

# FPGA in HEP experiments: challenges and radiation effects

Tullio Grassi, Univ. of Maryland

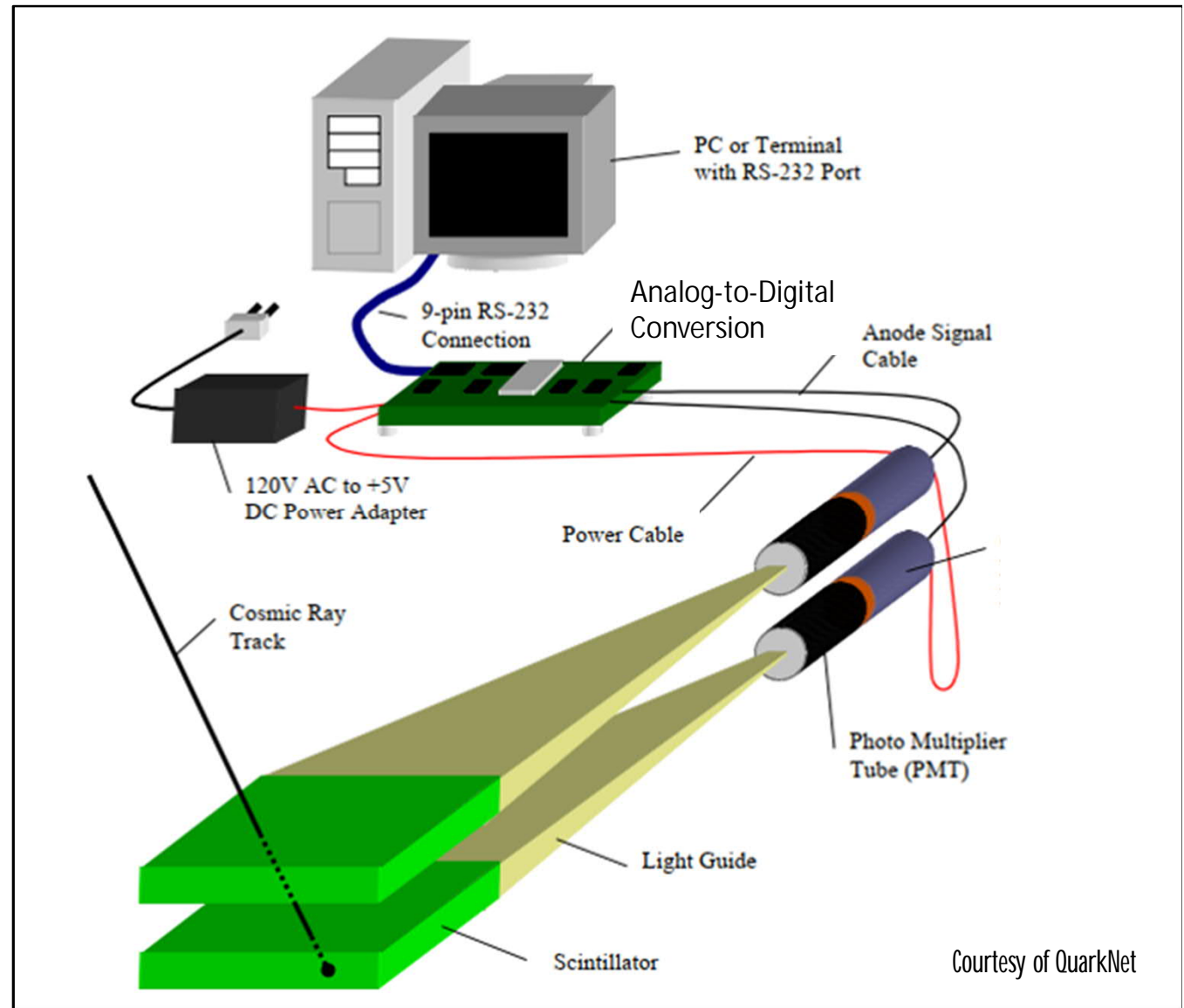
INFN National Laboratories of Legnaro, 4th of April 2019

# Why we use FPGAs in HEP ?

Example:

a cosmic ray detector commonly used in student laboratories.

The detector system includes two scintillators, PMTs, Analog-to-Digital Conversion, data acquisition (in this case a computer).



Courtesy of QuarkNet

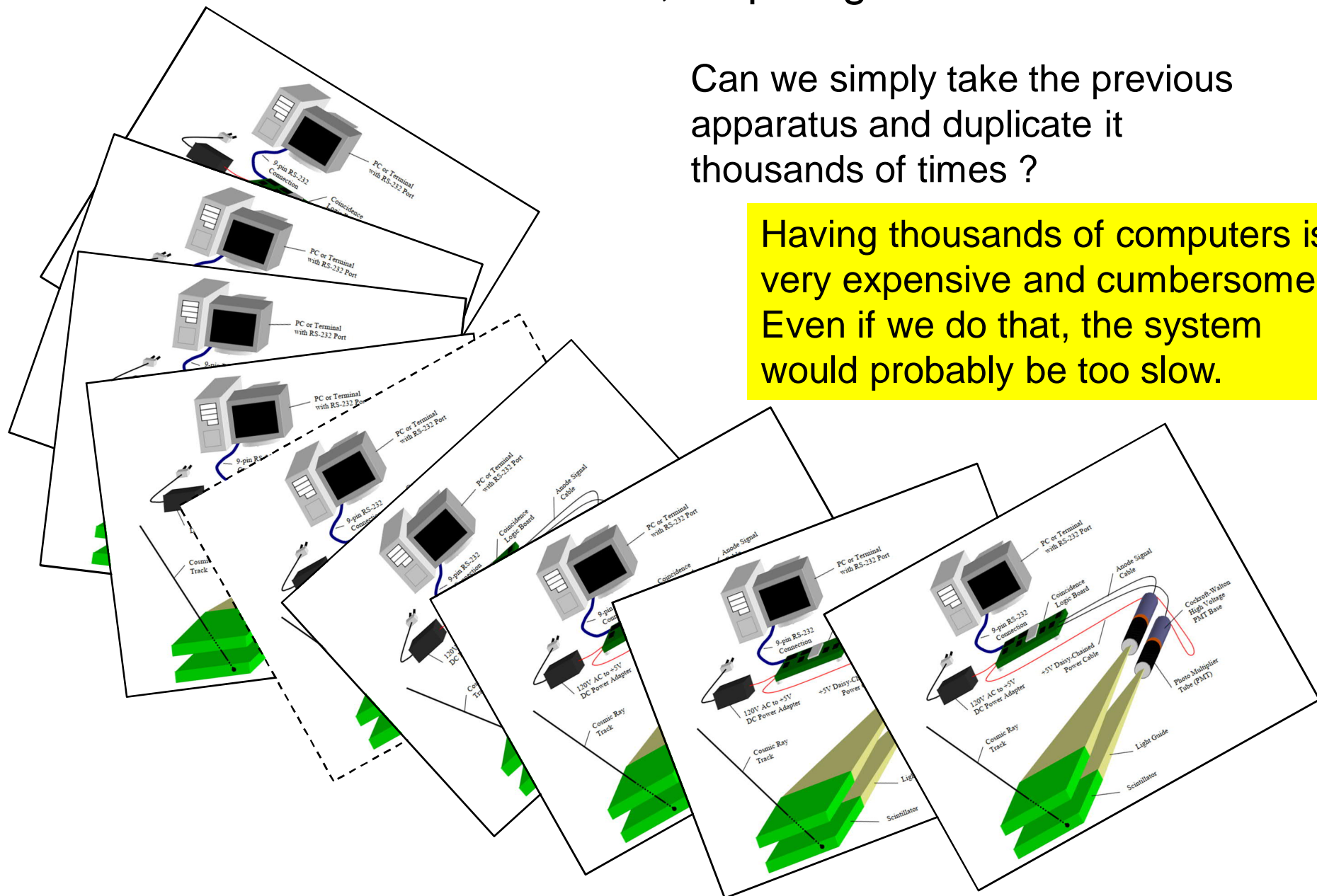
Real detector systems often have thousands of channels, acquiring events at  $\sim$  MHz rates.

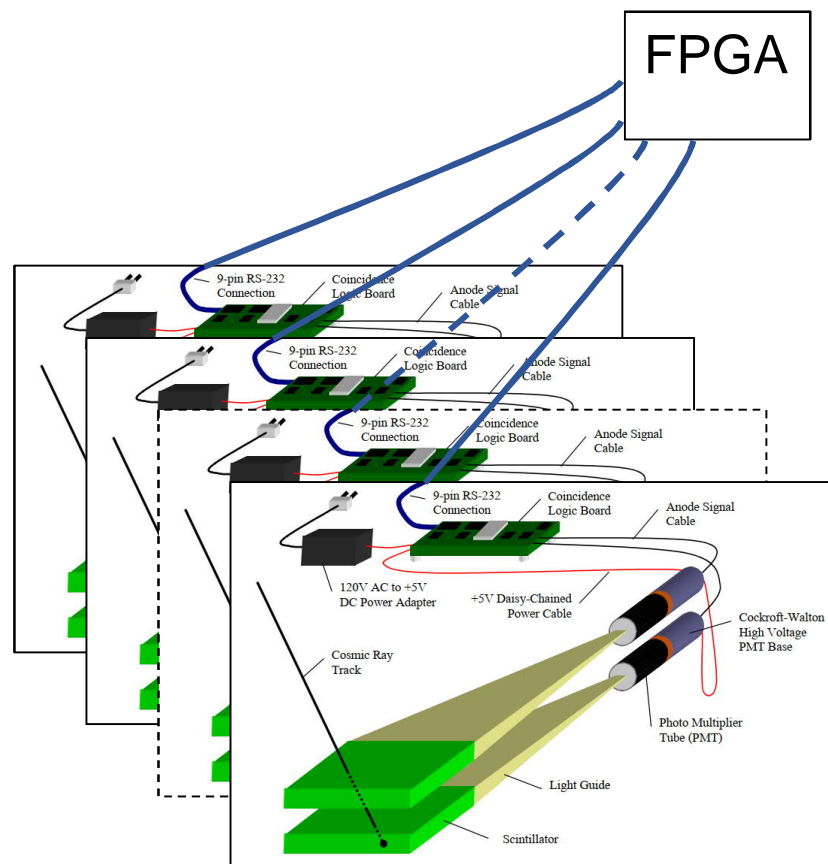
Can we simply take the previous apparatus and duplicate it thousands of times ?

Real detector systems often have thousands of channels, acquiring events at  $\sim$  MHz rates.

Can we simply take the previous apparatus and duplicate it thousands of times ?

Having thousands of computers is very expensive and cumbersome. Even if we do that, the system would probably be too slow.





In most cases, we can:

- buy one or a few FPGAs at a price similar or lower than computer systems
- program them
- obtain performances better than computers.

In HEP detectors and accelerators (as well as in related fields) FPGAs have become extremely useful to build digital hardware tailored to specific requirements.

But what are FPGAs ?

The core of FPGAs is made of lots of small, programmable digital circuits.

→ So, let's look at what are digital circuits

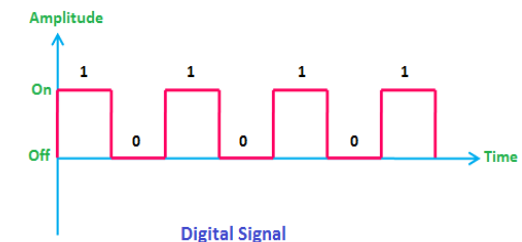
# Analog versus Digital electronic circuits

Both analog and digital circuits use time-varying electrical signals in order to represent information.

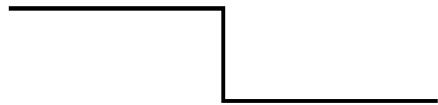


However:

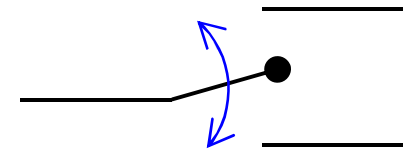
- Analog circuits use signals that can take any value within a continuous range
- Digital circuits use signals that typically can take only one value between two possible values
  - The values are called 1 and 0 (ON and OFF, HIGH and LOW, TRUE and FALSE, etc.) . In certain circumstances this is called a bit
  - “Logic circuit” is a synonym of “digital circuit”



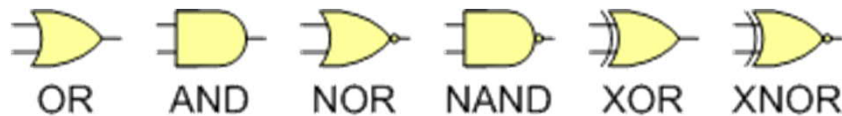
# Digital Logic: basic blocks



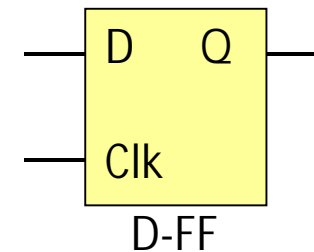
**Wires:**  
carry signals



**Simple switches:**  
(dis)connect wires



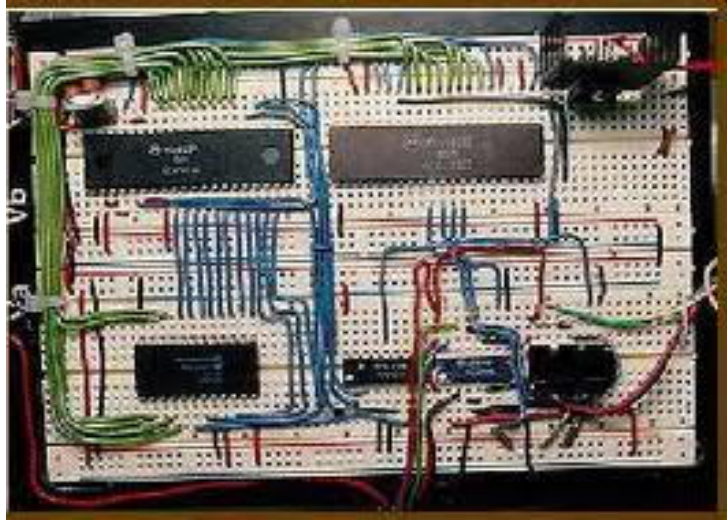
**Logic Gates:**  
do a simple data processing



**Various types of memory element (registers, Flip-Flop, RAM, etc):** can store bits



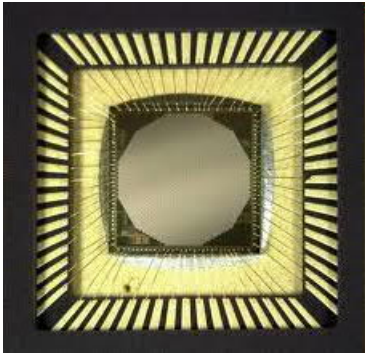
# Options for building digital circuits



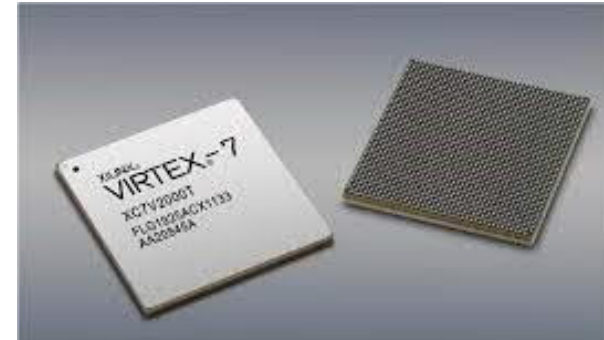
Breadboard circuit



PCB = Printed Circuit Board



ASIC = Application Specific  
Integrated Circuit



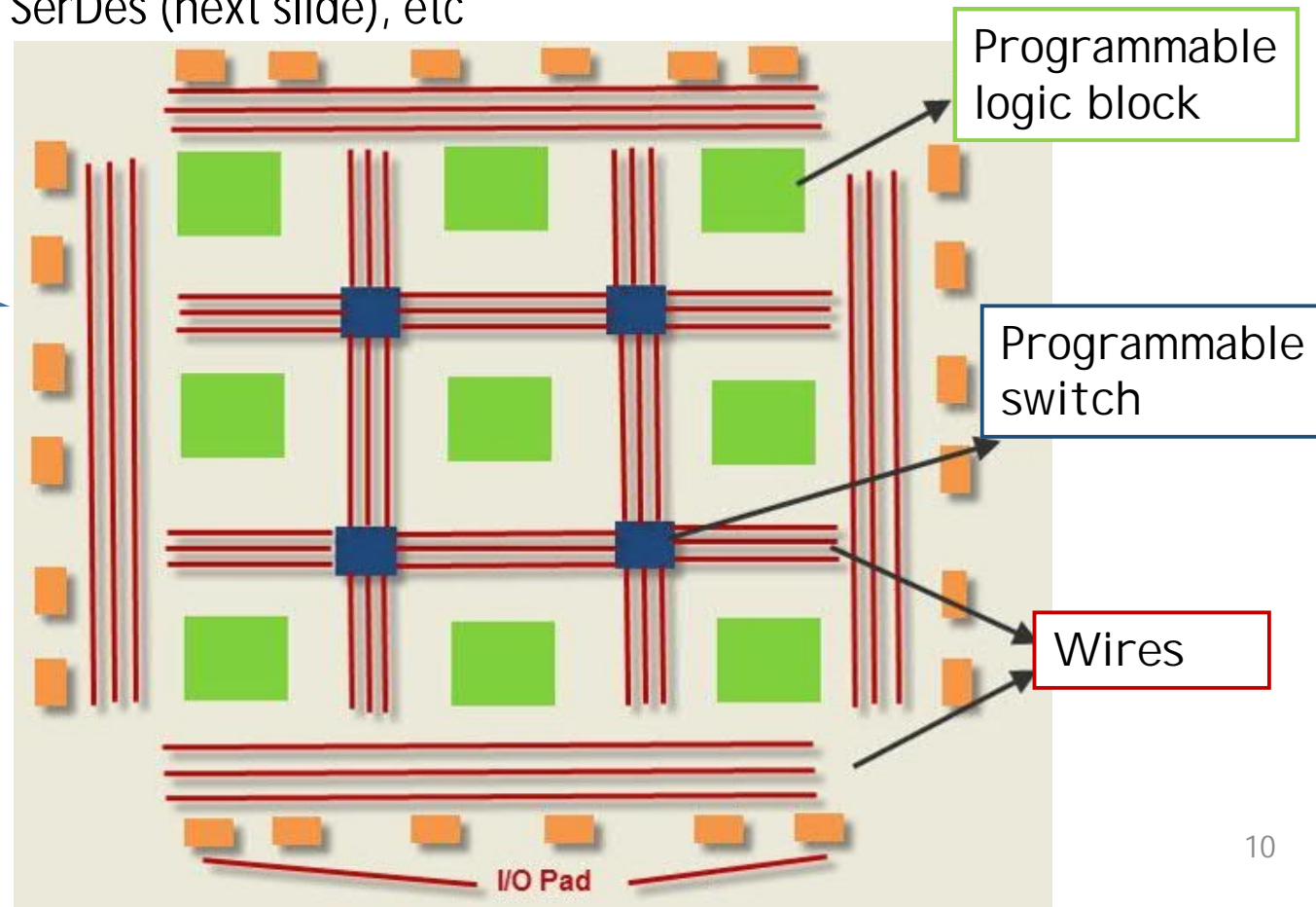
FPGA = Field Programmable  
Gate Array

# FPGAs are a special type of ASIC

- many simple Programmable Logic Blocks (logic gates + flip-flops)
- many simple programmable switches
- many, many wires
- an FPGA designer can create a complex logic circuit just programming the FPGA
- over the years, other (analog or digital) blocks have been added: PLL, RAM, multipliers, CPU, SerDes (next slide), etc

## Simple FPGA architecture

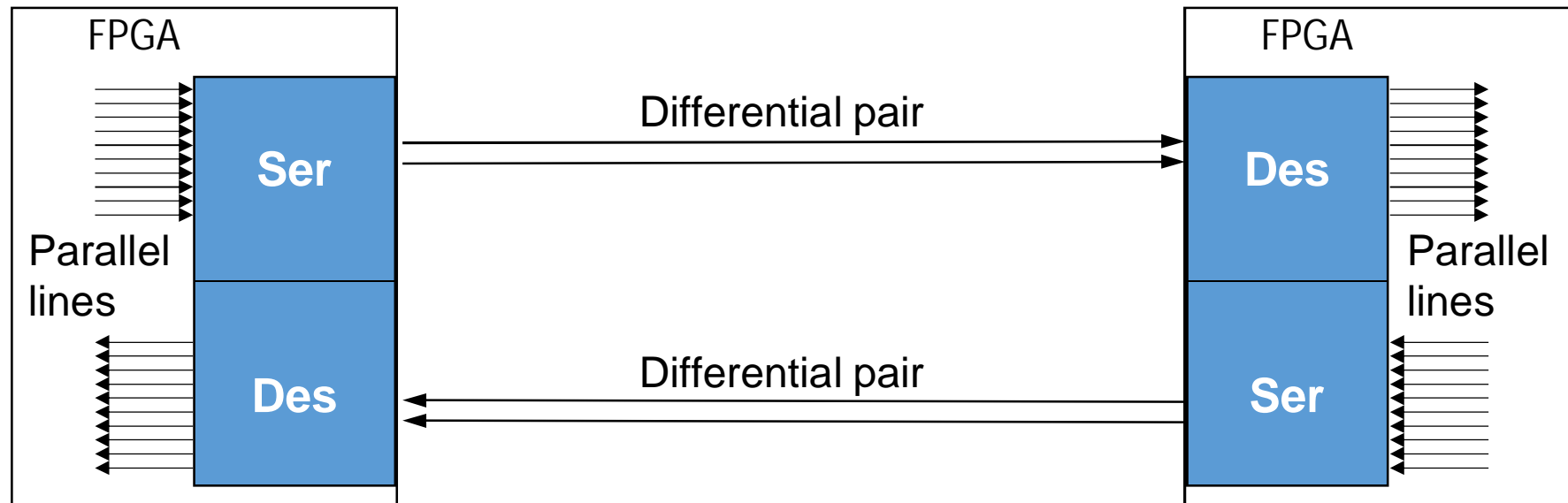
All programmable logic blocks and programmable switches together are sometimes called “user logic” or “FPGA fabric”.



# SerDes (Serializer/Deserializer)

A Serializer/Deserializer is a pair of blocks commonly used in high-speed communications. One block converts parallel data into serial data; the other block converts serial data into parallel data.

This allows to transmit lots of bits using a small number of FPGA input/output pins. Normally the serialized signal is sent out to a differential pair (= two electrical lines with controlled characteristics).



Sometimes the differential electrical signal is converted to an optical signal, which can travel over longer distances.

The details of SerDes are much more complex than shown in the picture (clocking, etc).<sup>11</sup>

# Comparing FPGA, CPU (and GPU)

CPU advantages	FPGA advantages
<ul style="list-style-type: none"><li>• better with floating point numbers</li><li>• programming a CPU is normally easier than programming an FPGA (does not require to understand digital electronics)</li><li>• faster compilation</li><li>• easier code portability</li><li>• lower unit cost</li></ul>	<ul style="list-style-type: none"><li>• more flexible processing</li><li>• more flexible input/output</li><li>• parallel processing</li><li>• better with multi-clock systems</li><li>• better with time-critical operations</li></ul>

The comparison would be similar for GPU vs FPGA.

if you can completely reach your targets with a CPU, then use a CPU.

If you can not, then consider a GPU or an FPGA.

More and more often, FPGAs and CPUs (or GPUs) are complementary: they co-exist in the same system and perform different tasks.

# FPGA vs ASIC

FPGA Advantages	ASIC Advantages
<ul style="list-style-type: none"><li>•<b>Faster time-to-market</b> - no layout, masks or other manufacturing steps are needed</li><li>•<b>Lower constant/initial cost</b></li><li>•<b>Simpler design cycle</b> - due to software that handles much of the routing, placement, and timing</li><li>•<b>More predictable project cycle</b> due to elimination of potential re-spins, wafer capacities, etc.</li><li>•<b>Reprogramability</b>: a new configuration can be uploaded</li></ul>	<ul style="list-style-type: none"><li>•<b>Full custom capability (including analog)</b> - since device is manufactured to design specs</li><li>•<b>Lower unit costs</b> - for very high volume designs</li><li>•<b>Smaller form factor</b> - since device is manufactured to design specs</li><li>•<b>Higher clock speeds</b></li></ul>

# Types of FPGAs

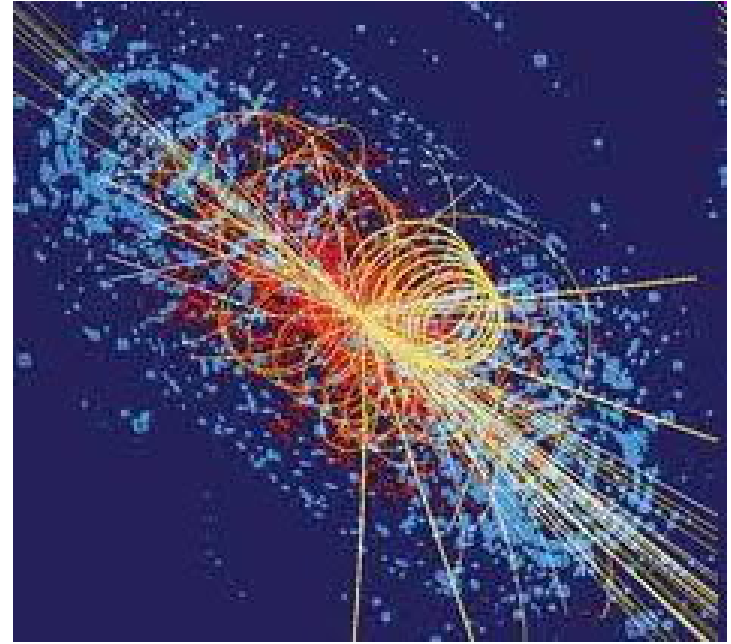
All FPGAs presently on the market are build using CMOS process technologies.  
But the technology used for the programming element (a.k.a. configuration memory) can vary.

Technology of the programming element	Vendors
SRAM (Static RAM). Standard CMOS process	Atmel, Intel, Lattice,Xilinx
Anti-fuse: non reversible (one-time-programmable). Non-standard CMOS process	MicroSemi, Aeroflex?
Flash: flash cells with floating gate. Non-standard CMOS process	MicroSemi
(IPs for SRAM FPGAs)	(NanoExplore)

# Uses of FPGAs outside HEP

- Telecommunication
- Automotive
- Aerospace and Defense
- Medical Electronics
- ASIC Prototyping
- Audio
- Broadcast
- Consumer Electronics
- Data Center
- Distributed Monetary Systems
- High Performance Computing
- Industrial
- Scientific Instruments
- Security systems
- Video & Image Processing
- digital signal processing
- bioinformatics
- controllers
- computer hardware emulation
- voice recognition
- cryptography

# FPGAs in HEP



In HEP detectors and accelerators, commercial-grade FPGAs have become extremely useful to build digital hardware tailored to specific requirements, including non-standard data processing and interfaces.

In HEP, FPGAs are used both in areas with ionizing radiation, and without ionizing radiation.

Being commercial products, they are not (necessarily) tolerant to ionizing radiations.



# Environment of high-energy physics experiments

Case study: the environment of the LHC accelerator at CERN. This is the accelerator producing the highest radiation levels as of today. Experiments can experience the following conditions:

- Radiation: up to ~200 kGy (=20 Mrad) and  $10^{14}$  n/cm<sup>2</sup> over a 10 year period.
- Magnetic field: up to 5 Tesla
- Limited access: one access every ~5 years
- Limited space → limited cabling, limited cooling
- Limited material: this is to avoid modifications in the trajectory of the particles

# Problems induced by radiation on digital circuits

## 1. Cumulative effects

- TID = Total Ionization Dose. Measured in Grey = Gy = J/kg (or in rad: 1 rad = 100 Gy)
- Displacement damage: defects caused by Non Ionizing Energy Loss. It does not affect CMOS processes so it is not relevant for FPGAs.

## 1. SEE = Single-Event Effects are effects caused by a single ionizing particle, as:

- SEU = Single-Event Upset : a bit stored in a memory element is modified
- SET = Single-Event Transient: a digital signal is temporarily modified
- SEFI = Single-Event Functional Interrupt
- SEL = Single-Event Latchup : a short-circuit is created inside an integrated circuit
- SEGR = Single-Event Gate Rupture
- SEB = Single-Event Burnout → on high-voltage or high-power electronic circuits

# SRAM-based FPGAs: pros and cons

- Re-programmable
  - Can add features after installation, allows to try more complex logic
- Fastest FPGAs
  - Latest CMOS silicon processes (example: 16 nm for Xilinx UltraScale+)
  - LVDS ports: 1.6 Gbps. SERDES: 32 Gbps.
- Best TID tolerance
  - At least 5 kGy (some parts much more)
- Drawbacks
  - SEUs in the Configuration: for example a connection or a logic-gate can be modified
  - Worst power consumption (from SRAM configuration memory)
  - Need to reconfigure frequently ("scrubbing") , which requires even more power [11]

# Antifuse FPGAs: pros and cons

- Reliability:
  - No SEU on configuration
- Drawbacks:
  - Limited resources
  - Non-reprogrammable (a.k.a. One time Programmable)
  - Speed : 700 Mbps LVDS ports
  - TID tolerance : 800 Gy

→ Not much use on new HEP projects

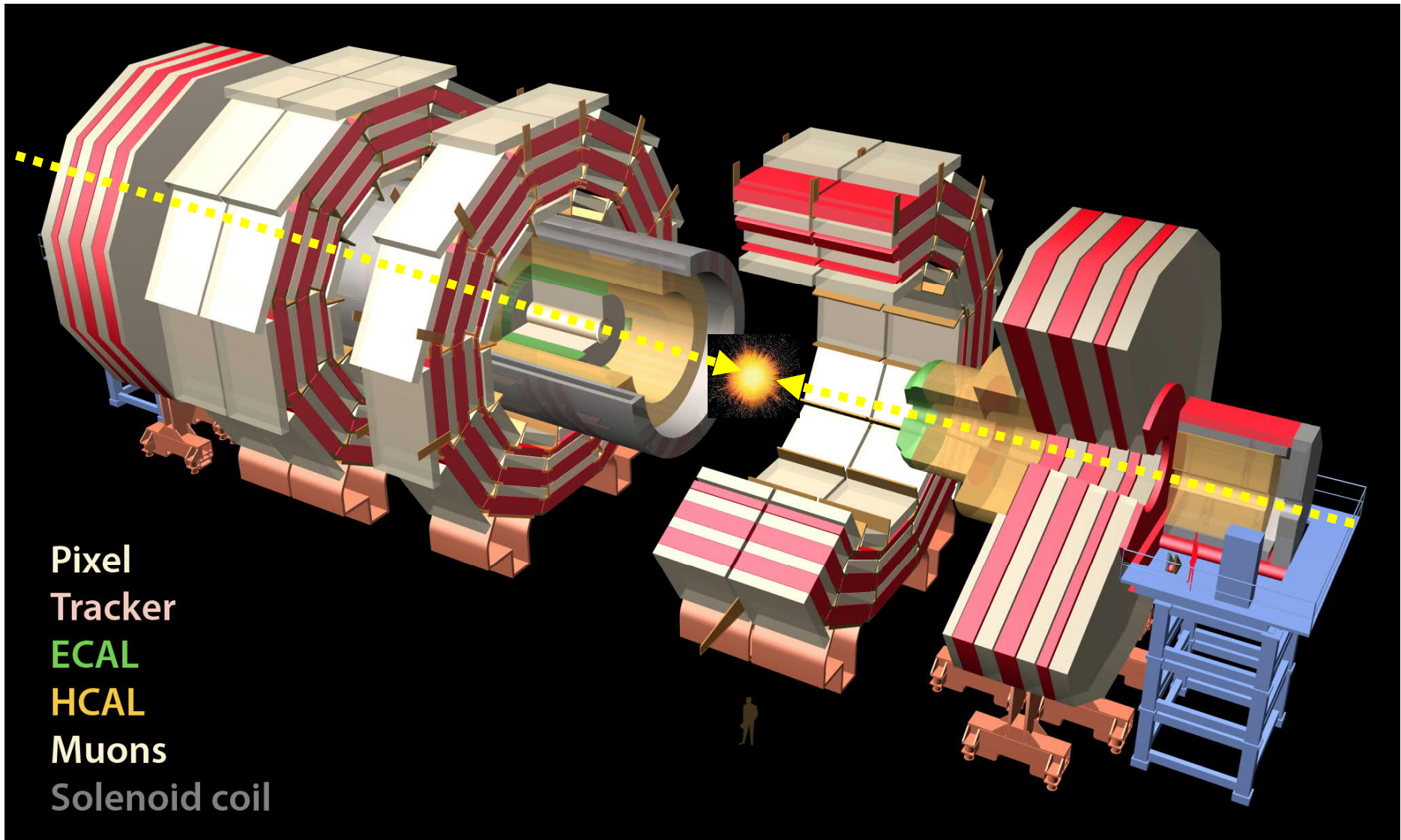
# Flash-based FPGAs: pros and cons

- Re-programmable
  - Can add features after installation, allows to try more complex logic
  - Normally it is the first function damaged by TID
- Reliability:
  - No SEU on configuration memory
- intermediated density and speed:
  - LVDS port: 1.25 Gbps. SERDES 12.7 Gbps (on PolarFire)
- Drawbacks
  - TID tolerance : ~700 Gy (on igloo2, SmartFusion2)
  - ability to reprogram fails at much lower TID
  - design tools not as mature as for SRAM FPGAs

# Use cases at CERN

System	Present FPGA family	Plans for next generation
nanoFIP (accelerators)	ProASIC3. [1]	investigating Smartfusion2 and nanoXplore FPGA [2]
LHC SciFi , Cal, and Muons	Antifuse AX , ProASIC	Igloo2
LHCb RICH	Antifuse AX , ProASIC	Xilinx Kintex7 [4]
ALICE ITS [7]	no FPGAs	Xilinx ultraScale(+) , ProASIC3L (scrubber).
ALICE TOF	ProASIC	Igloo2
ALICE TPC	SmartFusion2	
ATLAS muon RPC	Xilinx	Xilinx
ATLAS TGC Muon	Antifuse AX	Plan A : Xilinx Kintex-7 Plan B : PolarFire
CMS RPC Muon	Xilinx SPARTAN3; Actel ProASIC+ as blind scrubber, every 10 minutes.	Plan A : Xilinx Kintex7 and SmartFusion2. Plan B : PolarFire
CMS DT Muon	no FPGAs	PolarFire
CMS HCAL [10]	ProASIC3L, igloo2	

# A study case: the CMS detector



# Commercial grade vs RadTol grade

Commercial-grade FPGAs are industrial products not specifically developed to tolerate radiation.

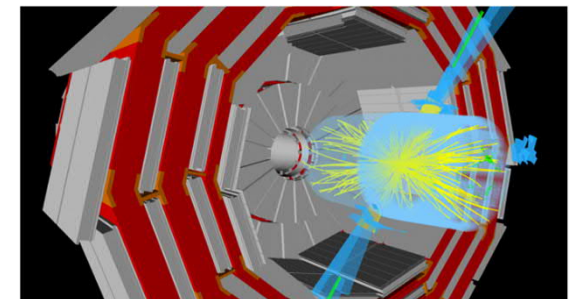


Some vendors also offer FPGAs specifically developed for radiation environments. These FPGAs can cost about 100 times more than commercial-grade FPGAs. They are targeted to space systems, where very small quantities of FPGAs are used per system.






HEP systems often use lots and lots of FPGAs, and cannot afford to buy the radiation-tolerant grade

→ on the rest of this presentation will focus on commercial-grade.





## Summary Radiation effects on present CMOS-process commercial-grade FPGAs (list not exhaustive)

	Destructive effects (SEL, etc)	SET	SEU on configuration	TID
<b>SRAM</b> [12, 13]	No	Yes	Yes	
<b>Anti-fuse</b> [12]	No	Yes	No	
<b>Flash</b> [8]	No	Yes	No	

SEU on the user logic depends on the design programmed in the FPGA, but it is similar for all commercial-grade FPGAs.

There are more effects, often family-specific, for example they are related to PLLs, internal voltage regulators, etc.

# SEU-mitigation techniques

Techniques to mitigate SEUs on the user logic:

- TMR
- fault-tolerant FSMs
- EDAC codes (mostly for SEUs on user memory)
- Watchdogs

Techniques to mitigate SEUs on the SRAM configuration memory:

- Scrubbing

Each of this technique can be implemented in a number of different way.  
Moreover these techniques depend on the FPGA technology.

# SEU-mitigation tools (1/2)

There are two commercial synthesisers that can help with SEUs on the user logic:

1. **Synplify Pro®** (cannot do TMR on IP blocks)
2. **Precision® Hi-Rel** (can do TMR on IP blocks)

Do not blindly trust these tools, they had bugs in the past. It is recommended to understand the concept of the technique that will be used by the tool, what the tool exactly does and occasionally double-check the results.

Example

**Synplify Pro®** has a directive named "syn\_safe\_case" , for FSM synthesis. NASA experts and also support people from the company advised NOT to use it because it is not safe !!

# SEU-mitigation tools (2/2)

There are also SEU-mitigation tools developed by academia and research lab.



- Tools from the HEP community
  - ❑ do they have long-term support, in order to work with new FPGAs?
- Tools from the space community
  - ❑ do they remain freely available once they have good funding and results?

# Radiation-mitigation techniques : warnings

- **incorrect implementation can increase errors !!!**
- **no strategy is 100% fail-safe**

## Warning

Do not mitigate failure mechanisms that have insignificant contribution to the overall failure rate. This:

- adds risk
- slows down system
- can provide a false sense of protection.

# Design techniques: SEU prevention on SRAM configuration memory (= scrubbing)

The configuration of SRAM-based FPGAs can be upset by a particle.

“Naïve” solution: we can reprogram the FPGA periodically:

Problem: sometimes the configuration will be upset much before reprogramming it, other times it will be reprogrammed when there is no need : not very efficient

Better solution: certain FPGAs allow to monitor the status of their own configuration, so an external controller can detect when the configuration is upset and trigger the reprogramming : ACTIVE RECONFIGURATION

Problem: we need a radiation-tolerant controller.

Problem #2: during the reprogramming phase, the FPGA will loose all data and capability (dead-time, data loss).

Even better: certain FPGAs allow to monitor and reprogram only a part of the FPGA itself, this reduced the data-loss : ACTIVE PARTIAL RECONFIGURATION .

Research is ongoing in order to eliminate the data-loss and dead-time, using redundancy strategies.

# SET-mitigation techniques

Prevention of SETs:

- TMR that includes combinatorial logic
- filtering with guard-gate

The Precision Rad-Tolerant tool can do TMR of combinatorial logic. But it works only for certain FPGA families.

Commercially available tools are evolving rapidly wrt SEU and SET → keep watching.

In the Microelectronics Section of CERN, some designers have been using a custom script that generates automatically TMR on registers and combinatorial logic.

The script supports only Verilog 1995 designs.

The script is available to people registered on the CERN FPGARadTol web page.

# SEL-mitigation techniques

Definition: in an integrated circuit, a latch-up is a unwanted short-circuit caused during the operation of the circuit. Traditionally (in non radiation environments) it can be caused by current injection or overvoltage.

A SEL is a latch-up caused by a particle crossing the circuit.  
It can happen on the internal nodes (while normal latchups occur mostly on the I/Os due to ESD).

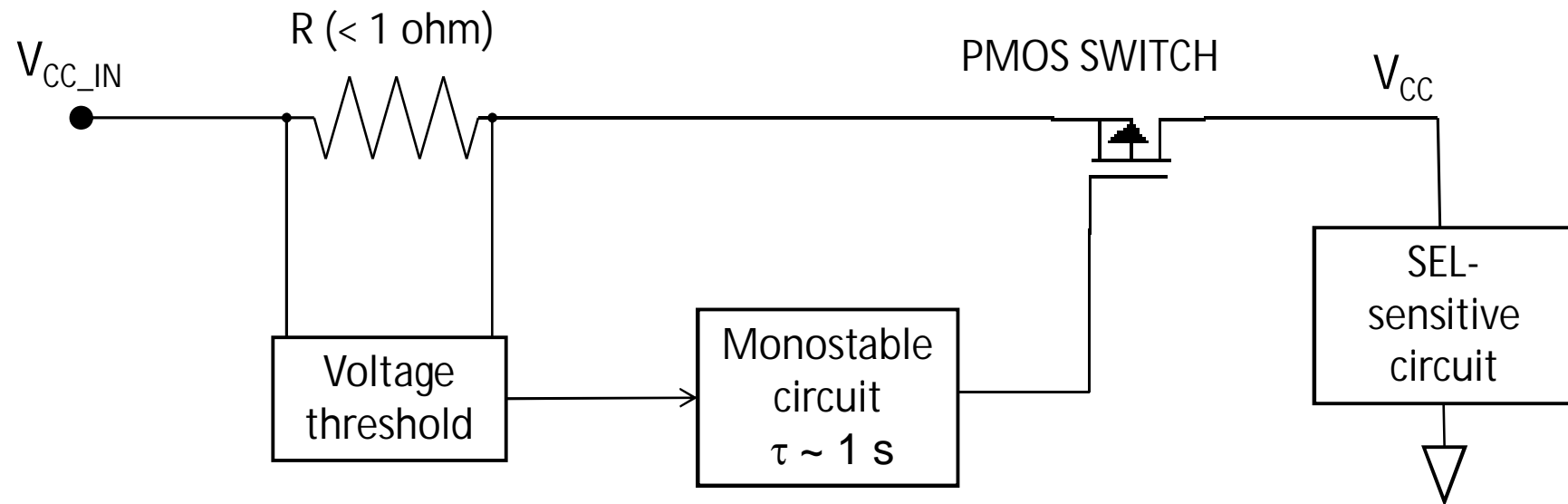
Most modern FPGAs are immune from SEL.

But other commercial components can be affected by SEL.

→ external SEL protection circuit (next slide).

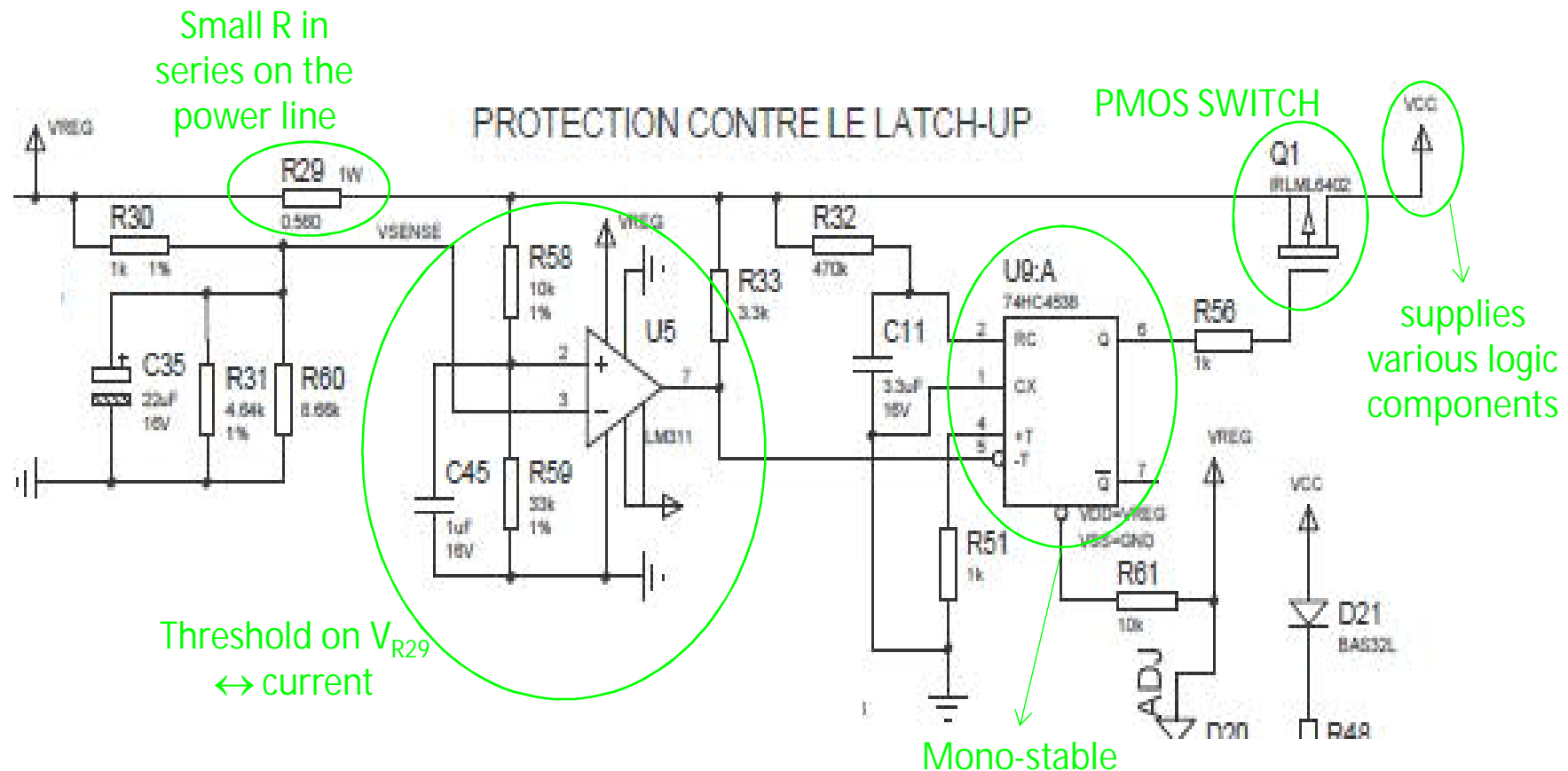


# SEL-protection circuits: a generic scheme



When a SEL-sensitive circuit develop a SEL, it draws more current. An external circuit can detect this situation. Then it can turn the power off and then on again.  
Problem: also the protection circuit can be affected by radiation. But being a simpler circuit, it is possible to design it so that it is very unlikely that it develops a problem.

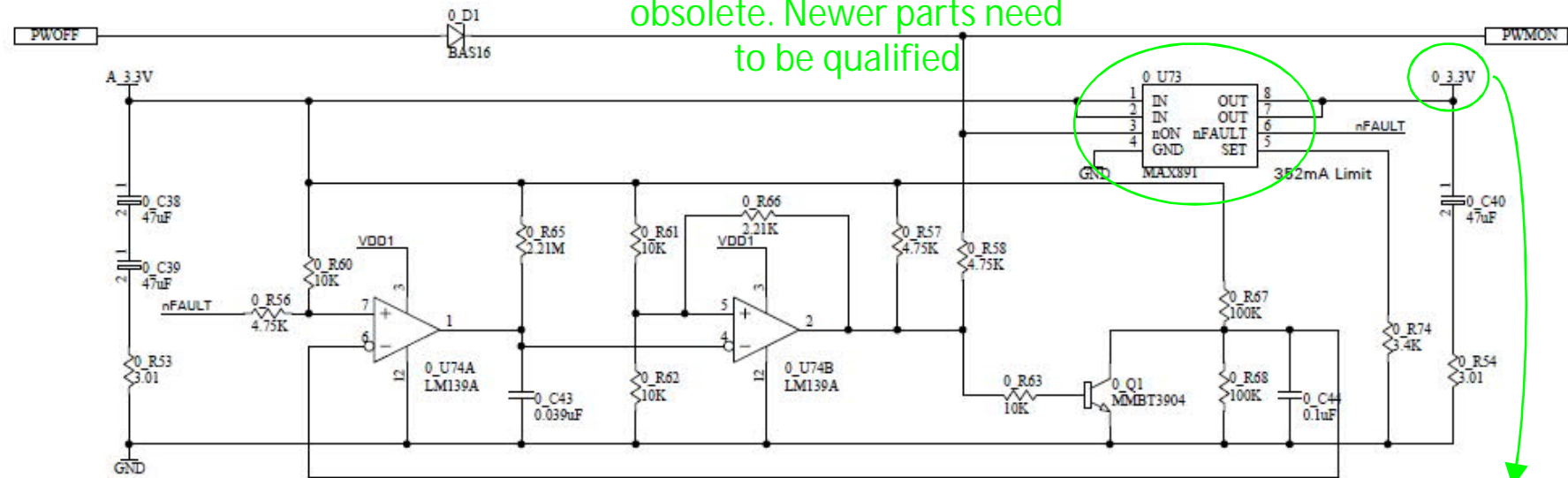
# Prevention of SEL [RepFIP card, LHC]



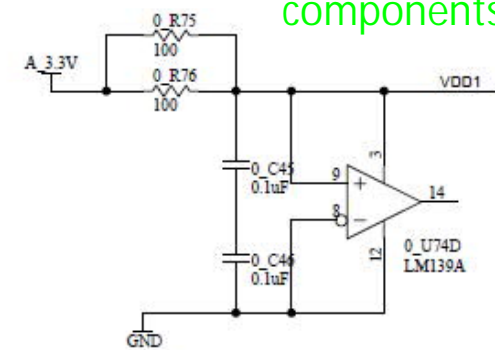
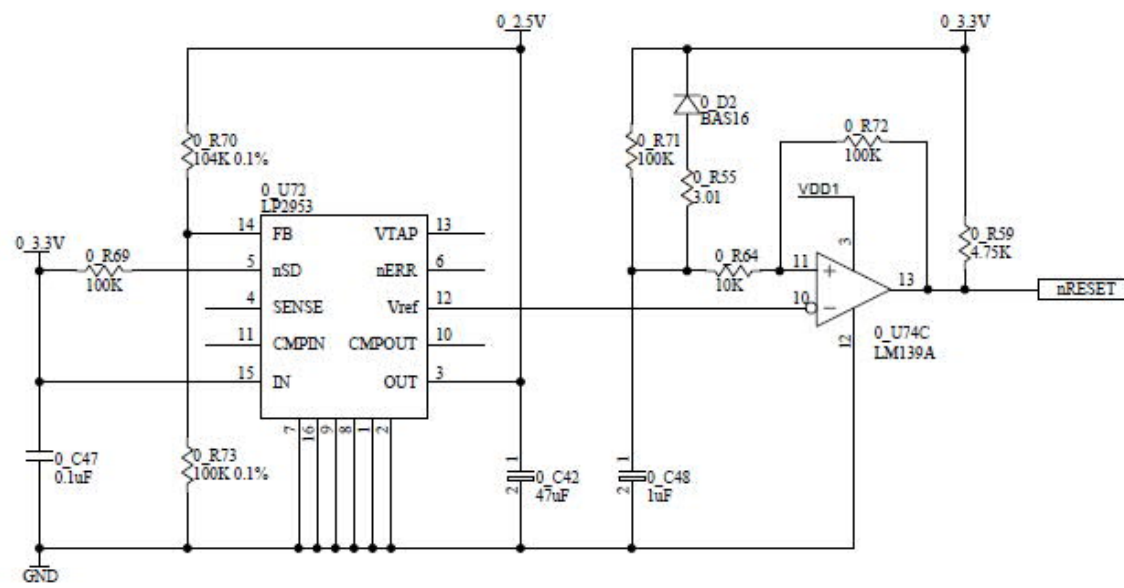
A few samples have been tested under radiation, in one case the component U9 (L4931CD25) has failed at  $2 \times 10^{11}$  p/cm<sup>2</sup> (corresponding to TID 100 Gy). A few dozen cards are presently installed.

MAX891 "smart" switch: now

obsolete. Newer parts need to be qualified



- supplies various logic components



Title C:\jinf\prote\jinf.ddb - Documents\ssf_0.sch		
Size B	Number	Revision 35
Date:	21-Jun-2003	Sheet 25 of 25
File:	C:\jinf\prote\jinf.ddb	Source: B...

# SEFI

SEFI = Single Event Functional Interrupt

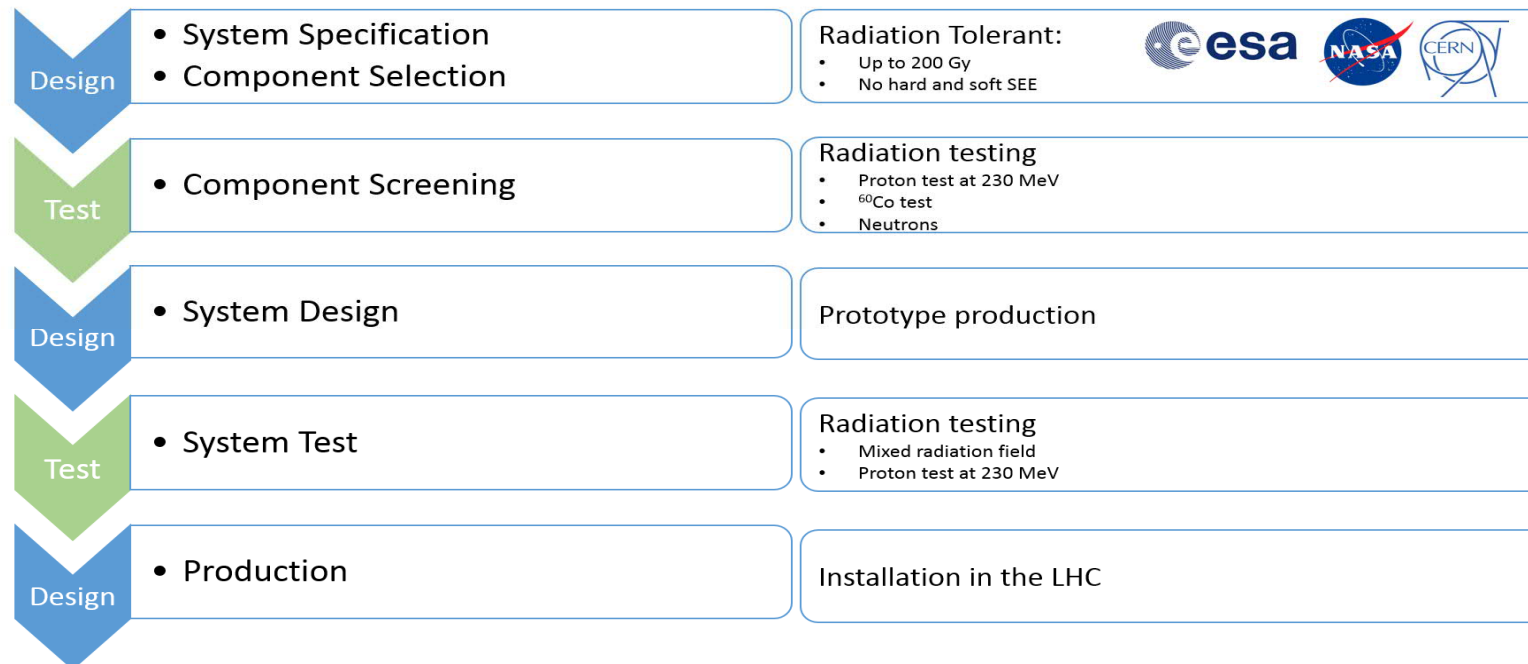
The definition can vary according to the authors, but it normally indicates an SEE which affects the entire device, for instance:

- power-on reset
- global reset,
- global tristate
- problems in the circuit that program the rest of the FPGA

For an FPGA, it is difficult or impossible to mitigate SEFI within the FPGA design. SEFI could be mitigated at the system level.

# Testing FPGAs in a radiation environment

- Commercial FPGAs are a special case of commercial components (a.k.a COTS)
- There are typical procedures for radiation testing and qualification of COTS. These procedures can be applied or adapted to FPGAs
- Need to know the expected radiation levels
- Initial radiation tests are normally TID tests: done with gamma or x-rays
- Main radiation test usually done in proton beam
- Often FPGAs need external components very near them (memories, clock oscillators, jitter cleaners, voltage regulators, etc). This is (part of) an electronic system, and it is normally too big to be tested with a proton beam.
- Once a (prototype) system is ready, it should be tested: where?



# CHARM: a radiation facility for system testing at CERN

CHARM = Cern High Energy Accelerator Mixed-Field Facility

## Main purpose

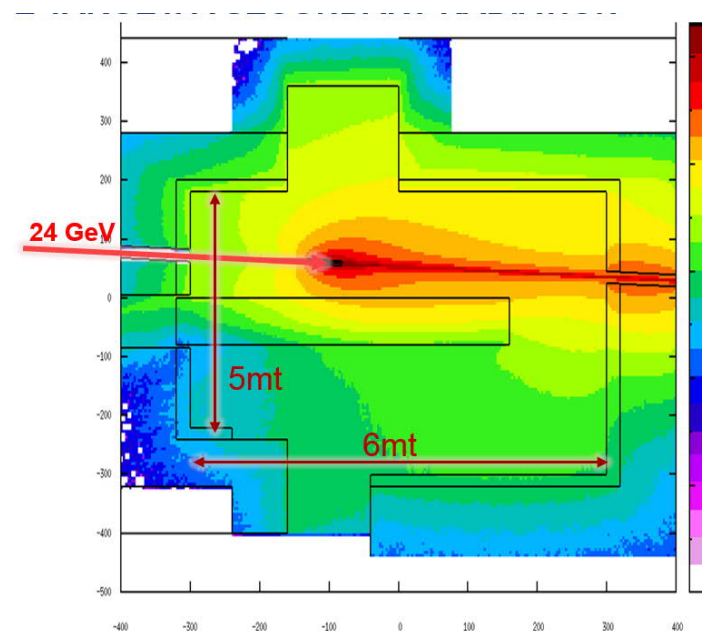
Radiation tests of electronic equipment and components in a radiation environment similar to the one of the accelerator

## Large dimension of the irradiation room

- Large volumes electronic equipment
- High number of components
- Full systems

## Numerous representative radiation fields

- Mixed-Particle-Energy: Tunnel and Shielded areas, atmospheric and space environments
- Direct beam exposure (proton beam 24 GeV)



# Future of FPGAs

- Present FPGAs are built using CMOS silicon processes
- In the near future, some FPGAs will be built using FinFET processes
  - The effects of radiation will change !
  - This will affect mitigation techniques, testing procedures, etc

# References

- [1] [https://edms.cern.ch/ui/file/1183301/1/LargeScale\\_RadTests\\_PSI\\_17Dec2011.pdf](https://edms.cern.ch/ui/file/1183301/1/LargeScale_RadTests_PSI_17Dec2011.pdf)
- [2] <https://wikis.cern.ch/display/HT/Distributed+IO+Tier>
- [4] [https://indico.cern.ch/event/608587/contributions/2614176/attachments/1521369/2376930/Radiation\\_Hardness\\_Studies\\_and\\_Evaluation\\_on\\_SRAM-Based\\_FPGA\\_for\\_HEP\\_Vlad\\_Placinta.pdf](https://indico.cern.ch/event/608587/contributions/2614176/attachments/1521369/2376930/Radiation_Hardness_Studies_and_Evaluation_on_SRAM-Based_FPGA_for_HEP_Vlad_Placinta.pdf)
- [7] [https://indico.cern.ch/event/698929/contributions/2866600/attachments/1626142/2583382/WP10\\_PRR\\_Environment\\_v0.pdf](https://indico.cern.ch/event/698929/contributions/2866600/attachments/1626142/2583382/WP10_PRR_Environment_v0.pdf)
- [8] Private communication, Federico Faccio  
[https://twiki.cern.ch/twiki/pub/FPGARadTol/InformationOfInterest/SEL\\_on\\_ProASIC3\\_by\\_Faccio.pdf](https://twiki.cern.ch/twiki/pub/FPGARadTol/InformationOfInterest/SEL_on_ProASIC3_by_Faccio.pdf)
- [10] [https://indico.cern.ch/event/677825/contributions/2775524/attachments/1557646/2450397/hirschauer\\_HB\\_PRR\\_overview.pdf](https://indico.cern.ch/event/677825/contributions/2775524/attachments/1557646/2450397/hirschauer_HB_PRR_overview.pdf)
- [11] Stoddard, Aaron Gerald, "Configuration Scrubbing Architectures for High-Reliability FPGA Systems" (2015). All Theses and Dissertations. 5704.  
<https://scholarsarchive.byu.edu/etd/5704>
- [12] "Radiation Effects in FPGAs," J. Wang, in 9th Workshop on Electronics for LHC Experiments, October 2003  
<http://cdsweb.cern.ch/record/712037/files/p34.pdf?version=1>
- [13] "Radiation tolerance tests of SRAM-based FPGAs for the potential usage in the readout electronics for the LHCb experiment", [http://iopscience.iop.org/1748-0221/9/02/C02028/pdf/1748-0221\\_9\\_02\\_C02028.pdf](http://iopscience.iop.org/1748-0221/9/02/C02028/pdf/1748-0221_9_02_C02028.pdf)
- [14] <https://indico.esa.int/indico/event/130/session/7/contribution/34/material/slides/0.pdf>



*Backup slides*

# Types of TMR

# Digital systems in the CMS detector

Sub-system	TID. Neutrons > 100 keV	Present system, 2008-2017
Tracker [2]	200 kGy. $10^{14}$ n/cm <sup>2</sup>	ASICs only
ECAL [3,4]	25 kGy.	ASICs only
HCAL [5]	3 Gy. $10^{11}$ n/cm <sup>2</sup>	ASIC, Actel anti-fuse FPGA, commercial components
Muon detectors [6, 7]	0.4 Gy. $5 \times 10^{10}$ n/cm <sup>2</sup>	ASIC, , including SRAM FPGAs.
Counting room	~0	All, not radiation-qualified

Obsolete

# Digital systems in the LHCb detector

L0 sub-system [9, 10]	TID. Neutrons (1 Mev equiv)	Present system, 2008-2017
Inner Tracker	60 kGy. $10^{14}$ n/cm <sup>2</sup>	ASIC
Outer Tracker	70 Gy. $10^{12}$ n/cm <sup>2</sup>	ASIC
Calorimeters	50 Gy. $10^{12}$ n/cm <sup>2</sup>	Antifuse FPGA (Actel AX)
Muon detectors [11]	80 Gy. $10^{12}$ n/cm <sup>2</sup>	ASIC, Actel ProAsicPlus, commercial components
Counting room	~0	All (not radiation-qualified)

Obsolete