Electronic *CAD* & *Reliability* Group

Estimating the reliability of DNNs regarding permanent faults on GPUs

Juan David Guerrero Balaguera

Politecnico di Torino
Department of Control and Computer Engineering
Torino, Italy

PitchD
the PhD's pitch

IEEE POLITO Student Branch
Politecnico di Torino IEEE Student Branch

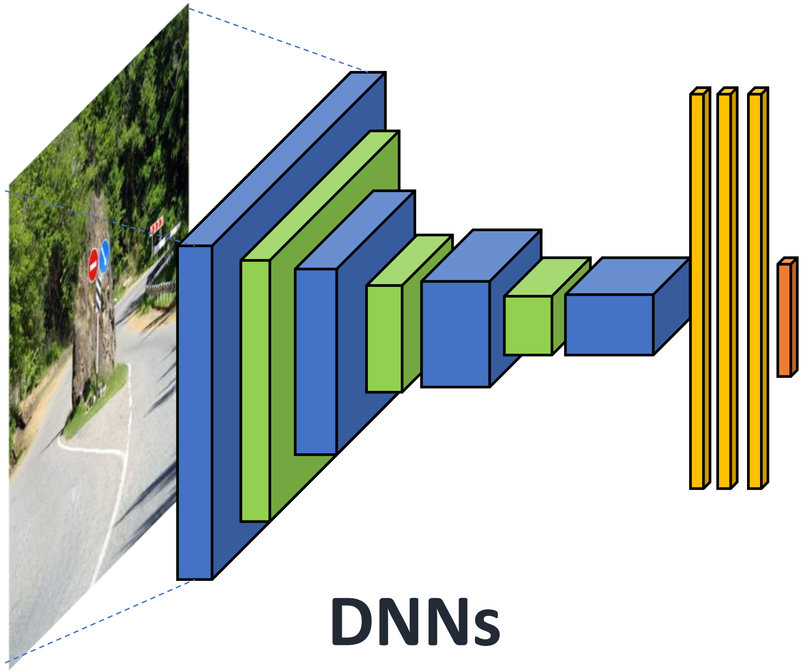
IEEE
Advancing Technology for Humanity

Outline

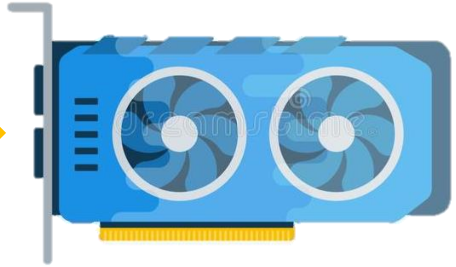
- **Introduction and background**
- Evaluation methodology
- Experimental results
- Conclusions.

Introduction

Graphic Processing Units (GPUs) are widely used as **AI accelerators** for deploying **Neural Networks** in **safety-critical applications**

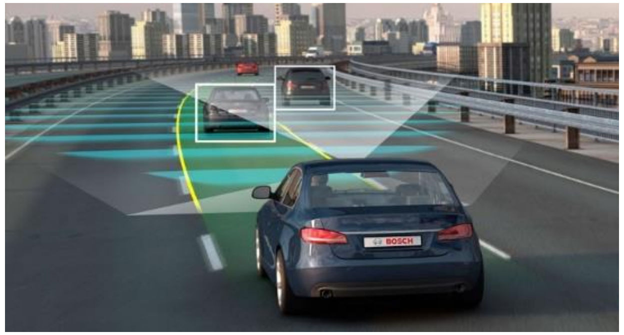


DNNs



GPU device

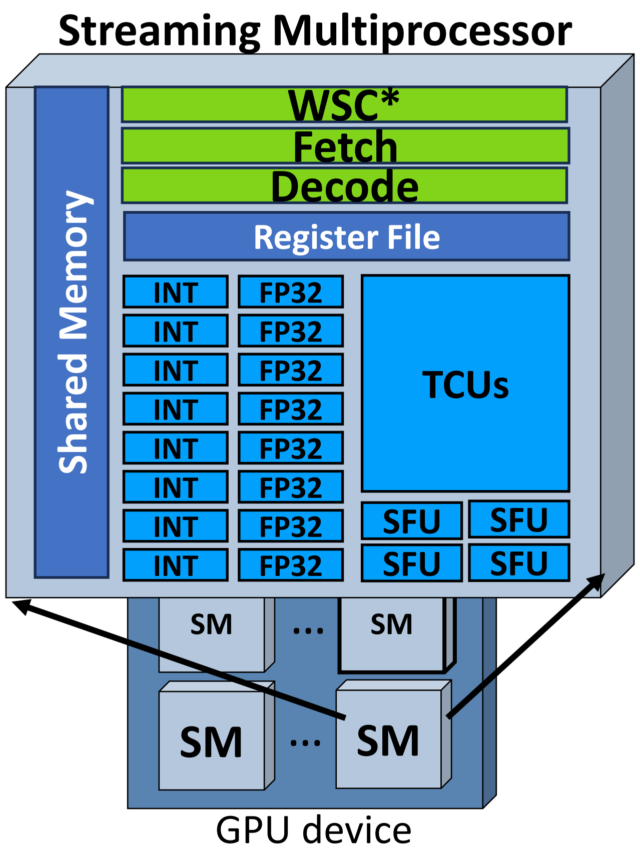
Robotics



Autonomous Driving Systems

Introduction

- **GPUs** are composed of a **hierarchical interconnection** of **Streaming Multiprocessors (SMs)**



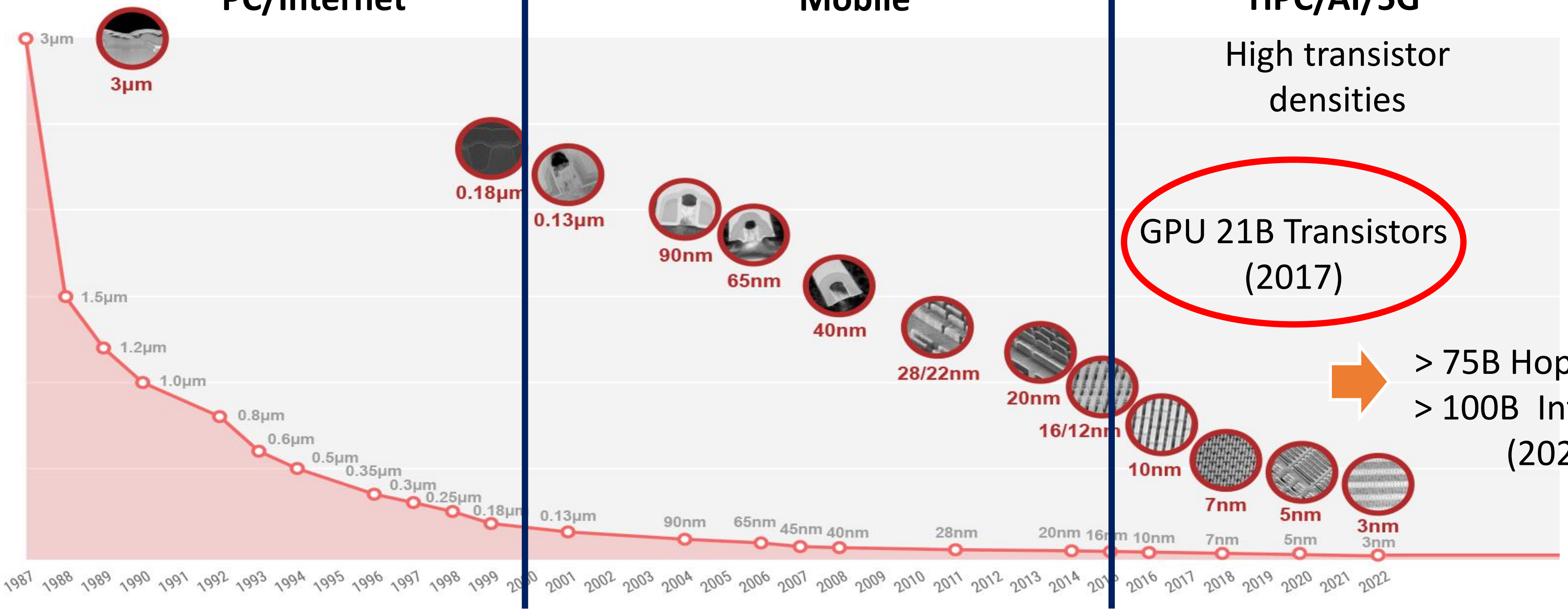
- The SM comprises multiple units such as:
  - Control Units (Warp Scheduler, Fetch, Decode)
  - Scalar Units (INT, FP32)
  - Special Function Units (SFUs)
  - Tensor Core Units (TCUs)
  - Storage Units (Memories, Registers)

- **Single Instruction Multiple Thread – SIMT**
  - Parallel Kernels (i.e., Cooperative Threads Arrays – CTAs)

Introduction

- The application landscape has driven semiconductor technology evolution, including GPUs.

PC/Internet | **Mobile** | **HPC/AI/5G**



High transistor densities

GPU 21B Transistors (2017)

> 75B Hoper GPUs
> 100B Intel GPUs (2023)

Introduction

Are modern chips reliable?

The New York Times

Tiny Chips, Big Headaches

As the largest computer networks continue to grow, some engineers fear that their smallest components could prove to be an Achilles' heel.

- Modern **microelectronic** devices, including GPUs, can be affected by **hardware defects** caused by several phenomena:
  - Manufacturing defects
  - Material stress
  - Test Escapes
  - Environmental harshness
  - Wear-out
  - Aging

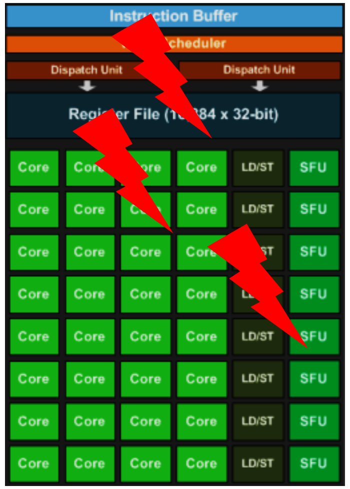
Introduction

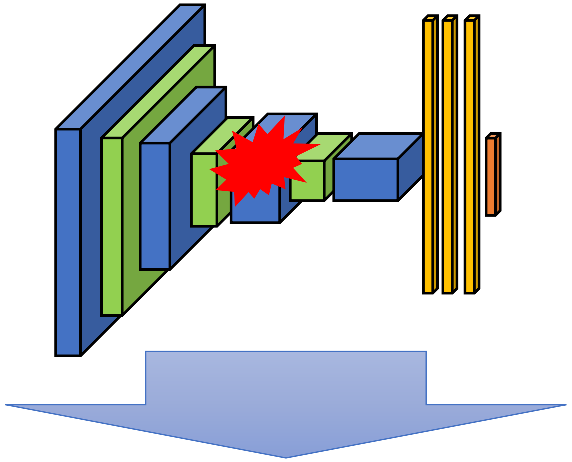
A **fault in a digital device** may induce **“Silent Data Errors” (SDEs)** at the application level producing undesired results and **jeopardizing the reliability** of the whole system

It is crucial to assess the impact of such hardware faults on the operation of cutting-edge electronic devices, in order to devise suitable fault tolerance mechanisms

Introduction

Hardware Permanent Fault → **Induces** → **DNN failures** → **Producing** → **Catastrophic results**

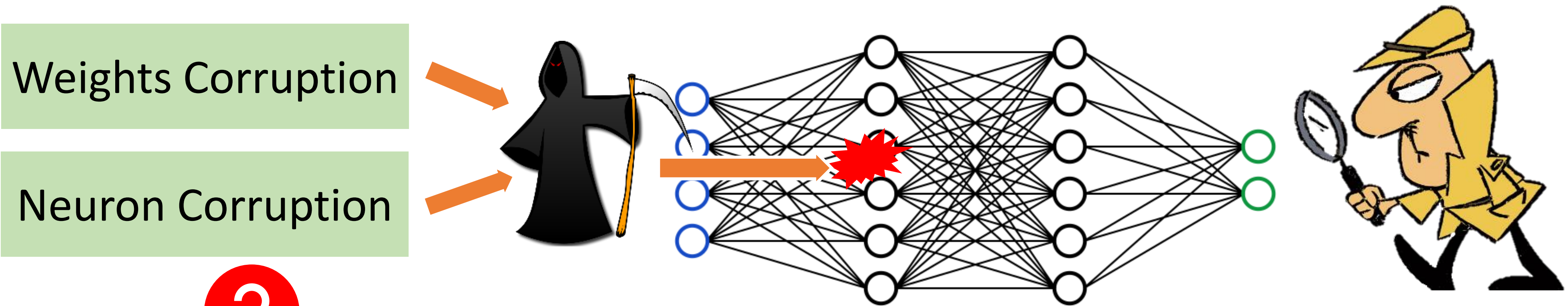


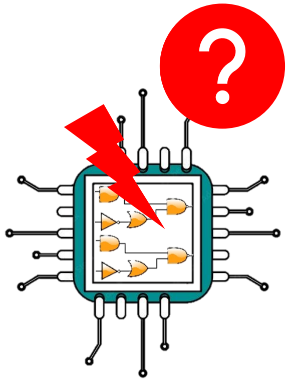


The **reliability** estimation of **DNNs** w.r.t. hardware **faults** in **GPUs** is mandatory to fulfill the requirements of the **safety standards** (e.g., ISO26262 for automotive domain).

Introduction

- The reliability estimation of DNNs w.r.t. PFs mainly resorts to Fault Injections (FI) at the application-level

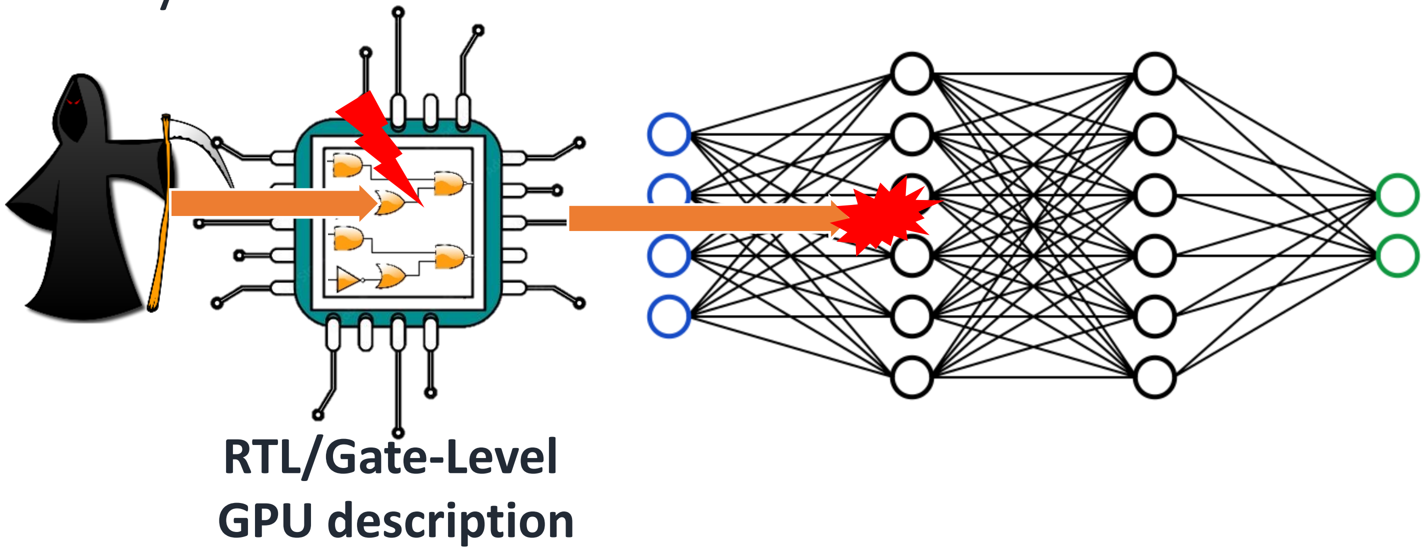




Unrealistic evaluations

Introduction

- Stuck-at
- Delay Faults



> 10,000 days for the evaluation of permanent faults on an RTL-based GPU model running LeNet

Realistic evaluations

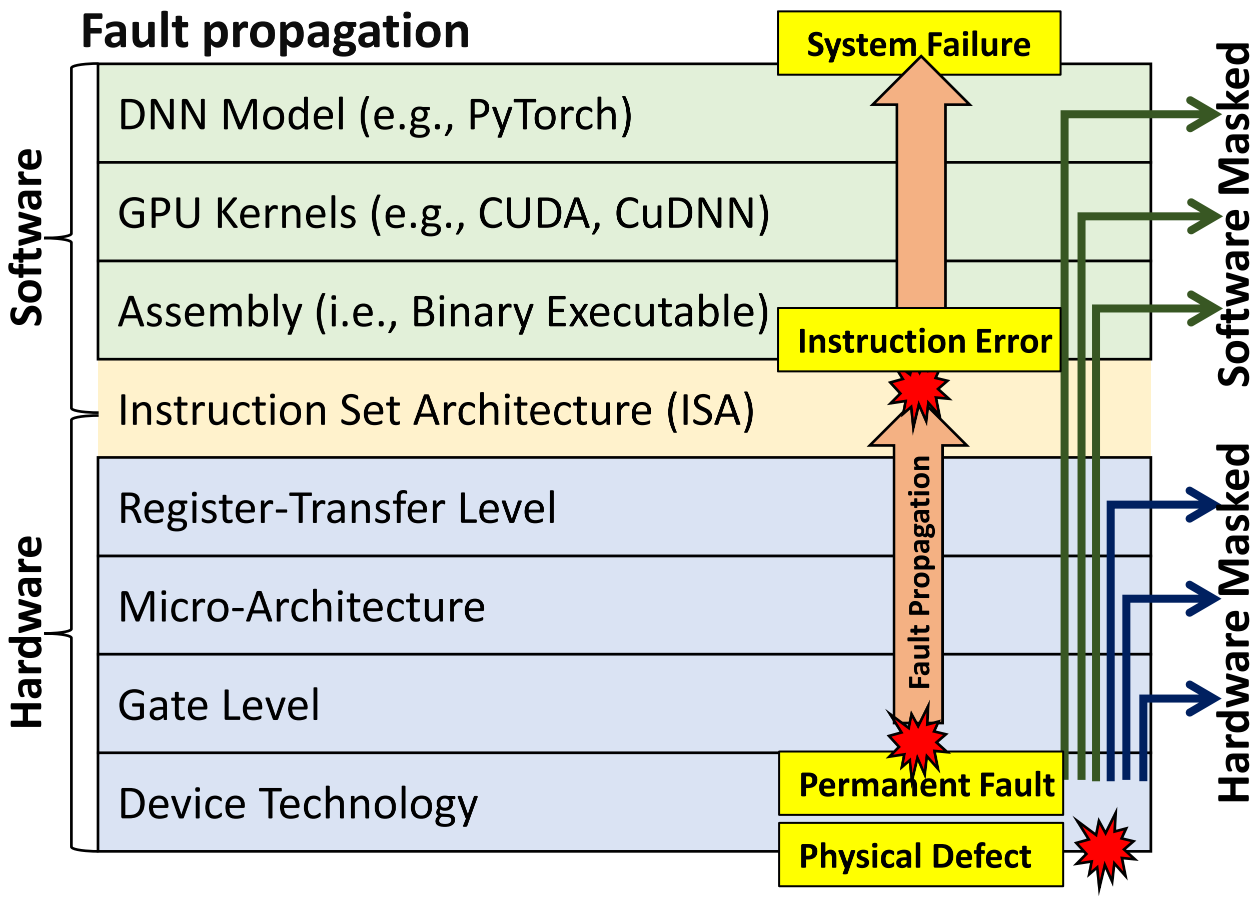
Prohibitive Time

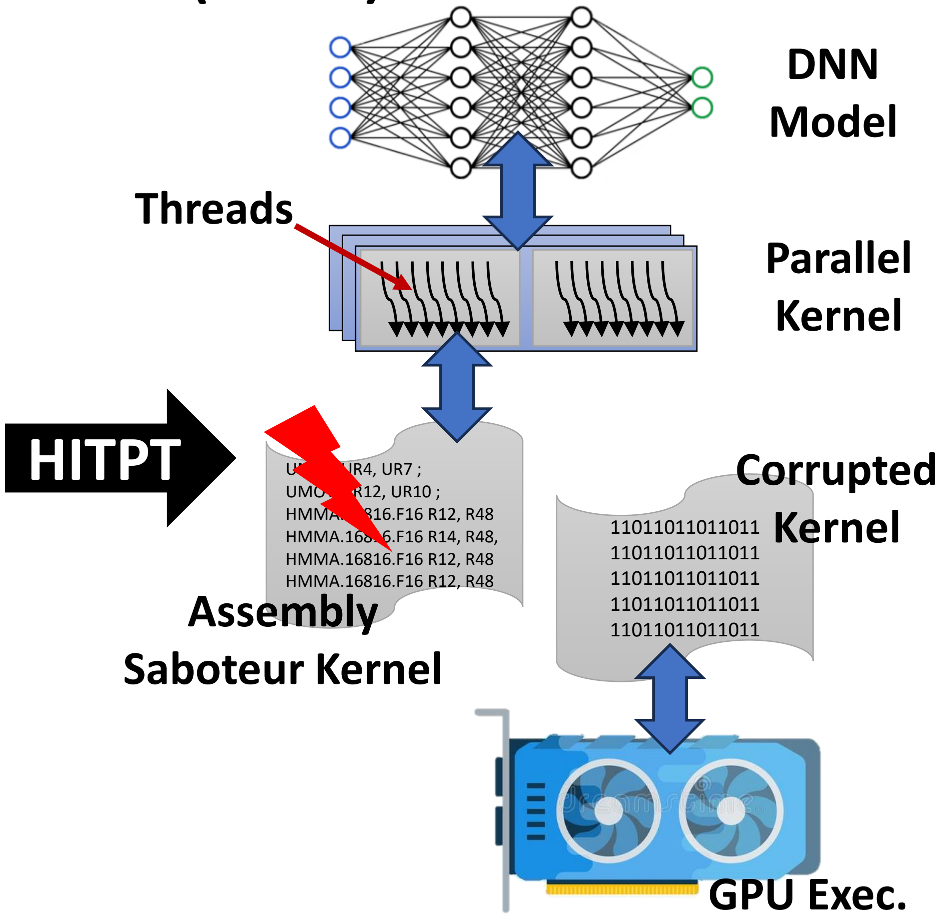
Outline

- Introduction and background
- **Evaluation methodology**
- Experimental results
- Conclusions.

Evaluation methodology

- **Hardware Injection Trough Program Transformation (HITPT)**





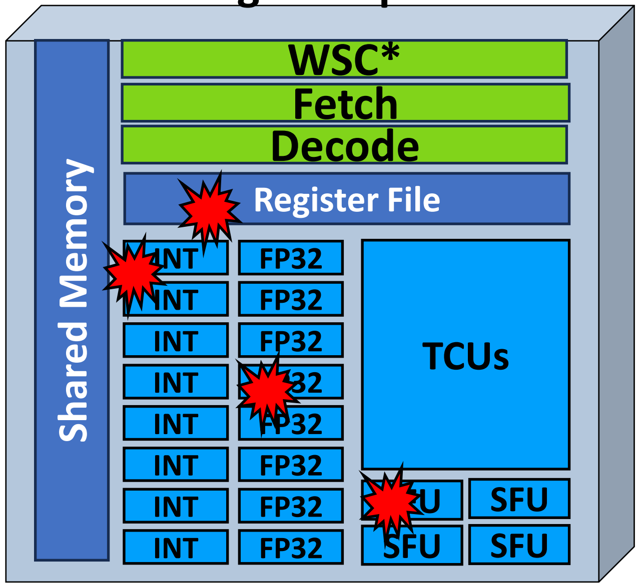
Evaluation methodology

- This research proposes a HITPT methodology to evaluate the resilience of DNNs concerning permanent faults on GPUs

- The proposed approach insert saboteur functions at the instruction level of the GPU kernels

- Fault injections (FI) targets several GPU structures:
  - Register Files (RFs)
  - Integer Cores (INT)
  - Floating Point Cores (FP32)
  - Special Function Units (SFU)

Streaming Multiprocessor



Evaluation methodology

the PhD's pitch

Each fault is classified according to **its impact on the accuracy** of the DNN w.r.t. the fault-free scenario. The **Relative Accuracy Degradation (RAD)** metric indicates the degree of severity of the fault in the DNN.

$$RAD = \frac{ACC_{Gold} - ACC_{Faulty}}{ACC_{Gold}}$$

Faulty CNN's Acuracy

Fault-free CNN's Acuracy

Fault classification

Device Unrecoverable Error (DUE)	GPU crashes or hangs
Silent Data Corruption (SDC-Critical)	RAD > 0.0. At least one image was wrongly classified with respect to the fault-free scenario
Silent Data Corruption (SDC-Safe)	RAD = 0.0. The confidence prediction values for at least one image differ from the fault-free scenario but the classification is still correct
Masked	RAD = 0.0. No difference is observed between the faulty scenario and the golden one

Outline

- Introduction and background
- Evaluation methodology
- **Experimental results**
- Conclusions.

Experimental Results

- We developed a **prototypical tool** based on an improved version of the NVBitFI tool

- Four DNNs were evaluated using an RTX 3060TI GPU, NVIDIA Ampere architecture
  - **LeNet**: classifies images of **handwritten digits (0 to 9)** using the **MNIST** dataset
  - **AlexNet**: classifies images from **1,000 categories** from the **ImageNet** dataset
  - **DarkNet19**: classifies images from **1,000 categories** from the **ImageNet** dataset
  - **VGG-16**: classifies images from **10 different classes** defined by the **CIFAR-10** dataset.

Experimental Results

- Two fault injection (FI) campaigns
  - Exhaustive FI for all registers in one resident thread of an SM
  - Exhaustive FI for all instructions in one INT, FP32 and SFU core of an SM

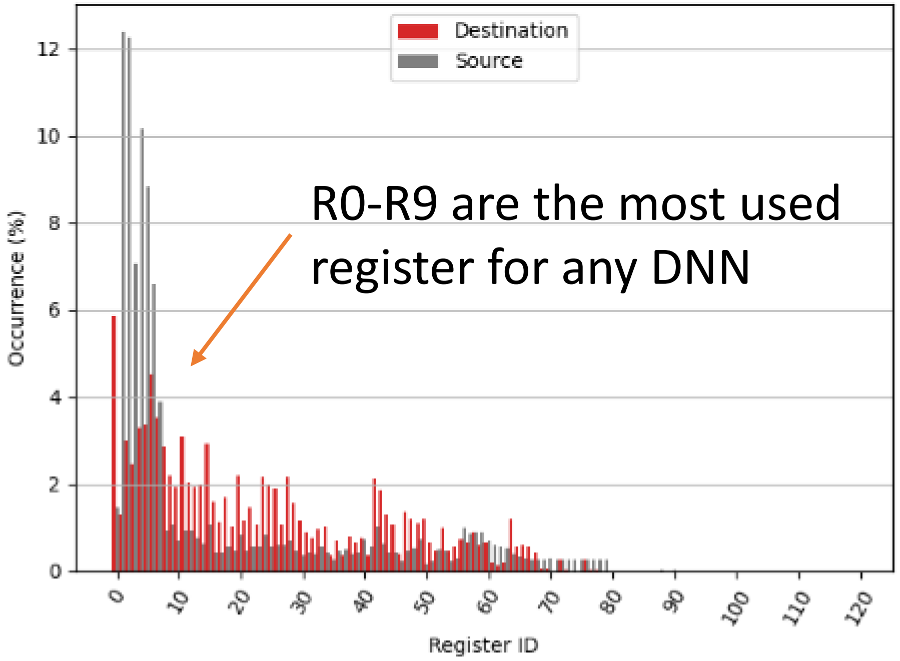
FI Campaing	LeNet	AlexNet	DarkNet19	VGG-16
SM0, Warp0, Thread0	7,233	7,296	10,945	7,233
INT_0, FP32_0, SFU_0	2,880	2,880	2,880	2,880

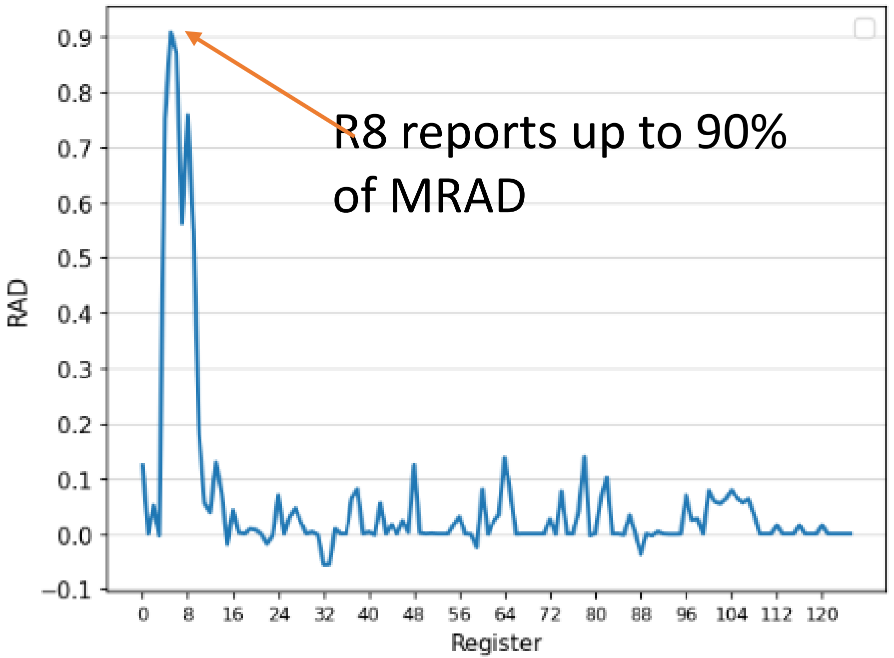
- Experimental evaluation takes around 107 Hours.

Experimental Results

the PhD's pitch

- The results show that faults in the first ten registers are the most critical, generating more than 20% of Mean Relative Accuracy Degradation (MRAD), and up to 90% of MRAD for R8.

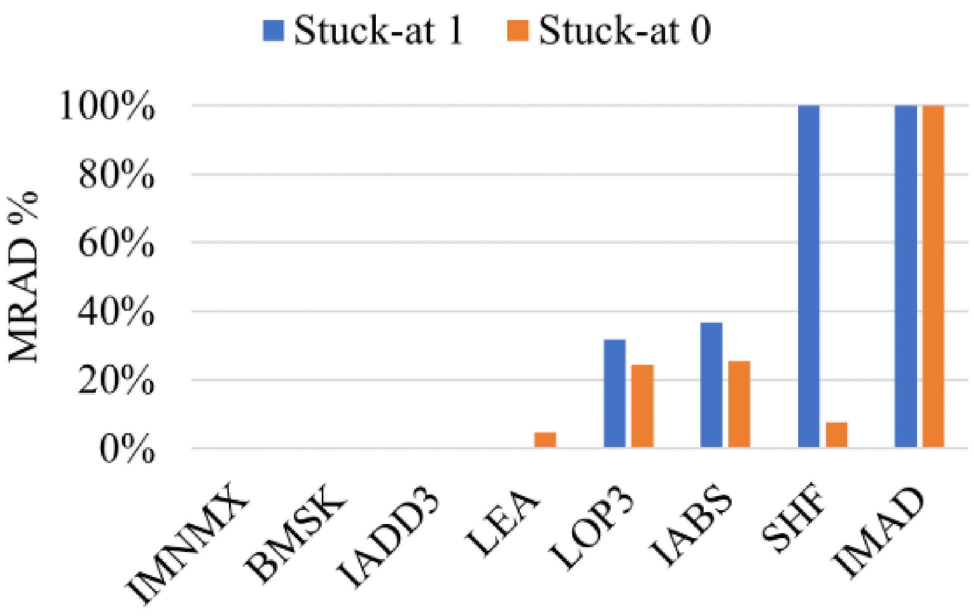




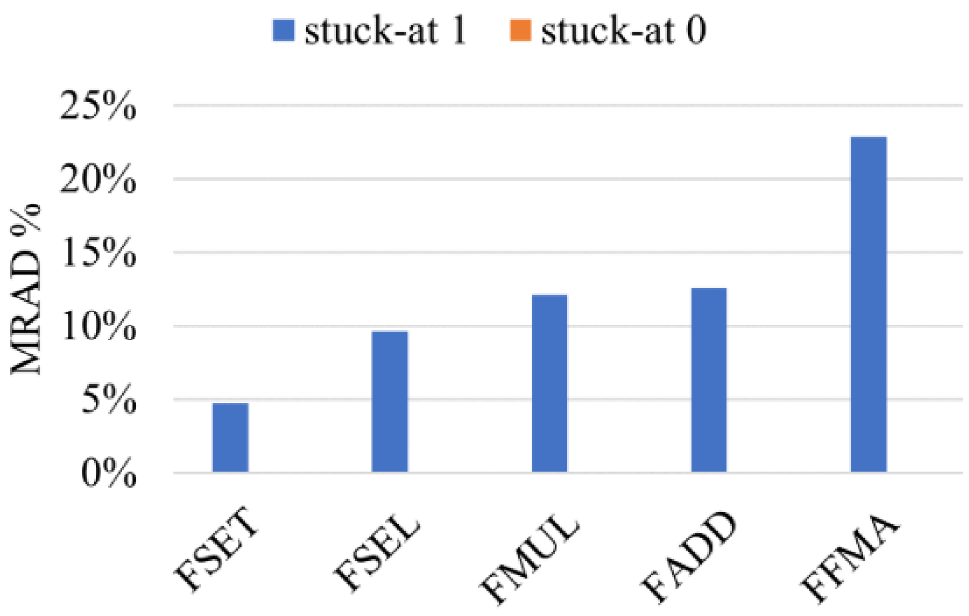
Politecnico di Torino
IEEE Student Branch

Experimental Results

- Stuck-at-1 faults lead to a higher impact on the DNN's accuracy for both integer and floating-point instructions than stuck-at-0 faults



(a) INT core



(b) FP32 core

Outline

- Introduction and background
- Proposed fault injection methodology
- Experimental results
- **Conclusions**.

Conclusions

- This paper presents **a method for realistic evaluation of DNNs regarding permanent faults** (PFs) in **GPU devices**

- The **proposed solution** adopts the Hardware Injection Trough Program Transformation (**HITPT**)
    - Register Files (**RFs**)
    - Functional units (**INT**, **FP32**, and **SFU**)

- The results show that DNNs are highly sensitive to faults on the first ten registers per thread in a GPU

- **Stuck-at-1** faults on both INT and FP32 cores induce a high accuracy degradation in DNNs

Future Works

the PhD's pitch

- Future activities aim to devise effective hardening techniques to counteract the impact of permanent faults on the reliability of DNNs

Thank you!

Questions?

juan.guerrero@polito.it