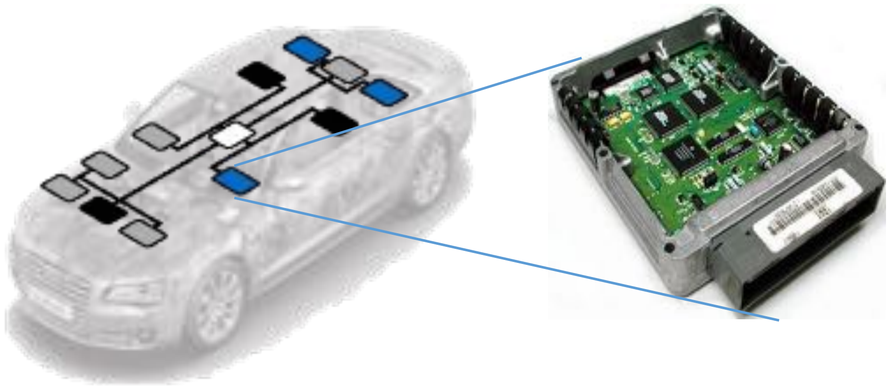
Next Generation Automotive Networks for ADAS and Autonomy

Suhaib A Fahmy
School of Engineering, University of Warwick

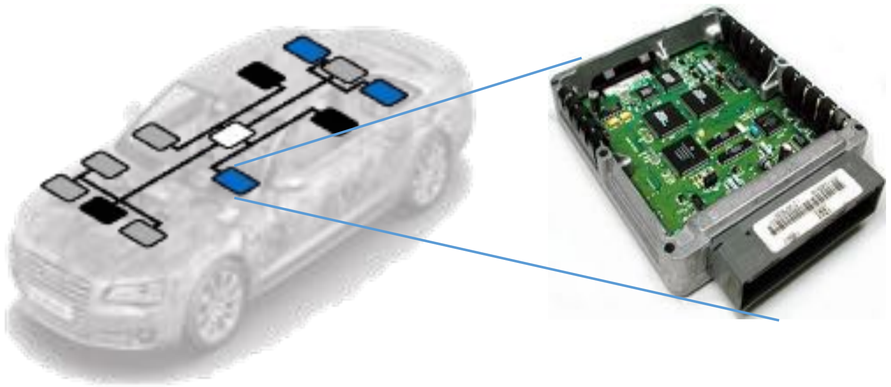
Automotive networks

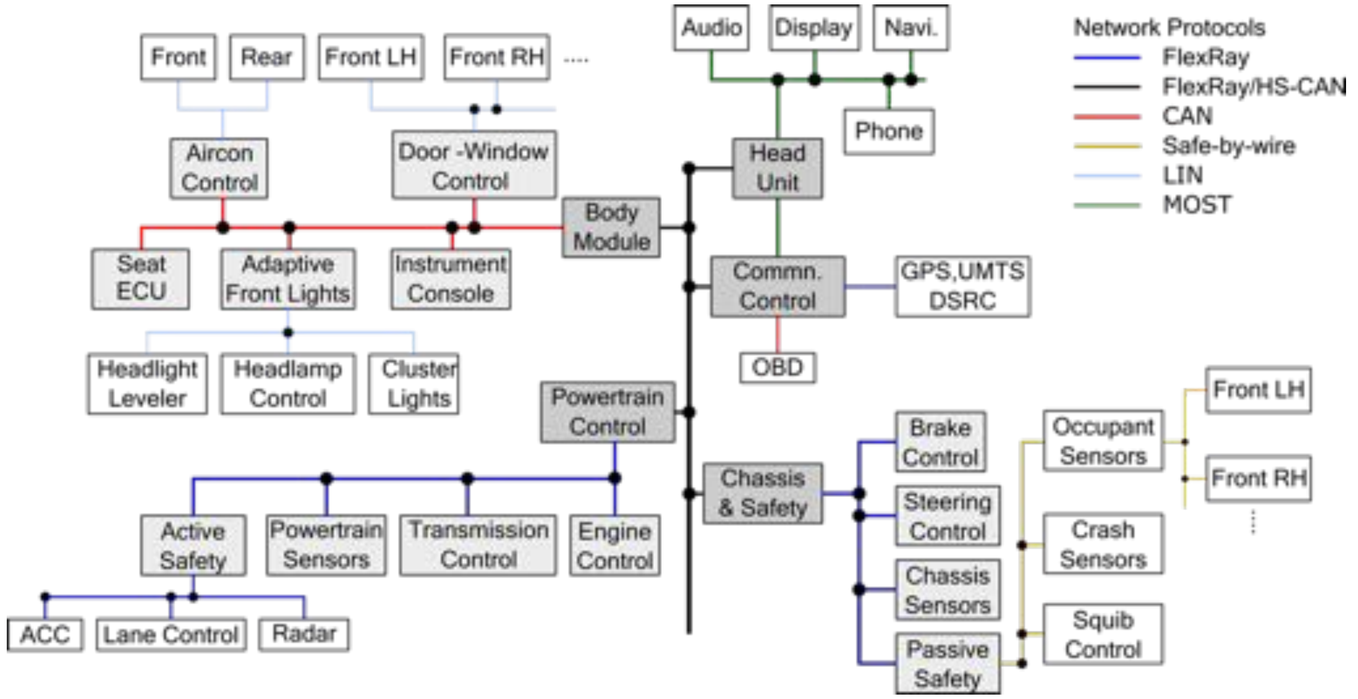
- Complex distributed control systems
- Built out of a succession of subsystems interconnected



Automotive networks

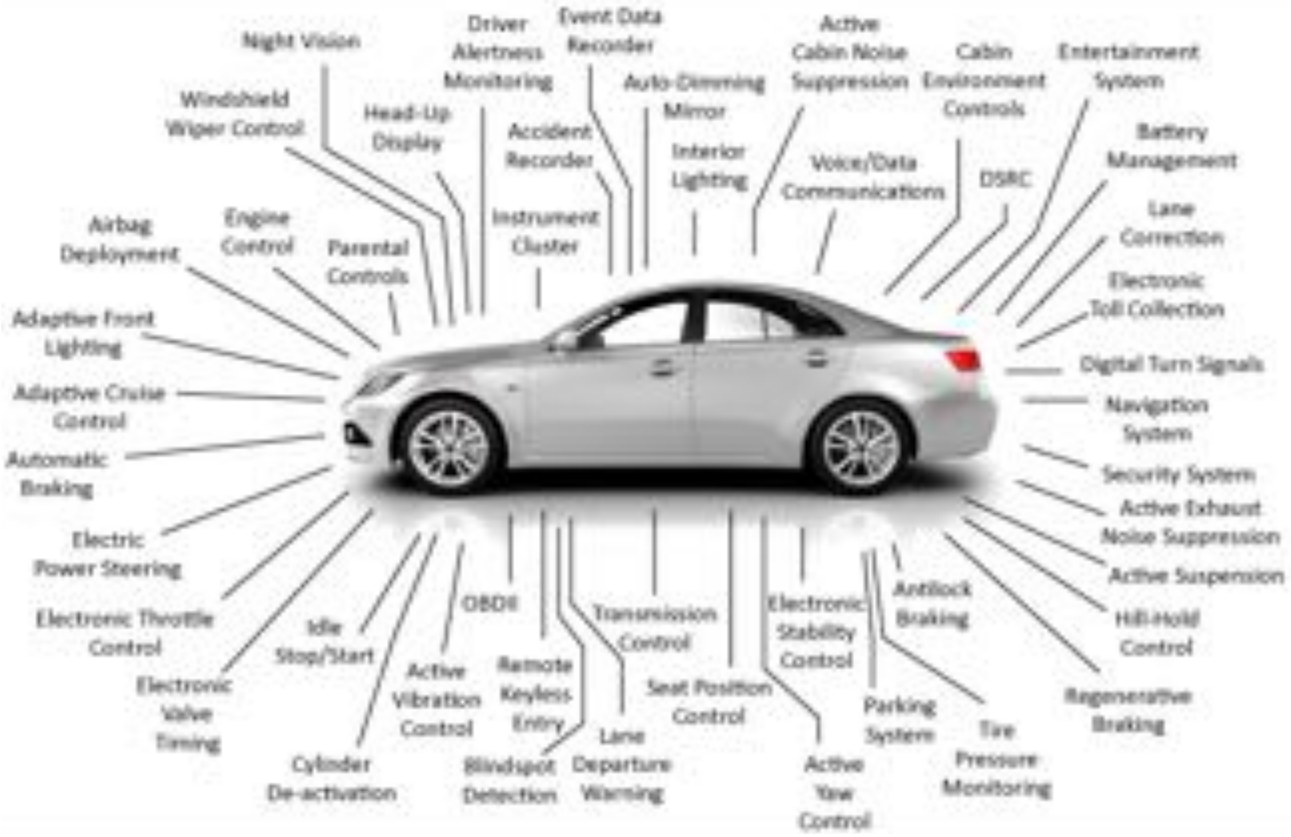
- Complex distributed control systems
- Built out of a succession of subsystems interconnected





Automotive networks

- Over 100 ECUs in higher end vehicles
- Over 100 million lines of code – *more than an aircraft!*
- Over 100kg of electronics
- Over 6km of cabling!



Automotive networks

- Traditionally split into:
  - **Critical networks:** generally assumed to be low to medium bandwidth but with strict requirements for key subsystems
  - **Non-critical networks:** lightweight comfort features and audiovisual without strict requirements
- More specifically, the *domains* specify a required combination of bandwidth, reliability, and other QoS properties
  - Body, chassis, powertrain, telematics, occupant safety

Automotive network constraints

- Key drivers of network choice in the automotive domain:
  - Cost
  - Safety
  - Weight

- Ethernet was problematic for a long time, due to cabling cost; BroadCom developed a new physical layer allowing unshielded twisted pair wiring, standardised by OPEN Alliance

ADAS and autonomy requirements

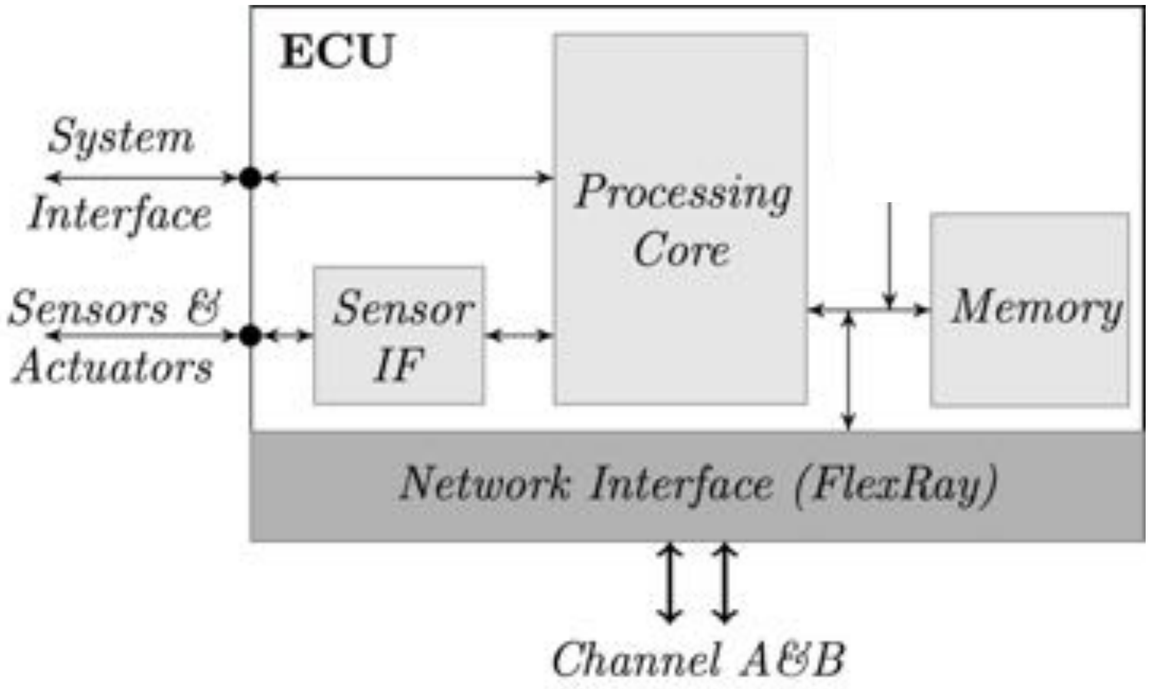
- Emergence of advanced driver assistance systems and autonomous driving bring with them the following

  1. A larger number of data-intensive sensors,
  2. distributed more widely around the vehicle,
  3. requiring significant computation to determine outcomes,
  4. which must be communicated within strict deadlines

- This impacts the networking and computation requirements in vehicles

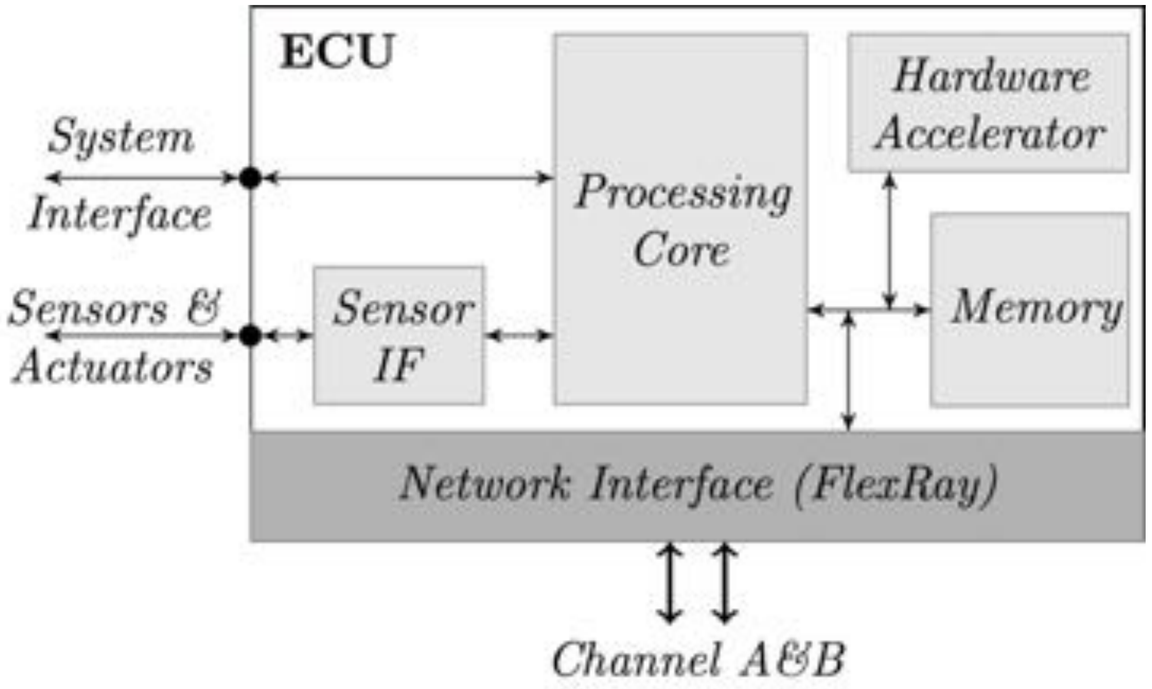
Network architecture

- ECU architecture includes
  - Single/multi-core processor
  - Memory
  - Sensor interfaces
  - Network interface
  - Some hardware accelerators



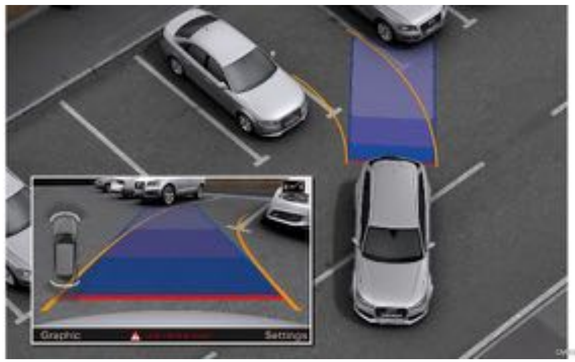
Network architecture

- ECU architecture includes
  - Single/multi-core processor
  - Memory
  - Sensor interfaces
  - Network interface
  - Some hardware accelerators



Network architecture

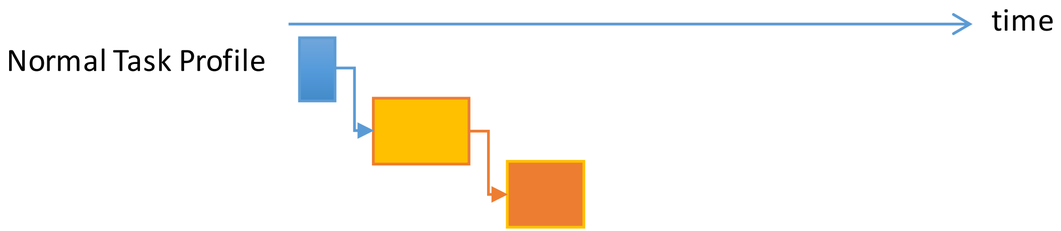
- Currently, adding new functionality to a vehicle means adding more ECUs to perform the required computation → number of ECUs 1.45×/year
- There is an interest in **ECU consolidation**
  - Non-concurrent functions can share the same hardware
  - But isolation problematic even with multi-core ECUs due to shared resources, e.g. memory, network interface
- ADAS and autonomous driving are ideal for this due to shared sensors and multiple modes



Courtesy Audi

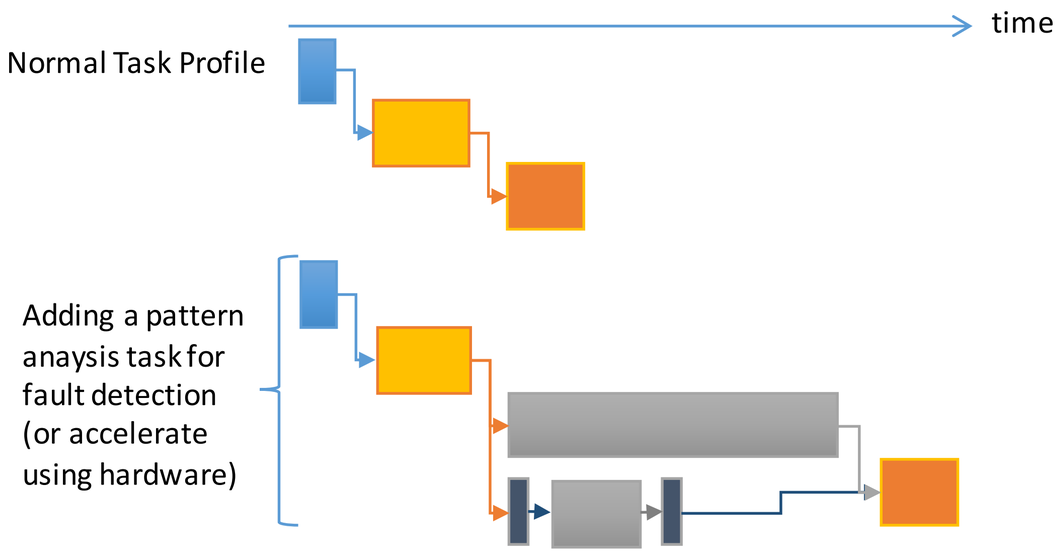
Compute limitations

- Many applications require additional compute tasks to be added
  - Add encryption to critical messages
  - Add preprocessing to sensor data
- These adversely affect timing, requiring recertification
- Even hardware coprocessors have have this problem



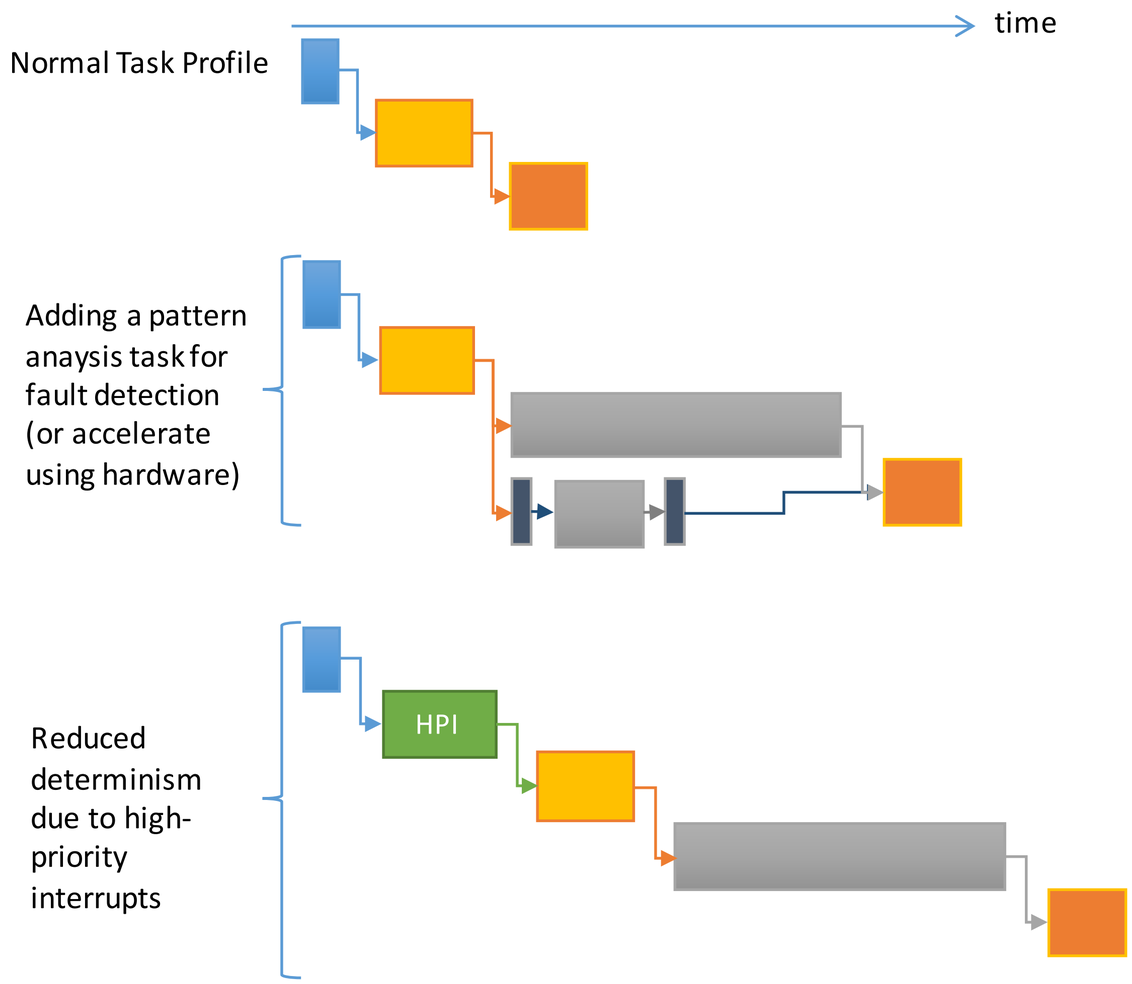
Compute limitations

- Many applications require additional compute tasks to be added
  - Add encryption to critical messages
  - Add preprocessing to sensor data
- These adversely affect timing, requiring recertification
- Even hardware coprocessors have have this problem



Compute limitations

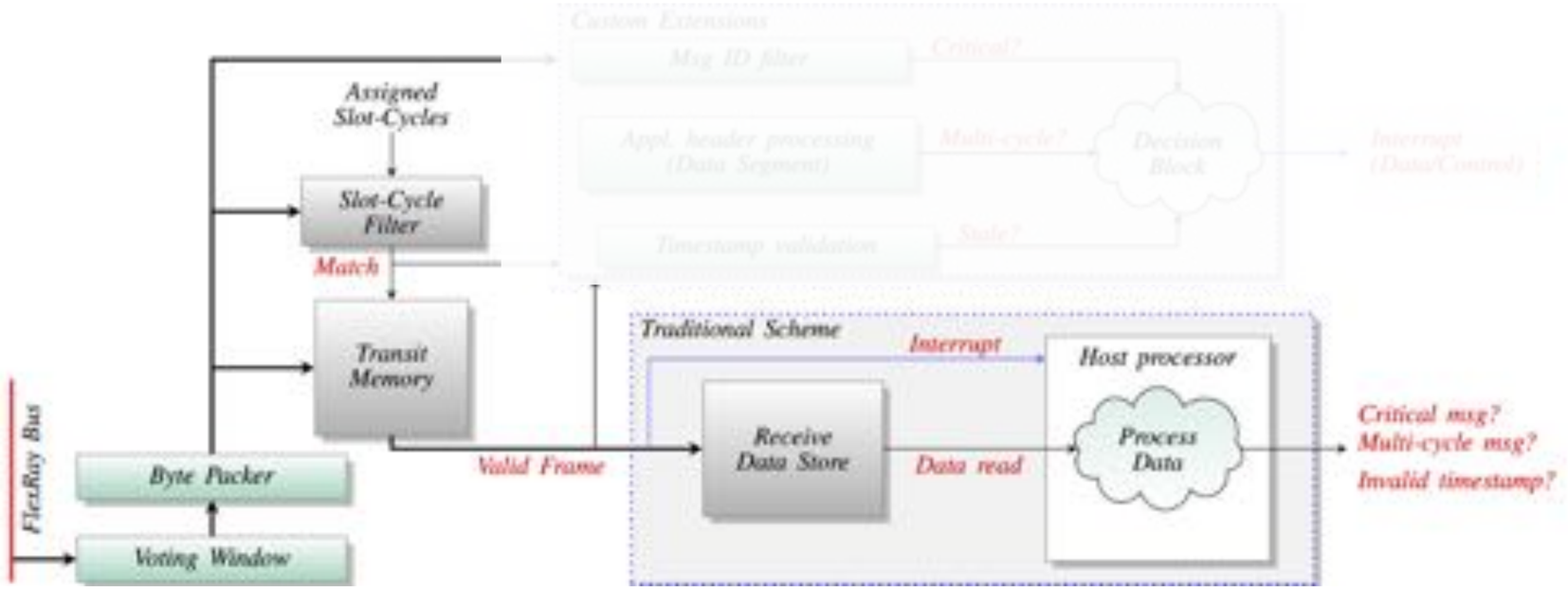
- Many applications require additional compute tasks to be added
  - Add encryption to critical messages
  - Add preprocessing to sensor data
- These adversely affect timing, requiring recertification
- Even hardware coprocessors have have this problem



Smart network interfaces

- Normally network interface simply moves messages between network and ECU processor

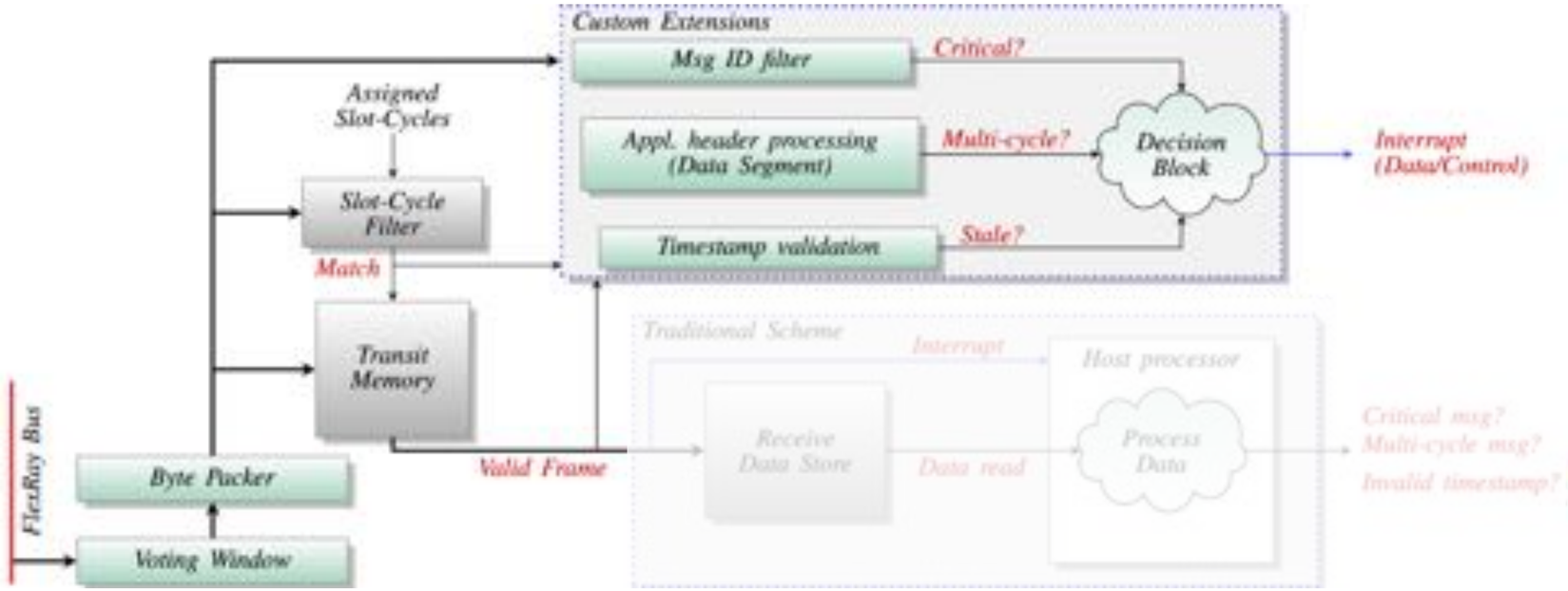
- Our proposal: add an extensible, programmable datapath inside the controller



Smart network interfaces

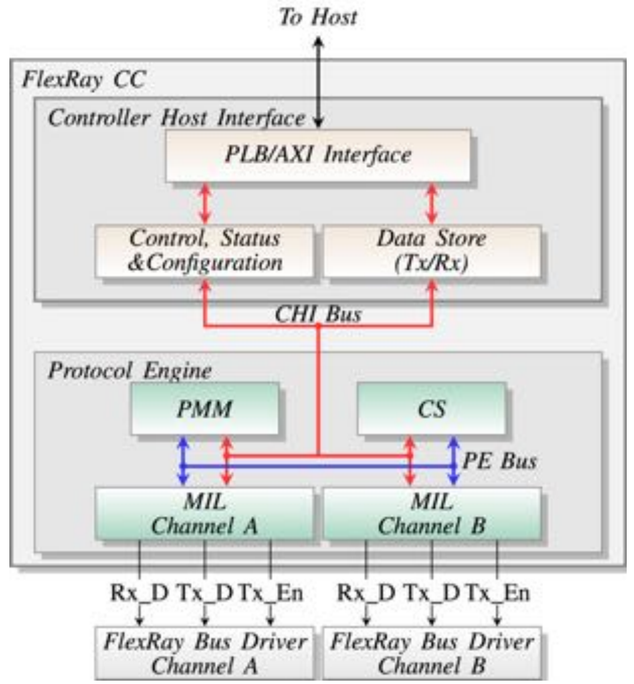
- Normally network interface simply moves messages between network and ECU processor

- Our proposal: add an extensible, programmable datapath inside the controller



Prototyping platform

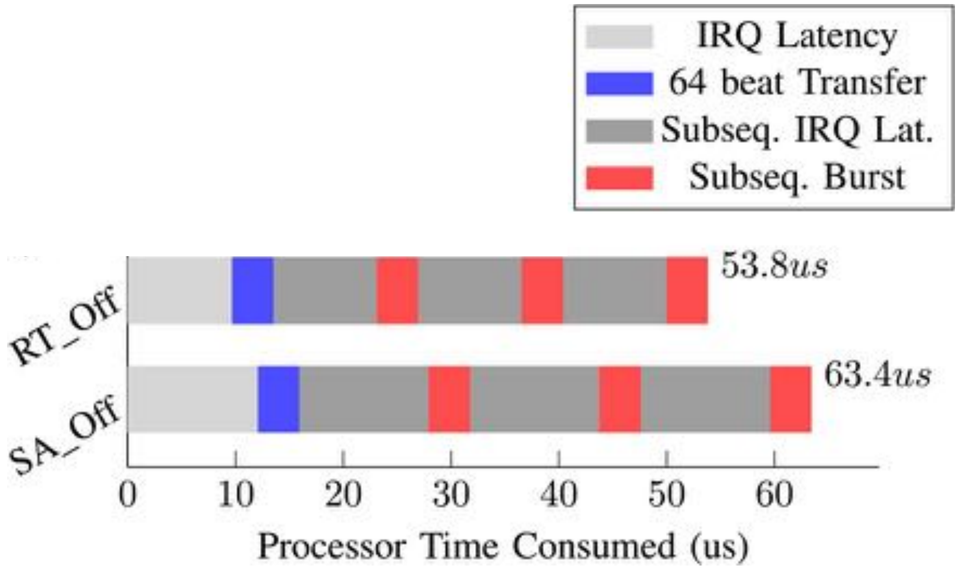
- We built a standards-compliant FlexRay controller to explore these ideas
- Comparable power consumption to a discrete ASIC controller (Bosch eRay)
- Additional features added to controller to quantify overheads



Moving large data streams

- How do we move large-volume sensor data more efficiently?
- Currently frame size determined by standard
- Each frame initiates an interrupt to the ECU processor
- Usable amount of data typically involves multiple frames, latency
- Adding a suitably sized buffer in the network controller means it only initiates an interrupt once sufficient data has been received

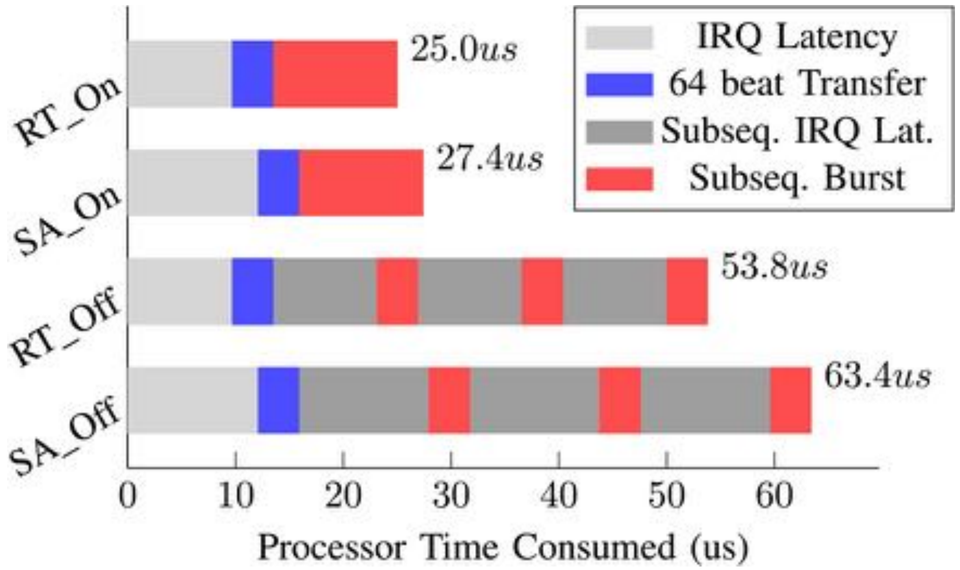
On Microblaze



Moving large data streams

- How do we move large-volume sensor data more efficiently?
- Currently frame size determined by standard
- Each frame initiates an interrupt to the ECU processor
- Usable amount of data typically involves multiple frames, latency
- Adding a suitably sized buffer in the network controller means it only initiates an interrupt once sufficient data has been received

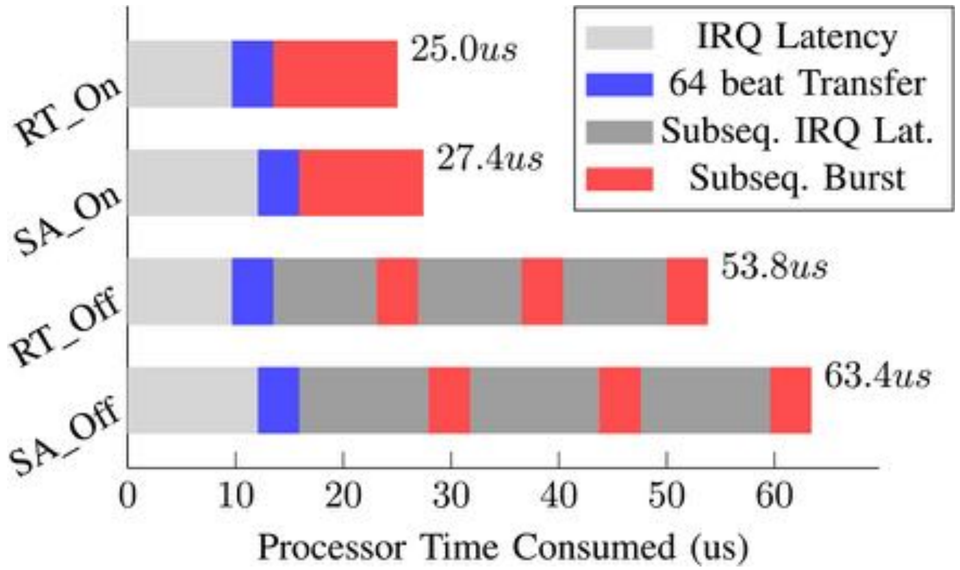
On Microblaze



Moving large data streams

- How do we move large-volume sensor data more efficiently?
- Currently frame size determined by standard
- Each frame initiates an interrupt to the ECU processor
- Usable amount of data typically involves multiple frames, latency
- Adding a suitably sized buffer in the network controller means it only initiates an interrupt once sufficient data has been received

On Microblaze



On Zynq ARM

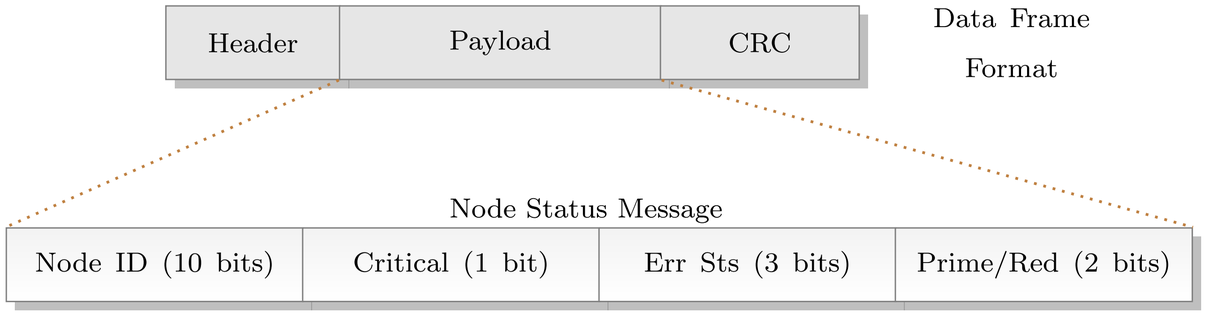
Mode	Latency components		Total time	Change
	Interrupt	Data Movement		
Software	2.96 µs × 8	0.3 µs × 8	26.08 µs	
Extension	2.96 µs × 1	0.3 µs × 8	5.36 µs	−79%

Adapting ECU functionality

- The advent of ADAS and autonomous driving means we need to adapt ECU functionality depending on operating context
- The same set of sensors can support multiple mutually-exclusive system functions
- The amount of signal processing required suggests a place for hardware acceleration, and perhaps FPGAs
- Ideally, we could consolidate these functions on a single platform, requiring a trigger to switch between modes

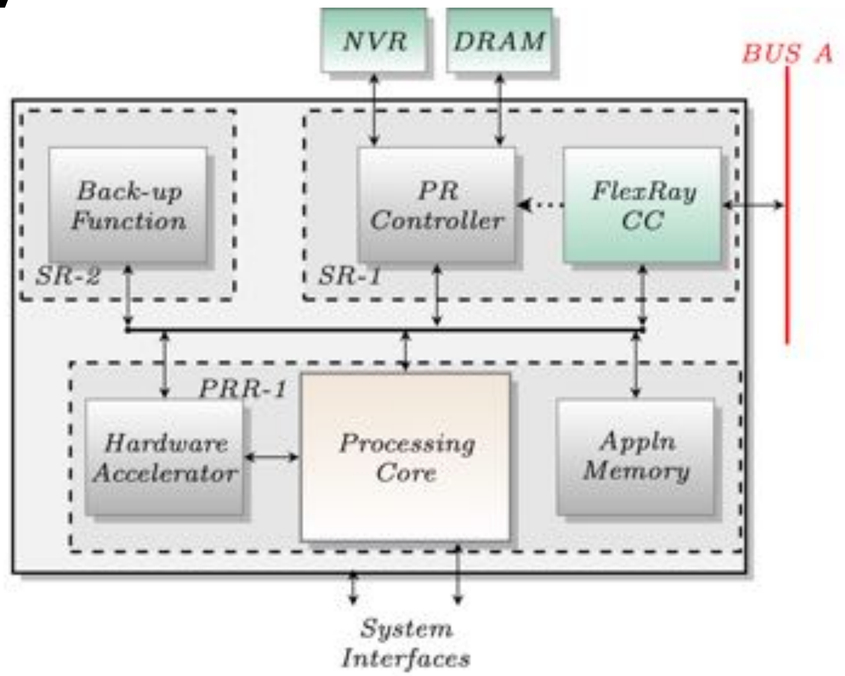
Adapting ECU functionality

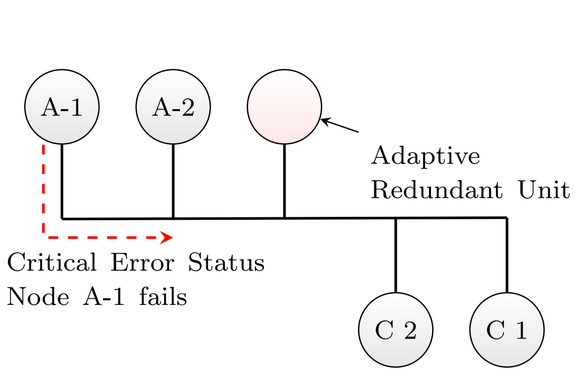
- To maintain standards-compliance, we can extend the data frame format within the payload, allowing the network controller to add/read data there
- Extensions in the network interface manage this data transparently to the application
- Special status bits allow interfaces to communicate with each other without affecting processor operation

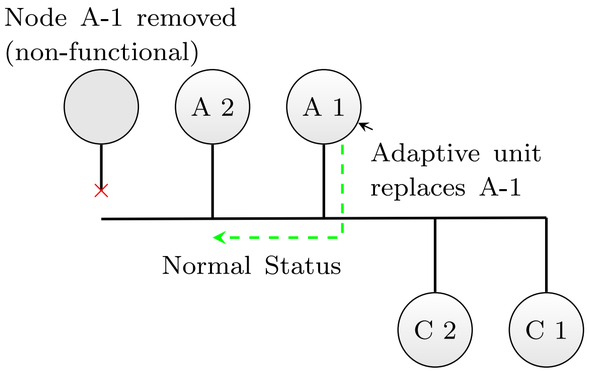


Adapting ECU functionality

- We propose FPGA-based ECUs to provide a redundant backup that can be adapted to errors/mode changes in real time
- The redundant unit can be partially reconfigured with the appropriate backup function, in direct response to a critical error message on the network
- A high performance accelerator can be replaced as needed for different modes







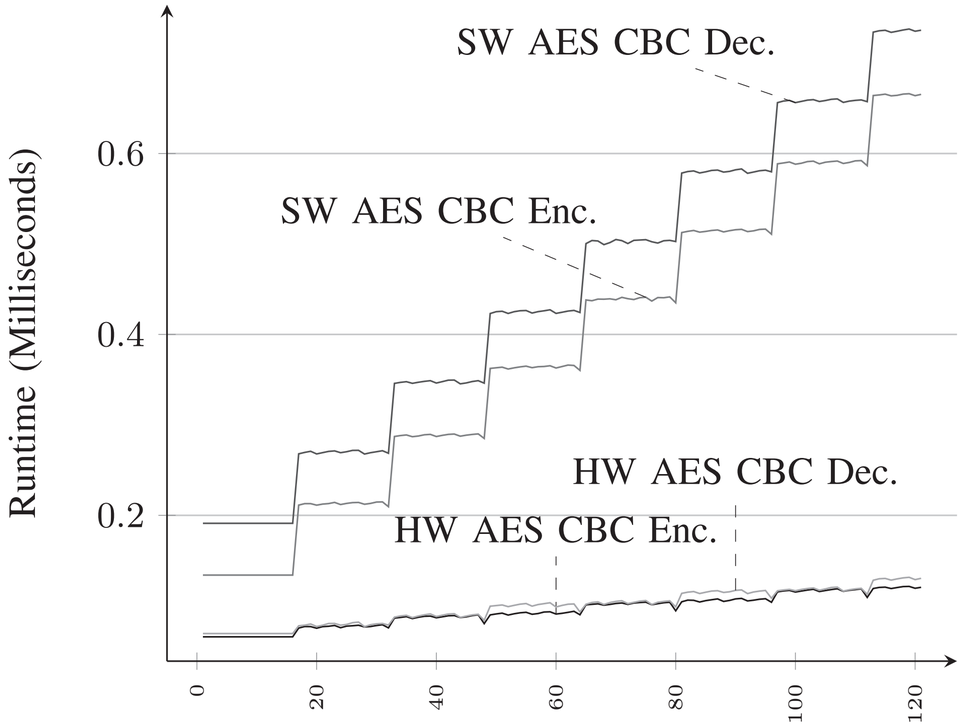
Adapting ECU functionality

- Enabling FPGA reconfiguration from within the network controller improves the response time significantly
- Adding the decision making time that would otherwise be done within an ECU processor increases this even further

Mode	Latency components			Total time	Change
	Interrupt	Data Movement	Reconfig.		
Software (PCAP)	2.96 µs	0.3 µs	2257.9 µs	2261.1 µs	
H/W intelligence with custom ICAP	NA	NA	759.4 µs	759.4 µs	−66%

Securing automotive networks

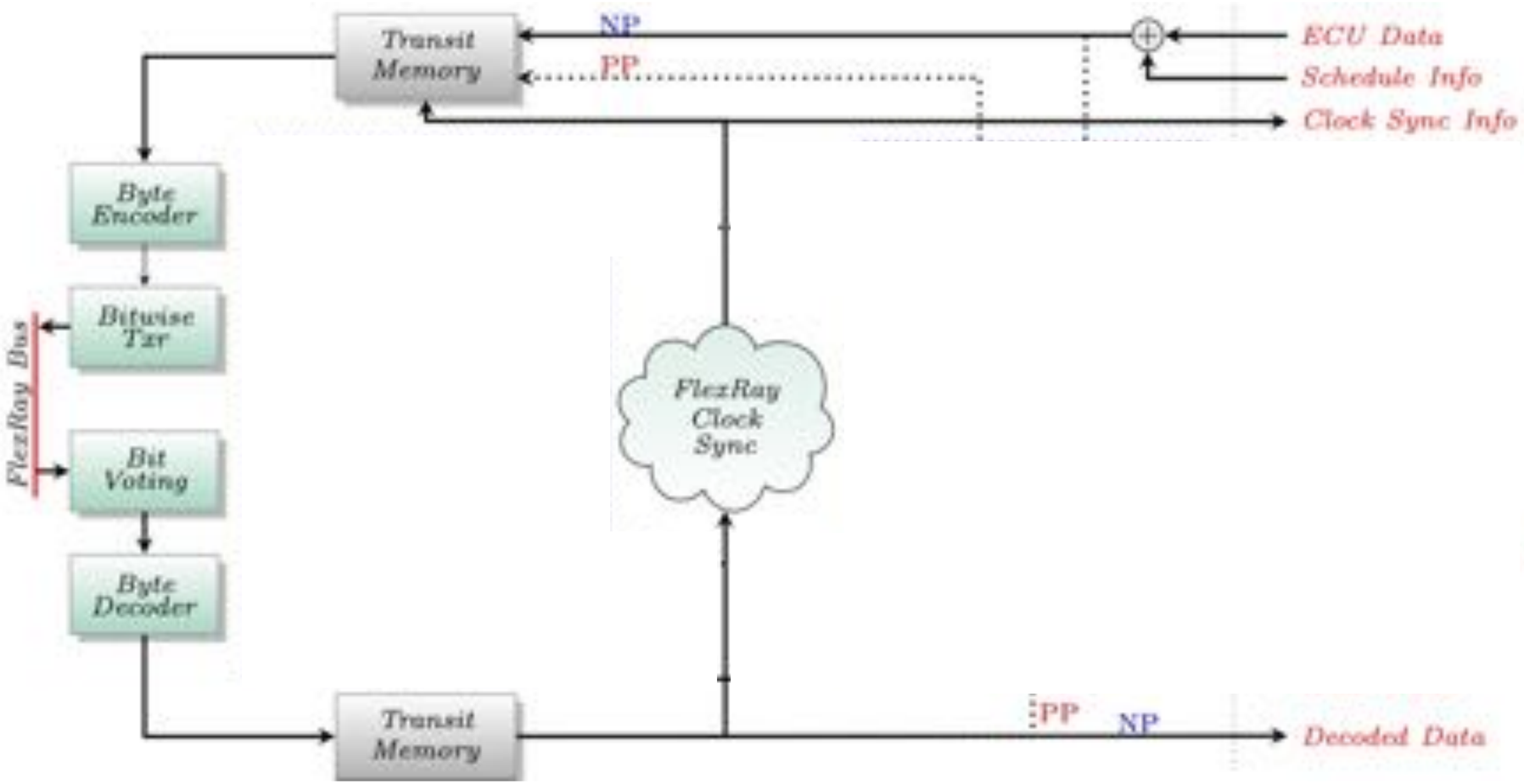
- Wider connectivity will underly the ADAS and autonomy trends – sensitive data
- This challenges the previous approach of only protecting networks external interfaces
- One compromised ECU can bring down the whole network
- Adding security entails significant software overhead; hardware accelerator adds less overhead but inflexible in terms of protocol



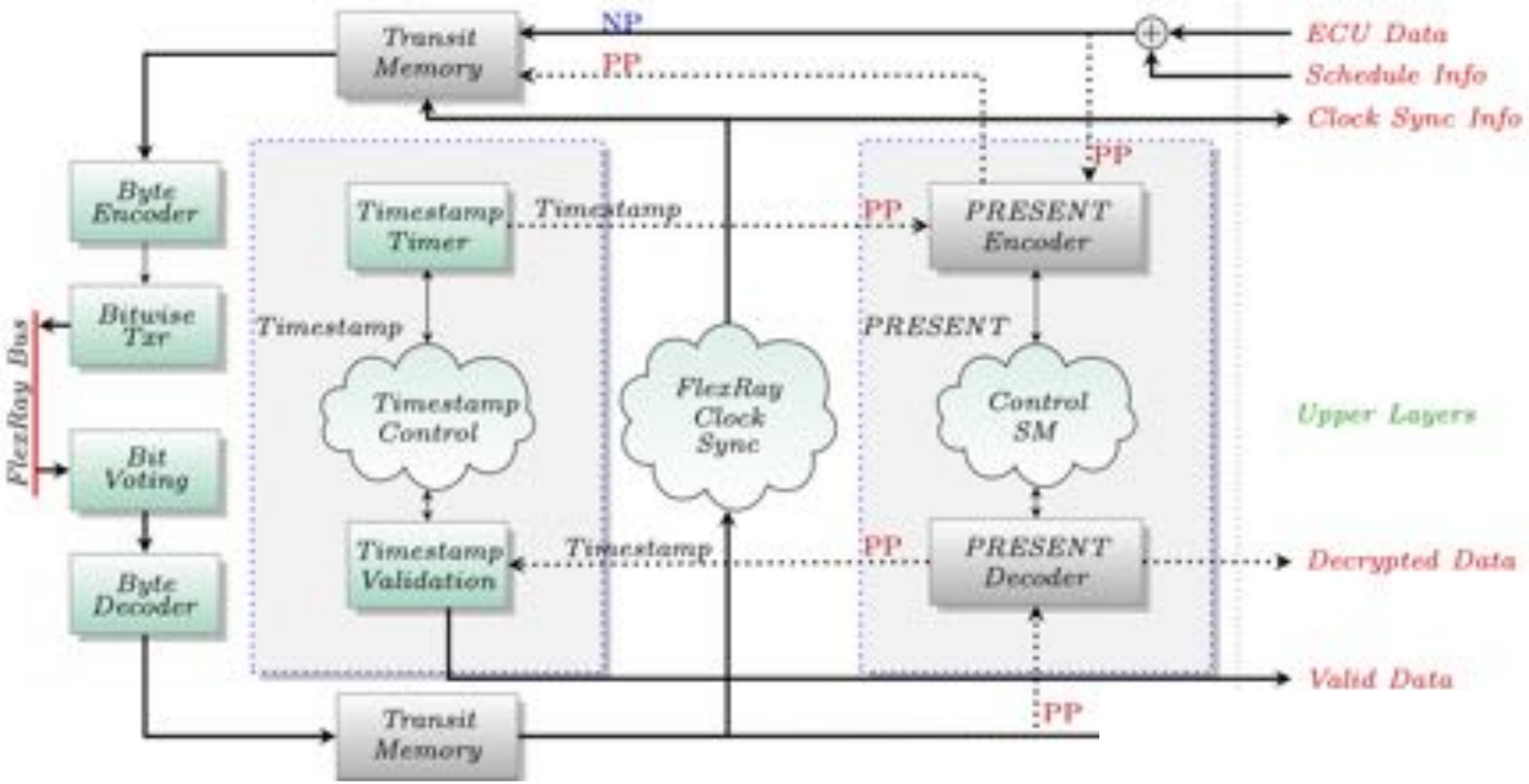
Securing automotive networks

- We similarly explore the ability to move security functions into the network controller as opposed to additional software
- Motivations:
  - Time triggered controllers have a synchronised time-base, software had jitter
  - Network headers can be obfuscated to prevent attack nodes from joining the network – access control
  - Data can be encrypted transparently to the ECU processors

Securing automotive networks

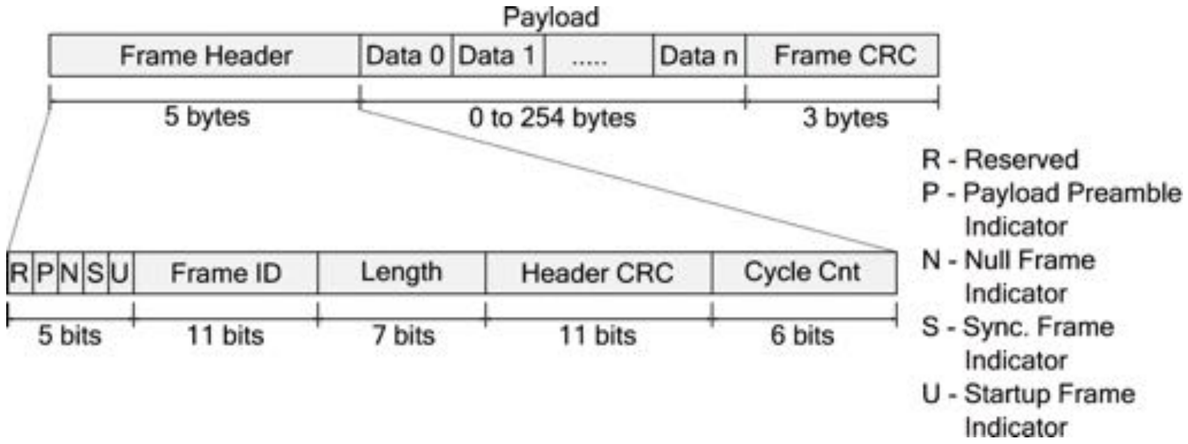


Securing automotive networks



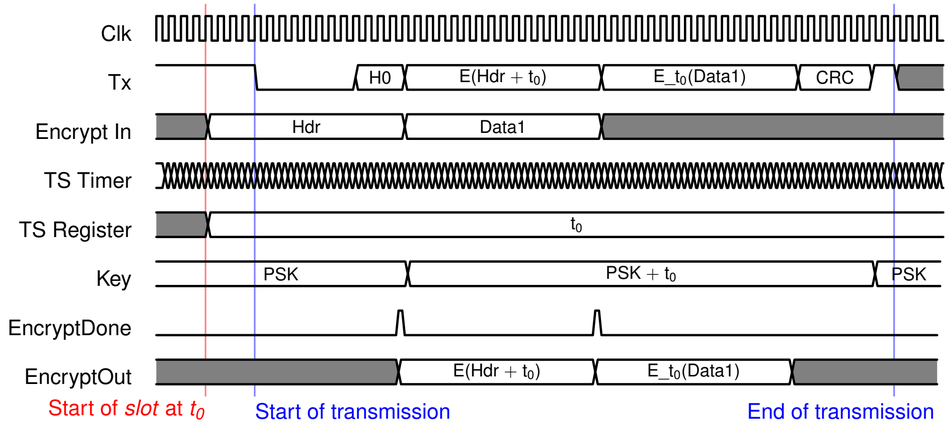
Securing automotive networks

- We add obfuscation of the network headers with a pre-shared key
- Only devices with this key can integrate onto the network
- Header manipulation cannot be done in software since this is managed in the interface



Securing automotive networks

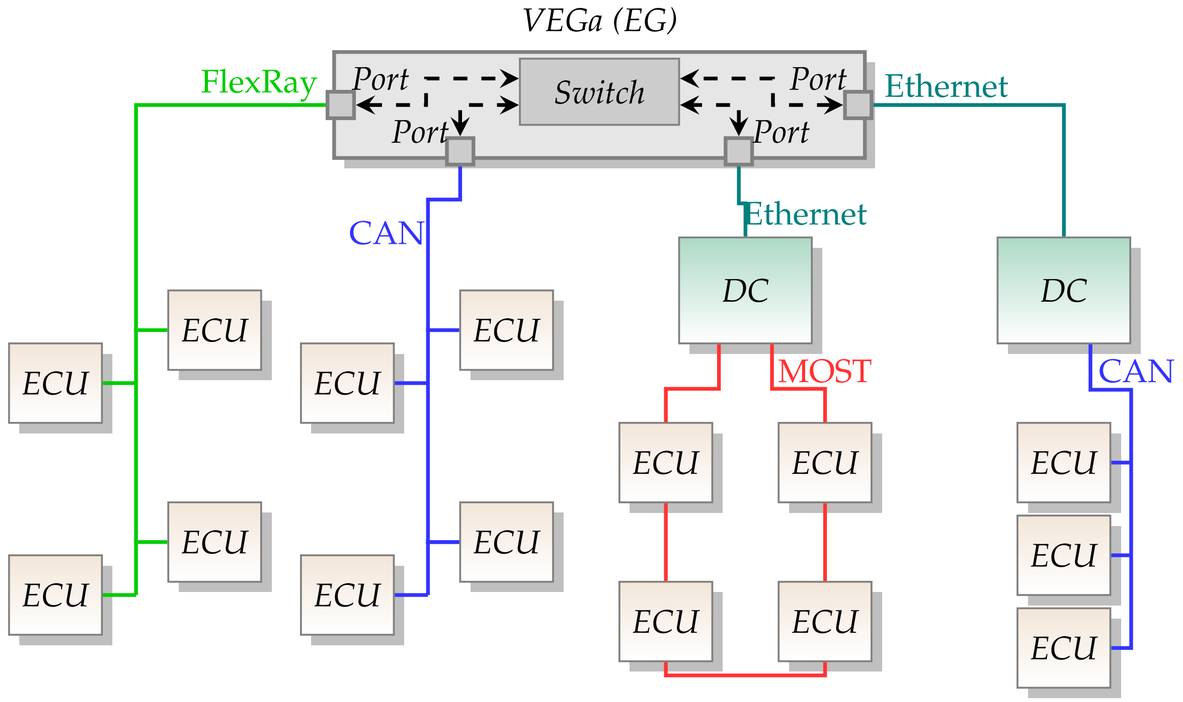
- Adding a lightweight cipher primitive in the controller enables seamless encryption and decryption of messages
- Adding timestamp information increases entropy significantly
- This can be done within the buffer time for a single frame, effectively *zero latency*



Task	Rounds	Latency components			Total delay
Encyption		TS Read	Encrypt	Writeback	
Software	32	0.3 µs	40.9 µs	0.3 µs	41.5 µs
	64	0.3 µs	82.6 µs	0.3 µs	83.2 µs
Extension	up to 470	NA	0 µs *Overlaps with txn*	0.3 µs	0.3 µs
Decyption		Data Read	TS read	Decrypt	
Software	32	0.6 µs	0.3 µs	42.1 µs	43.0 µs
	64	0.6 µs	0.3 µs	85.2 µs	86.1 µs
Extension	up to 470	0.3µs	NA	0 µs *Overlaps with rxn*	0.3 µs

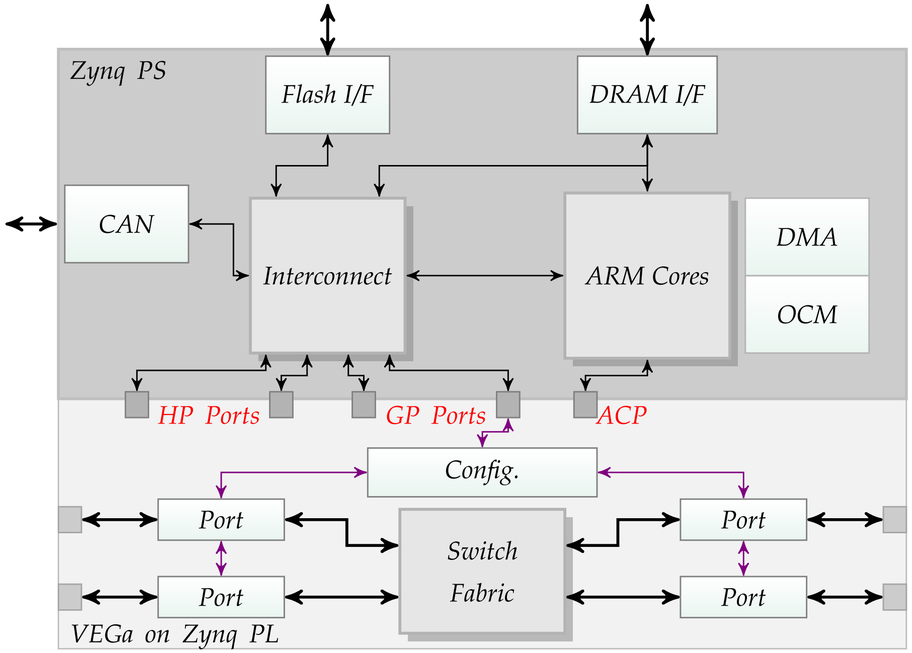
Integrating networks

- These more capable networks need to interact with legacy systems
- We expect some subsystems to retain these legacy architectures while the more advanced functions leverage more expensive modern architectures



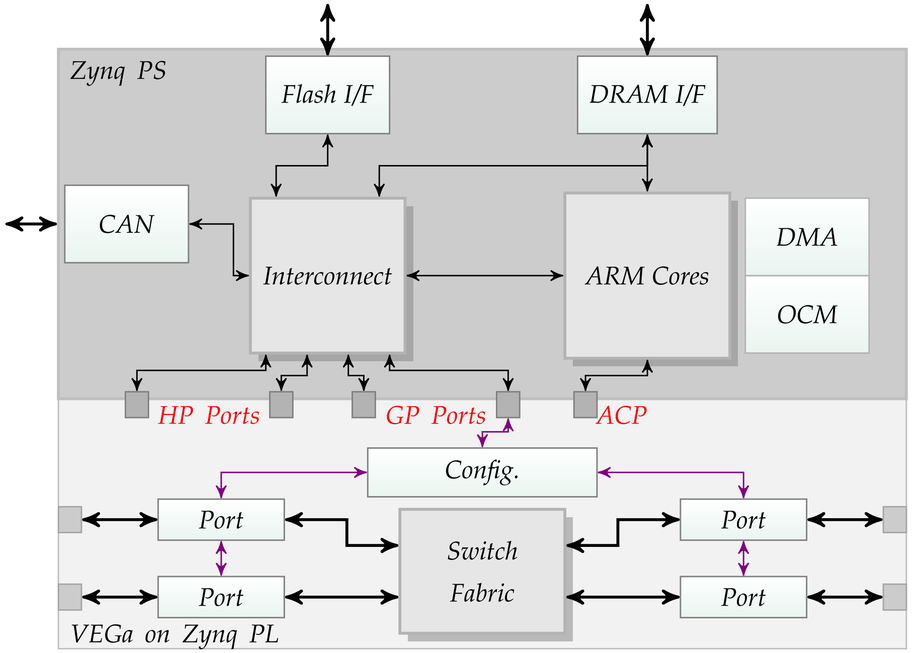
Integrating networks

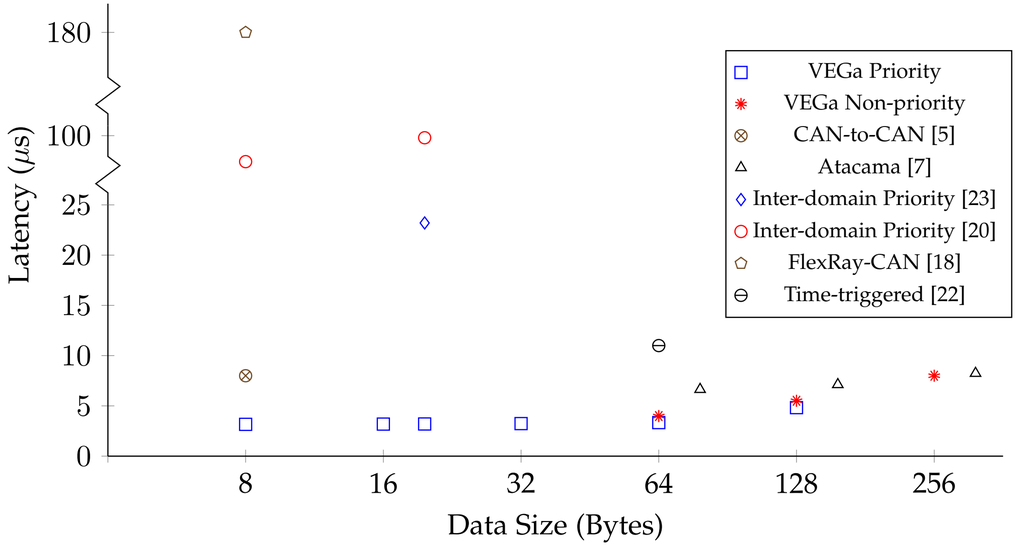
- Built the VEGa Ethernet gateway with reconfigurable switch architecture
- Enables low-latency switching of legacy networks with Ethernet backbone, including priority messages
- Flexible switch enables application of same hardware in different products



Integrating networks

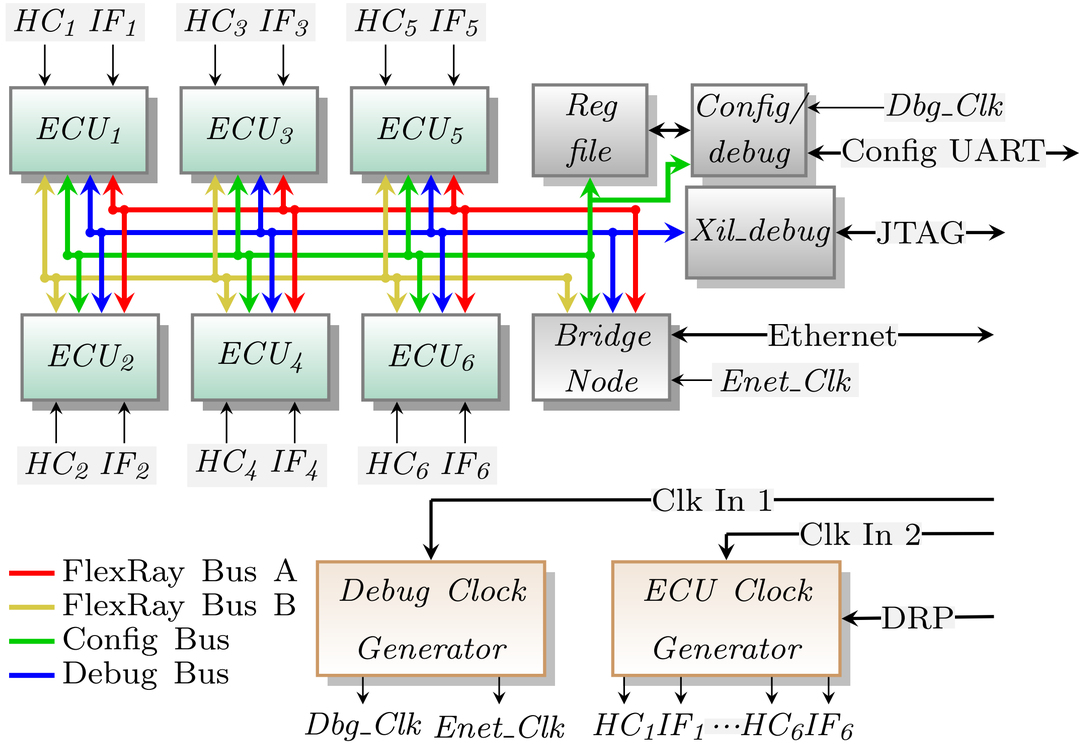
- Built the VEGa Ethernet gateway with reconfigurable switch architecture
- Enables low-latency switching of legacy networks with Ethernet backbone, including priority messages
- Flexible switch enables application of same hardware in different products





Validation Platform

- Another benefit of having access to low level controller design is the ability to prototype a full cluster on a single (large) FPGA
- Apply a variety of network tests
  - Bit errors
  - Frame drops
  - Babbling idiot
  - Frequency drift
- Accelerated super-real time validation



Smart network interfaces

- Such capability in the network interfaces key to re-architecting automotive networks for upcoming applications
- We are interested in porting this approach to new Ethernet TSN
- FPGAs offer a significant opportunity in terms of processing ability and flexibility
- Require significant effort to address concerns of the automotive community: functional safety, etc.

Acknowledgements

- Work with TUM CREATE, Singapore
  - Shreejith Shanker, PhD, now at Warwick
  - Philipp Mundhenk, PhD, now at Audi
  - Martin Lukasiewycz, now at Google
  - Sebastian Steinhorst, now at TUM
  - Samarjit Charkraborty, TUM

Publications

- “Reconfigurable Computing in Next-Generation Automotive Networks”, in **IEEE Embedded Systems Letters** 2013.
- “Extensible FlexRay Communication Controller for FPGA-Based Automotive Systems”, in **IEEE Transactions on Vehicular Technology** 2015.
- “An Approach for Redundancy in FlexRay Networks Using FPGA Partial Reconfiguration”, in **Design, Automation and Test in Europe Conference (DATE)** 2013.
- “Security Aware Network Controllers for Next Generation Automotive Embedded Systems” in **Design Automation Conference (DAC)** 2015.
- “Accelerating Validation of Time-Triggered Automotive Systems on FPGAs”, in **Field Programmable Technology (FPT)** 2013.
- “VEGa: A High Performance Vehicular Ethernet Gateway on Hybrid FPGA”, in **IEEE Transactions on Computers 2017.**
- “Smart Network Interfaces for Advanced Automotive Applications”, in **IEEE Micro**, to appear 2018.