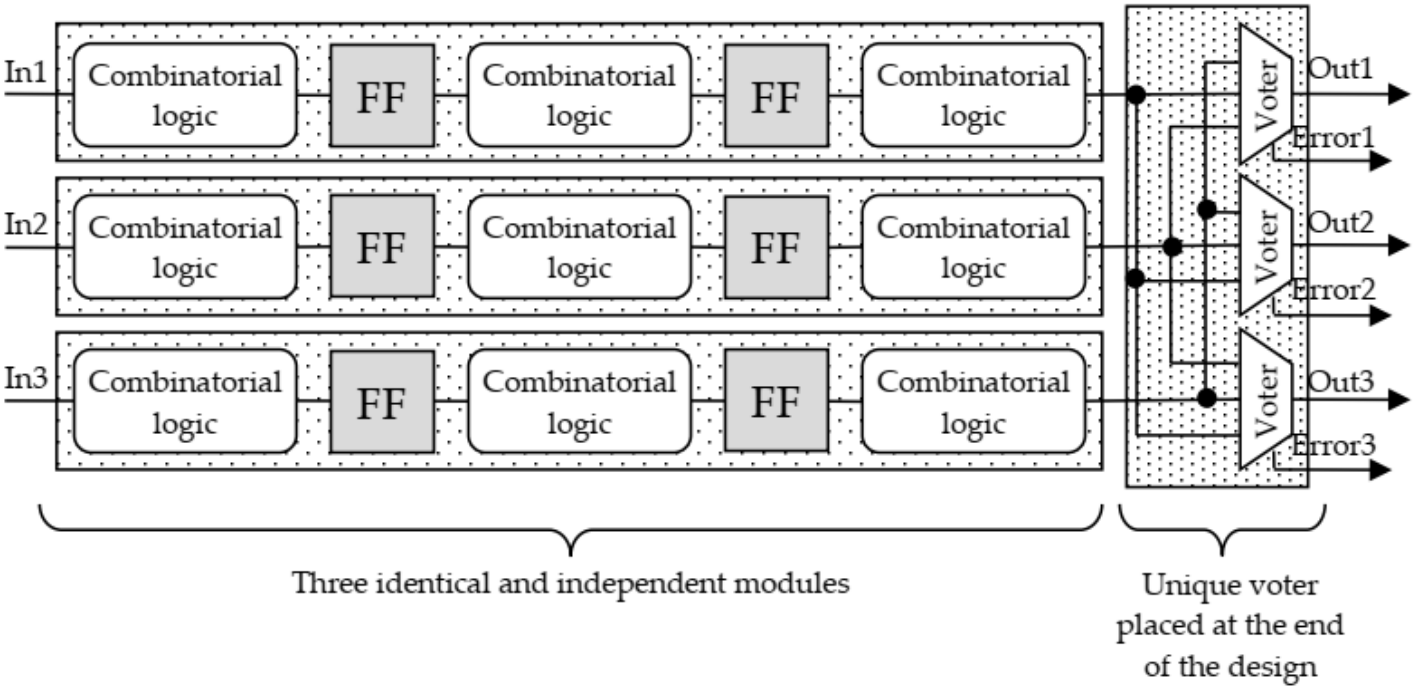
- Global TMR implies having frequent interconnections between 3 TMR replicas
- Almost impossible to physically separate the 3 replicas in FPGA implementation of the design
- Design with TMR after FPGA implementation: the 3 replicas overlap and FPGA resources from the 3 domains are mixed within same logic blocks. This has two consequences:
  - a) partial scrubbing cannot be used and
  - b) increased risk to encounter domain crossing



Physical Implementation of Global TMR inside an FPGA

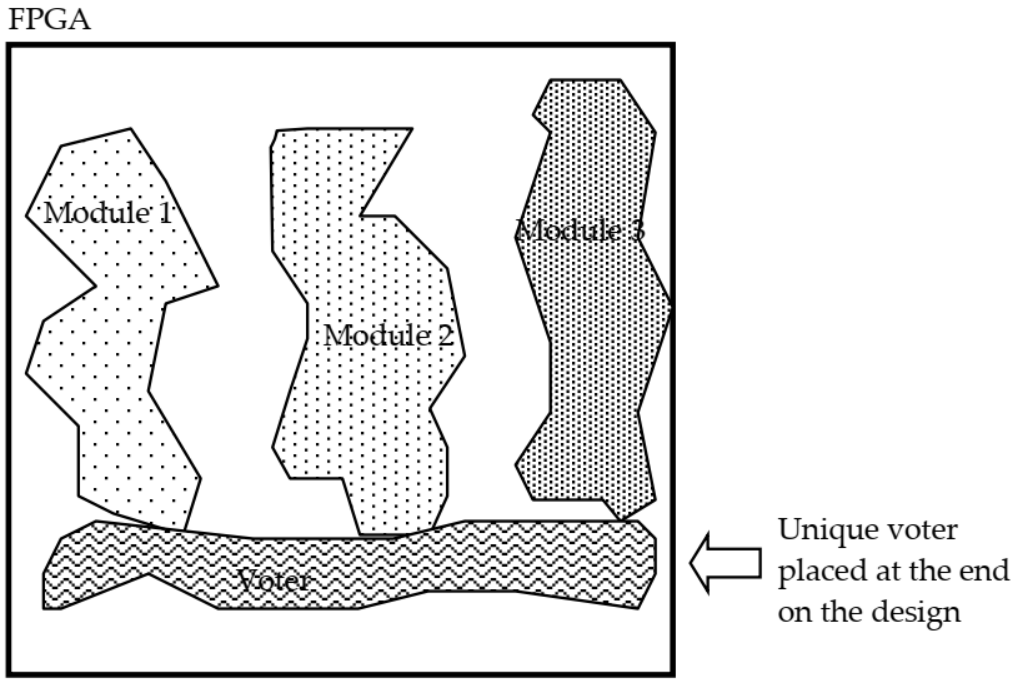
Large grain TMR

- Large grain TMR consists in triplicating a design, but unlike local and global TMR, the FFs are not voted
- Instead, a unique voter is placed at the end of an entire module

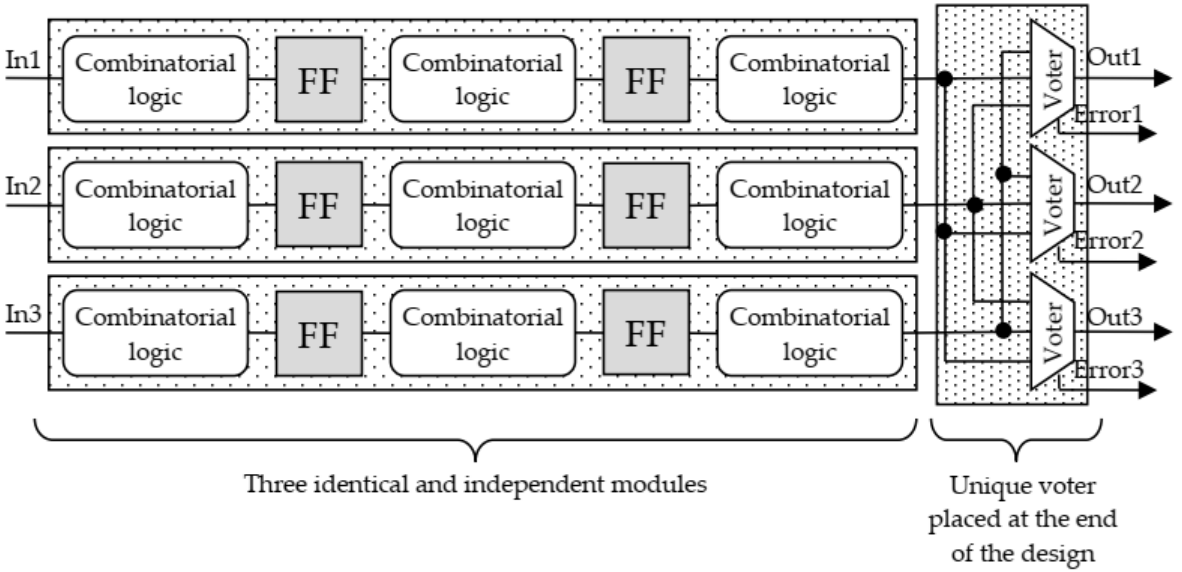


Large grain TMR

- **Challenge**: resynchronize an erroneous replica with the others
- **Solution**:
  - Identify the erroneous module by modified majority voter
  - Reconfigure the faulty module if the upset took place in the configuration memory
  - Synchronize the module with the other two



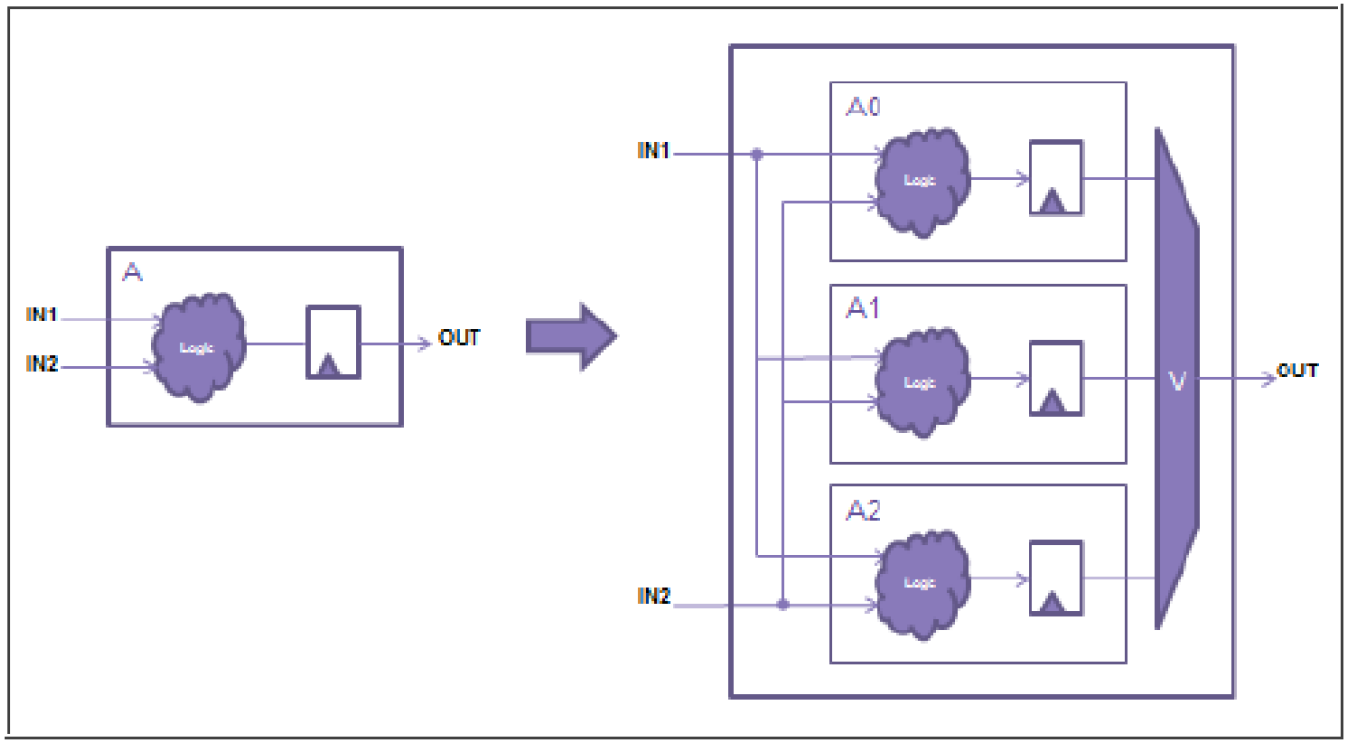
Large grain TMR



- Benefits
  - Helps mitigating SEU in the configuration memory and in the user logic
  - Local placement and routing for each TMR redundant domain allowing physical separation of each replica. There is the possibility of using partial reconfiguration to scrub only a redundant domain that has the error, reducing scrubbing time and energy
  - Minimal points of domain crossing means reduced vulnerable bit-flips that can upset TMR
- Weaknesses
  - Increased area overhead
  - Timing overhead
  - Can fail if two sensitive bits belonging to two different replicas are upset

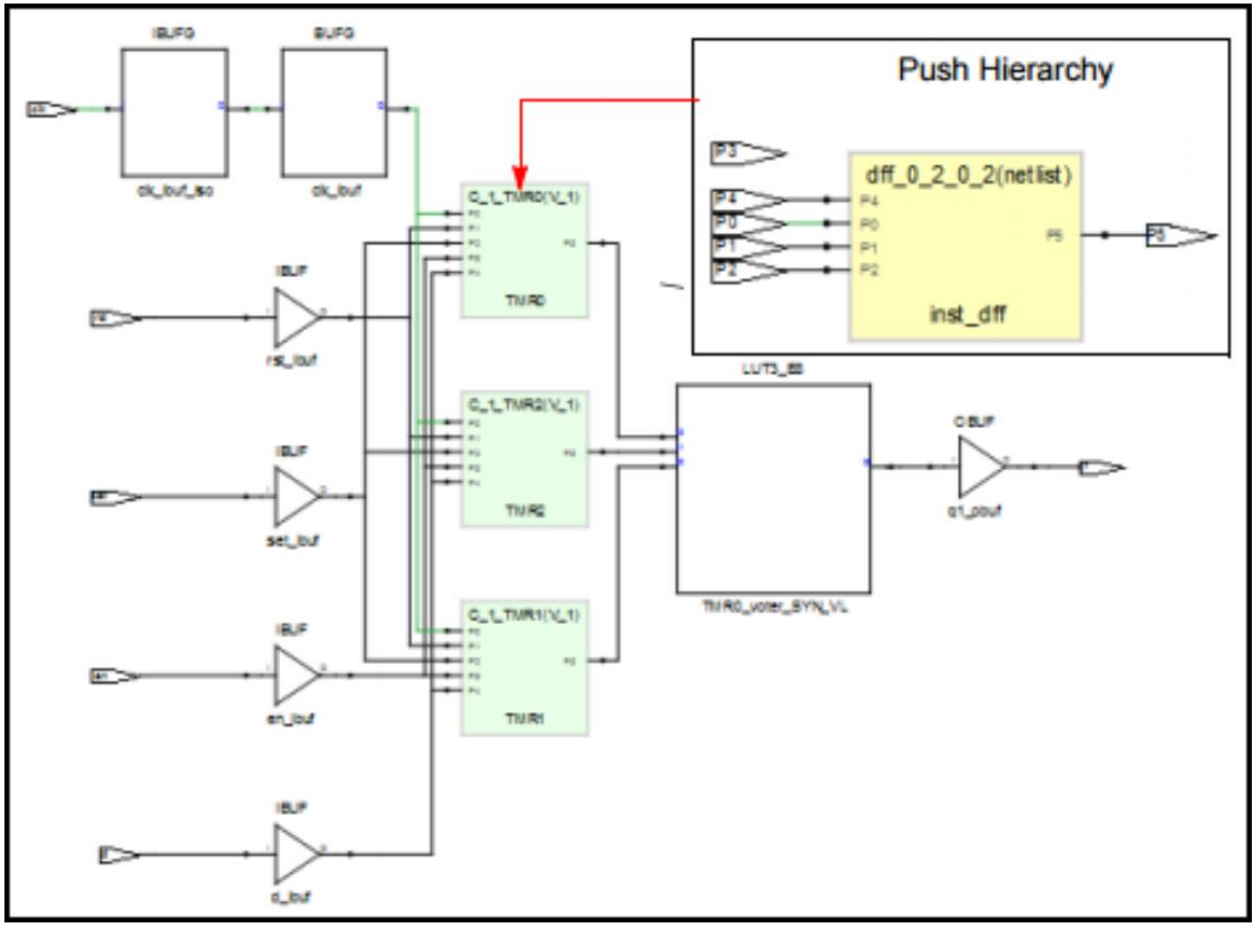
Example: Synopsys Distributed TMR

- Three separate instances created for the specified module within the original module hierarchy
- Inputs to the original module are connected to all three instances
- Outputs are fed to a majority voter, which produces a single output connected to the fanout of the original module



Credit: Synopsys

Example: Synopsys Distributed TMR



Credit: Synopsys

Ensuring Safe Operation in FPGA designs

- When applying redundancy-based error mitigation like TMR, do following:
  - Set direct connections from inputs or outputs to the voter
    - Ensures error-corrected output has less chance being corrupted or subject to skew
  - Ensure clocks are synchronized
    - Clocks that drive logic cones not been synchronized with each other can cause meta instability
  - Minimize clock skews on triplicated circuitry
    - E.g. , if block TMR is used to triplicate IP/large blocks, synchronize all data inputs with a single clock, whenever possible
  - If you insert custom error mitigation circuitry or probes into design, make sure synthesis tool preserves them using the *syn_keep* or *syn_preserve* attribute
    - Otherwise, this circuitry is at risk of being optimized away.

Credit: Synopsys

Error Correcting Codes (ECC)

- Error Correcting Codes (ECC) are algorithms capable of detecting and/or correcting errors in data by adding redundant data or parity data to the original data
  - ECC is a mitigation technique based on information redundancy
  - When errors are detected & corrected, term **EDAC (Error Detection And Correction)** is used
- Each ECC has its own characteristics in terms of fault detection and fault correction
  - All impact the system by adding an area overhead to store the redundant data
  - All add time overhead to compute these data and check original data for consistency
- There are two main ECC families: block codes and convolutional codes
  - Convolutional mainly used for data transfer (digital video, mobile communication and satellite communication)
  - Block codes are rather used for protection of data storage
- Benefits
  - ECC protect against SET, SEU, MBU/MCU in data storage logic including embedded memories (FPGA BRAMs)
- Weaknesses
  - There is an increased area and time overhead depending on the ECC and the amount of redundant data

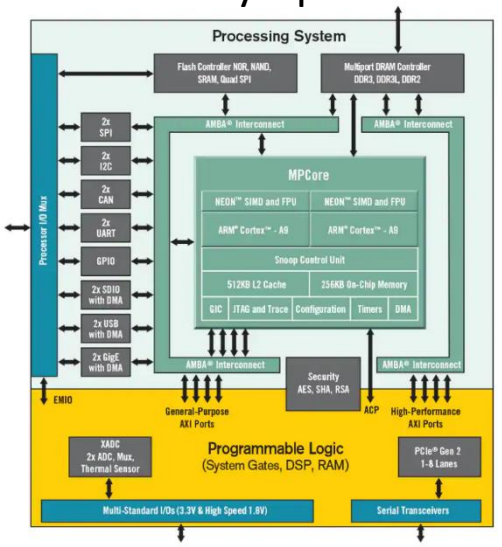
ECC	Error detection	Error correction
Parity check	X	
Cyclic Redundancy Check	X	
BCH codes	X	X
Hamming codes	X	X
Reed-Solomon codes	X	X
Low Density Parity codes	X	X

Most commonly used ECC in space applications

Embedded Processor Protection

- Recent FPGAs embed hardwired processors sensitive to SETs, SEUs and SEFIs since these FPGAs are implemented in commercial processes without any built-in protections against radiation effects
- Several techniques can mitigate radiation effects in embedded processor:
  - Purely SW-based (e.g. **Software-Implemented Hardware Fault Tolerance - (SIWFT)**
    - No HW overhead but SW modifications (application of instruction, task or application-level redundancy)
  - Spatial redundancy-based (i.e. DMR such as **Lockstep** where a primary processor and a backup one run the same SW)
    - Such solutions generally involve multiple processors at system-level performing the same task in order to compare their outputs, and thus to detect faults. In case of mismatch the task can be performed again
  - Hybrid approach
- Benefits
  - Embedded processor protection mitigation techniques protect against SET, SEU and SEFI.
- Weaknesses
  - Increased memory penalty (≈2x to 3x code and data size) and similar time penalty.
  - There is also a FPGA resource penalty depending on the processor core

Xilinx Zynq FPGA

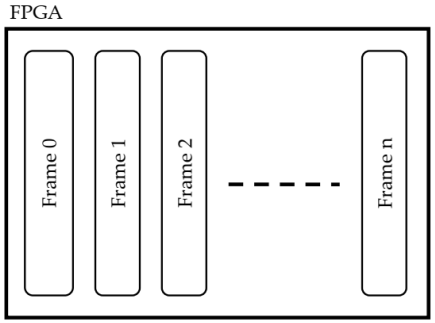


Configuration Memory Scrubbing

- Spatial redundancy by itself is not sufficient to provide mitigation against radiation induced errors in SRAM-based FPGAs
  - It allows rejecting SET in the combinatorial logic and SEU in registers
- SRAM-based FPGAs configuration memory is sensitive to radiation effects
  - Can create a permanent malfunction of the system programmed in the FPGA by changing, for example, the logical function implemented in a LUT or the type of an IO port in use
- Important: periodically reload the configuration bitstream of the FPGA:
  - to over-write the configuration bits with the golden ones
  - avoid the accumulation of faults in the configuration memory
- **This continuous loading of the bitstream is popularly called "scrubbing"**
- Scrubbing allows a system to repair bit-flips in configuration memory without disrupting its operation
  - Includes memory cells that configure: LUT, control routing and CLB customization
- Configuration scrubbing prevents multiple configuration faults and reduces the time in which an invalid circuit configuration is allowed to operate

Configuration Memory Scrubbing

- Configuration memory of a Xilinx Virtex FPGA is divided into several frames representing the minimal amount of resources which can be configured. Such structure allows reconfiguring either:
  - Full device (**full scrubbing**) or
  - Only a part of the design (**partial scrubbing**)
  - Selection of the scrubbing mode mainly depends on the selected spatial redundancy scheme



- Scrubbing is not sufficient to completely protect SRAM-based FPGAs as it only avoids accumulation of faults in configuration memory
  - Faults can occur between two scrubbing cycles and provoke errors in the application until the next refresh of the configuration memory
- Scrubbing will not correct faults in user registers nor in embedded RAM
  - It is important to protect against faults both in configuration memory and user logic
  - It is important to apply additional mitigation techniques as a complement to scrubbing
- Good practice: Scrub >10 times faster than expected worst-case SEU rate
  - Scrubbing frequency depends on the particle flux and cross-section of the device

Partial Reconfiguration

- Some SRAM FPGAs allow user to perform a re-write of only a fraction of reconfiguration bits
  - Overwrite bits of configuration bitstream that have been flipped by radiation with the good values (remain safely stored in a memory outside the FPGA)
  - This operation is referred as "**partial scrubbing**"
- Partially reconfiguring the FPGA can provide TID effects mitigation
  - If a **permanent fault** is in a given area of the FPGA, and the user has been able to detect this, a possible mitigation technique could be to relocate the affected functions that were mapped in faulty area to another (not used and fault-free)
  - This requires a dedicated partial reconfiguration controller and logic to detect permanent faults across used and unused areas
- Dynamic partial reconfiguration can be used to apply more or less mitigation, depending on radiation environment and/or how many radiation induced faults can be tolerated along the space mission life
  - Reliable FPGA design: mitigation is applied according to the worst-case condition
  - Mitigation overheads (e.g. in performance, power) during relaxed conditions can be optimized by not always applying mitigation for the worst-case conditions

Configuration Memory Scrubbing

- Benefits
  - Scrubbing prevents SEU accumulation in the configuration memory.
  - Recent Xilinx devices provide internal scrubbing using the HWICAP and an internal scrubbing controller, therefore not using an external processor for the scrubbing
  - Scrubbing application is not interrupted
  - Scrubbing helps mitigating upsets in the configuration memory but not in the user logic.
  - Dynamic partial reconfiguration can also be used to mitigate permanent faults caused by TID effects (combined with aging and wear out). This is useful for non-rad hard FPGAs
- Weaknesses
  - Scrubbing does not correct SEU in embedded memories (e.g. BRAMs) nor in user's flip-flops. BRAM TMR is recommended in embedded user memory
  - Upsets occurring between two scrubbings can provoke errors. Additional mitigation techniques (i.e. TMR) can be implemented as a complementary technique to scrubbing
  - Dynamic partial reconfiguration mitigation may involve having to store large amount of bitstreams, and maybe slow (generating, programming and running fault detection circuits before implementing the partial reconfiguration of the FPGA).
  - Some permanent faults in some FPGA resources (e.g. user's memory or DSP routing resources) may not be easy to detect

Fail-safe, Dead-lock Free FSMs

- FSMs implement system control functionality
- Radiation induced failures in FSMs can have severe consequences on system operation
- For an FSM with N states, at least log2(N) bits are used to store state vector
  - Unless N is a power of two, “illegal states” exist
  - Illegal states are state vector values which can never be reached by the FSM
  - States where the FSM is not supposed to enter during its intended normal operation

Fail-safe, Dead-lock Free FSMs

- SEUs in the FSM state elements can cause the following problems:
  - "**illegal transitions**" between legal states are those which occur when the nominal sequence of states is modified
    - The new state, resulting from an illegal transition is a legal state, but it is not what is expected according to the previous state and the inputs
    - This can result in malfunction in rest of the logic affected by FSM state and output vectors
  - transitions into "**illegal states**". Depending on how the FSM is implemented and on the input vectors following the entry into an illegal state, two cases can be distinguished:
    - The illegal state reverts to a legal state after one or more clock cycles.
      Malfunctions can occur before the FSM goes back to a correct state.
    - The illegal state is persistent, the FSM remains locked in this state and can be recoverable only by a system reset.
      If this persistent FSM "deadlock" is not detected and reset, malfunctions can occur
  - the FSM output vector can also take an undefined (illegal) value

Fail-safe, Dead-lock Free FSMs

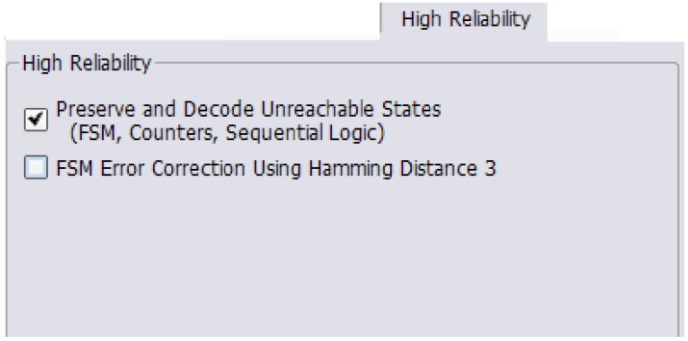
- Several FSM-specific techniques can used mitigate the FSM problems in order to create what is sometimes called "**fail-safe**", "**fault tolerant**" or "**deadlock-free**" state machines
- Commercial HDL synthesis tools usually recognize FSMs and their unreachable/illegal states and can be configured by the user to create additional circuitry that:
  - brings the FSM out of any illegal state into, for example, a legal idle state or another FSM deadlock recovery procedure.
  - to create an illegal-state-reached signal
- Term "fail-safe" can be misleading…
  - Even if a full deterministic decoding of all possible illegal states is done to avoid that FSM enters a persistent dead-lock situation, disruption in the nominal sequence of states and an eventual corruption of the output vectors can always lead to a temporary malfunction of the system

Fail-safe, Dead-lock Free FSMs

- FSMs can be also protected against radiation effects by other more generic mitigation methods
  - E.g. TMR, DMR with parity, Hamming codes etc.
- In many cases, the pure control FSMs only take a minor part of the resources when compared to the resources used for the data path and data storage
  - Selecting the highest available protection level for FSMs (i.e. applying several mitigation techniques) in a design is often also an affordable choice because it does not introduce significant area, power or performance overheads
  - E.g. it has been seen in some designs that all FSM FFs were protected by TMR, whereas the data-path FFs use a 'lighter' protection, such as error detection with a simple parity bit

Use of “when others”

- Using the “when others” section in VHDL is a prerequisite to ensure a deterministic way out of illegal states
- It is however – in general – not sufficient because present synthesis tools are able to recognize illegal states and therefore can optimize away the associated logic unless certain HDL coding and synthesis steps are done
- Synopsys Synplify Premier option “***Preserve and Decode Unreachable States***” prevents the tool from optimizing away logic associated to illegal states
  - Note: that this works only in conjunction with the proper coding “when others” to define what to do next when an illegal state is reached
  - The equivalent to this user-configurable option is attribute “***syn_safe_case = 1***” which can be added as a comment in the HDL code of the FSM and which affects only locally to an HDL architecture or module declaration, where the FSM is coded



Credit: Synopsys

Use of VHDL “when others”

- FSMs can be coded with a “Hamming distance 3” by setting the Synopsys Synplify Premier synthesis tool option (globally) or by setting locally in the HDL code the attribute “***syn_fsm_correction***”
- Implementation of TMR and DMR or DWC (Duplication With Compare), can be performed in the Synopsys Synplify Premier synthesis tool by using the attribute ***'syn_radhardlevel***' at architecture/module level
- It is important to enable the voting of feedback loops (‘***'syn_vote_loops***’)
- User must carefully study what attributes and synthesis constraints options are offered by the HDL synthesizer tool in order to recognize FSMs, implement them with the desired code-style and Hamming distance and to ensure a proper way out of illegal states

Synopsys Synplify attribute *syn_fsm_correction*

Implements FSM double-bit error detection (DED) and single-bit error correction (SEC) logic using Hamming 3 encoding.

syn_fsm_correction Values

Value	Description
hamming3	Builds single-bit error correction logic with Hamming 3 encoding.
hamming3_ded	Builds single-bit error correction and double-bit error detection logic with Hamming 3 encoding.
hamming3_ded_recovery	Builds single-bit error correction and double-bit error detection logic with Hamming 3 encoding along with default state recovery for double-bit error.

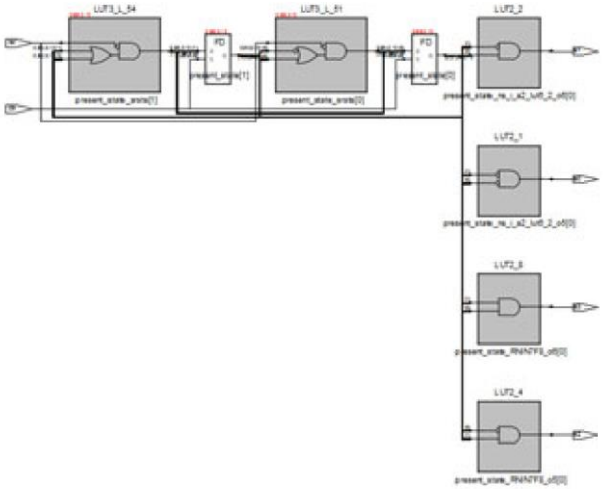
```
entity test is
port (a input std_logic;
      b out std_logic);
end test;


architecture rtl of test is
attribute syn_fsm_correction : string;
attribute syn_fsm_correction of rtl : architecture is "hamming3";
```

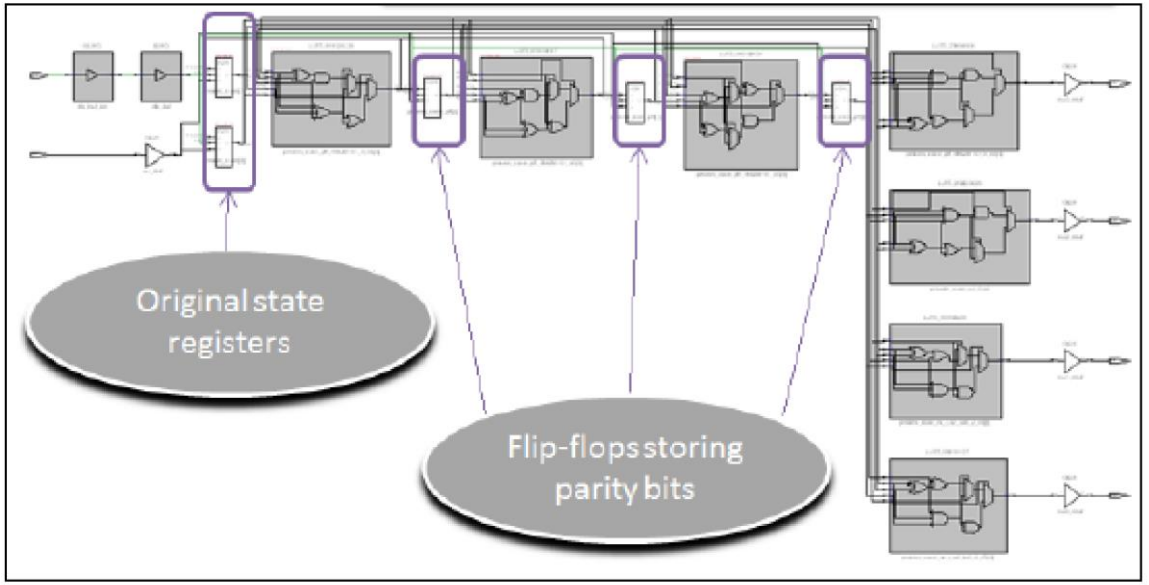
Credit: Synopsys

Synopsys Synplify attribute *syn_fsm_correction*

Without error correction:



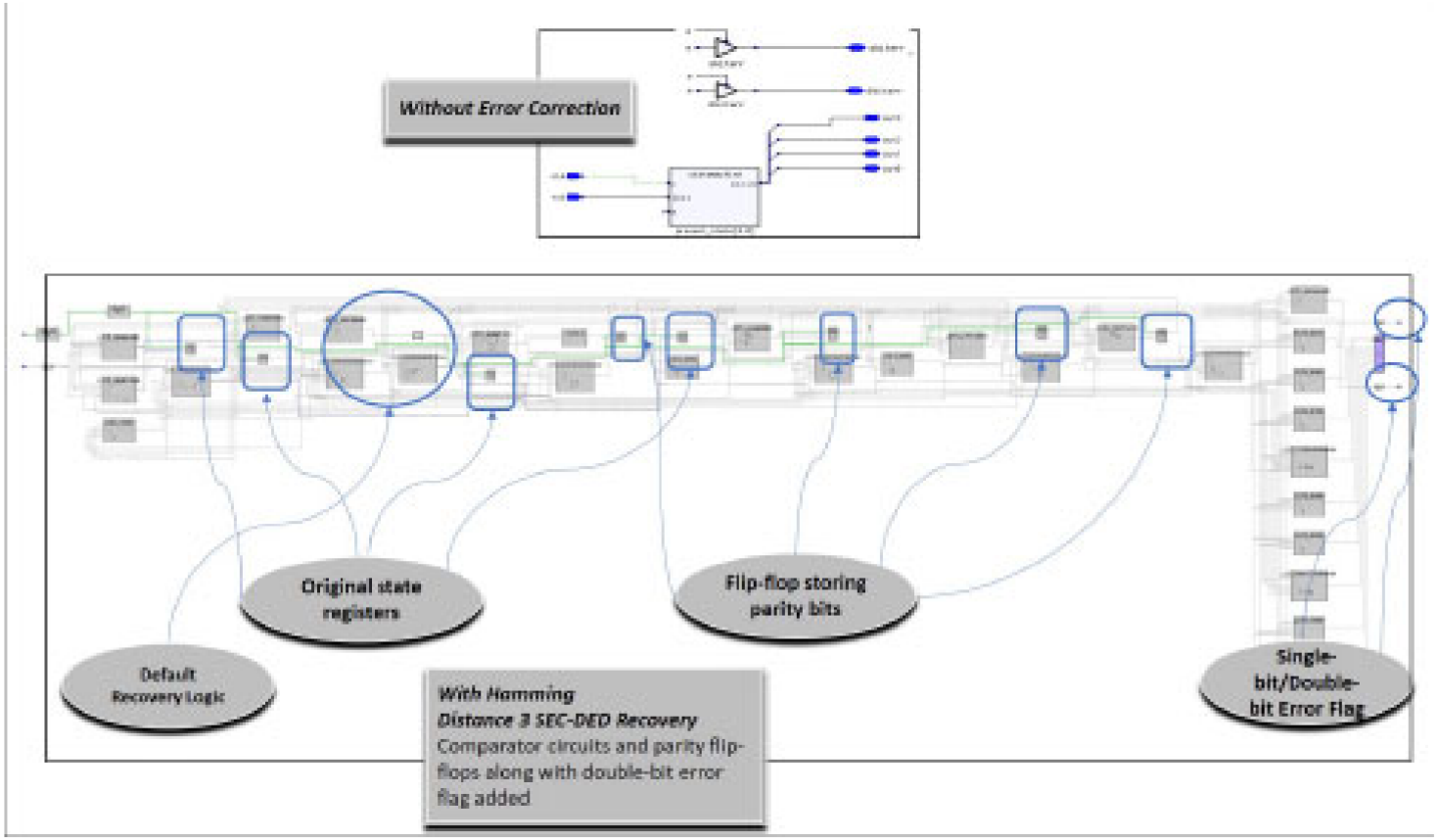
With Hamming-3 EC comparator circuits and parity flip-flops added:



Credit: Synopsys

Synopsys Synplify attribute *syn_fsm_correction*

With Hamming3_DED_RECOVERY, EC comparator circuits and parity flip-flops along with default state recovery on double error detection are added:



Credit: Synopsys

Synopsys Synplify attribute *syn_radhardlevel*

The syn_radhardlevel attribute can use the following options:

syn_radhardlevel = none | block_tmr | distributed_tmr | duplicate_with_compare | tmr

VHDL

```
attribute syn_radhardlevel : string;
attribute syn_radhardlevel of topLevelModule: architecture is
"none|block_tmr|distributed_tmr|duplicate_with_compare";
attribute syn_radhardlevel: string;
attribute syn_radhardlevel of Object:Object Type is
"none|block_tmr|distributed_tmr|duplicate_with_compare";
```

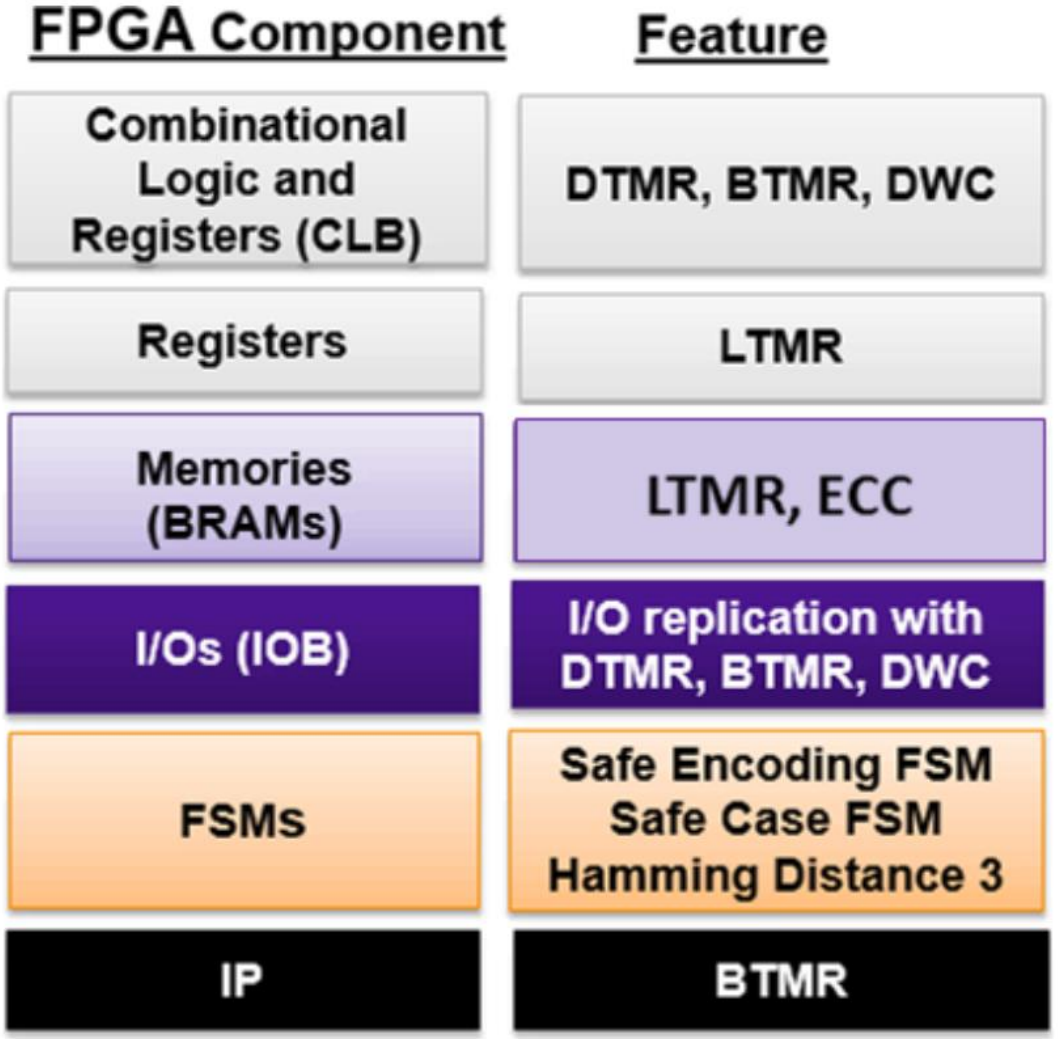
Credit: Synopsys

Ensuring Safe Operation in FPGA designs

- Best mitigation methodology to use: consider the following:
  - First, decide whether you want to trade off reliability for area, performance, and throughput
  - Then, determine where and what protection is needed, depending on the type of device you are using
  - Finally, select type of error mitigation methodology to use. You can insert circuitry or have the software automatically create error recovery

Credit: Synopsys

Which high reliability features to use to protect critical design components



Credit: Synopsys