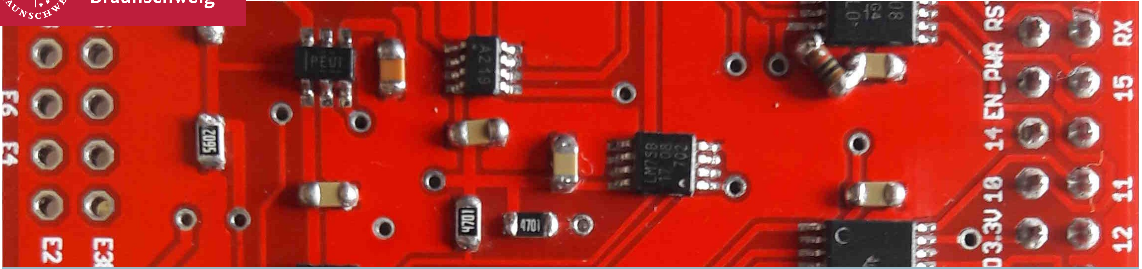
Institute of Operating Systems
and Computer Networks

Technische
Universität
Braunschweig



Effective but Lightweight Online Selftest for Energy-Constrained WSNs

SenseApp 2018

Ulf Kulau, Daniel Szafranski and Lars Wolf, 01.10.2018

Technische Universität Braunschweig, IBR

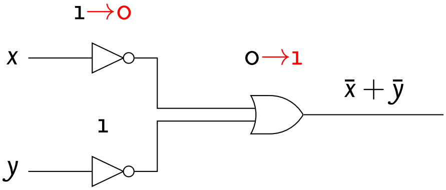
Soft Errors

What are Soft Errors?

Soft Errors

What are Soft Errors?

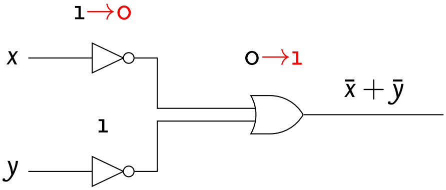
- Unexpected state changes in digital circuits
  - Bit-flips, stuck-at errors, ...
- Occur randomly and temporary
- Can lead to malfunction of components
- Overall system is often not affected
  → Very hard to detect



Soft Errors

What are Soft Errors?

- Unexpected state changes in digital circuits
  - Bit-flips, stuck-at errors, ...
- Occur randomly and temporary
- Can lead to malfunction of components
- Overall system is often not affected
  → Very hard to detect
- Causes
  - In space applications: cosmic rays
  - Undervolting
  - Large temperature variations
  - Faulty units, ageing and wear, ...



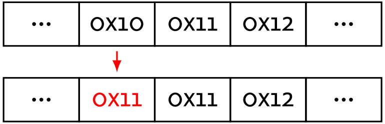
Soft Errors

Effects of soft errors in computer systems

Soft Errors

Effects of soft errors in computer systems

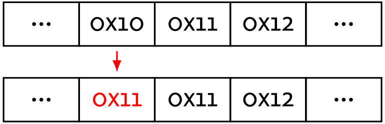
- Soft errors can occur in any component
  - Program Counter (PC) → unpredictable program flow
  - Arithmetic logic unit (ALU) → incorrect calculations
  - RAM → data corruption
  - Peripheral components, ...



Soft Errors

Effects of soft errors in computer systems

- Soft errors can occur in any component
  - Program Counter (PC) → unpredictable program flow
  - Arithmetic logic unit (ALU) → incorrect calculations
  - RAM → data corruption
  - Peripheral components, ...
- Can propagate and lead to unpredictable malfunction of the entire system



Soft Errors

Effects of soft errors in WSNs

- Disadvantages
  - Malfunction of WSN nodes
  - Can significantly decrease the overall energy efficiency of WSNs

Kulau et.al., *Energy Efficiency Impact of Transient Node Failures when using RPL*, WoWMoM, 2017

Soft Errors

Effects of soft errors in WSNs

- Disadvantages
  - Malfunction of WSN nodes
  - Can significantly decrease the overall energy efficiency of WSNs

Kulau et.al., *Energy Efficiency Impact of Transient Node Failures when using RPL*, WoWMoM, 2017

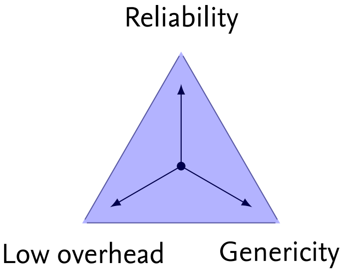
- Advantages
  - Safe indicator of an malfunctioning MCU due to undervolting
  - Can be used to detect unreliable WSN node

Kulau et.al., *IdealVolting – Reliable Undervolting on Wireless Sensor Nodes*, ACM Transactions on Sensor Networks (TOSN), 2016

→ Soft error detection is both, necessary and beneficial

Soft Errors

Project goal: Effective but Lightweight Online Selftest for Energy-Constrained WSNs



- **Reliability**

  → High soft error detection rates
- **Low overhead**
  - Focus on most error-prone components

  → Increase energy efficiency
- **Genericity**
  - Use a software implementation instead of addidtional hardware

  → usable on different MCUs

Online Selftest - Basics

Online Selftest - Basics

Online Selftest - Basics

Online Selftest - Basics

- Well-known Algorithm-Based Fault Tolerance (ABFT)
  - Used for fault tolerance in complex computer systems
  - Error correction requires huge overhead

Online Selftest - Basics

Online Selftest - Basics

- Well-known Algorithm-Based Fault Tolerance (ABFT)
  - Used for fault tolerance in complex computer systems
  - Error correction requires huge overhead
- Online Selftest for energy-constrained MCUs
  - ALU is one of the most error-prone components
    → Based on checksum based fault tolerant matrix multiplication
  - Checksum is implemented as a sum function
  - Small dimensional matrices
  - Online capable implementation
  - Focus on error detection instead of correction

Online Selftest - Basics

1) Starting with 2 Matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times r}$

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \in \mathbb{R}^{m \times n}, \qquad B = \begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,r} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,r} \end{pmatrix} \in \mathbb{R}^{n \times r}$$

Online Selftest - Basics

2) Adding column and row sums to A and B

$$A_c = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \\ a_{m+1,1} & a_{m+1,2} & \cdots & a_{m+1,n} \end{pmatrix} \in \mathbb{R}^{(m+1)\times n}, \qquad a_{m+1,j} = \sum_{i=1}^{m} a_{i,j} \quad \text{with} \quad j = [1, \ldots, n]$$

$$B_c = \begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,r} & b_{1,r+1} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,r} & b_{2,r+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,r} & b_{n,r+1} \end{pmatrix} \in \mathbb{R}^{n\times(r+1)}, \qquad b_{i,r+1} = \sum_{j=1}^{r} b_{i,j} \quad \text{with} \quad i = [1, \ldots, n]$$

Online Selftest - Basics

3) Multiplication and results review

$$A_c \cdot B_c = C_c = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,r} & c_{1,r+1} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,r} & c_{2,r+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,r} & c_{m,r+1} \\ c_{m+1,1} & c_{m+1,2} & \cdots & c_{m+1,r} & c_{m+1,r+1} \end{pmatrix}$$

- No soft errors

$$\forall j \in [1, \ldots, r+1] : c_{m+1,j} = \sum_{i=1}^{m} c_{i,j}$$

$$\forall i \in [1, \ldots, m+1] : c_{i,r+1} = \sum_{j=1}^{r} c_{i,j}$$

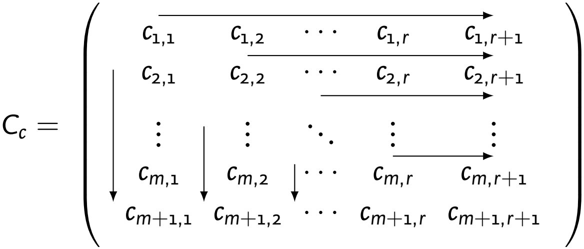
- At least one soft error

$$\exists j \in [1, \ldots, r+1] : c_{m+1,j} \neq \sum_{i=1}^{m} c_{i,j}$$

$$\exists i \in [1, \ldots, m+1] : c_{i,r+1} \neq \sum_{j=1}^{r} c_{i,j}$$

Online Selftest - Basics

Implementation Optimization



$$C_c = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,r} & c_{1,r+1} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,r} & c_{2,r+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,r} & c_{m,r+1} \\ c_{m+1,1} & c_{m+1,2} & \cdots & c_{m+1,r} & c_{m+1,r+1} \end{pmatrix}$$

- Alternating row and column multiplication
- Checksums can be verified online → reduce the overhead
- Square matrices ($n = m = r$) are used to simplify implementation

Online Selftest - Example

1) Starting with 2 Matrices $A, B \in \mathbb{R}^{3 \times 3}$

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 0 \\ 3 & 1 & 2 \end{pmatrix} \in \mathbb{R}^{3x3}, \qquad B = \begin{pmatrix} 0 & 4 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 4 \end{pmatrix} \in \mathbb{R}^{3x3} \tag{1}$$

2) Adding column and row sums to A and B

$$A_c = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 0 \\ 3 & 1 & 2 \\ 6 & 5 & 3 \end{pmatrix} \in \mathbb{R}^{4x3}, \qquad B_c = \begin{pmatrix} 0 & 4 & 1 & 5 \\ 1 & 2 & 3 & 6 \\ 1 & 3 & 4 & 8 \end{pmatrix} \in \mathbb{R}^{3x4} \tag{2}$$

Online Selftest - Example

3) Multiplication and results review

$$A_c \cdot B_c = C_c$$

Online Selftest - Example

3) Multiplication and results review

$$A_c \cdot B_c = C_c$$

- No soft errors

$$C_c = \begin{pmatrix} 2 & 13 & 9 & 24 \\ 3 & 10 & 10 & 23 \\ 3 & 20 & 14 & 37 \\ 8 & 43 & 33 & 84 \end{pmatrix} \in \mathbb{R}^{4x4}$$

Online Selftest - Example

3) Multiplication and results review

$$A_c \cdot B_c = C_c$$

- No soft errors

$$C_c = \begin{pmatrix} 2 & 13 & 9 & 24 \\ 3 & 10 & 10 & 23 \\ 3 & 20 & 14 & 37 \\ 8 & 43 & 33 & 84 \end{pmatrix} \in \mathbb{R}^{4x4}$$

- At least one soft error

$$C_c = \begin{pmatrix} 2 & 13 & 18 & 24 \\ 3 & 10 & 10 & 23 \\ 3 & 20 & 14 & 37 \\ 8 & 43 & 33 & 84 \end{pmatrix} \in \mathbb{R}^{4x4}$$

Online Selftest - Example

3) Multiplication and results review

$$A_c \cdot B_c = C_c$$

- No soft errors

$$C_c = \begin{pmatrix} 2 & 13 & 9 & 24 \\ 3 & 10 & 10 & 23 \\ 3 & 20 & 14 & 37 \\ 8 & 43 & 33 & 84 \end{pmatrix} \in \mathbb{R}^{4x4}$$

- At least one soft error

$$C_c = \begin{pmatrix} 2 & 13 & 18 & 24 \\ 3 & 10 & 10 & 23 \\ 3 & 20 & 14 & 37 \\ 8 & 43 & 33 & 84 \end{pmatrix} \in \mathbb{R}^{4x4}$$

- Online implementations allows to finish after first row calculation

Automatic test equipment

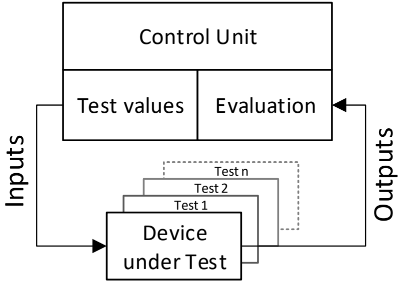
How to provoke soft errors?

- Under normal conditions, soft error only occur rarely
  - For evaluation purposes, they have to be provoked
  - **Undervolting** is used for this purpose
    - Lower supply voltage increases risk of soft errors

Automatic test equipment

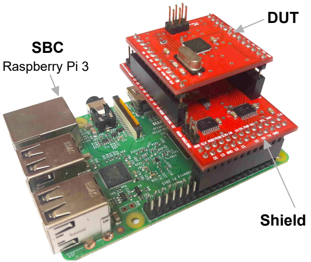
How to provoke soft errors?

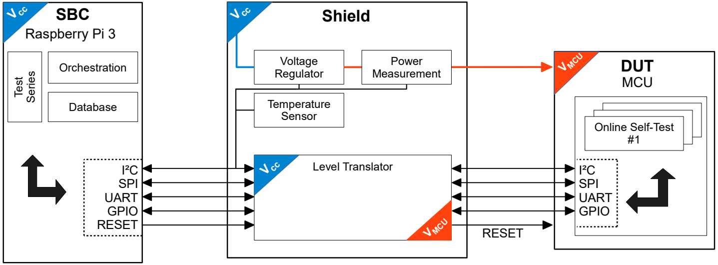
- Under normal conditions, soft error only occur rarely
  - For evaluation purposes, they have to be provoked
  - **Undervolting** is used for this purpose
    - Lower supply voltage increases risk of soft errors
- Test machine
  - Automatic testing platform for various MCUs
  - 100 test-iterations per voltage step
  - Multiple instances of common 8-bit MCUs were used
    - Atmel ATmega1284P
    - Microchip PIC18lf27j13



Automatic test equipment

Implementation of our automatic test equipment



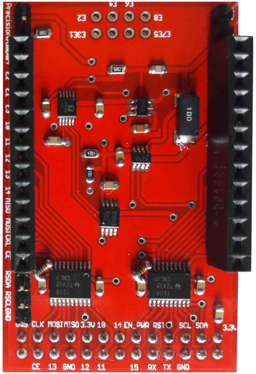

V_{cc}
SBC
Raspberry Pi 3
Test Series
Orchestration
Database
I²C
SPI
UART
GPIO
RESET

V_{cc}
Shield
Voltage Regulator
Power Measurement
Temperature Sensor
V_{cc}
Level Translator
V_{MCU}
RESET

V_{MCU}
DUT
MCU
Online Self-Test #1
I²C
SPI
UART
GPIO

Automatic test equipment

Evaluation of our automatic test equipment

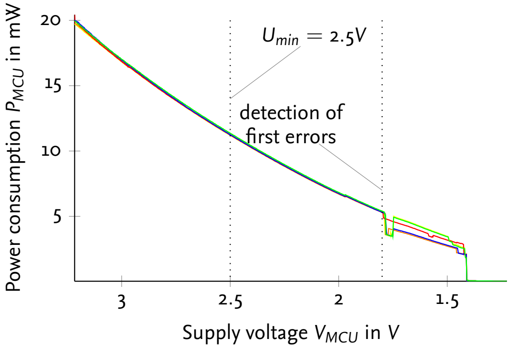


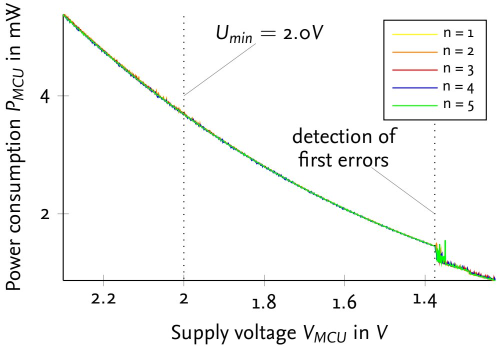
- Two voltage ranges: $3.3V - 1.2V$ and $2.3V - 1.2V$
- Resolution: $2mV$ and $1mV$
- Mean error in voltage regulation: $0.004V$
- Mean error in current measurement: $0.0052mA$

Evaluation

Power consumption during the selftest for different matrix sizes *n*

- Atmel ATmega1284P ($T = 20^\circ$C)
- Microchip PIC18lf27j13 ($T = 20^\circ$C)

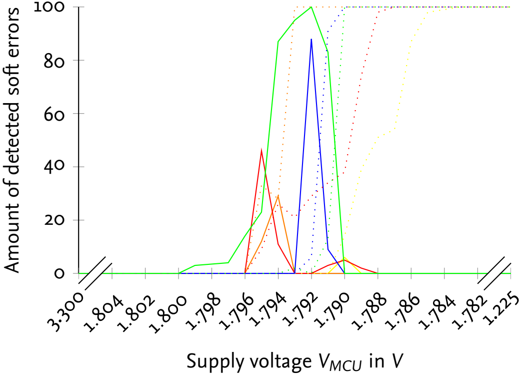




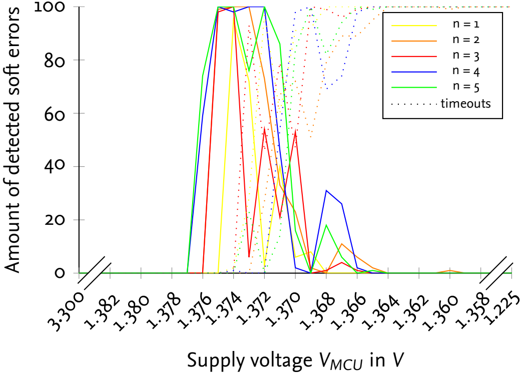
Evaluation

Evaluation for different matrix sizes

- Atmel ATmega1284P ($T = 20^\circ$C)



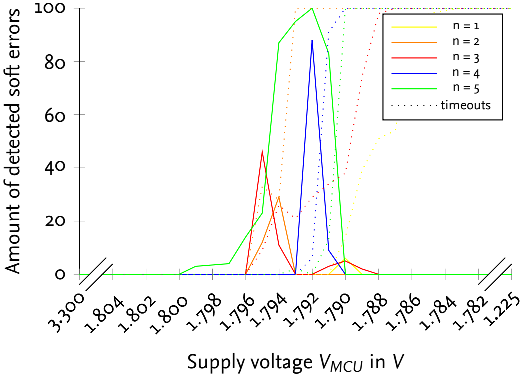
- Microchip PIC18lf27j13 ($T = 20^\circ$C)



Evaluation

Evaluation for different matrix sizes

- Atmel ATmega1284P ($T = 20^\circ$C)



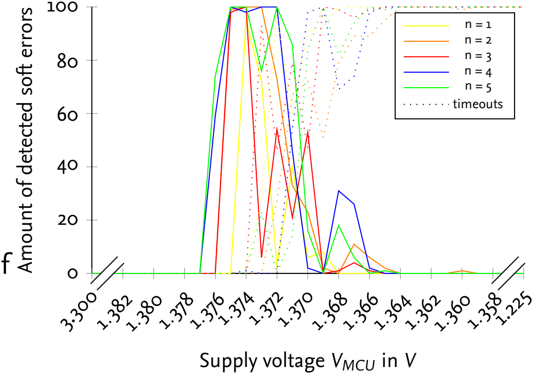
- MCU shows unreliable behaviour for $V_{MCU} \lesssim 1.8V$
  - soft errors
  - timeouts
- Amount of detected soft errors increases with lower voltage
- Amount of detected soft errors increases with higher matrix sizes

Evaluation

Evaluation for different matrix sizes

- MCU shows unreliable behaviour for $V_{MCU} \leq 1.377V$
  - soft errors
  - timeouts
- Amount of detected soft errors increases with lower voltage
- Weaker relationship between the amount of soft errors and higher matrix sizes

- Microchip PIC18lf27j13 ($T = 20°C$)



Evaluation

Comparison with other methods

- Online Selftest
  - with matrix size n = 5

Evaluation

Comparison with other methods

- Online Selftest
  - with matrix size n = 5
- Software TMR
  - exemplary function

Evaluation

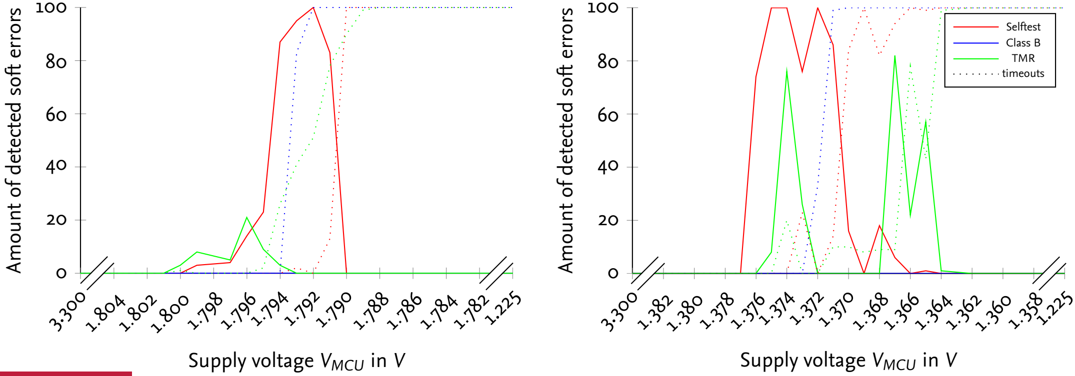
Comparison with other methods

- Online Selftest
  - with matrix size n = 5
- Software TMR
  - exemplary function
- Modified Class B Test
  - focus on SRAM
  - iterate through the entire SRAM, toggle and check every bit

Evaluation

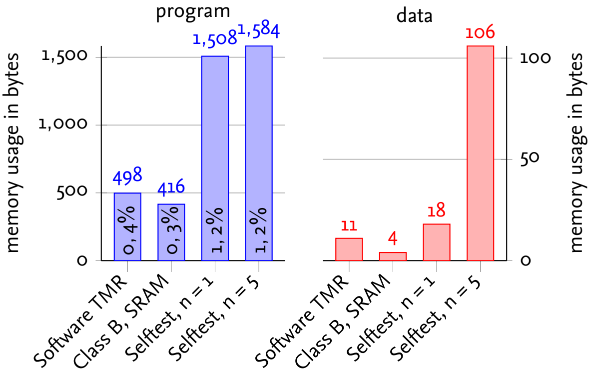
Comparison with other methods

- Atmel ATmega1284P (20°C)
- Microchip PIC18lf27j13 (20°C)



Evaluation

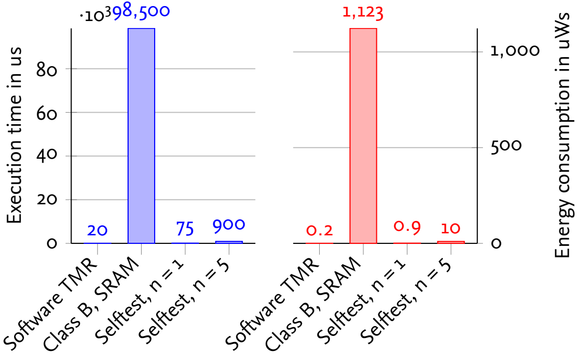
Memory requirement - Atmel ATmega1284P



- Class B uses the least memory usage
- Software TMR uses slight more memory
- Selftest needs the most memory, but still only 1.2% of ATmega1284P’s program memory

Evaluation

Execution time and energy requirement - Atmel ATmega1284P



- Class B shows (by far) the longest execution time
- Software TMR offers the lowest execution time
- Selftest slightly worse then Software TMR, much better then Class B
  → very reasonable overhead

Summary

Soft errors

- Soft errors in computer systems

Summary

Soft errors

- Soft errors in computer systems

Online Selftest

- Software based method to detect soft errors
- Focus on the most error-prone component: ALU
- Use of fault-tolerant matrix multiplication, online capable implementation

Summary

Soft errors

- Soft errors in computer systems

Online Selftest

- Software based method to detect soft errors
- Focus on the most error-prone component: ALU
- Use of fault-tolerant matrix multiplication, online capable implementation

Evaluation

- Compared against well-known methods for error detection (TMR, Class B)
- Online Selftest showed the highest detection rates and medium overhead
  → best compromise for Energy-Constrained WSNs

Summary

Soft errors

- Soft errors in computer systems

Online Selftest

- Software based method to detect soft errors
- Focus on the most error-prone component: ALU
- Use of fault-tolerant matrix multiplication, online capable implementation

Evaluation

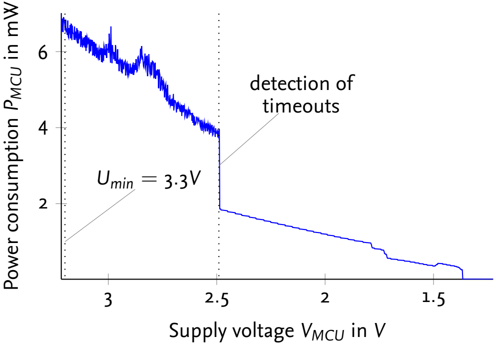
- Compared against well-known methods for error detection (TMR, Class B)
- Online Selftest showed the highest detection rates and medium overhead
  → best compromise for Energy-Constrained WSNs

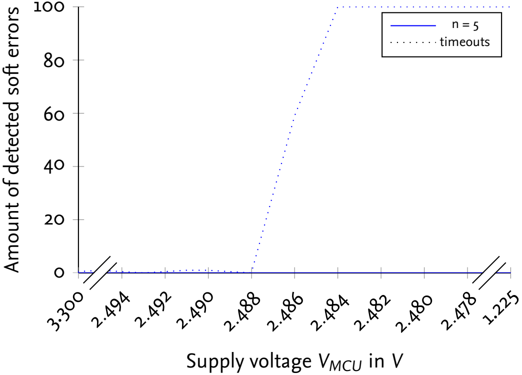
Thank you for your attention! Questions?

Ulf Kulau, Daniel Szafranski
`kulau@ibr.cs.tu-bs.de, d.szafranski@tu-bs.de`

Evaluation

Texas Instruments MSP430F2013 Results ($f_{CPU} = 16Mhz$, $T = 23^{\circ}$C)





Evaluation

Atmel ATtiny85-20PU Results ($f_{CPU} = 8Mhz$, $T = 19^{\circ}$C)