KU LEUVEN

Let’s Meet – Inside the Campus

26 September 2023

Presented by: Mohaddaseh Nikseresht

Outline

- Let’s Explore who I am
- Introduction
- Bitflips
- Software Protection Techniques
- A look into my PhD
- Overview
- References

KU LEUVEN

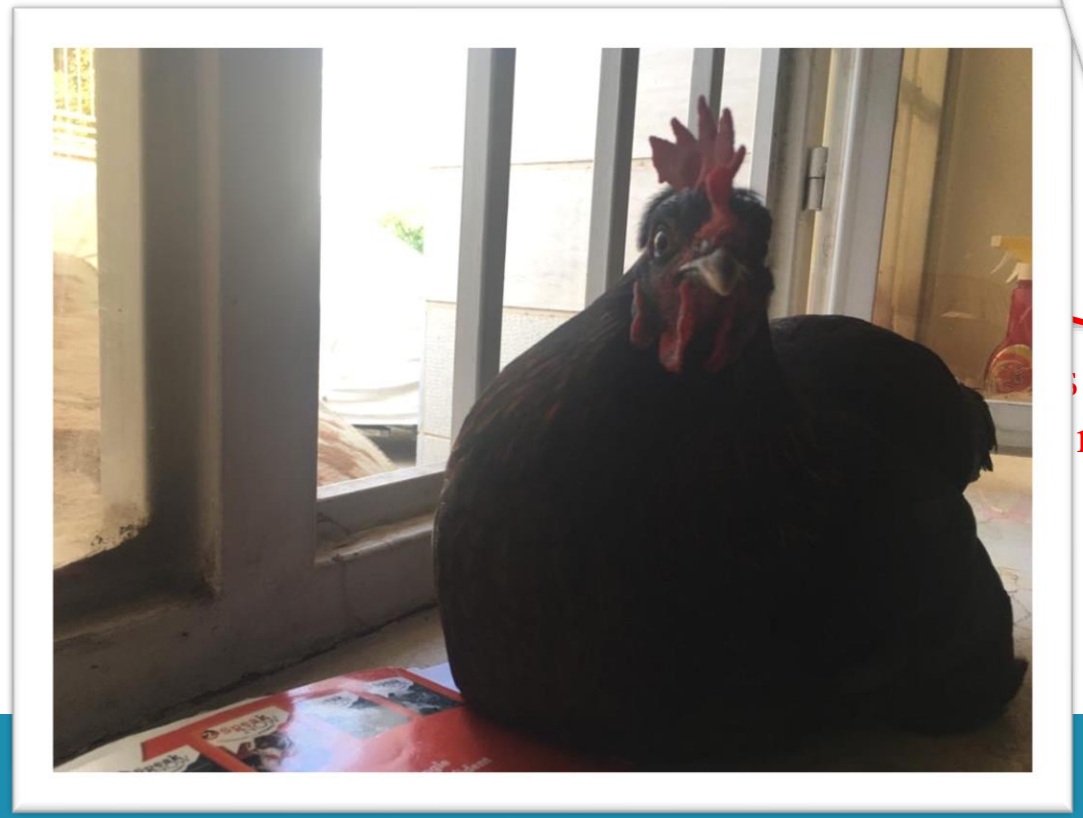
Let’s Explore who I am

Let's explore who I am

Basics

I love to play video games.

And I love chicken.



s is my pet.
name is “The Lady”.

KU LEUVEN

Let's explore who I am

Hometown: Shiraz, Iran

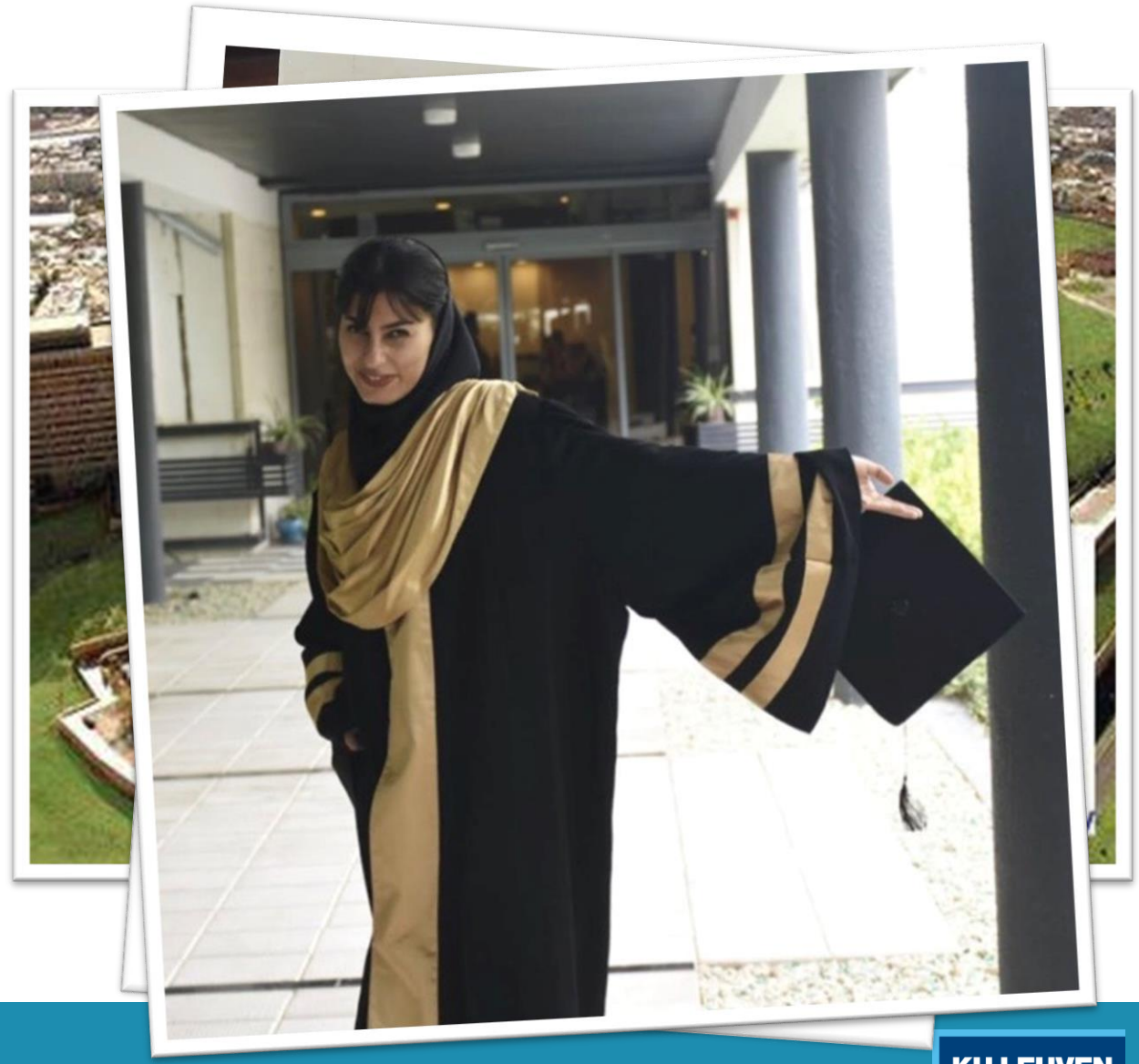
Let's explore who I am

Academic background

I studied **Computer Engineering** at **Shiraz university.** One of the **highest-ranked public** universities in Iran.

My interest in **embedded systems** began …

In 2019, I have graduated as **ranked one** with **several publications** in **well-known** conferences and journals.



KU LEUVEN

Let's explore who I am

Academic background

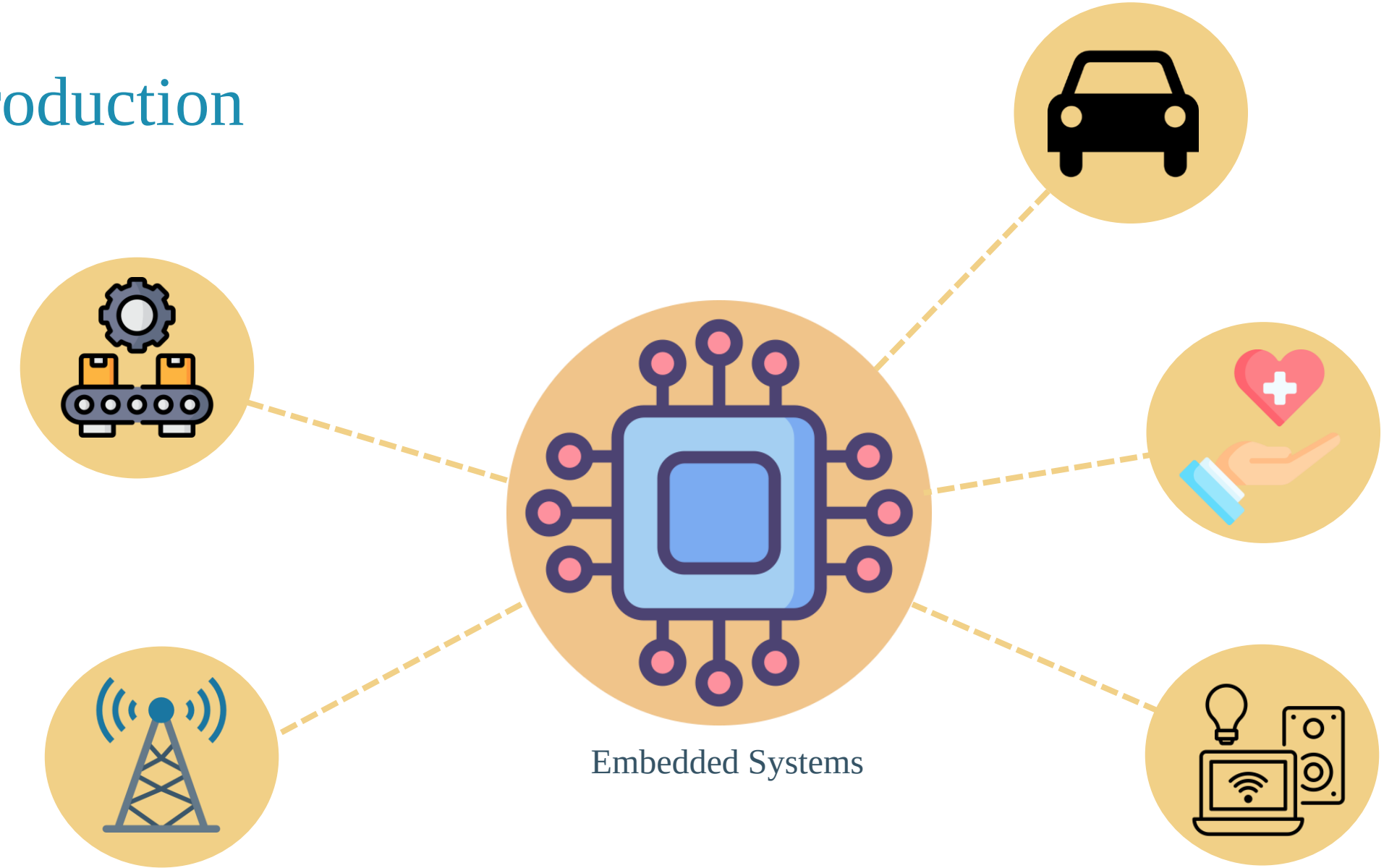
In 2021, I have started to pursue my PhD at KU Leuven, under Supervision of **Prof. Jeroen Boydens**.

Almost three years into my PhD …

KU LEUVEN

Introduction

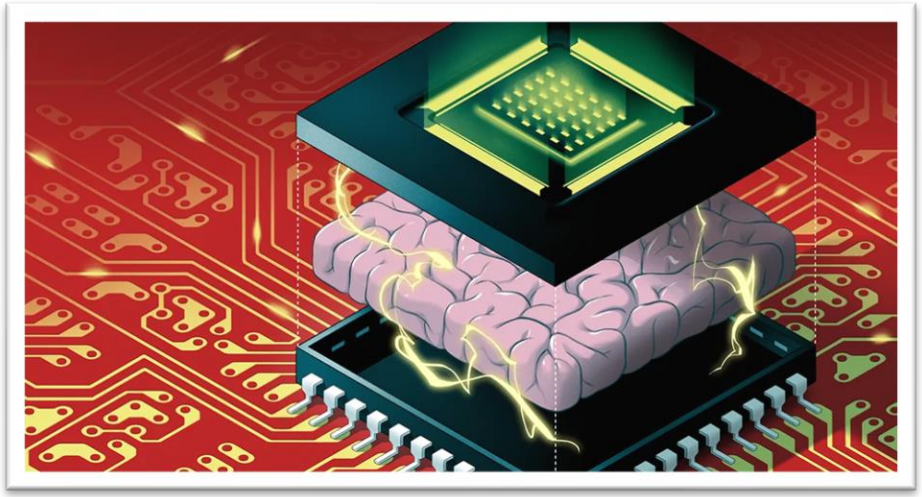
Introduction



Introduction

Advance in technology made it possible to build electronic components which are:

- Programable (+)
- efficient (+)
- cost-effective (+)



However, these changes lead to **limit** the processor systems' reliability by

Shrinking of transistor sizes (-)

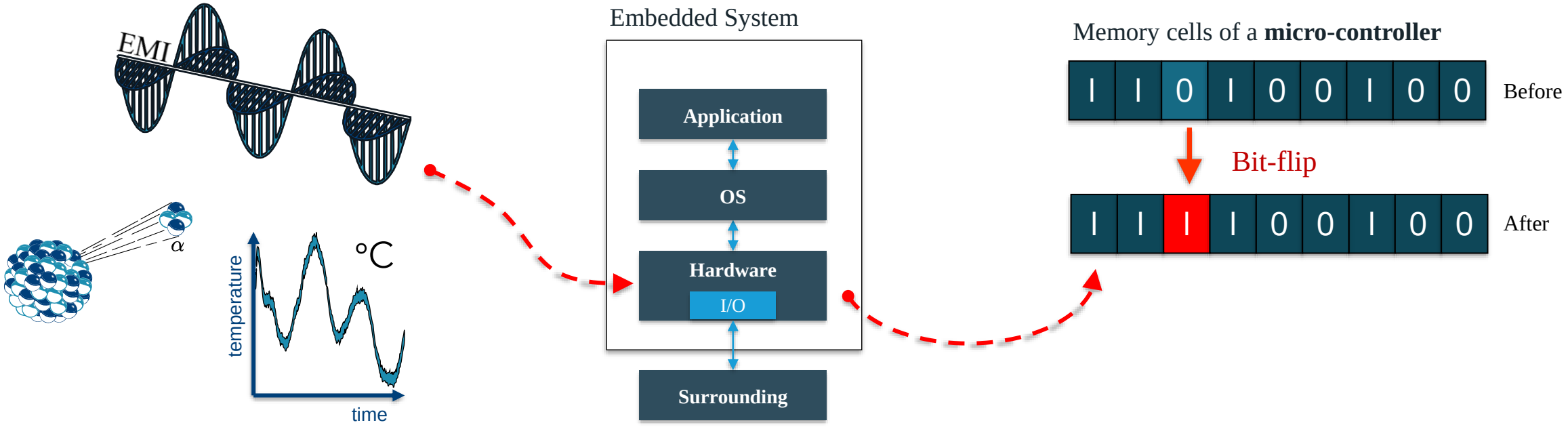
Increase registers vulnerability to radiation-induced Single Event Effects (SEE)

Soft Errors (Bitflips)

KU LEUVEN

Bitflips

Bitflips



KU LEUVEN

Bitflips

Different types of software errors:

0	**0**	1	Original value
0	**1**	1	Modified value

```
1d0:  MOV   r3,  r0
1d2:  CBZ   r0,  1e2
1d4:  MOVS  r0,  #0
1d6:  SUBS  r2,  r3,  #1
1d8:  ADDS  r3,  r1
1da:  ADD   r0,  r0,  #1
1de:  BNE   1d6
1e0:  ADD   r0,  #1
1e2:  BX    lr
```

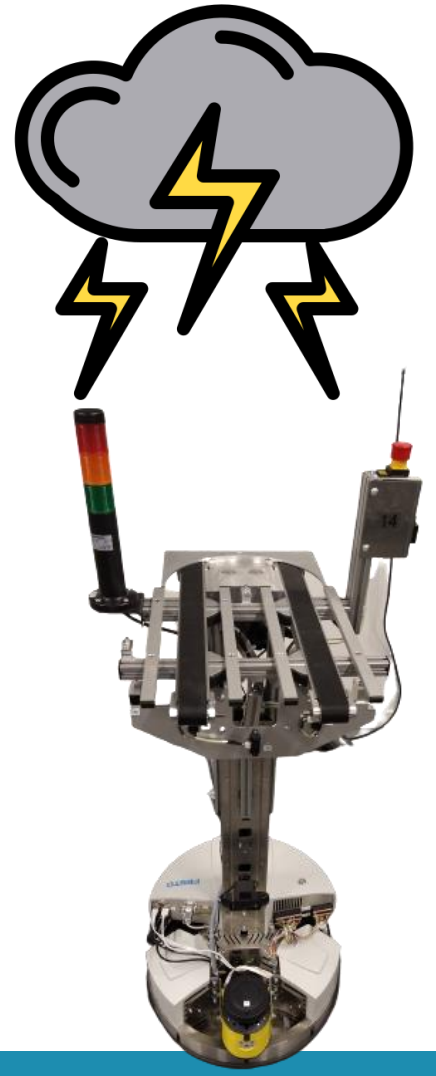
Original program

```
1d0:  MOV   r3,  r0
1d2:  CBZ   r0,  1e2
1d4:  MOVS  r0,  #0
1d6:  SUBS  r2,  r3
1d8:  ADDS  r3,  r1
1da:  ADD   r0,  r0,  #1
1de:  BNE   1d6
1e0:  ADD   r0,  #1
1e2:  BX    lr
```

Control Flow Error (CFE)

```
1d0:  MOV   r3,  r0
1d2:  CBZ   r0,  1e2
1d4:  MOVS  r0,  #0
1d6:  SUBS  r2,  r3
1d8:  ADDS  r3,  r1
1da:  ADD   r0,  r0,  #1
1de:  BNE   1d6
1e0:  ADD   r0,  #1
1e2:  BX    lr
```

Data Flow Error (DFE)

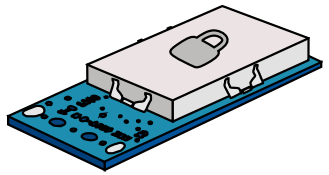


KU LEUVEN

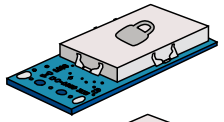
Bitflips

Protecting embedded systems

Hardware techniques

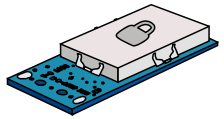


Adding extra physical component to the system to increase the reliability of the system.



F(A)

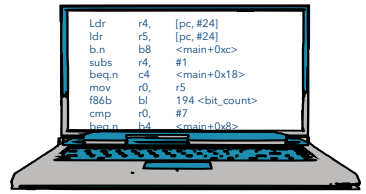
F(A')



F(A'')

Majority voting → **Result**

Software techniques



Adding extra codes to the application to increase the reliability of the system.

KU LEUVEN

KU LEUVEN

Software Protection Techniques

Software Protection Techniques

Protecting embedded systems

```
subs  r2, r3, #1
adds  r3, r1
add   r0, r0, #1
bne   BB2
```

Original code

A DFED protection technique is applied

```
subs  r2, r3, #1
subs  r8, r9, #1
cmp   r3, r9
bne   error
cmp   r1, r7
bne   error
adds  r3, r1
adds  r9, r7
add   r0, r0, #1
adds  r6, r6, #1
bne   BB2
```

Protected code

Pros and Cons

- **Flexible** and can **be automated** (+)
- Suitable for COTS systems (+)
- **Extra code size Overhead** imposed on the system (-)
- **Execution time** of the application may be **increased** (-)

KU LEUVEN

Software Protection Techniques

Required registers for DFED techniques

Several researches have been done… 

Technique	Required registers
EDDI	N
VAR3+	N
ED4I	N

```
../drivers/SPI.cpp:47:1: error: unable to find registers to spill
 }
 ^
../drivers/SPI.cpp:47:1: error: this is the insn:
(insn 64 80 18 2 (set (mem/f:SI (plus:SI (reg/f:SI 152 [orig:116 this ] [116])
                (const_int 4 [0x4])) [9 this_3(D)->_spi.spi+0 S4 A32])
        (reg:SI 140 [139])) "../drivers/SPI.cpp":41 878 {*thumb2_movsi_insn}
     (expr_list:REG_DEAD (reg:SI 140 [139])
        (nill))))
../drivers/SPI.cpp:47: confused by earlier errors, bailing out
make[1]: *** [mnt/data/Documents/.../DistributionStation/makefile:428: drivers/SPI.o] Error 1
make: *** [makefile:26: all] Error 2
```

Microcontroller might not have enough registers to compile

KU LEUVEN

KU LEUVEN

A look into my PhD:
Selective Implementation of Software Resilience Techniques to Increase Low-Level Code Reliability

A look into my PhD

Selective Protection

Software Redundancy

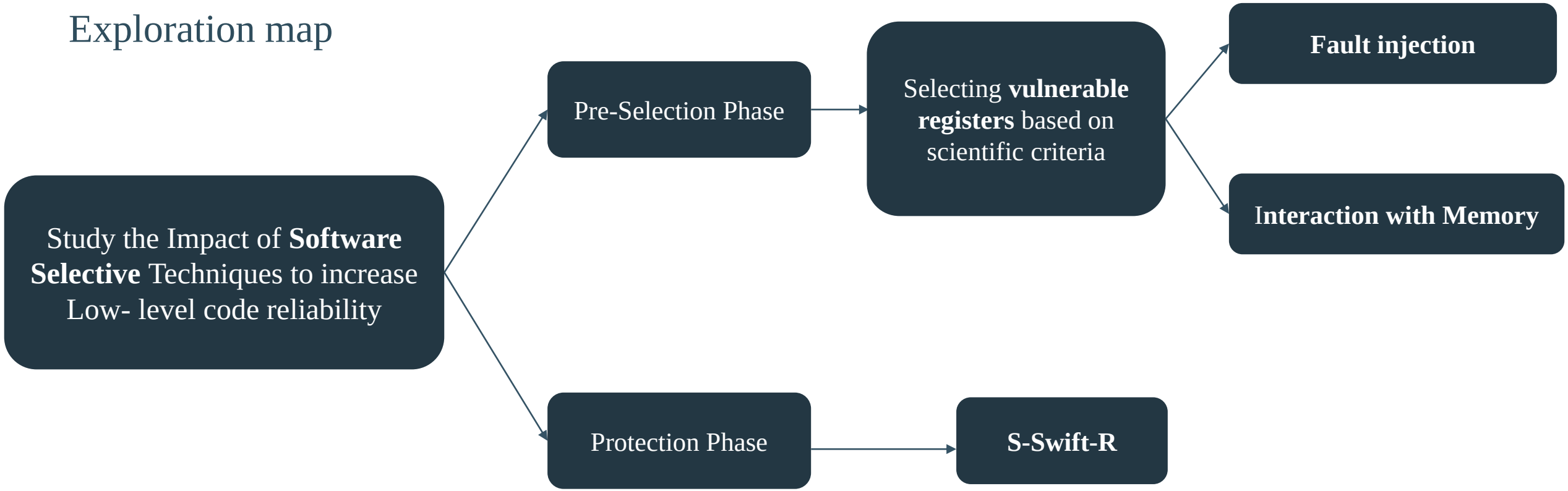
Software Implemented Hardware Fault Tolerance (SIHFT)

- Give systems the ability to detect and correct faults (+)
- Memory usage (-)
- Storage usage (-)

SOLUTION: Partially protecting the registers

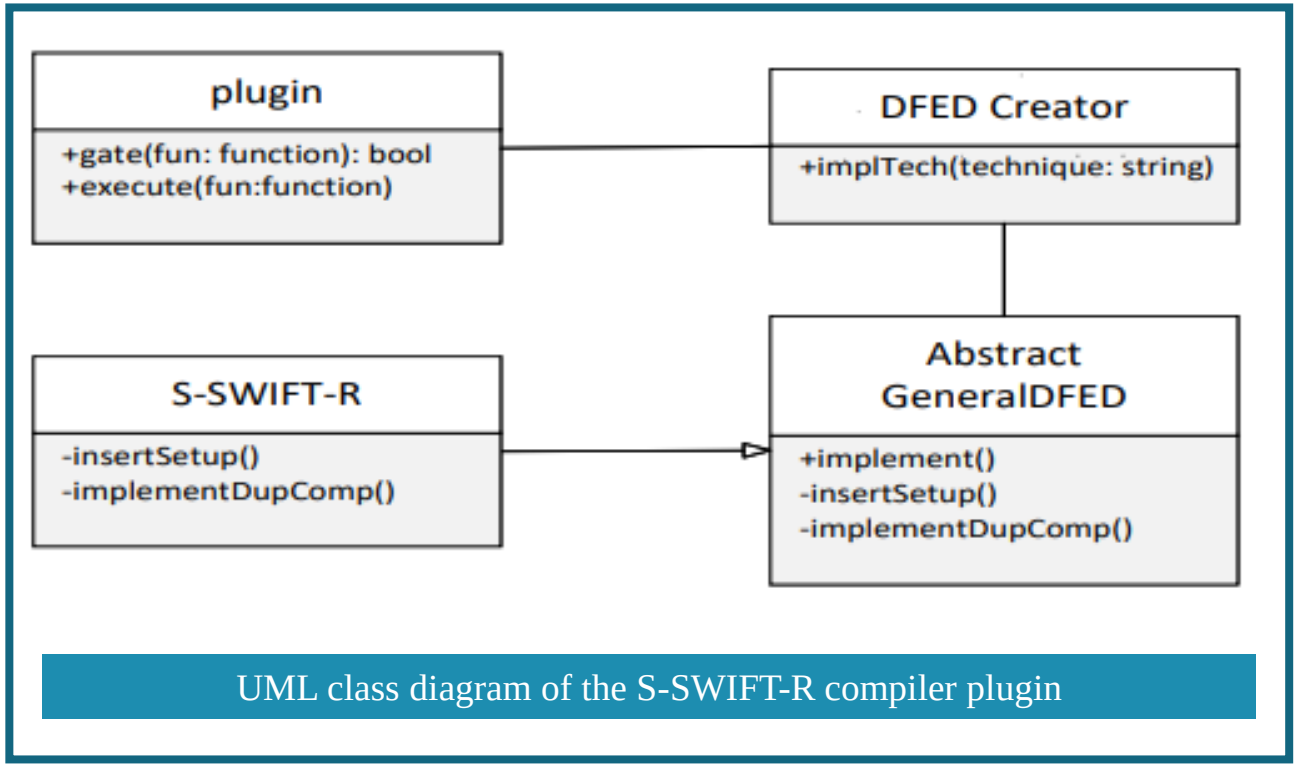
A look into my PhD

Exploration map



A look into my PhD

Protection phase - Automatic Implementation



UML class diagram of the S-SWIFT-R compiler plugin

A look into my PhD

Example

#	Non-Hardened	Selective SWIFT-R		SWIFT-R
		Protected register: s0	Protected register: s1	Protected registers: s0, s1
1	**LOAD s0, 00**	**LOAD s0, 00**	**LOAD s0, 00**	**LOAD s0, 00**
2		*create s0 copies*		*create s0 copies*
3	**LOAD s1, 2A**	**LOAD s1, 2A**	**LOAD s1, 2A**	**LOAD s1, 2A**
4			*create s1 copies*	*create s1 copies*
5			*majority voter for s1*	*majority voter for s1*
6	**ADD s0, s1**	**ADD s0, s1**	**ADD s0, s1**	**ADD s0, s1**
7		*ADD s0′, s1*		*ADD s0′, s1*
8		*ADD s0″, s1*		*ADD s0″, s1*
9		*majority voter for s0*		*majority voter for s0*
10			*majority voter for s1*	*majority voter for s1*
11	**STORE s0, (s1)**	**STORE s0, (s1)**	**STORE s0, (s1)**	**STORE s0, (s1)**

KU LEUVEN

A look into my PhD

Example

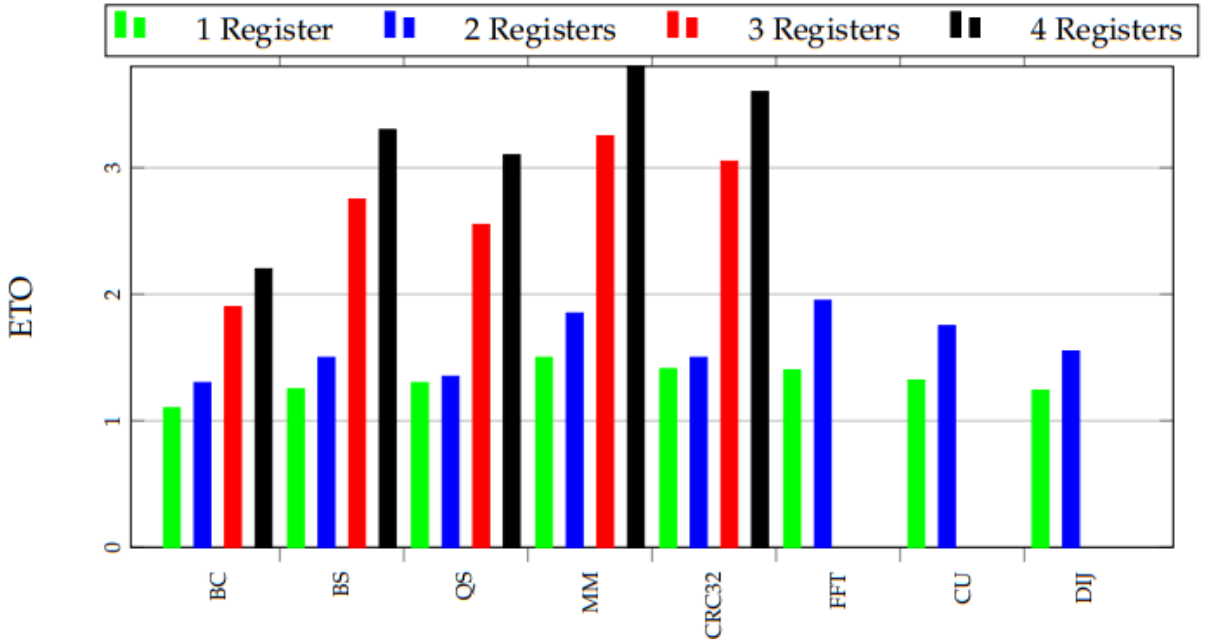
Number of Register	SDC	NE	Number of Register	SDC	NE
1 Register			**2 Registers**		
S_SWIFT_R_F	Min: 14.5% Avg.: 16.3% Max: 36.2%	Min: 62.3% Avg.: 72.1% Max: 84.3%	**S_SWIFT_R_F**	Min: 10.2% Avg.: 12.3% Max: 21.7%	Min:69.4% Avg.: 80.1% Max: 88.9%
S_SWIFT_R_M	Min: 24.2% Avg.: 19.7% Max: 49.7%	Min:50.3% Avg.:66.5% Max:73.9%	**S_SWIFT_R_M**	Min: 12.1% Avg.: 14.9% Max: 20.5%	Min: 66.2% Avg.: 78.6% Max:79.2%
Unprotected (baseline)	Min: 29.1% Avg.: 47.8% Max: 59.34%	Min:29.1% Avg.: 41.8% Max: 59.34%	**Unprotected (baseline)**	Min:29.1% Avg.: 47.8% Max: 59.34%	Min: 29.1% Avg.: 47.8% Max: 59.34%
3 Registers			**4 Registers**		
S_SWIFT_R_F	Min: 14% Avg.: 9.7% Max: 18%	Min: 81.9% Avg.: 87.3% Max:89.1%	**S_SWIFT_R_F**	Min: 2.9% Avg: 3.1% Max: 4.2%	Min: 80% Avg.: 89.3% Max: 90.7%
S_SWIFT_R_M	Min: 9.3% Avg.: 10.1% Max: 12.13%	Min: 77.9% Avg.: 83.2% Max: 85.9%	**S_SWIFT_R_M**	Min: 5.3% Avg.:4.1% Max: 6.8%	Min: 79.3% Avg. :86.1% Max: 80.2%
Unprotected (baseline)	Min: 40.23% Avg.: 58.6% Max: 67.3%	Min: 29.8% Avg.: 39.2% Max: 60.9%	**Unprotected (baseline)**	Min: 40.23% Avg.: 58.6% Max: 67.3%	Min: 29.8% Avg.: 39.2% Max: 60.9%

*code size and execution time overhead increase exponentially as we increase the number of registers to protect

A look into my PhD:

Example

A. Measured code size for different case studies



B. Measured execution time overhead for different case studies

KU LEUVEN

Overview

Overview

- The role of Embedded systems and the impact of downscale technology on their reliability
- Bitflips:
  - Definition
  - Different type of Soft errors: DFE / CFE
  - Hardware/ Software techniques to protect against bitflips
- Software Protection Techniques
  - The Pros and Cons
  - The problem of full software protection techniques
- A look into my PhD: Selective protection
  - Exploration map
  - An example

References

[1] Mohaddaseh Nikseresht, Jens Vankeirsbilck, Jeroen Boydens, A Study on Selective Implementation Approaches for Soft Error Detection Using S-SWIFT-R, Electronics, volume 11, issue 20, 21 pages, Multidisciplinary Digital Publishing Institute (MDPI), October 19, 2022

[2] Mohaddaseh Nikseresht, Jens Vankeirsbilck, Davy Pissoort, Jeroen Boydens, A Selective Soft Error Protection Method for COTS Processor-based Systems, XXX International Scientific Conference Electronics (ET), 2021 XXX International Scientific Conference Electronics (ET), pages 1-5, Sozopol, Bulgaria, September 15-17, 2021

[3] Mohaddaseh Nikseresht, Brent De Blaere, Jens Vankeirsbilck, Davy Pissoort, Jeroen Boydens, Impact of Selective Implementation on Soft Error Detection Through Low-level Re-execution, DASC (Workshop), 2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 6 pages, AB, Canada, October 25-28, 2021

Questions?