

Automotive Embedded Systems Research: Recent Developments and Future Infrastructure Needs

Dr. Mohammed A. S. Khalid
April 16, 2018

*University of Windsor
Department of Electrical and Computer Engineering*



Seminar Outline

- Automotive Embedded Systems (AES) Overview
- Open Research Problems in AES (my view)
- Recent Research: HLS for a computational task used in ADAS
- Infrastructure for AES Research
- Conclusions
- References



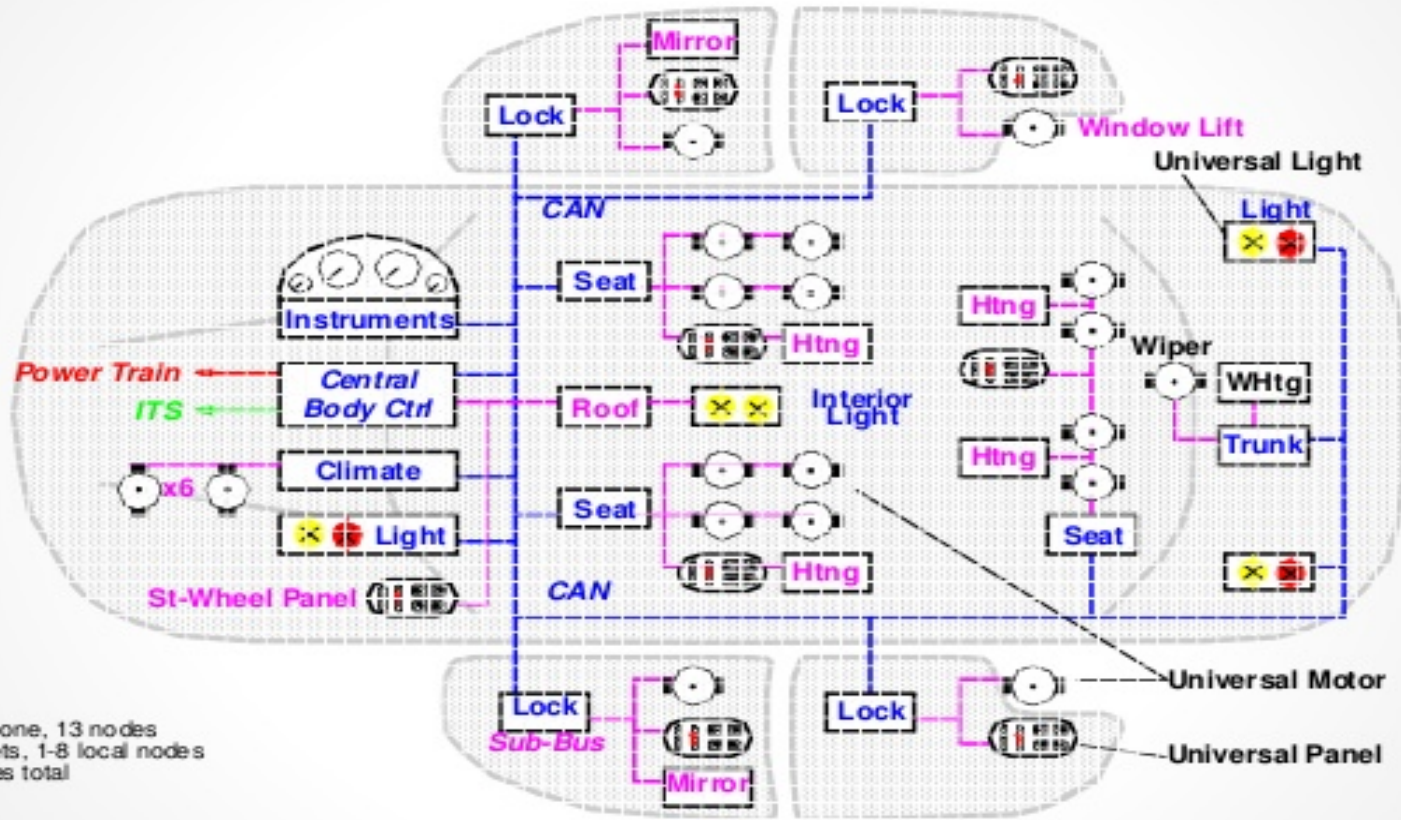
Automotive Embedded Systems (AES) Overview



- Today's cars are basically very complex embedded systems on wheels!
 - Contain tens of Electronic Control Units (ECUs) based on low to high end processors.
 - Each ECU can be considered an embedded system by itself
 - Many cameras, radars, sensors and actuators distributed throughout the vehicle
 - ECUs are connected using sophisticated in-vehicle networks
 - CAN, LIN, MOST, FlexRay and most recently Ethernet for automotive
- Features and Constraints of AES
 - High safety and reliability requirements
 - Real time constraints for many ECUs, e.g. engine controller
 - Harsh operating environment (both hot and cold temperature extremes)
 - Very cost sensitive, compared to areas such as Avionics

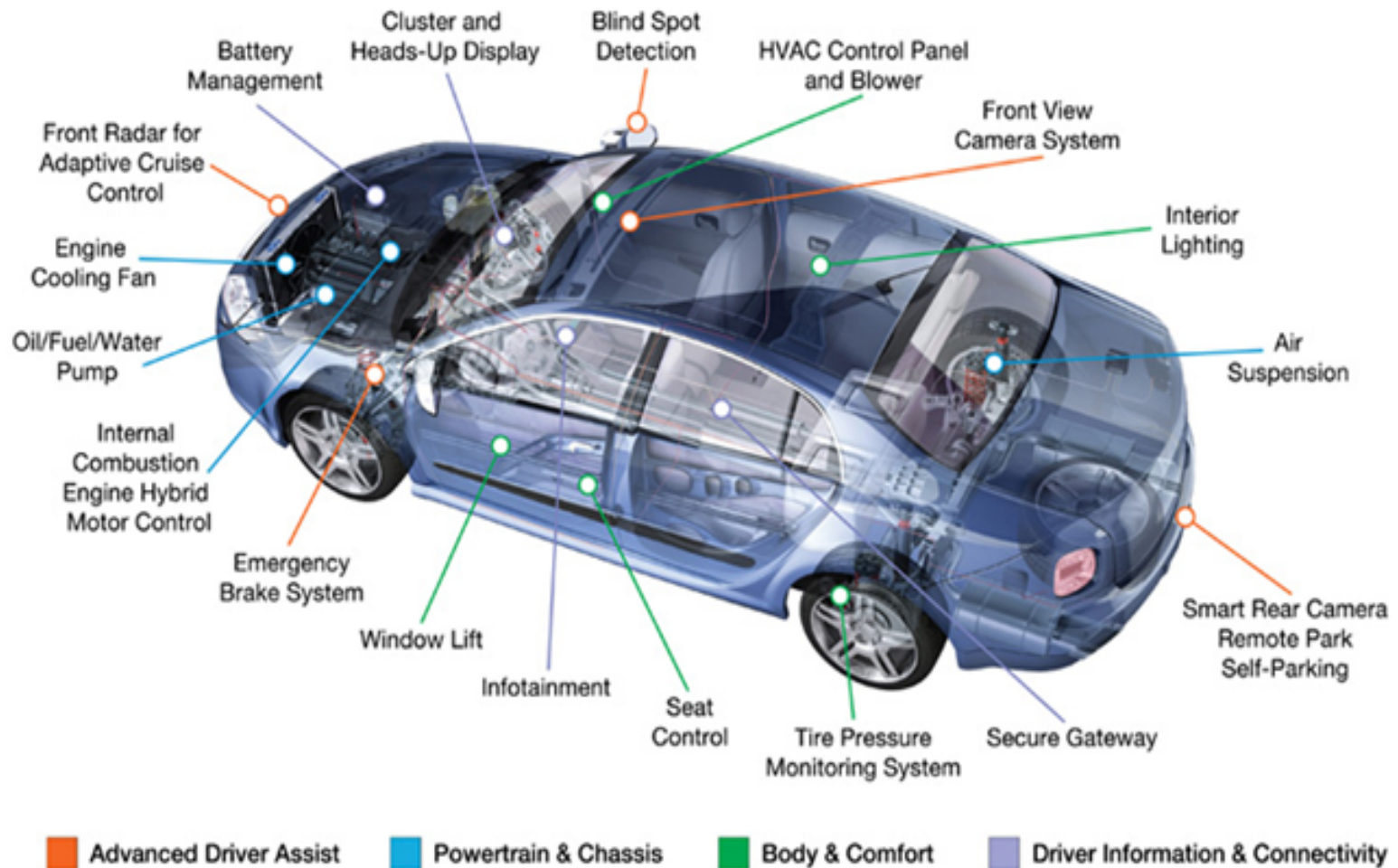


Automotive Embedded Systems



- Main tasks performed by AES
 - **Powertrain and chassis control:** engine, brake, steering, automatic transmission, etc.
 - **Vehicle body functions:** door locks, sliding windows, airbag, lighting, instrument display, etc.
 - **Infotainment:** vehicle navigation system, audio, cameras, video displays, etc.
 - **Integrated systems and services:** park assist, lane departure warning and assist, traction control, electronic stability control, automatic braking system, etc.





Design Challenges and Open Research Problems in AES (my view)



- **Disclaimer:** not including all open problems
- **Autonomous Vehicles (AV)** and **Advanced Driver Assistance Systems (ADAS)** are two of the most important drivers of R&D in AES
 - Entails lots of computationally intensive tasks that require powerful computing platform that provides real time execution of these tasks
 - Possibly heterogeneous computing systems (HCS) based on multi-core CPUs, GPUs, FPGAs, DSP processors, etc.
 - Need powerful but low cost and energy efficient computing platforms
- Increasing number of ECUs to handle new features is not sustainable.
 - There is considerable interest in “ECU consolidation”, i. e. map multiple tasks to a single ECU
 - This introduces it’s own technical challenges such as changes in design flow, impacts on timing and security, etc. [1]



- Large scale and complex software development for AES is a big challenge
 - How to achieve high reliability and safety
 - Software reuse for reduced design / verification cost and time
 - AUTOSAR (AUTomotive Open System ARchitecture) standard is a good start by automotive industry to address this challenge
 - See [\[2\]](#) for more details
- Challenges in design and verification of in-vehicle networks – especially support for automotive ethernet [\[3\]](#)
- Support for short range Vehicle-to-Everything (V2X) communication (includes V2V, V2I, V2D, V2P, V2G)
- Security is a major issue – keep the vehicle safe from hacking and even hijacking



Recent Research:
High Level Synthesis (HLS) and
Evaluation of Radar Signal
Processing Algorithm for FPGAs [4]



Introduction:

Background on High Level Synthesis

- What is HLS?
 - Optimized hardware synthesis from high level specification (C, C++, SystemC etc.)
 - Increases designer productivity by enabling design and verification at a higher level of abstraction
 - Enables reduced time to market and extensive design space exploration (DSE)
 - In recent years HLS CAD tools have become increasingly effective in terms of Quality of Results (QoR) achieved
 - Competitive with manual HDL-based design at the RTL level using VHDL or Verilog
 - Many options can be given manually to help HLS CAD tool optimize the design.



Introduction:

Why High Level Synthesis?

- Advantages of HLS
 - Design using High Level Languages (C, C++, etc.)
 - Scalability, portability and extendibility
 - Easy to learn/code.
 - High Productivity
 - Faster time to market
 - Lower NRE design cost
 - Utilization by Software Engineers
 - Do not need extensive hardware design knowledge
 - Automated Optimization and Design Space Exploration (DSE)
 - Compiler handles most of the optimizations
 - Fast design synthesis enables extensive DSE which is not feasible with RTL level design methodology



Introduction:

Why not Hardware Description Language?

- Disadvantages of HDL
 - Hardware Description Languages (Verilog, VHDL etc.)
 - Full access but difficult to design Hardware
 - Not scalable
 - Time consuming
 - Specific training required
 - Slow time to market
 - High cost
 - Software Engineers
 - Will require extensive training to effective use HDLs
 - Optimization
 - Manually optimize HDL code
 - Low productivity



Introduction:

High Level Synthesis tools

- There are quite a few HLS compilers, examples:
 - **Xilinx: Vivado HLS**
 - Intel (Altera): FPGA SDK for OpenCL
 - ADS: Agility
 - BlueSpec Inc: BlueSpec
 - Calypto Design Systems: Catapult C
 - Cadence: C to Silicon
 - UofT LegUp
- All HLS tools work in a similar manner
 - Synthesize High level specification into an optimized RTL design
 - Provide the HDL files in VHDL/Verilog, etc.



Introduction:

HLS tools – Why Xilinx Vivado HLS [6]

- Why Vivado HLS over others?
 - HDL code for the DSP algorithm was originally written/synthesized with Xilinx Vivado
 - Xilinx Vivado and Xilinx Vivado HLS are cross compatible
 - It is one of the most popular choices for HLS
 - It will be a fair comparison between the HLS and HDL capabilities in general.



Introduction:

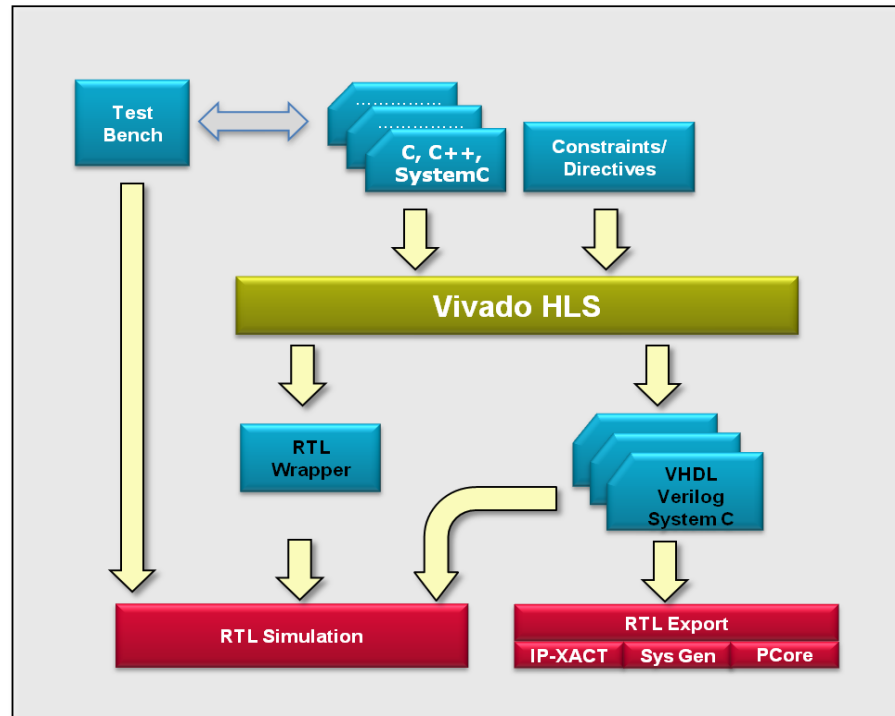
Xilinx Vivado HLS

- Major points
 - Similar to other tools
 - Converts code written in High Level Languages and converts it into RTL
 - Provides us with synthesis results like resource usage, latency etc.
 - Has support for C, C++ and SystemC
 - Traditional C/C++/System C will work with few minor changes due to the specific syntaxes for HW design
 - This was done to add support for bitwise operations and I/O etc.
 - Very huge library for pre-built common hardware designs (IP cores)
 - Eg: FFT, FIR etc.
 - Not compatible with other HLS tools.
 - RTL design available after synthesis



Introduction: Xilinx Vivado HLS

- HLS CAD flow:



RADAR Signal Processing

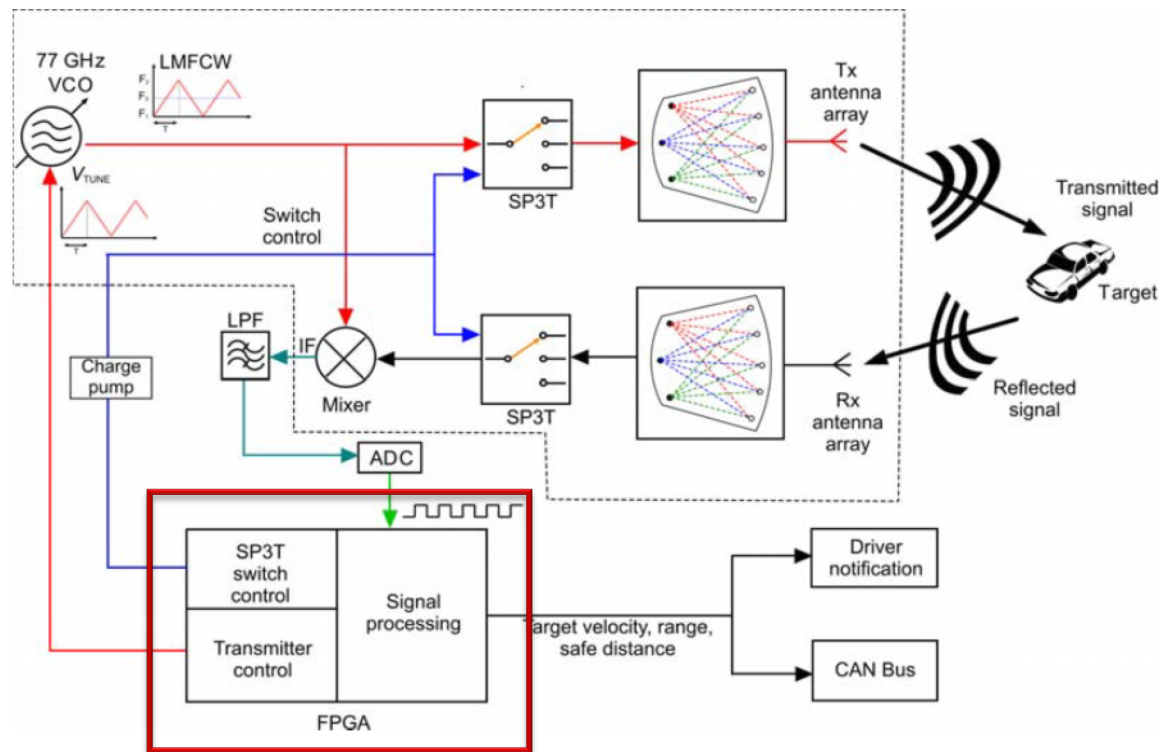
Introduction

- Controller for a Tri-mode Radar system
- High Level Synthesis of the controller
- Comparison between the HLS model and the existing HDL model [5]
- Radar System Applications: ADAS
 - For collision detection and/or avoidance, adaptive cruise control
- Algorithm (high level specification):
 - The algorithm is based on the existing HDL model of the FPGA based controller for the radar



RADAR Signal Processing Design Overview

- Tri-mode RADAR System overview



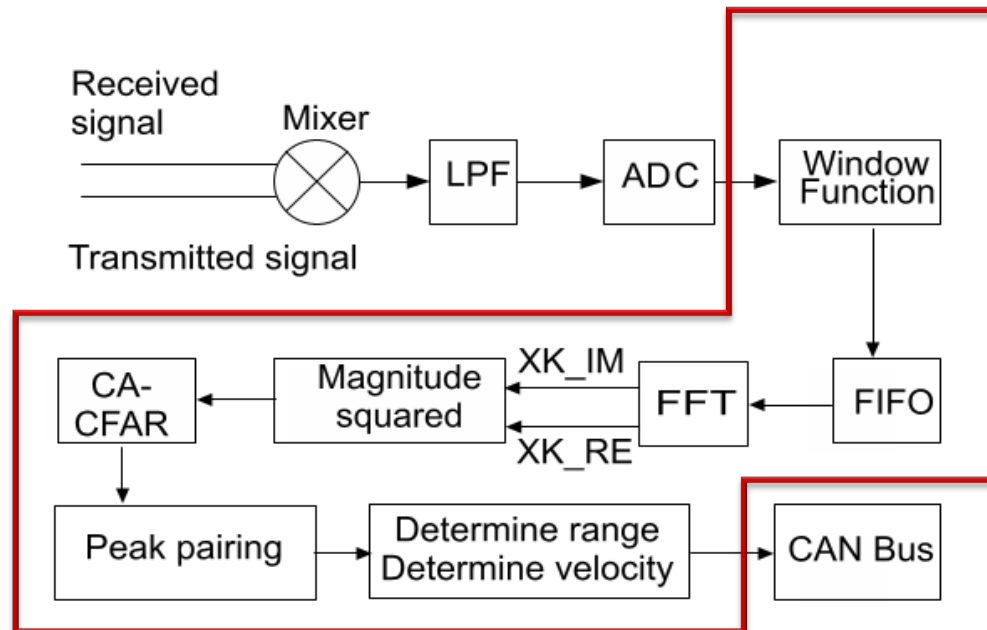
RADAR Signal Processing Algorithm

- Transmitter & switch control
 - Provides the modulating output to the DAC for VCO tuning
 - (4.5 V to 6.1 V) → (0 to 1023)
 - Controls which beam port to be used
 - 100 (Decimal equivalent of 4): For beam port 1
 - 010 (Decimal equivalent of 2): For beam port 2
 - 001 (Decimal equivalent of 1): For beam port 3
- Receiver Flow & Signal Processing
 - Provides the sampling clock to the ADC
 - 20 MHz generated from the system clock
 - RPU described in next slide



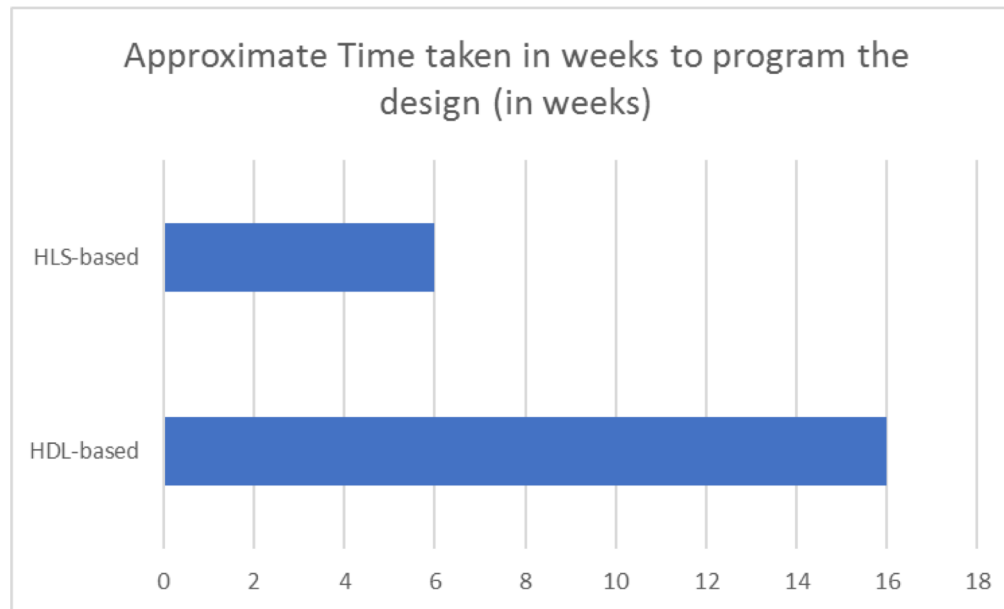
RADAR Signal Processing Algorithm - RPU

- The main signal processing flowchart for the system.



Evaluating HLS Results

Time-To-Market Metric



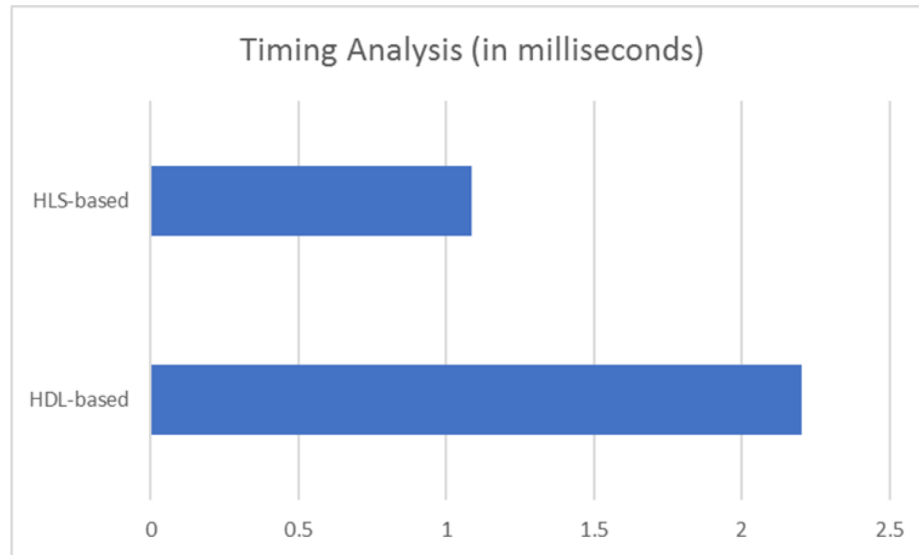
- Significantly less design time (62% reduction) required to synthesize the design using HLS-based design methodology.
 - **implies less time to market!**



Evaluating HLS Results

Performance Metric (Latency)

- Latency



Latency in	RTL-based design	HLS-based design
<i>Clock Cycles</i>	20185	10863
<i>Time in milliseconds</i>	2.20185 ms	1.0863 ms

- Around 2X speed up was achieved (for the same FPGA, Virtex 7)

Summary: HLS for RADAR SP

- HLS of a RADAR SP system was performed.
- Existing HDL-model of the system was used for comparison
- Significantly less time to market for the HLS-based design
- Achieved 2X speed up overall
- Significantly higher resource utilization when compared to the HDL-based design (synthesized for maximum possible speed)
- Despite that, no more than 5 % of the available resources on Viretx 7 FPGA were used.
- **We can conclude that HLS is much superior for hardware design for radar signal processing applications**



Infrastructure for AES Research



- Mentor Graphics Automotive Solutions
 - <https://www.mentor.com/mentor-automotive/>
- CAD tools and/or platforms available for research in following automotive areas
 - Connectivity (V2X)
 - Electrification (Electric vehicle related)
 - Autonomous (AV and ADAS)
 - Can look into acquiring DRS360 Platform for Autonomous Driving
 - Potential research enabler for Canadian universities in the areas of AV and ADAS
 - Architecture of automotive embedded systems
 - Volcano Automotive: design tools and run time software for automotive ECU design



- Xilinx Automotive Solutions
- <https://www.xilinx.com/applications/automotive.html>

- ADAS and AV
- Security
- Motor control
- In Vehicle Infotainment (IVI)
- Driver Information (DI) – HMI and UX
- Can look into [Xilinx Automotive Boards and Kits](#) offerings and evaluate them for their utility in automotive research



- Intel (Altera) Automotive Solutions
<https://www.altera.com/solutions/industry/automotive/overview.html>
- Intel automotive offerings seem to be more limited compared to Mentor and Xilinx
- Can look into **Intel Automotive Boards and Kits** offerings and evaluate them for their utility in automotive research



Conclusions

- Automotive Embedded Systems is a high growth area with lots of potential for researchers
 - ADAS and AV are key R&D drivers
 - Need new and innovative automotive system architectures
 - Platform based development will dominate
- Lots of open research areas for EDA/CAD and embedded system design research communities to explore and contribute innovative solutions for future Automotive Embedded Systems [1]
 - Need increased interaction with automotive community traditionally dominated by mechanical engineers
 - Retrofit existing solutions in embedded and EDA/CAD domains for the automotive domain



References

- [1] S. Chakraborty and S. Ramesh, *Guest Editorial: Special Section on Automotive Embedded Systems and Software*, IEEE Trans. On CAD of Circuits and Systems, Vol. 34, No. 11, November 2015.
- [2] H. Takada, *Introduction to Automotive Embedded Systems*, Online Presentation, Nagoya University, June 2012.
- [3] Zeng, Weiying; Khalid, Mohammed; Chowdhury, Sazzadur, In-Vehicle Networks Outlook: Achievements and Challenges, IEEE Communications Surveys and Tutorials, 18(3), 1552-1571, 2016, January
- [4] Siddhant Luthra, High Level Synthesis and Evaluation of an Automotive RADAR Signal Processing Algorithm for FPGAs, M.A.Sc. Thesis, ECE Dept., University of Windsor, August 2017.
- [5] Zereen, S; Lal, S.; Khalid, Mohammed; Chowdhury S., An FPGA-Based Controller for a 77 GHz MEMS Tri-Mode Automotive Radar, Microprocessors and Microsystems, Elsevier, 58, 34-40, 2018.



References

[6] Xilinx Vivado High Level Synthesis

<https://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html>

