Design and Evaluation of MPSoC ECU Architectures for Secure and Dependable Automotive CPS

Dr. Arslan Munir

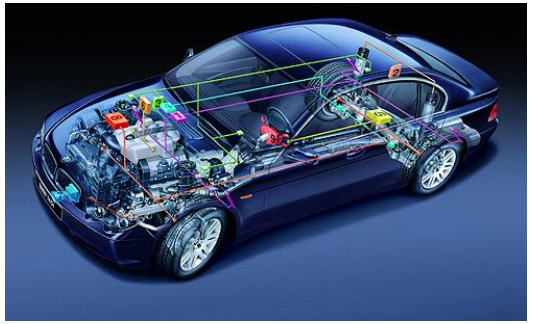
Intelligent Systems, Computer Architecture, Analytics, and Security (ISCAAS) Laboratory

Department of Computer Science
Kansas State University

Presentation for ARM Research Summit

This work was supported by the National Science Foundation (NSF) (NSF CNS 1743490)

Automotive CPS



Modern Automobiles

More than 100 electronic control units (ECUs)

Various distributed control applications

CAN with Flexible Data Rate (CAN FD) & FlexRay

Controller area network (CAN)

Next generation of automobiles

Further escalate the proliferation of ECUs

Cybercars

Applications

ECUs substitute the traditional mechanical or hydraulic systems

X-by-Wire

Steer-by-Wire (SBW)

Brake-by-Wire

Most prevalent protocol for communication among ECUs

Automotive CPS

Cybercar design challenges

Failure to meet

Catastrophic consequences

- Dependability
- Security
- Quality of service (QoS)
- Limited resources

ISO 26262 ➔ automotive safety integrity levels (ASILs)

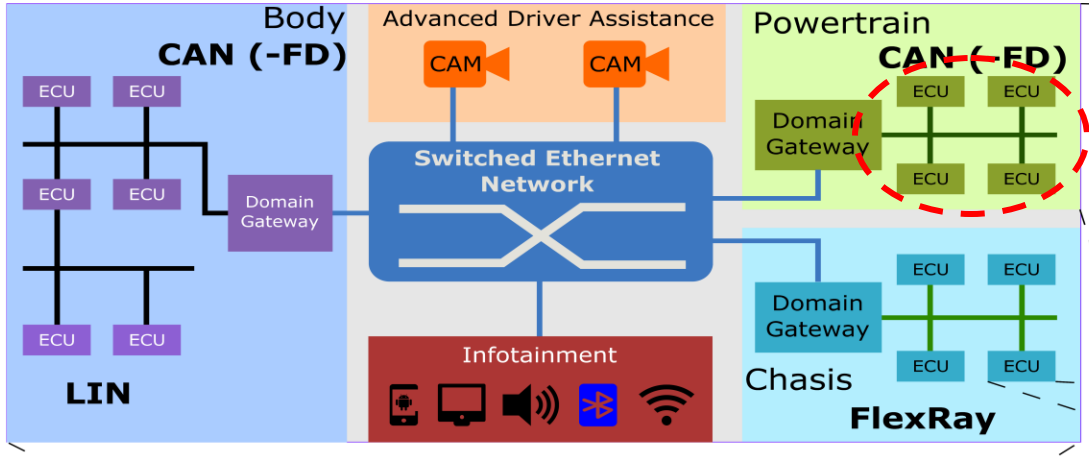
Biggest challenge

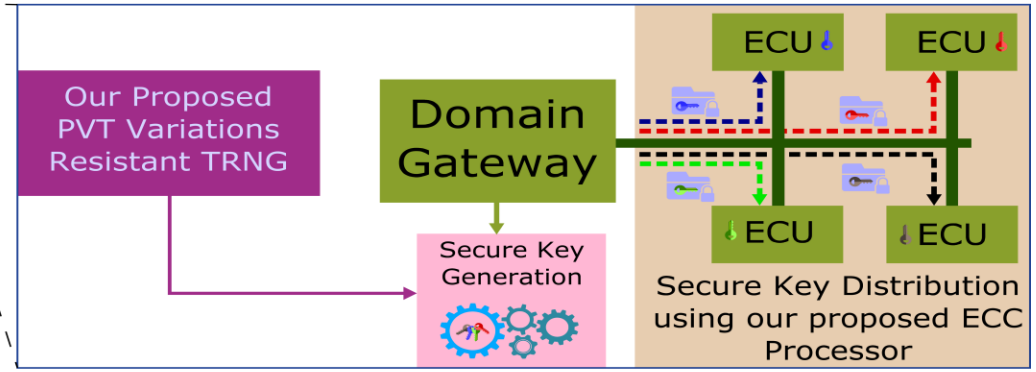
- Computing
- Memory
- Bandwidth
- Cost

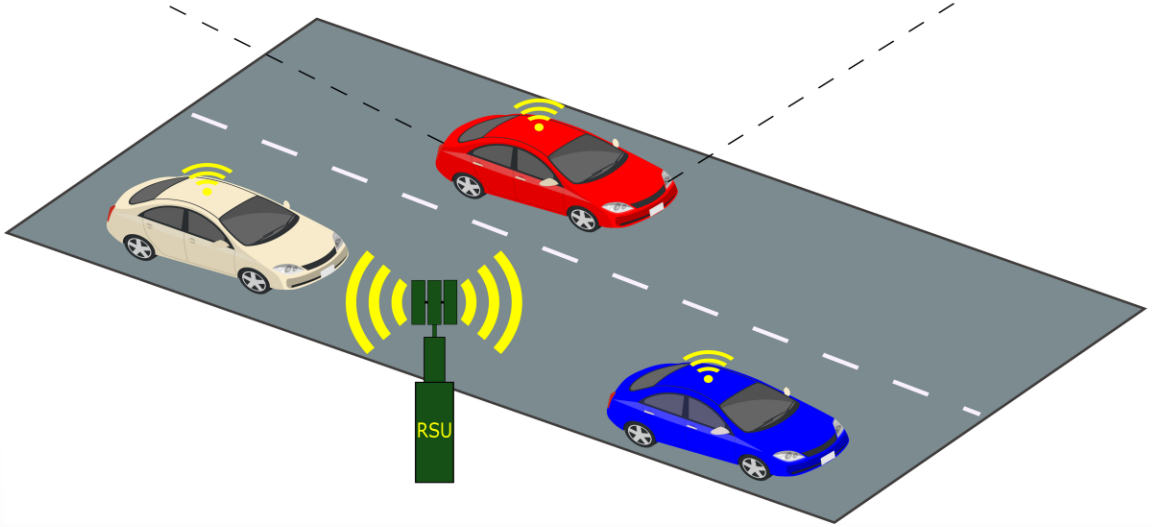
Simultaneous integration of security & dependability without violating hard real-time constraints imposed by desired QoS

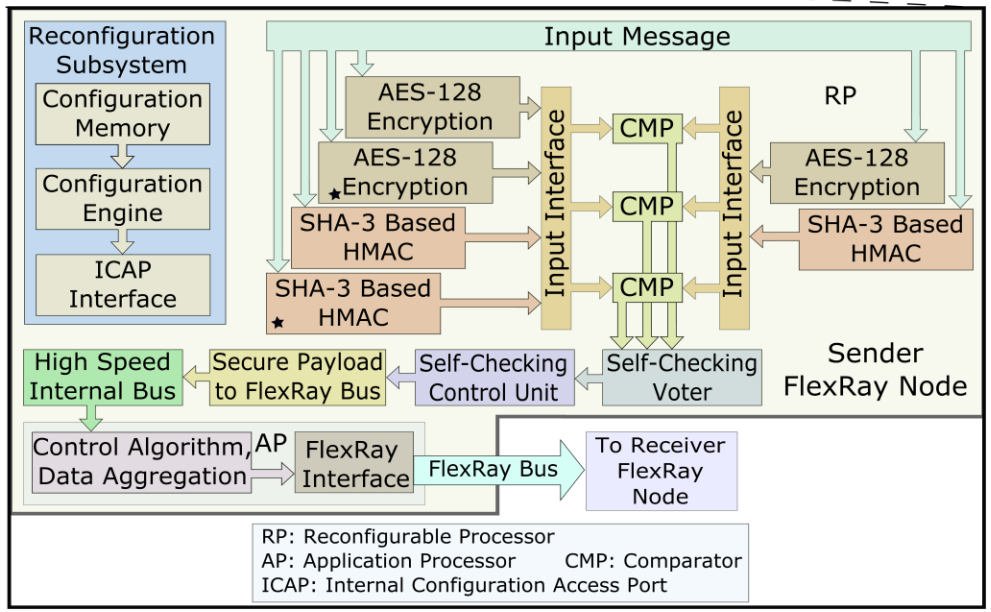
Focus of our work

Design of Secure and Dependable Automotive CPS









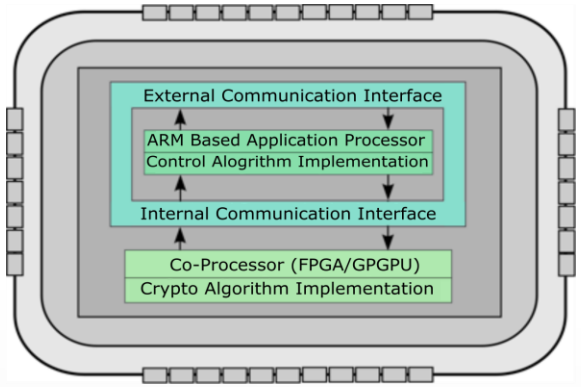
ECU Architectures for Secure and Dependable Automotive CPS

Contributions

An integrated Approach for Designing Secure & Dependable Cybercars

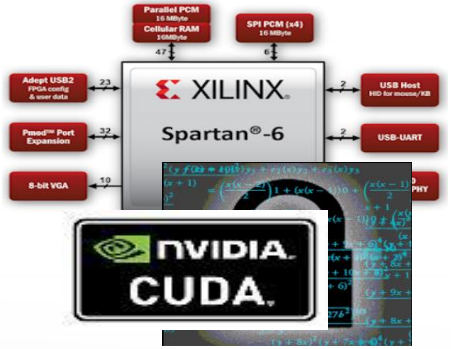
A. NOVEL ECU ARCHITECTURES

Proposal of novel ECU architectures that incorporate security and dependability primitives



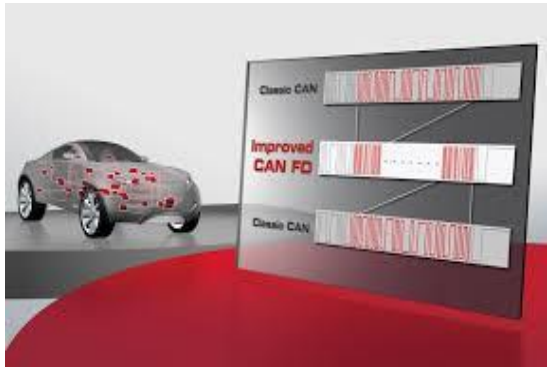
B. ECU IN FPGA & GPGPU

Implementation of cryptographic module of proposed ECUs in Xilinx automotive spartan-6 FPGA and Jetson TK1 GPU



C. CAN, CAN FD & FlexRay ECU COMMUNICATION BUS

Embedding & analyzing security primitives over CAN, CAN FD and FlexRay bus for the proposed ECUs



D. COMPARISON

Timing performance and energy efficiency comparison of the proposed ECUs with the prior design

Security and Dependability Primitives Used in Proposed ECU

Dependability → **Goal**

- Tolerate one permanent fault
- Tolerate multiple soft errors

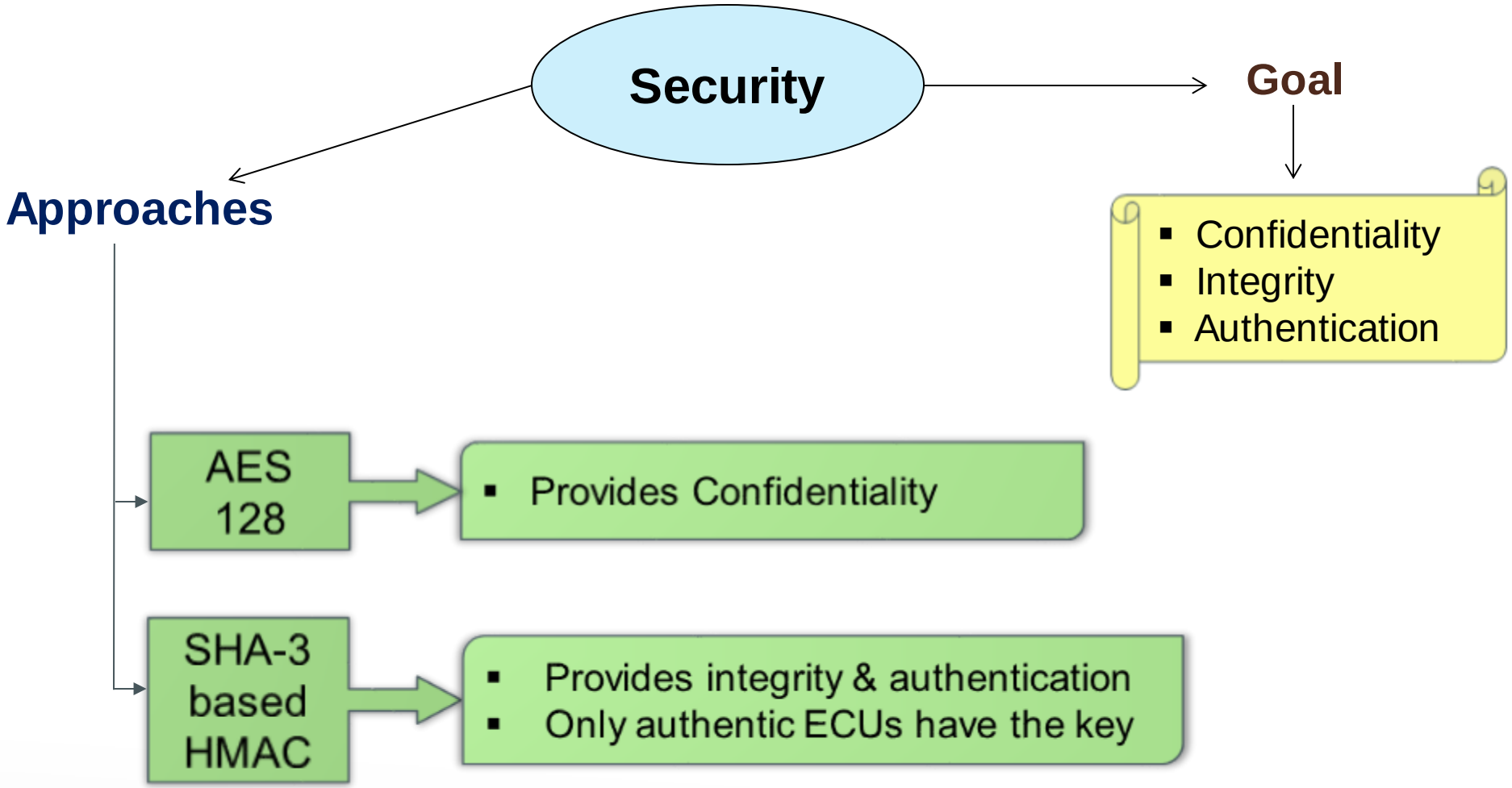
FT-RMT

- Fault Tolerance using Redundant Multithreading
- Recomputation to remove soft errors
- Tolerate one permanent fault
- Used in BED and GED

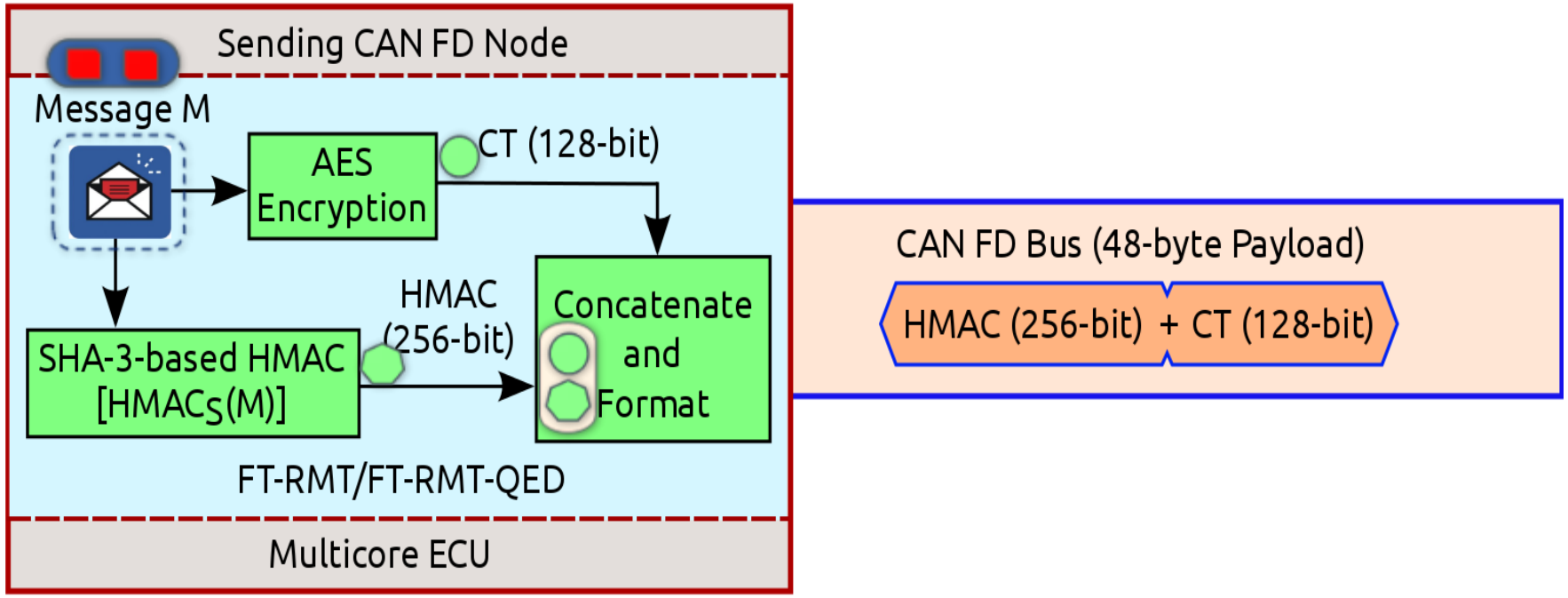
FT-SR-DMR

- Fault Tolerance using Self-Reconfiguration in Dual Modular Redundant System
- DMR is used to detect error in computation between two redundant modules
- Recomputation to remove soft errors
- Self-reconfiguration of faulty module for self-healing
- Used in RED

Security and Dependability Primitives Used in Proposed ECU

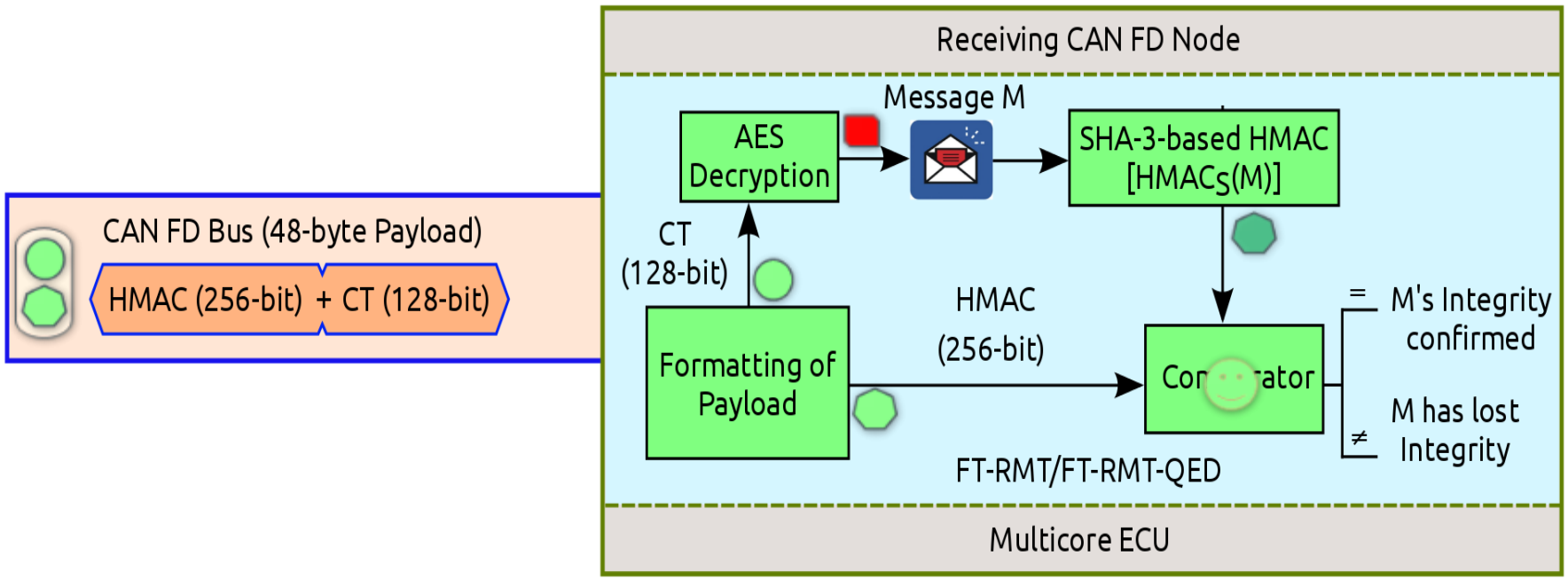


Baseline ECU Design

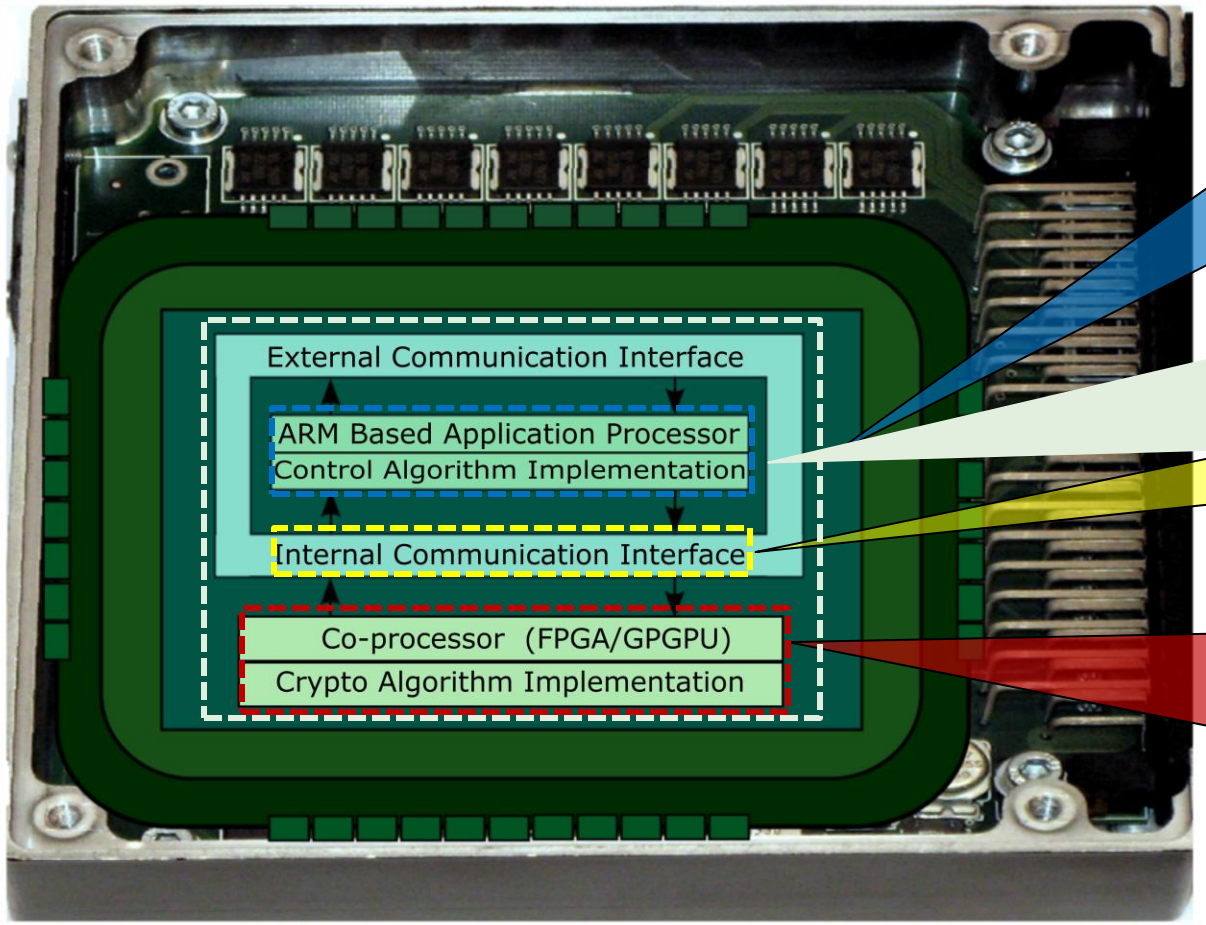


- Arslan Munir and Farinaz Koushanfar, "Design and Analysis of Secure and Dependable Automotive CPS: A Steer-by-Wire Case Study", *IEEE Transactions on Dependable and Secure Computing (TDSC), 2018.*

Baseline ECU Design



High Level Architecture of Proposed ECUs



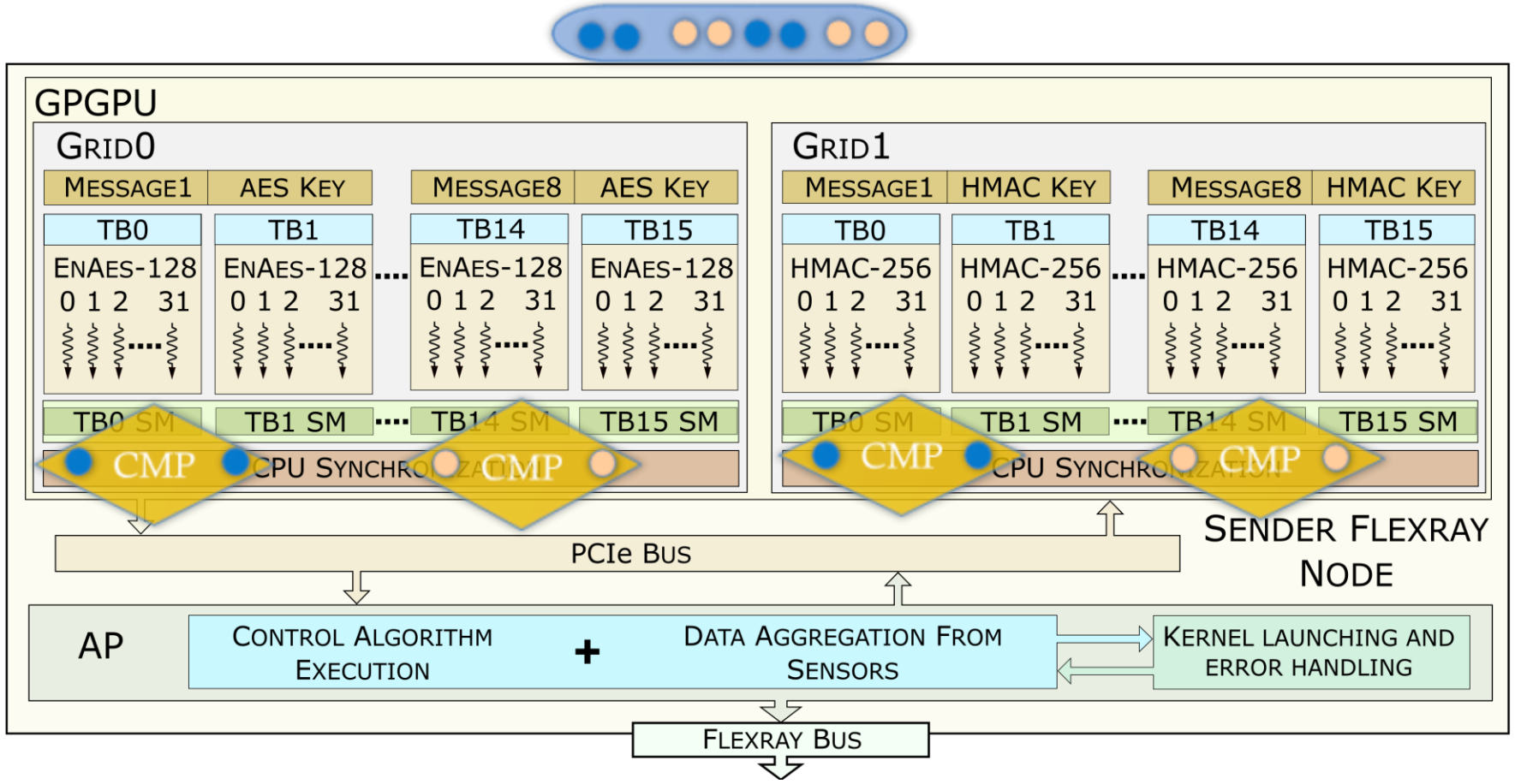
- **Main processor:** interacts with sensors and actuators
- Controls the functionality and communication with the co-processor
- Main function is to carry out control algorithm execution

- Two programmable processors
- **Application processor** is the main processor with the FPGA or GPGPU as **co-processor**
- Two processors communicate by some high-speed internal bus

- (CM) that can perform AES, HMAC, key managements, etc.
- Can be extended to implement compute-intensive algorithms without incurring extra cost

GPGPU ECU Design (GED)

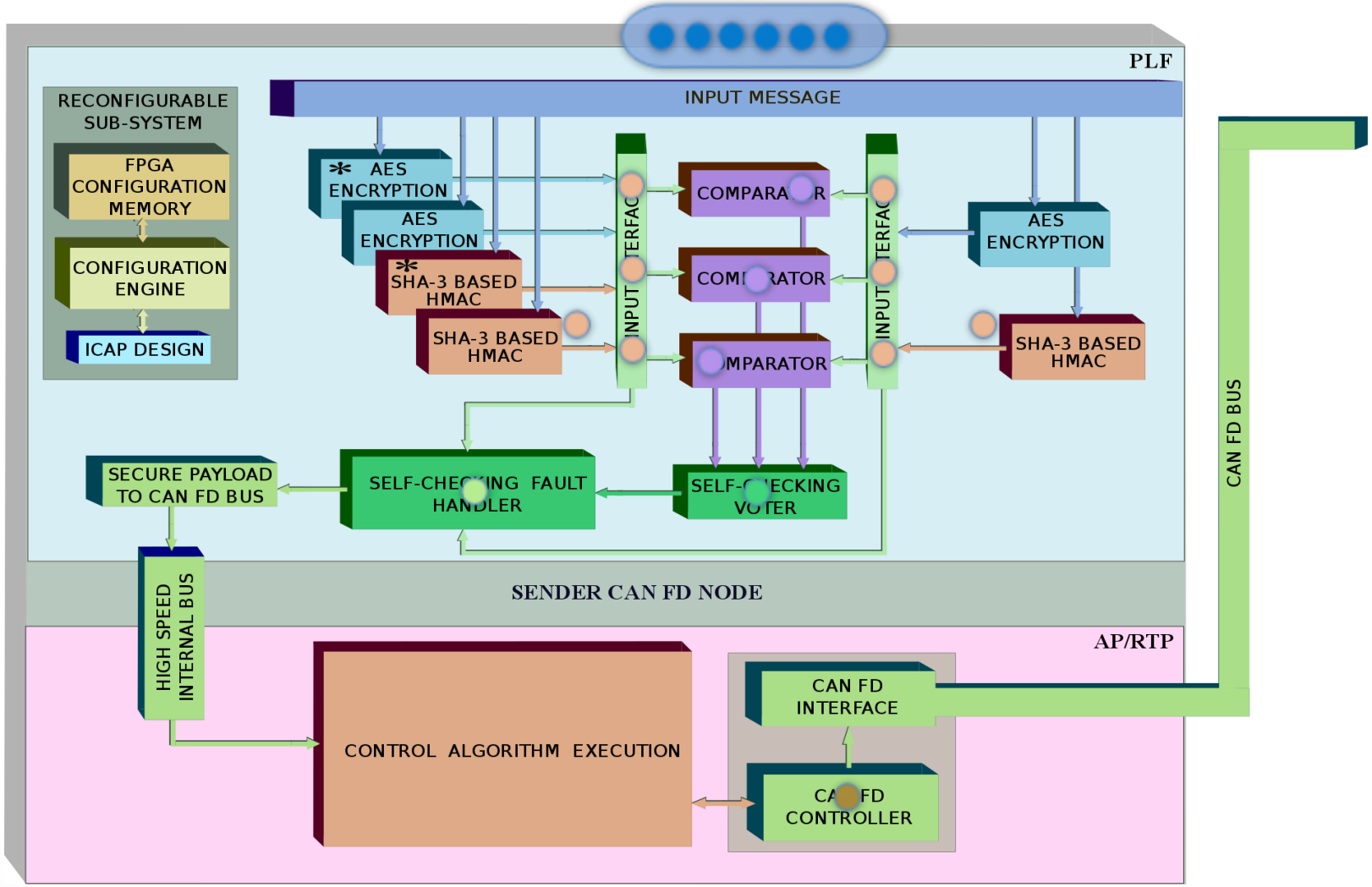
Sender Node



AP:APPLICATION PROCESS
ENAES: ENCRYPTION AES
TB: THREAD BLOCK
DEAES: DECRYPTION AES
SM:SHARED MEMORY
CMP: COMPARE

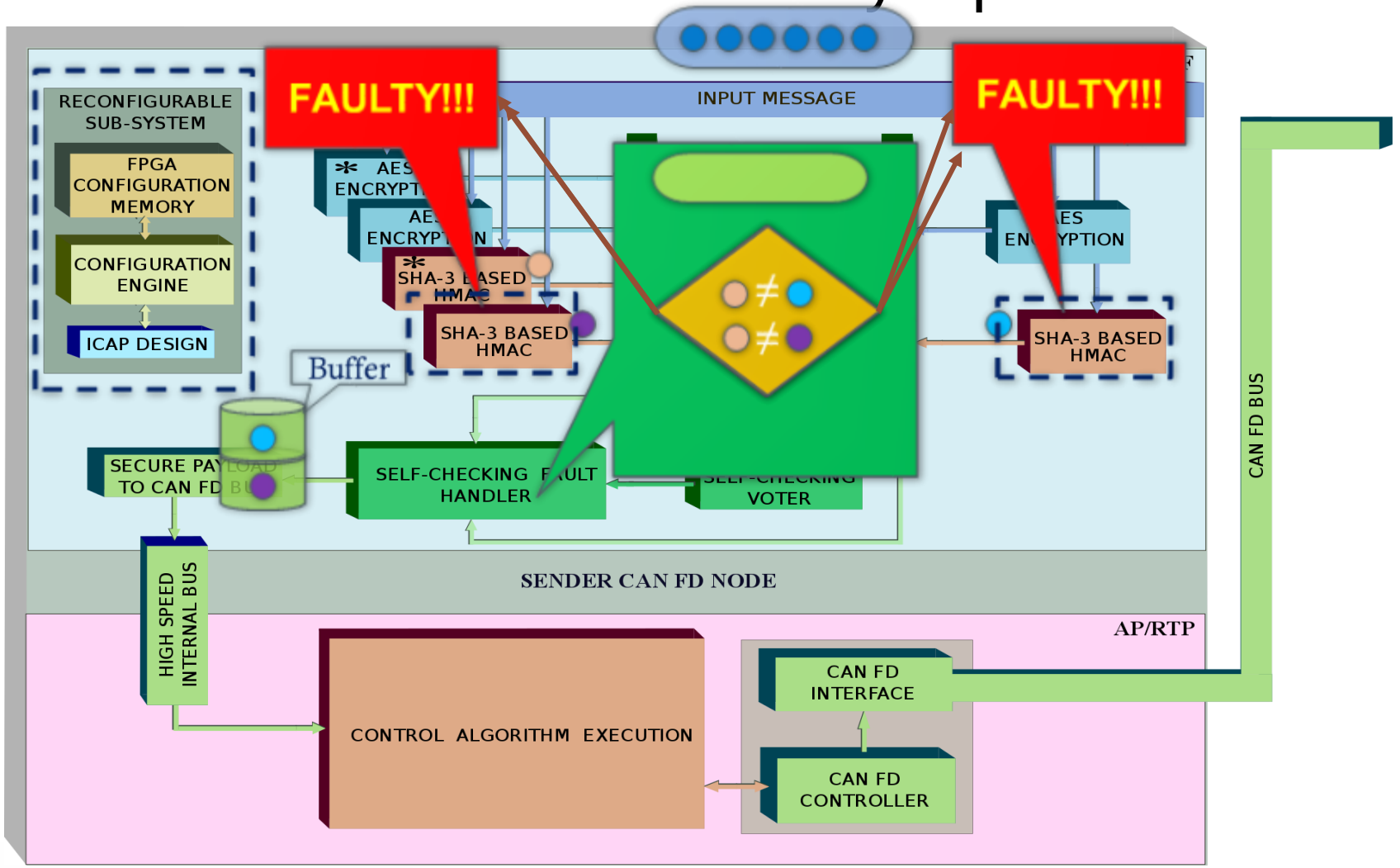
Reconfigurable ECU Design (RED)

Sender Node: Non-Faulty Operation



Reconfigurable ECU Design (RED)

Sender Node: Faulty Operation



Experimental Setup

❖32-bit NXP iMX6Q SABRE Board

❖Quad core symmetric multiprocessor architecture @396 MHz

❖OS - Ubuntu 14.04.4 LTS

❖Implemented BED

❖NVIDIA Tegra Jetson TK1

❖192 CUDA Cores

❖OS - Ubuntu 14.04.4 LTS

❖CUDA-C

❖Implemented GED

❖Xilinx Spartan-6 Automotive FPGA

❖@50 MHz

❖Verilog HDL-Xilinx ISE 14.7

❖Xilinx Xpower analyzer to measure power consumption

❖Implemented RED

❖Vector CANoe 8.5

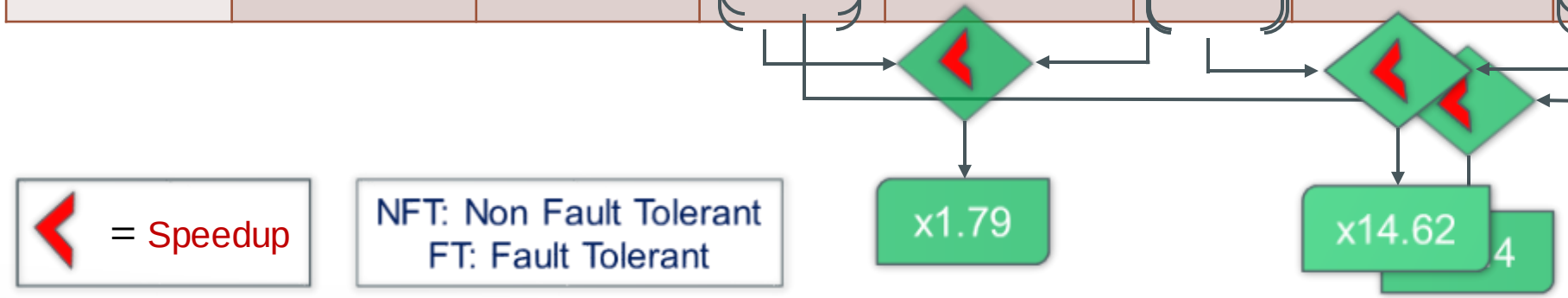
❖Simulation of SBW system

❖Used different baud rate for CAN, CAN-FD and FlexRay

❖CAPL to implement SBW functions on ECUs

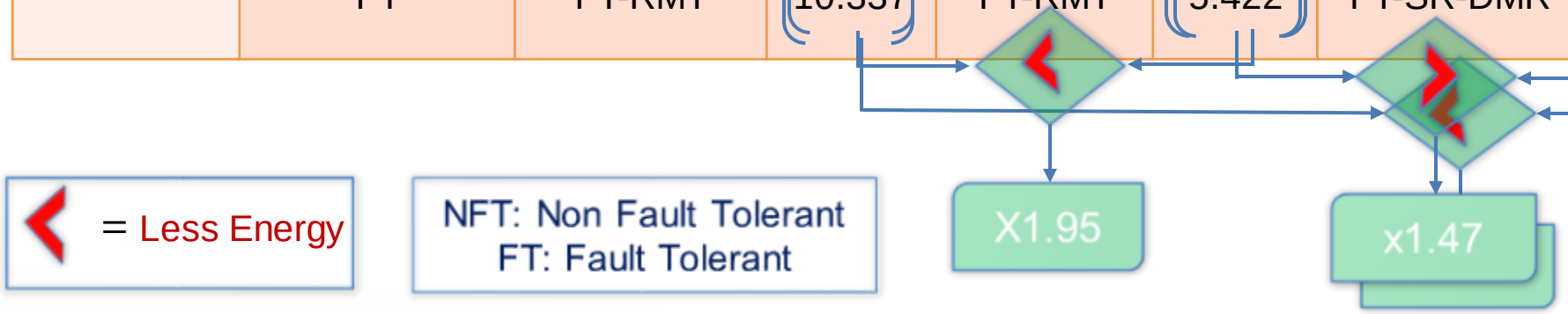
Results—Timing Analysis

CAN FD Node	Operational Mode	BED Implementation		GED Implementation		RED Implementation	
		FT Mode	Time (µs)	FT Mode	Time (µs)	FT Mode	Time (µs)
Sender Node	NFT	None	189	None	99.50	None	4.90
	FT	FT-RMT	207	FT-RMT	112.75	FT-SR-DMR	6.53
Receiver Node	NFT	None	184	None	102.2	None	9.00
	FT	FT-RMT	203	FT-RMT	115.42	FT-SR-DMR	9.63



Results—Energy Analysis

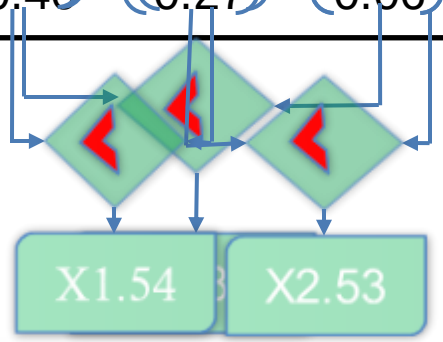
CAN FD Node	Operational Mode	BED Implementation		GED Implementation		FPGA Implementation	
		FT Mode	Energy (µJ)	FT Mode	Energy (µJ)	FT Mode	Energy (µJ)
Sender Node	NFT	None	9.661	None	4.674	None	2.17
	FT	FT-RMT	10.581	FT-RMT	5.297	FT-SR-DMR	6.040
Receiver Node	NFT	None	9.406	None	4.801	None	3.996
	FT	FT-RMT	10.337	FT-RMT	5.422	FT-SR-DMR	9.831



Results—End-to-End Response Time of SBW Subsystem

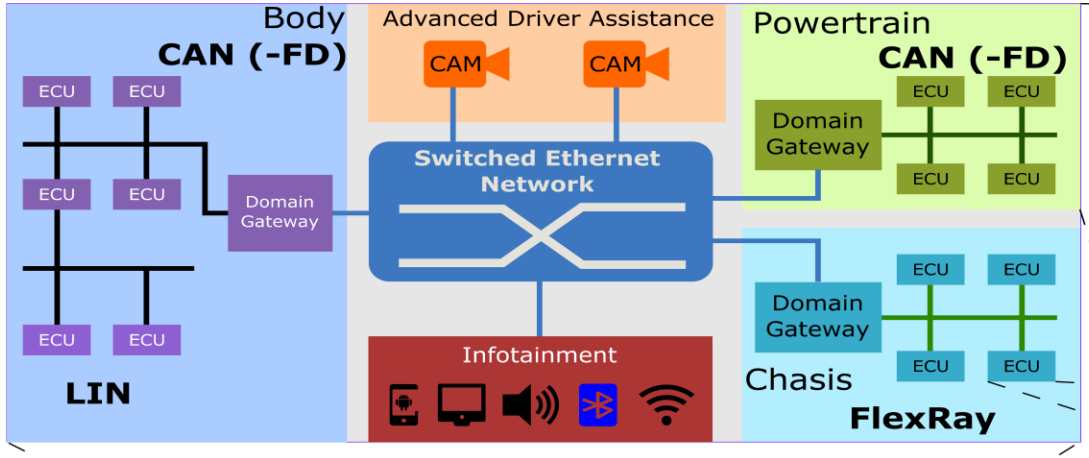
End-to-end delay or response time (in ms) for the three Implementations for CAN, CAN FD and FlexRay

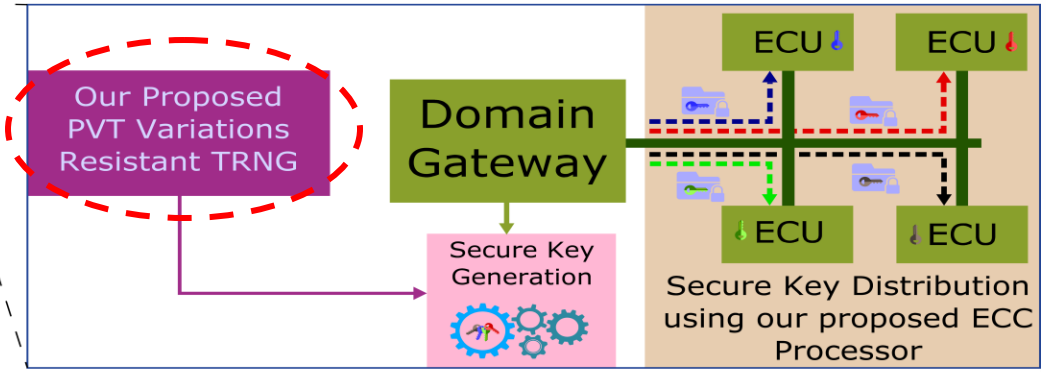
In-vehicle Bus	Operational Mode	BED	GED	RED
CAN Latency = 0.74ms	NFT	4.81	4.64	4.45
	FT	4.85	4.66	4.45
CAN FD Latency = 0.12ms	NFT	0.49	0.32	0.13
	FT	0.53	0.34	0.13
FlexRay Latency = 0.05ms	NFT	0.42	0.25	0.06
	FT	0.46	0.27	0.06

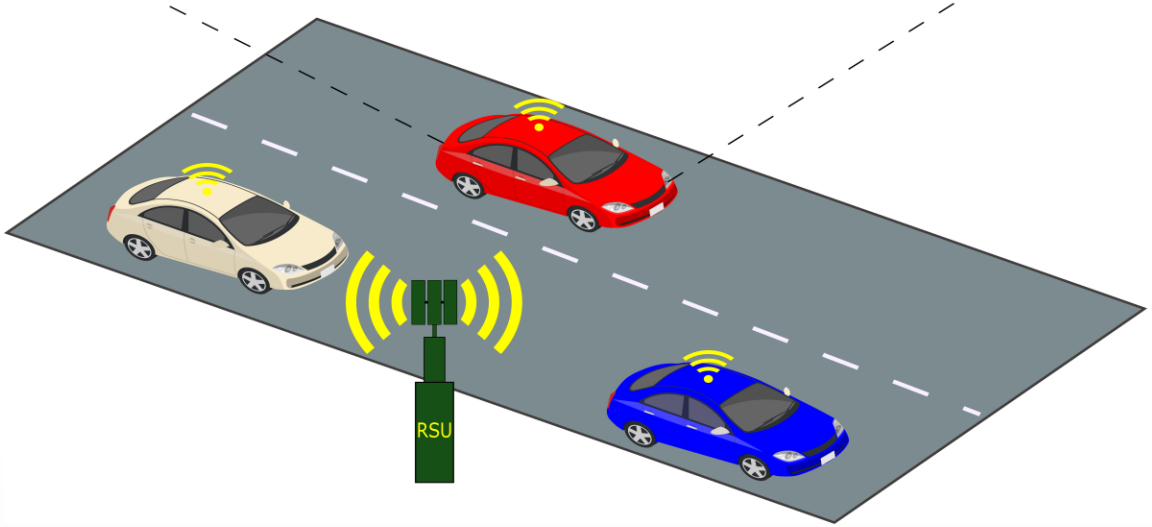


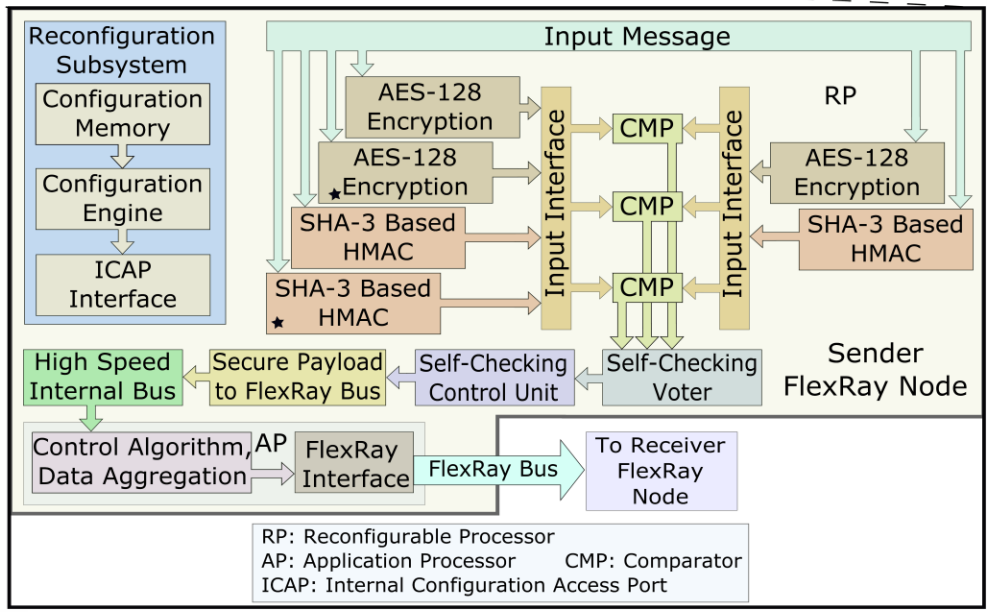
- Bikash Poudel, Naresh Kumar Giri, and Arslan Munir, "Design and Comparative Evaluation of GPGPU- and FPGA-based MPSoC ECU Architectures for Secure, Dependable, and Real-Time Automotive CPS", *Proc. of IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, Seattle, Washington, July 2017.

Design of Secure and Dependable Automotive CPS









Measure of Randomness of TRNG

Probability:

$\text{Prob}(X_{i+1} = \text{'1'} \mid X_i = \text{'1'}) = 0.5$ AND $\text{Prob}(X_{i+1} = \text{'1'} \mid X_i = \text{'0'}) = 0.5$

Entropy:

- Def1: Expected value of the information contained in a message
- Def2: measure of unpredictability of the state
- Entropy (H) can be expressed as,

$$H = -\sum_{k=0}^{n} P(x_i) * \log_b(P(xi))$$

- For a Bernoulli Process like Coin Tossing, $H = -p \log_2(p) - (1-p) \log_2(1-p)$
- For a perfect random source, H = 1

Challenges in Designing a TRNG

Aging →
- Wear-out effect

Operating Conditions →
- Ambient **T**emperature
- Heating of chip during operation
- Interference by neighboring sub-circuit
- Supply **V**oltage variation

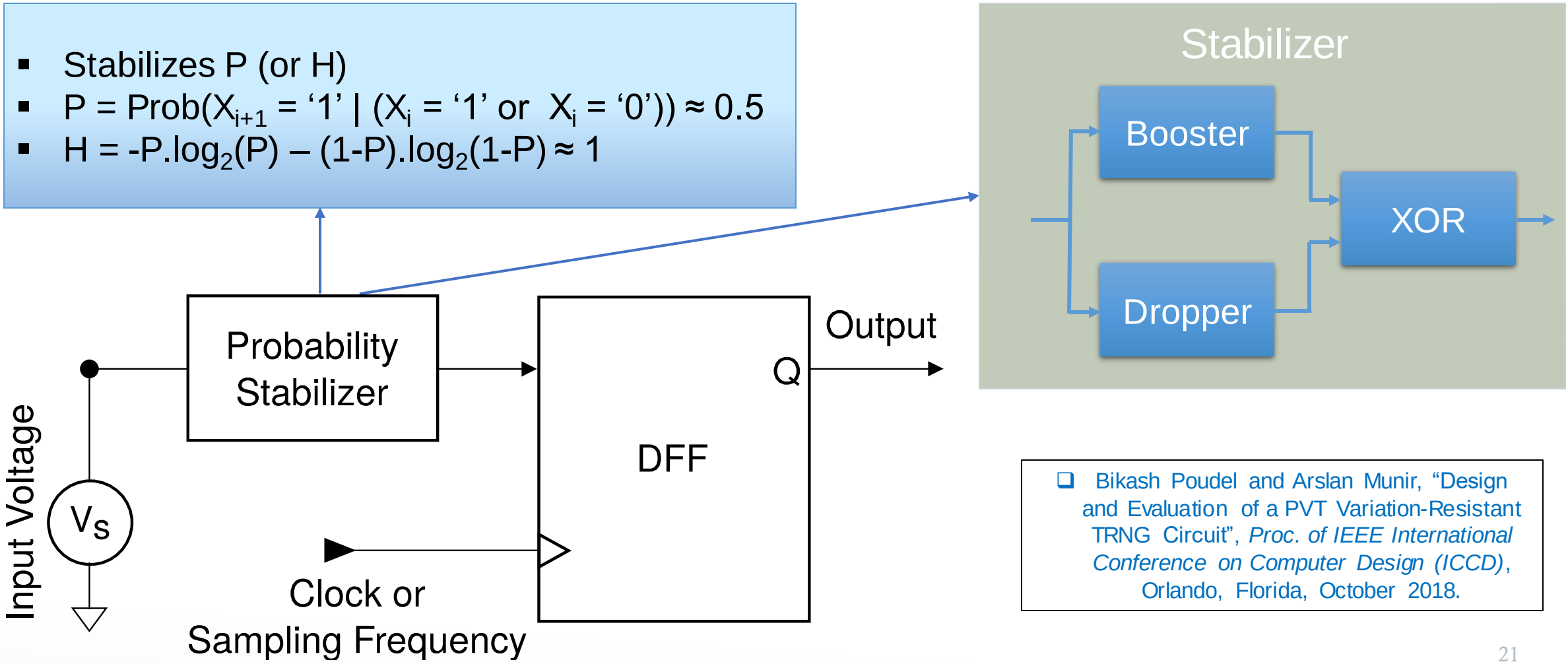
Process Variation →
- Same foundry + same process technology = slight variation in device parameters
- Two transistors manufactured from same foundry using the same process technology can have different *Threshold Voltage*

- Delay
- Power
- Reliability

→ Entropy

High Level Architecture of Proposed TRNG

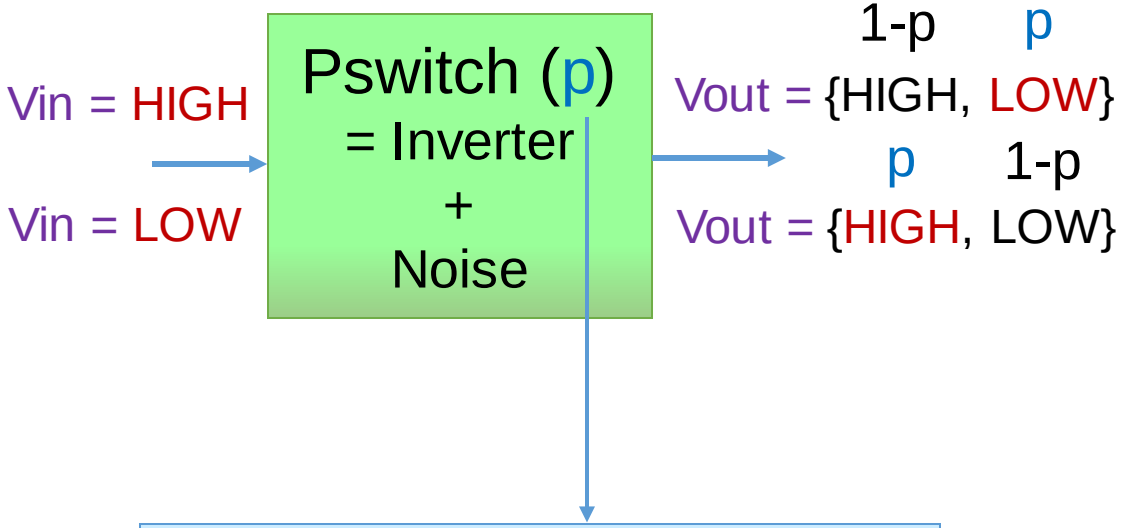
- Stabilizes P (or H)
- $P = Prob(X_{i+1} = '1' \mid (X_i = '1' \text{ or } X_i = '0')) \approx 0.5$
- $H = -P.\log_2(P) - (1-P).\log_2(1-P) \approx 1$



❑ Bikash Poudel and Arslan Munir, "Design and Evaluation of a PVT Variation-Resistant TRNG Circuit", *Proc. of IEEE International Conference on Computer Design (ICCD)*, Orlando, Florida, October 2018.

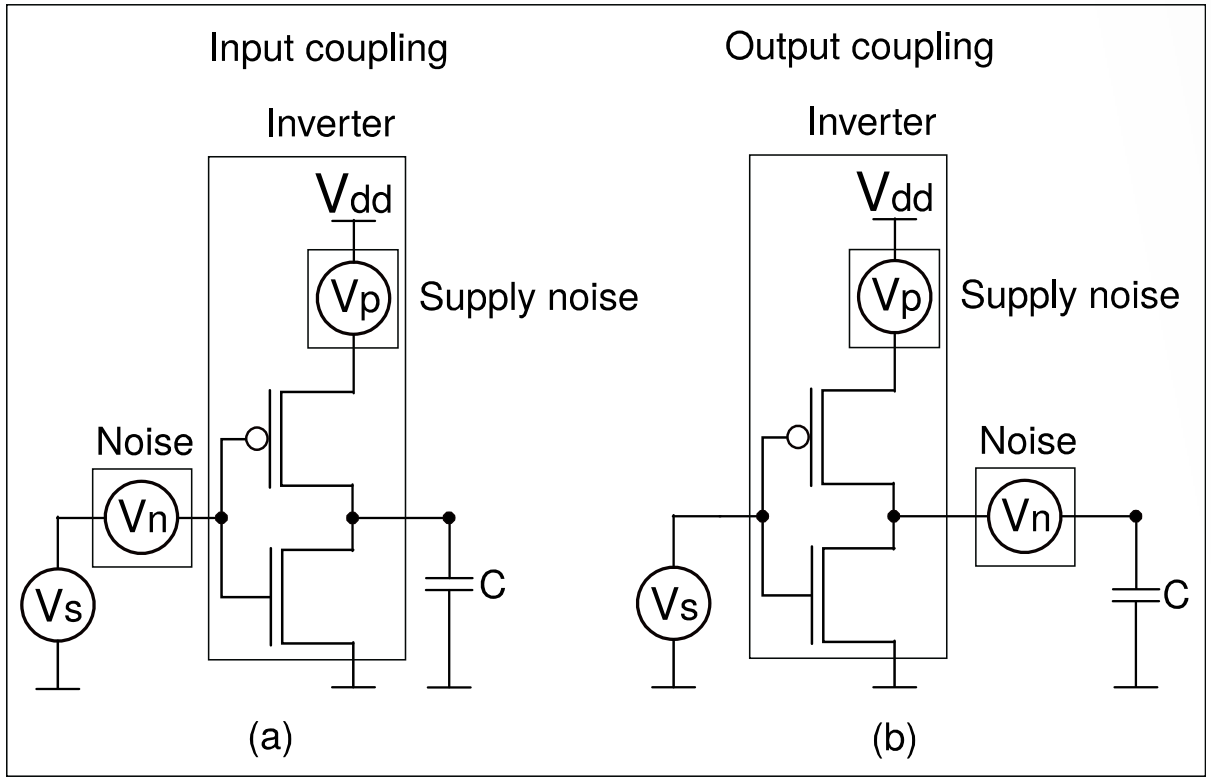
Basic Component: Pswitch

Probabilistic switch as source of entropy



Probability of Correctness:

Prob(Vout = LOW | Vin = HIGH)
OR
Prob(Vout = HIGH | Vin = LOW)



Reference: Korkmaz et. al "Advocating noise as an agent for ultra-low energy computing: Probabilistic complementary metal oxide semiconductor devices and their characteristics", Japanese Journal of Applied Physics, vol. 45, no. 4B, 2006, pp. 3307–3316.

Probability Booster Circuit

Modeling Probability Booster Circuit

B_p = probability of correctness of three parallel pswitches

B_s = probability of correctness of three pswitches in series

p_b = probability of correctness of probability booster circuit

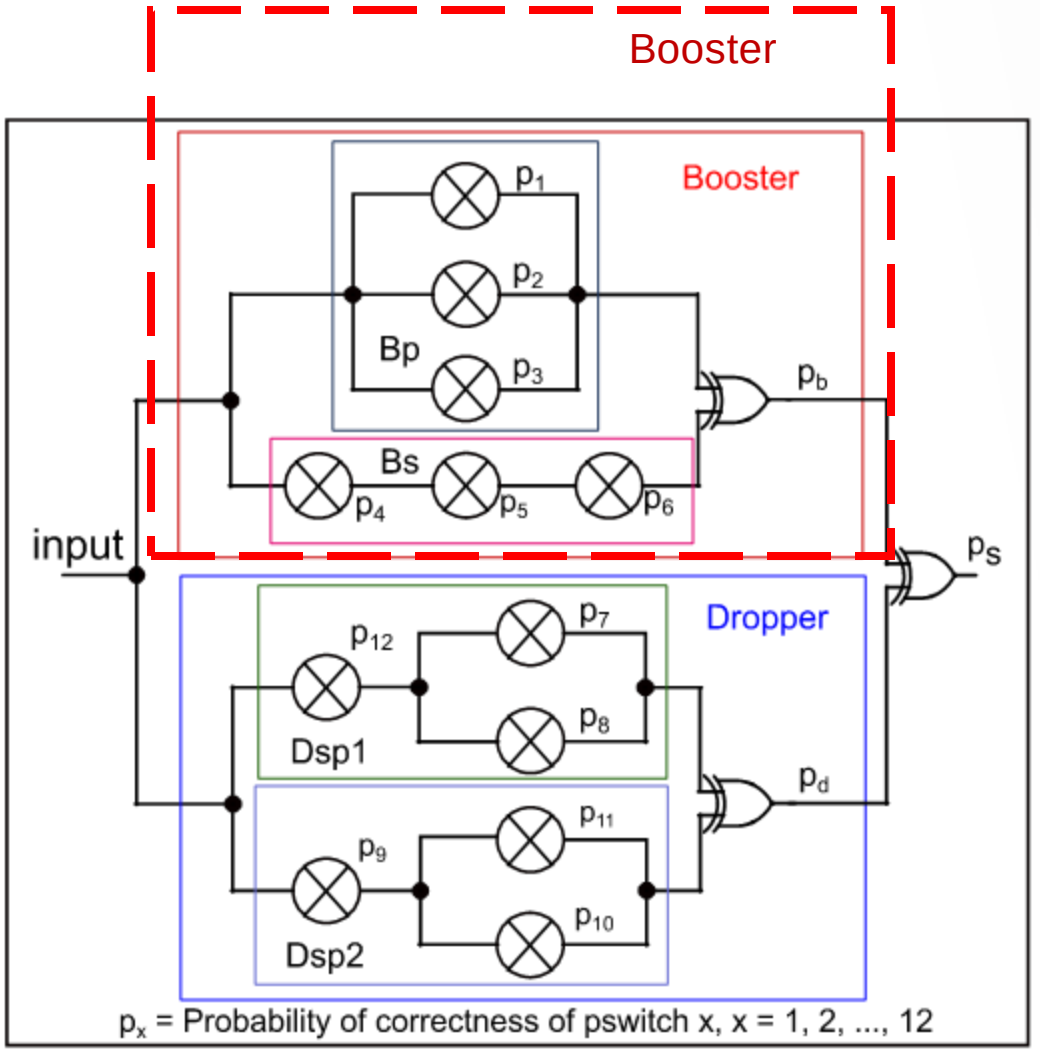
$$B_p = p_1 + p_2 + p_3 - (p_1\,p_2 + p_2\,p_3 + p_3\,p_1) + p_1\,p_2\,p_3$$

$$B_s = p_4\,p_5\,p_6$$

$$p_b = B_p + B_s - 2\,B_p\,B_s$$

If probability of all pswitches in booster circuit are equal,

$$p_b = 3p - 3p^2 + 2p^3 - 6p^4 + 6p^5 - 2p^6 \approx 3p$$



Probability Dropper Circuit

Modeling Probability Dropper Circuit

D_{spx} = probability of correctness of series-parallel pswitches

p_d = probability of correctness of probability dropper circuit

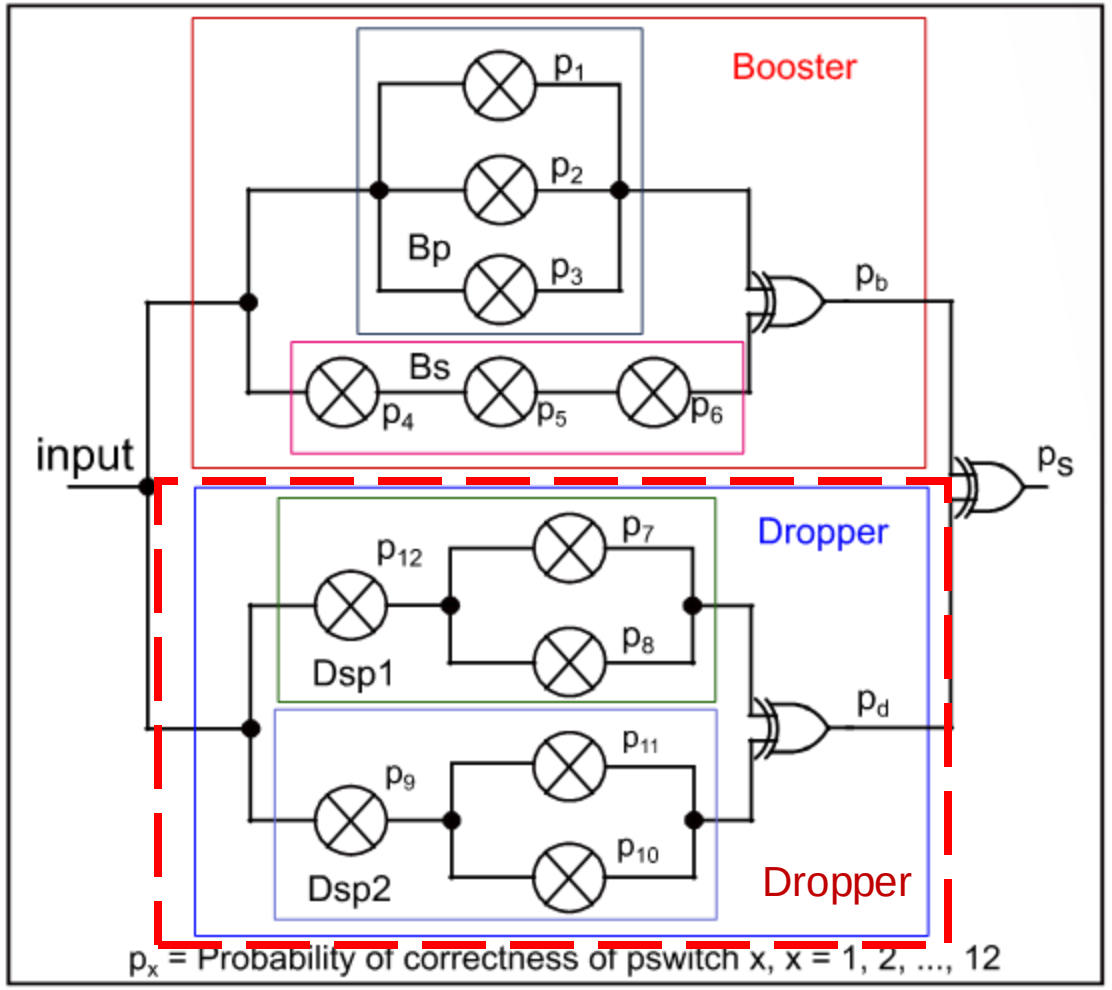
$$D_{sp1} = p_7\,p_{12} + p_8\,p_{12} - p_7\,p_8\,p_{12}$$

$$D_{sp2} = p_9\,p_{10} + p_9\,p_{11} - p_9\,p_{10}\,p_{11}$$

$$p_d = D_{sp1} + D_{sp2} - 2\,D_{sp1}D_{sp2}$$

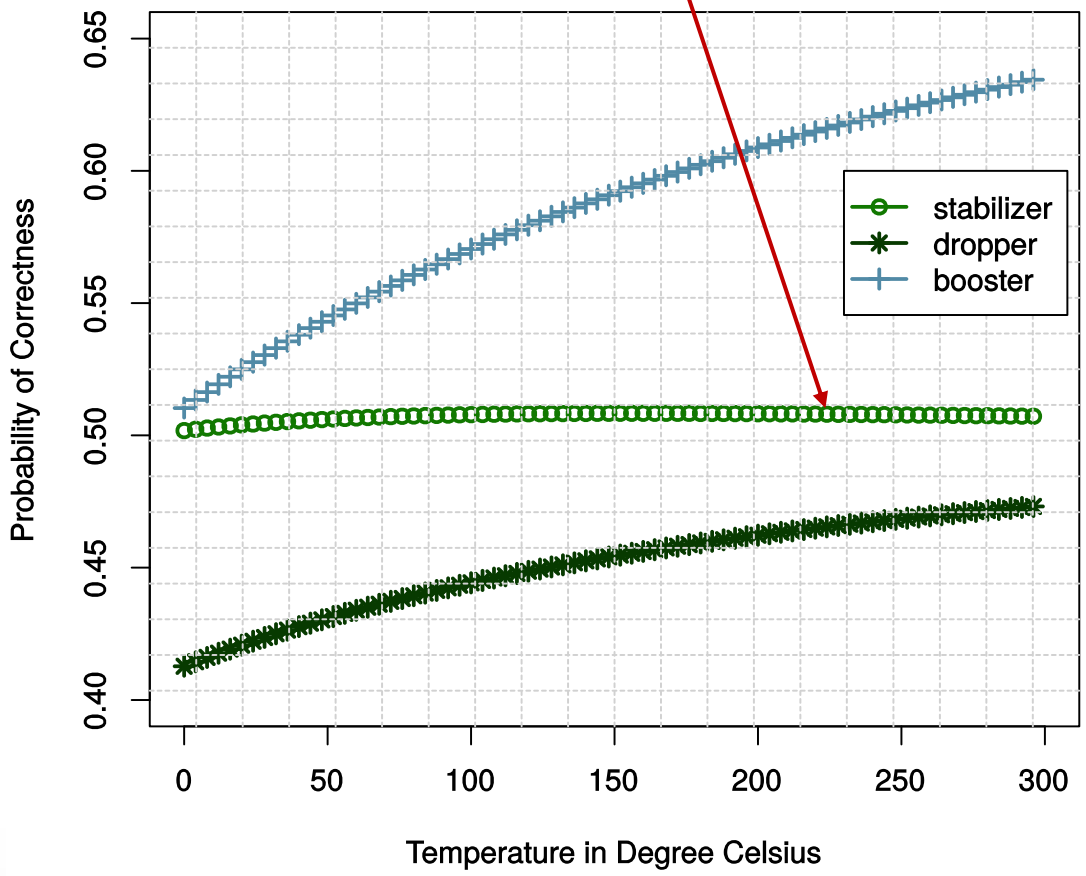
If probability of all pswitches in booster circuit are equal,

$$p_d = 4p^2 - 2p^3 - 8p^4 + 8p^5 - 26p^6 \approx p^2$$

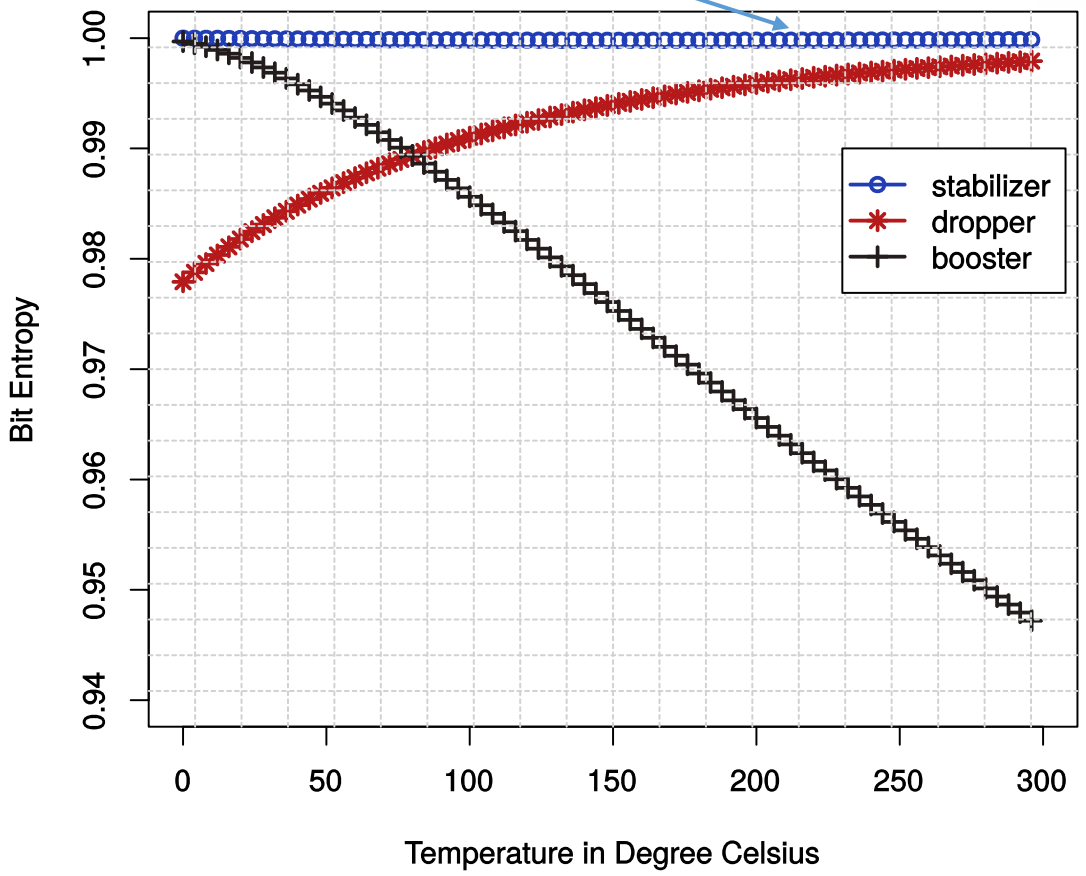


Resistance Against Temperature Variation

Probability of correctness of the stabilizer circuit remains nearly constant (around 0.5) as the temperature varies

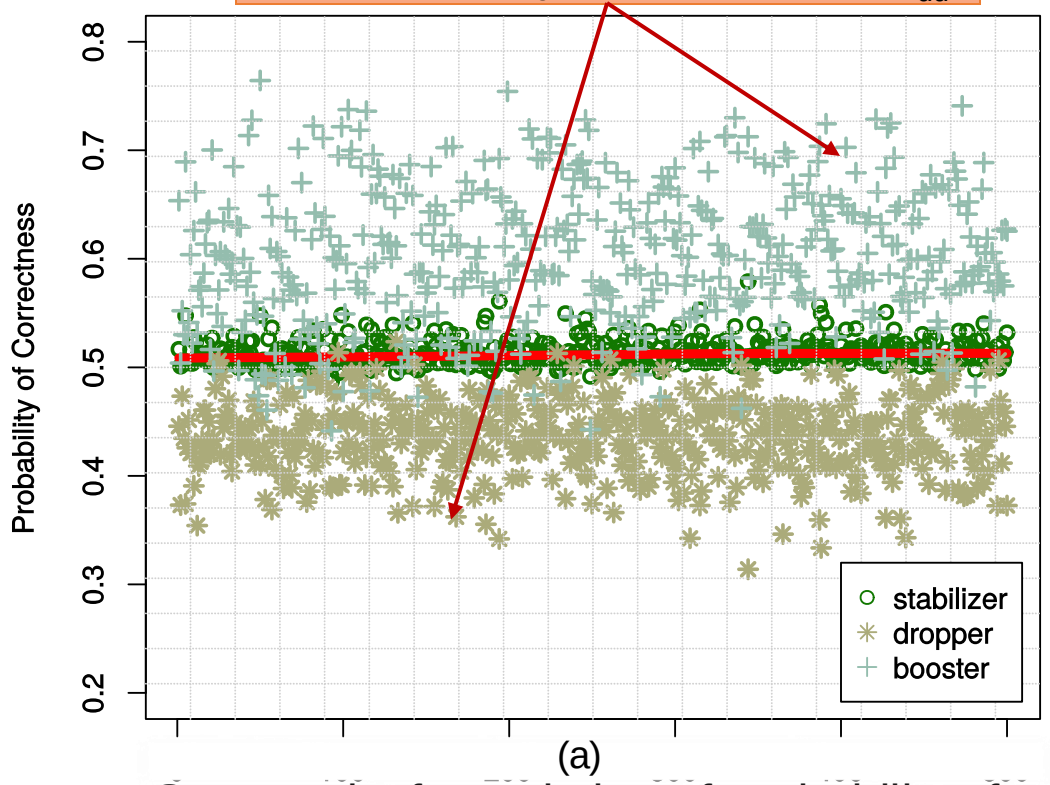


Entropy of the stabilizer circuit remains nearly constant (around 1.0) as the temperature varies



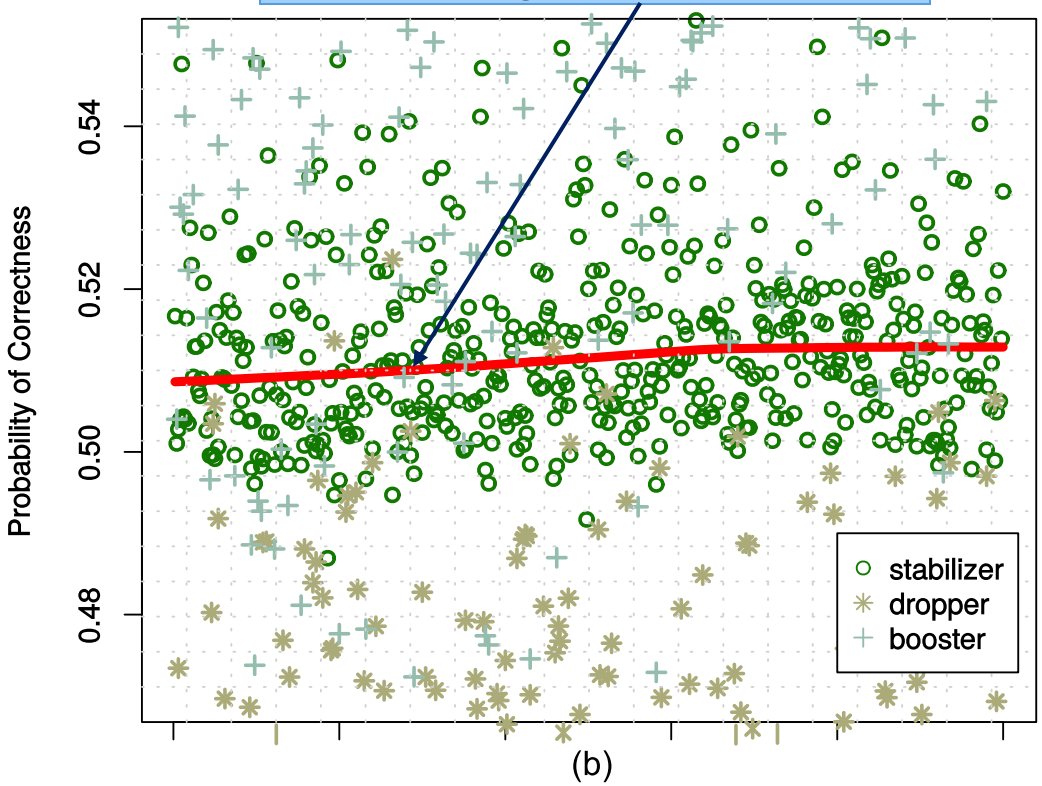
Resistance Against Supply Voltage Variation

The probability of correctness of the booster and dropper circuits are affected by the variation of V_{dd}



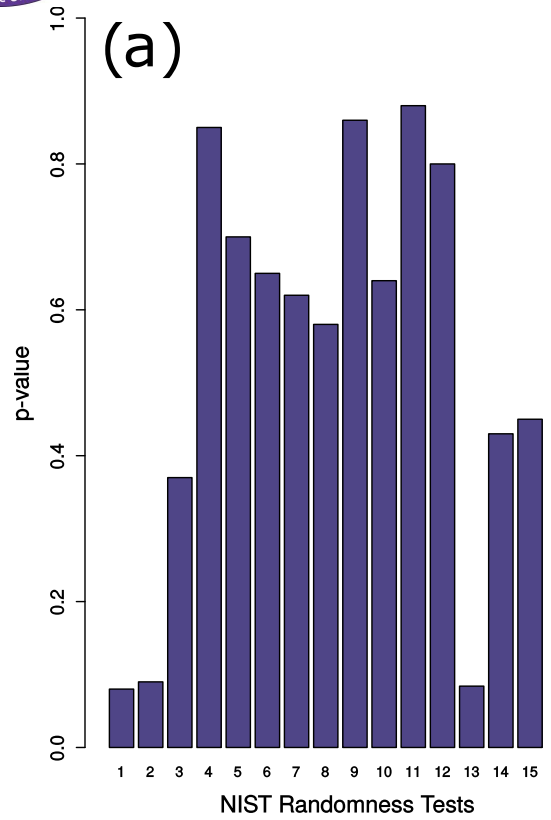
Scatter plot for variation of probability of correctness of booster, dropper, and stabilizer circuits for 28nm process due to V_{dd} variation

The probability of correctness of the stabilizer circuit remains in the neighborhood of 0.5

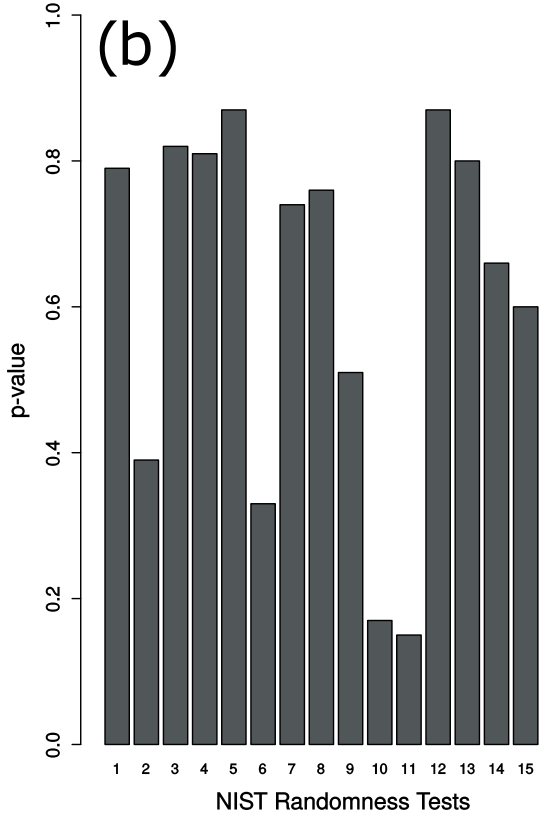


The probability of correctness variation plot in (a) zoomed around 0.5 value

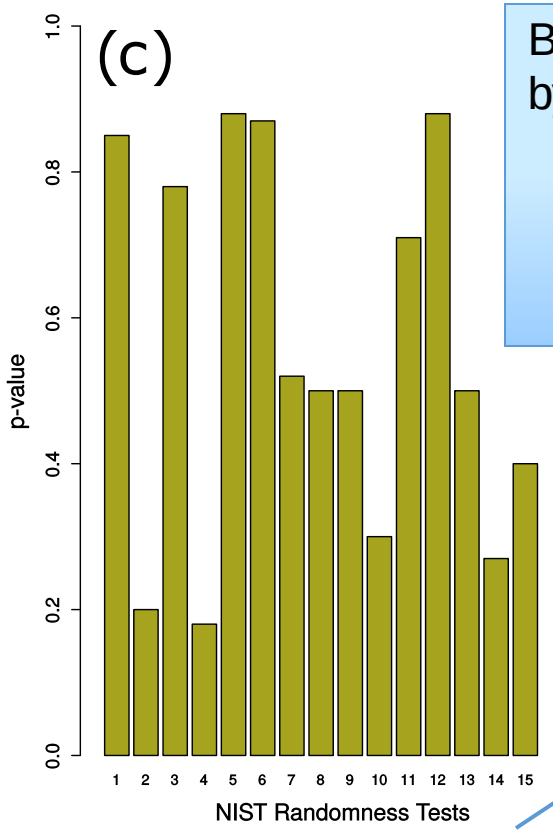
NIST Randomness Test



(a) 298K, nominal process



(b) 373K, +20% Δp



(c) 373K, -20% Δp

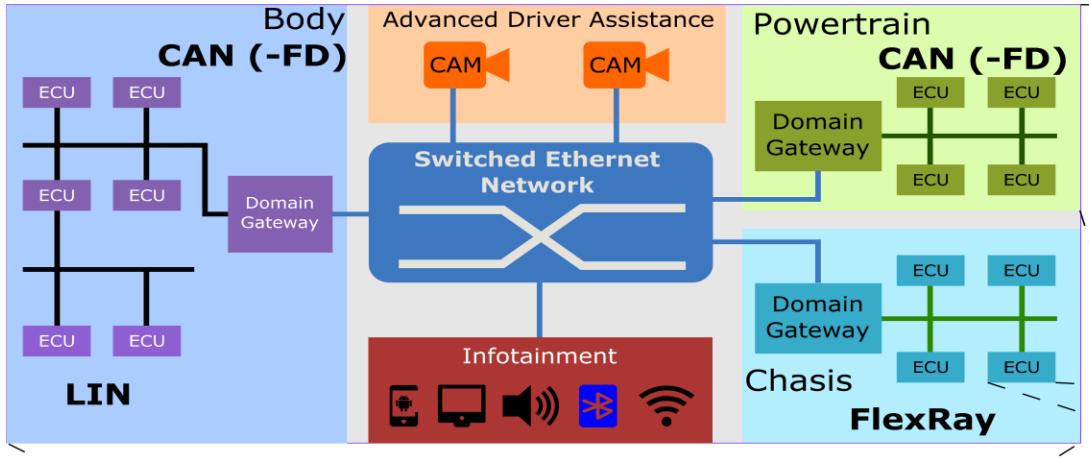
Bitstreams generated by our TRNG pass all the NIST tests for different process variations and temperatures

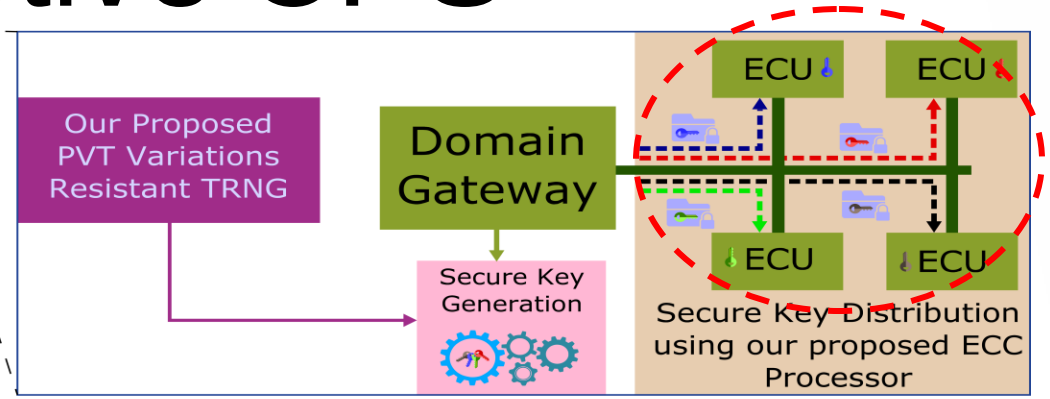
p-value ≥ 0.01 => 99% confidence levels

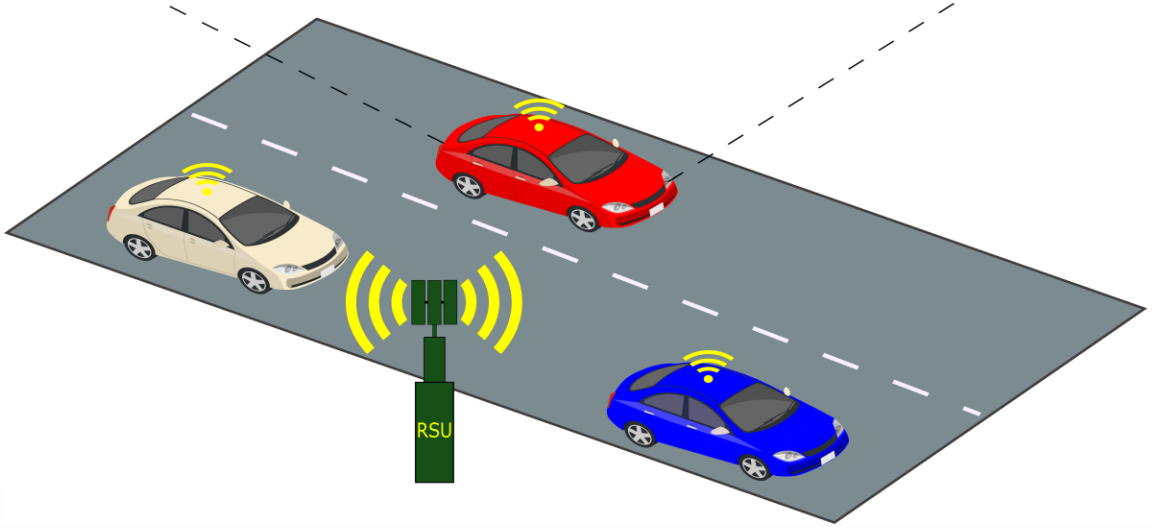
Process variation

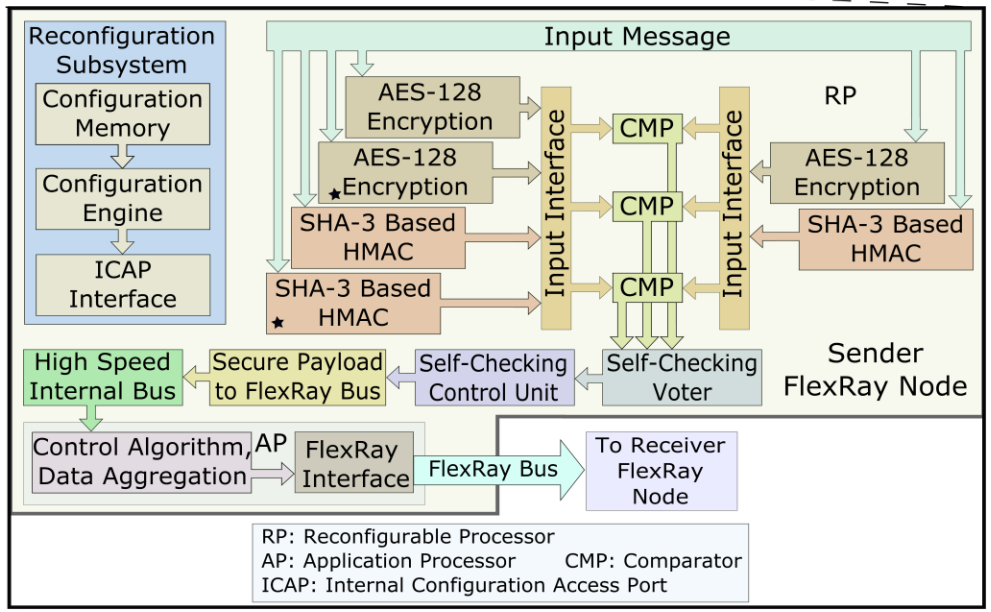
The numbers 1 to 15 respectively represents frequency, block frequency, runs, longest runs of 1s, binary matrix rank, DFT, non-overlapping template, overlapping template, Maurer's universal, linear complexity, serial, approximate entropy, cumulative sums, random excursions, and random excursion variant tests

Design of Secure and Dependable Automotive CPS









ECU Authentication and Secret Key Establishment

Certificate-based

- We have proposed an approach for ECU authentication and secret key establishment that leverages certificates and elliptic curve cryptography (ECC)
- Results verify that the proposed mechanisms do not violate the real-time constraints of automotive CPS even in the presence of errors in computation and transmission

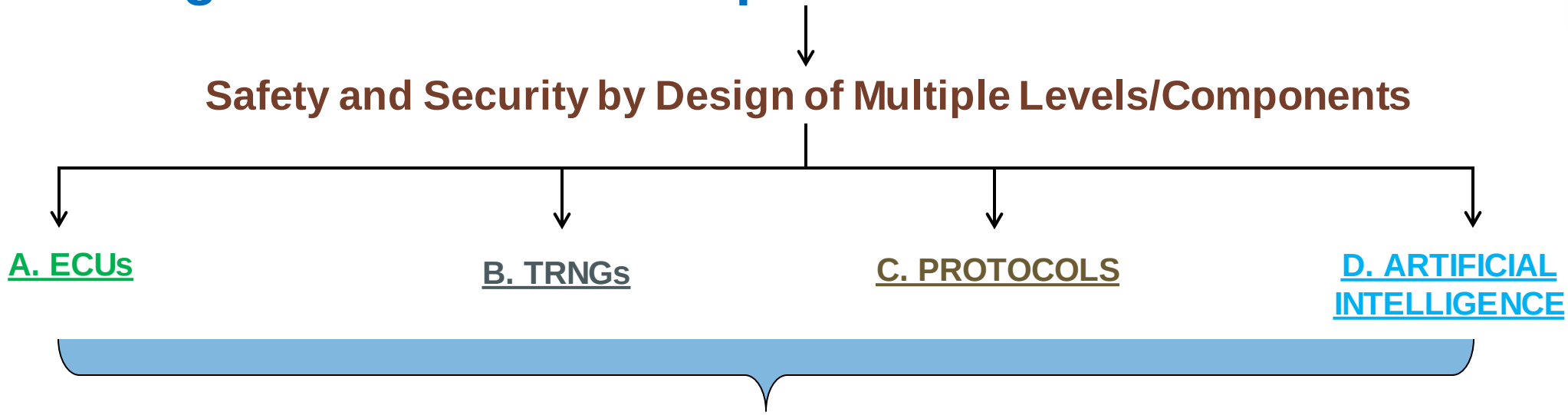
PUF-based

- We have proposed a PUF-based lightweight, highly reliable authentication scheme employing binary string shuffling
- The control logic keeps the PUF secure from brute force and modelling attacks
- The proposed approach provides a reduction of 63% and 74% for look-up tables (LUTs) and register count, respectively, in FPGA compared to a recently proposed approaches

PUF: Physically Unclonable Function

Conclusions

Design of Secure and Dependable Automotive CPS



Challenge: simultaneous integration of security and dependability without violating hard real-time constraints imposed by desired QoS

AI and ML present a new attack surface at both test-time and training-time

Pressing need for automotive CPS security and dependability research at all levels!

Questions