Multi-Vehicle Map Fusion using GNU Radio

Optimization and Acceleration Opportunities

Augusto Vega
Akin Sisbot
Alper Buyuktosunoglu
Arun Paidimarri
David Trilla
John-David Wellman
Pradip Bose

IBM T. J. Watson Research Center

Acknowledgment

- Thanks to the many IBM colleagues who contribute to and support different aspects of this work + our esteemed university collaborators at Harvard, Columbia, and UIUC (Profs. David Brooks, Vijay Janapa Reddi, Gu-Yeon Wei, Luca Carloni, Ken Shepard, Sarita Adve, Vikram Adve, Sasa Misailovic) + many brilliant graduate students and postdocs!

- Special thanks to **Dr. Thomas Rondeau**, Program Manager of the DARPA MTO DSSoC Program

This research was developed, in part, with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This document is approved for public release: distribution unlimited.

Outline

- **Part 1: DARPA-funded EPOCHS project**
  - Domain-specific (heterogeneous) SoC development

- **Part 2: EPOCHS Reference Application (“ERA”)**
  - Application domain: *multi-vehicle cooperative perception*

- **Part 3: 802.11p Transceiver**
  - Optimization and acceleration opportunities

Outline

- **Part 1: DARPA-funded EPOCHS project**
  - –Domain-specific (heterogeneous) SoC development
- **Part 2: EPOCHS Reference Application (“ERA”)**
  - –Application domain: *multi-vehicle cooperative perception*
- **Part 3: 802.11p Transceiver**
  - –Optimization and acceleration opportunities

DARPA's Domain-Specific System on Chip (DSSoC) Program

Program Manager: Dr. Tom Rondeau

- **Goal:** to develop a heterogeneous system-on-chip (SoC) comprised of many cores that mix general purpose processors, special purpose processors, hardware accelerators, memory, and input/output (I/O) devices to significantly improve performance of applications within a **domain** *

- **A domain is larger than any one application**
  - We target the "super" domain of embedded processors for autonomous/connected cars

"cooperative perception"

computer vision

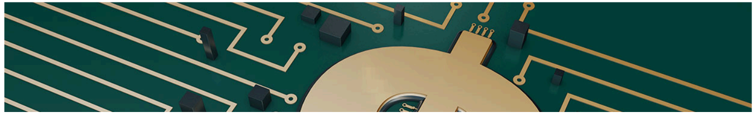
software radio

24 Jul 2018 | 17:00 GMT

DARPA Picks Its First Set of Winners in Electronics Resurgence Initiative

Teams announced in design, architecture, and materials and integration programs under the $1.5 billion effort to remake U.S. electronics

By **Samuel K. Moore**



Source: IEEE Spectrum (July 2018)

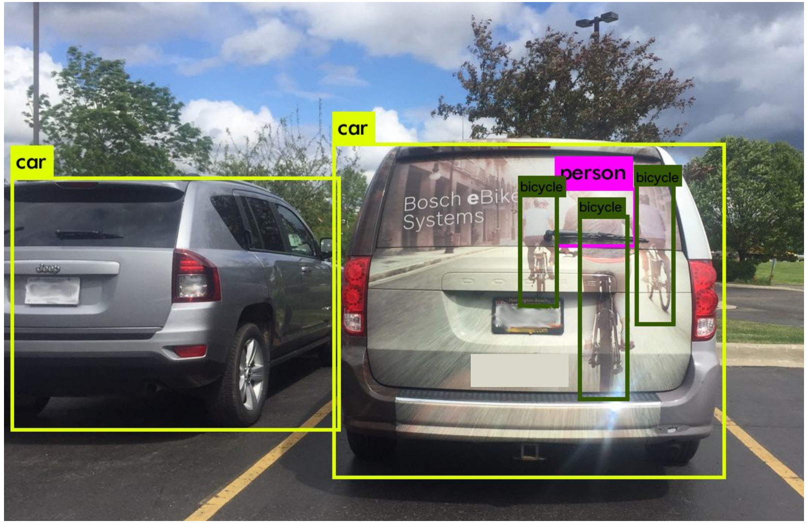
* *Source: https://www.darpa.mil/program/domain-specific-system-on-chip*

Application Domain: Cooperative Perception

- Automakers use arrays of sensors to build redundancy into their systems

This Image is Why Self-Driving Cars Come Loaded with Many Types of Sensors

When's a pedestrian not a pedestrian? When it's a decal.

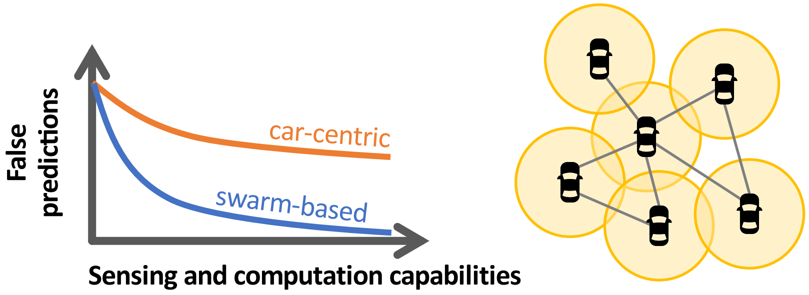


Source: MIT Technology Review

Application Domain: Cooperative Perception

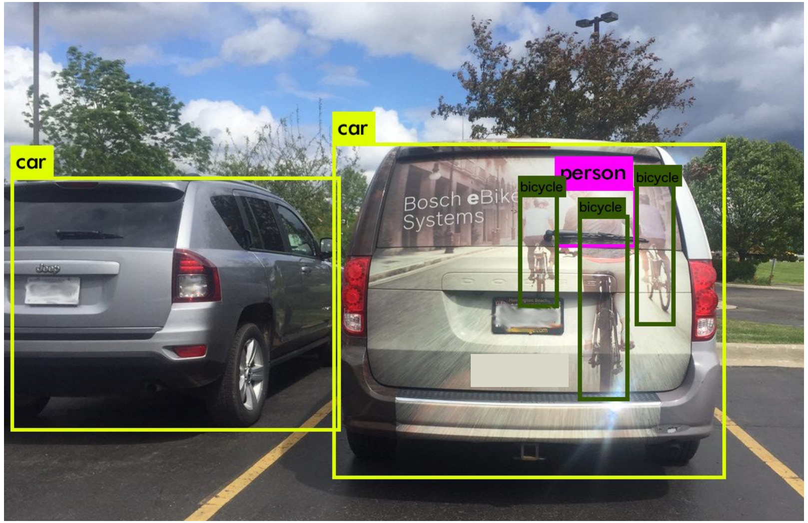
- Automakers use arrays of sensors to build redundancy into their systems

- We propose a complementary approach: **multi-vehicle (cooperative) perception**
  - Cars exchange locally-generated maps
  - Each vehicle merges its local map and the received ones in real time



This Image is Why Self-Driving Cars Come Loaded with Many Types of Sensors

When's a pedestrian not a pedestrian? When it's a decal.



Source: MIT Technology Review

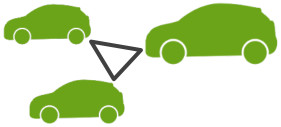
Efficient Programmability Of Cognitive Heterogeneous Systems

"EPOCHS" → our proposed solution for the design challenge presented by the DSSoC program

Efficient Programmability Of Cognitive Heterogeneous Systems

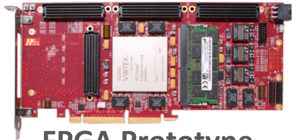
"EPOCHS" → our proposed solution for the design challenge presented by the DSSoC program

EPOCHS Reference Application



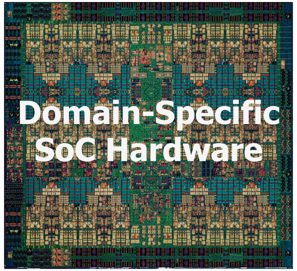
Compiler + Scheduler

Ontology & Design Space Exploration



FPGA Prototype

Agile Flow



Implementation

Accelerators + NoC + Memory Architecture

10X – 100X reduction in person-years

FPGA prototype, emulation, optimization, software bring-up

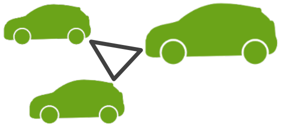
Agile methodology to quickly design and implement an easily programmed domain-specific SoC for real-time cognitive decision engines in connected vehicles

"Super"-Domain: Software-Defined Radio + Computer Vision

Efficient Programmability Of Cognitive Heterogeneous Systems

"EPOCHS" → our proposed solution for the design challenge presented by the DSSoC program

EPOCHS Reference Application



Compiler + Scheduler

Ontology & Design Space Exploration

FPGA Prototype

Agile Flow

10X – 100X reduction in person-years

FPGA prototype, emulation, optimization, software bring-up

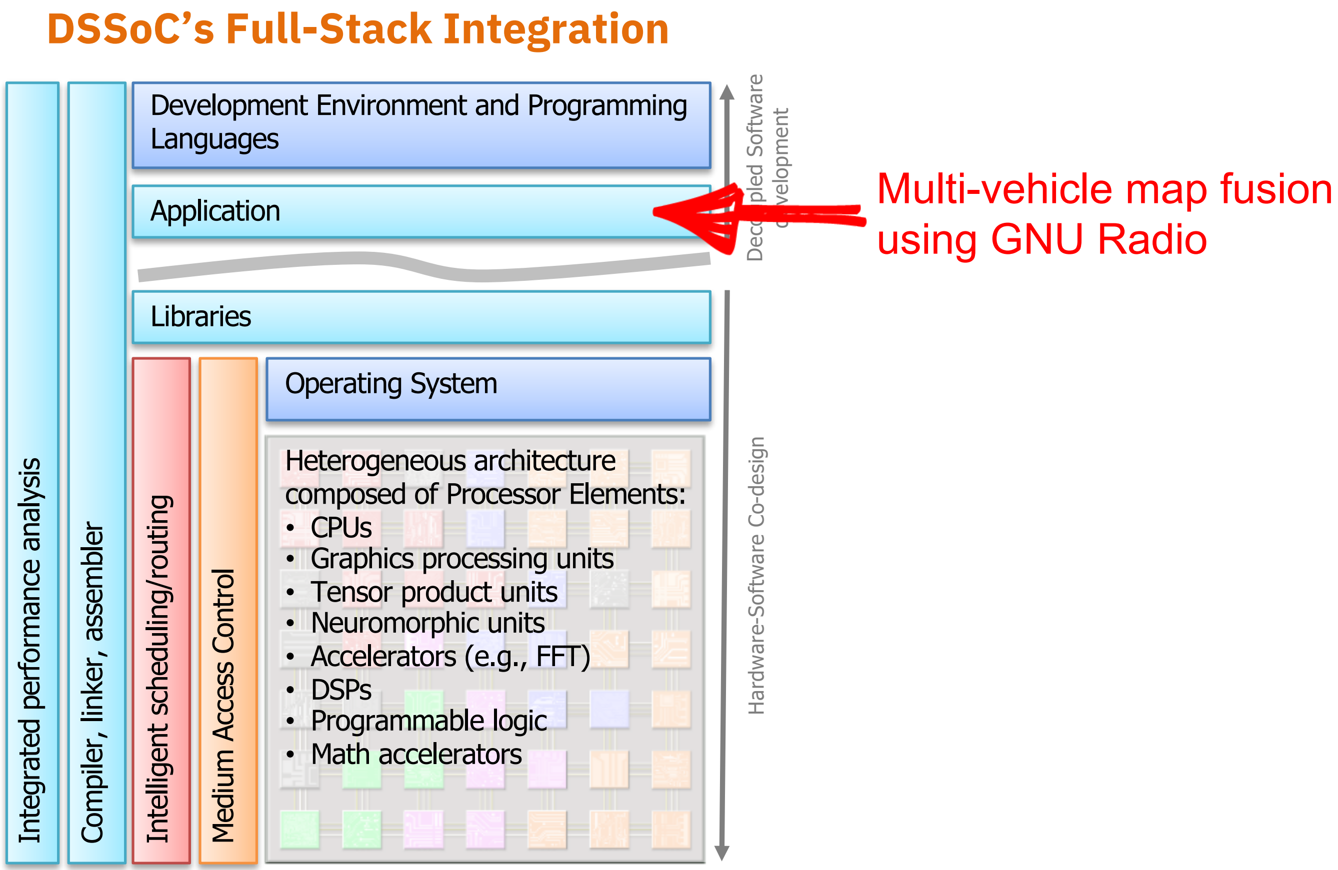
Implementation

Accelerators + NoC + Memory Architecture

Agile methodology to quickly design and implement an easily programmed domain-specific SoC for real-time cognitive decision engines in connected vehicles

"Super"-Domain: Software-Defined Radio + Computer Vision

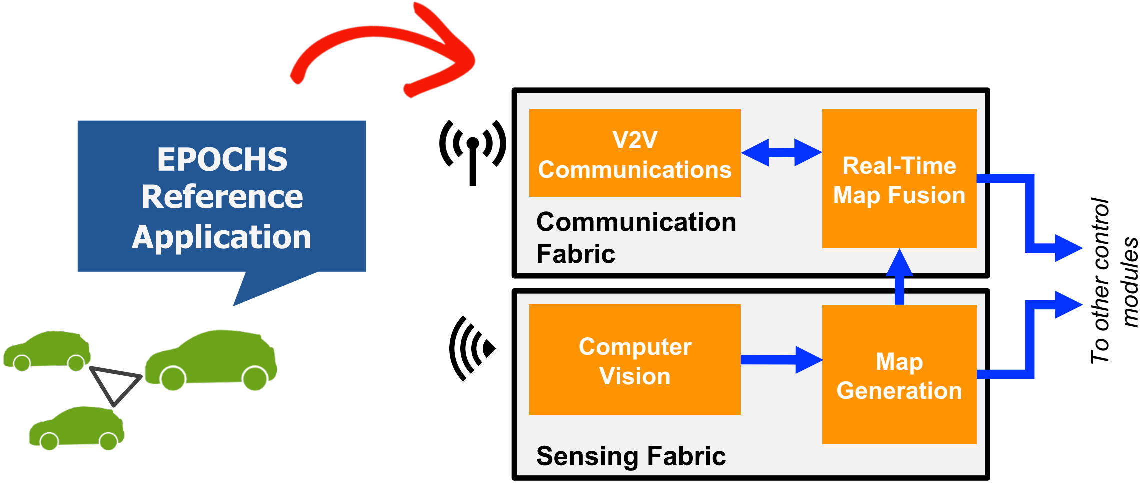
The Big Picture (Where Does This Talk Fit In?)



Outline

- **Part 1: DARPA-funded EPOCHS project**
  - Domain-specific (heterogeneous) SoC development
- **Part 2: EPOCHS Reference Application ("ERA")**
  - Application domain: *multi-vehicle cooperative perception*
- **Part 3: 802.11p Transceiver**
  - Optimization and acceleration opportunities

ERA: EPOCHS Reference Application

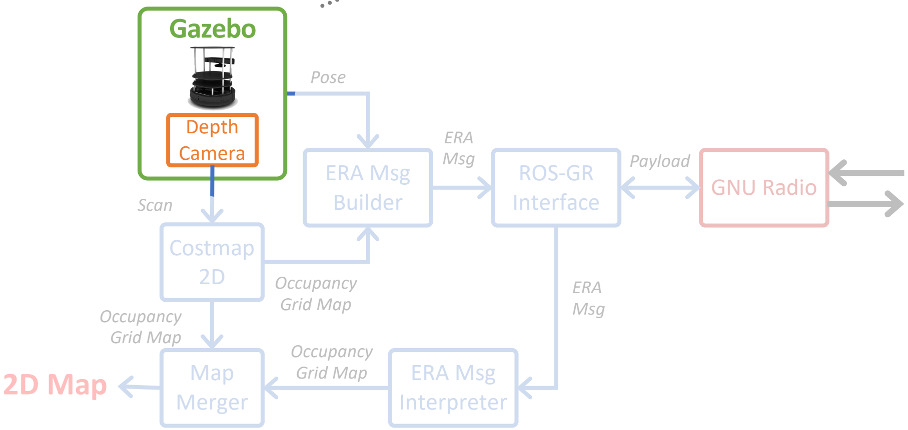


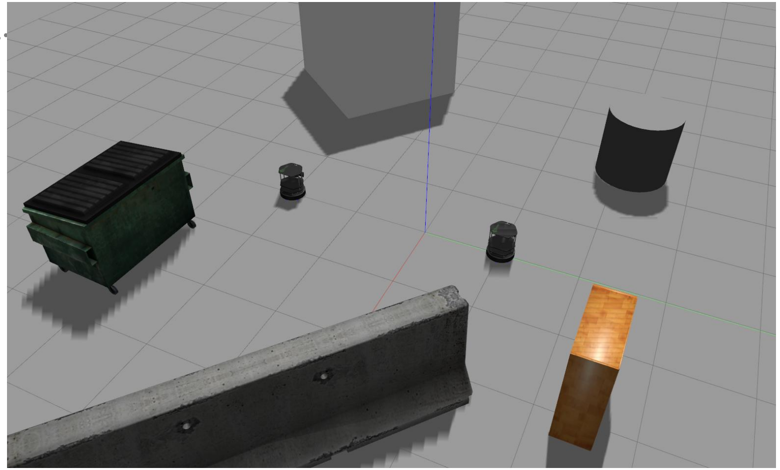
- **"Cooperative Perception" for connected/autonomous vehicles**
  - Multimodal sensing
  - Local occupancy map generation
  - DSRC-based V2V communication
  - Real-time map fusion

Contribute!
https://github.com/IBM/era

ERA Main Components (Single Robot's Viewpoint)

- Raw sensor data generated (simulated) using Gazebo in ERA v2
  – ERA v3 will replace Gazebo with an automotive simulation platform

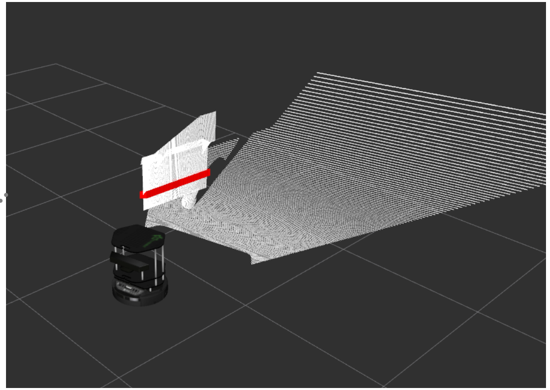




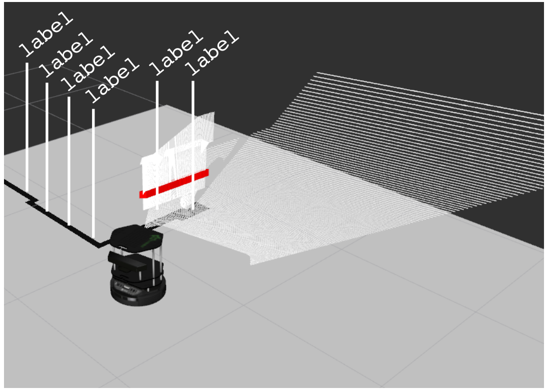
ERA Main Components (Single Robot's Viewpoint)

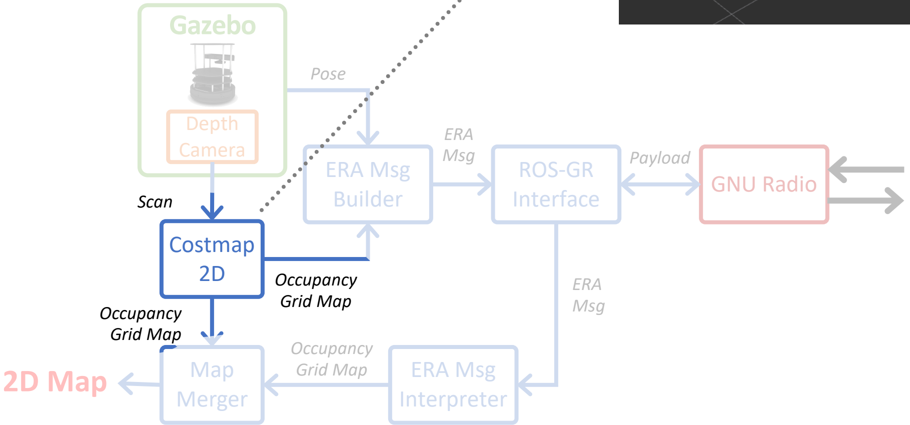
- Raw sensor data is first converted into laser scans which are used to generate a 2D occupancy grid map

Depth image → laser scans



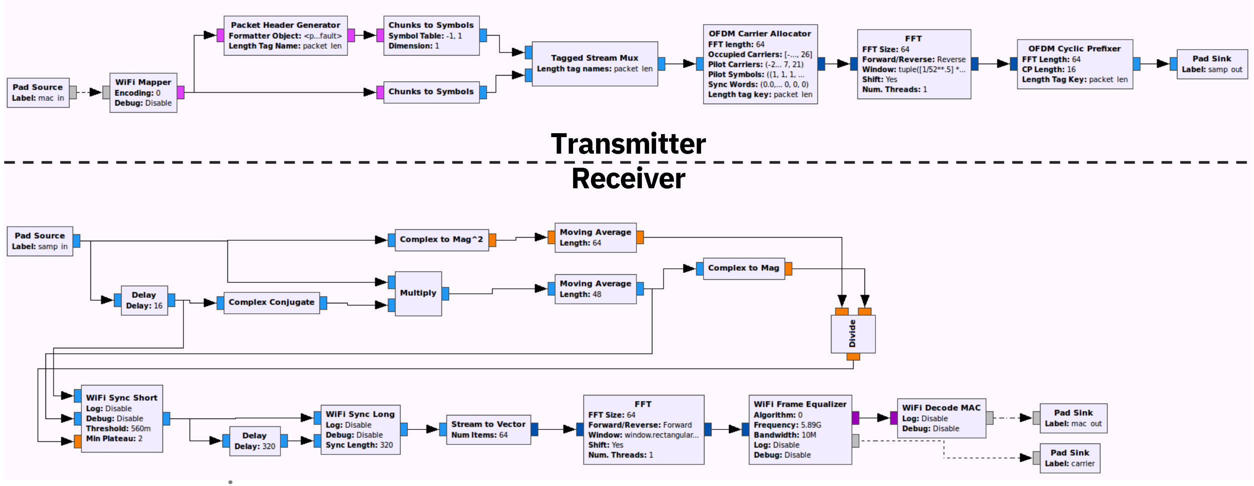
2D occupancy map generation

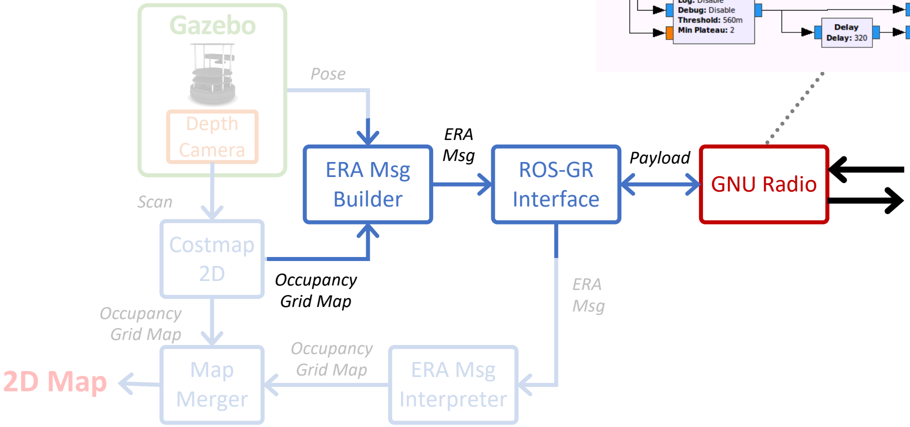




ERA Main Components (Single Robot's Viewpoint)

- Occupancy grid maps are serialized, compressed and put into a GNU Radio PDU
- Outbound PDUs are injected into the 802.11p transceiver

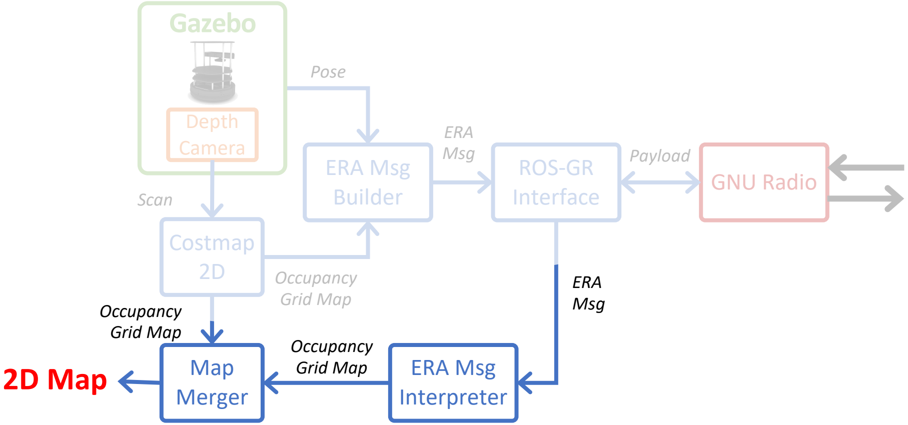




Open-source implementation by Bastian Bloessl
https://github.com/bastibl/gr-ieee802-11

ERA Main Components (Single Robot's Viewpoint)

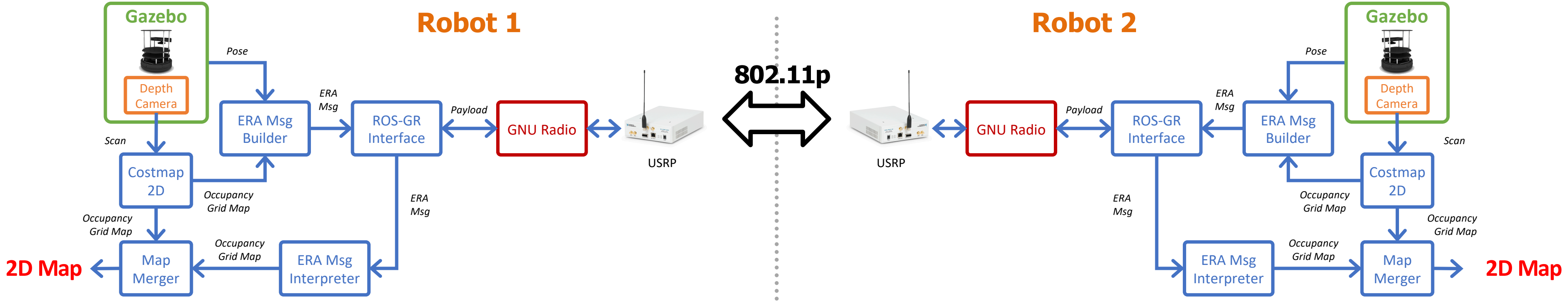
- Locally- and remotely-generated occupancy maps are merged in real time to improve the accuracy of the surroundings' view
- In ERAv2, *merging* is merely adding maps
  - Executed several times per second (!)

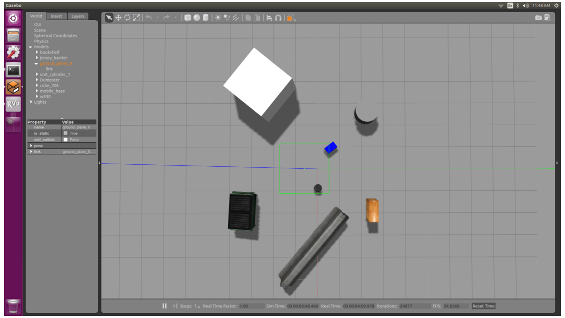


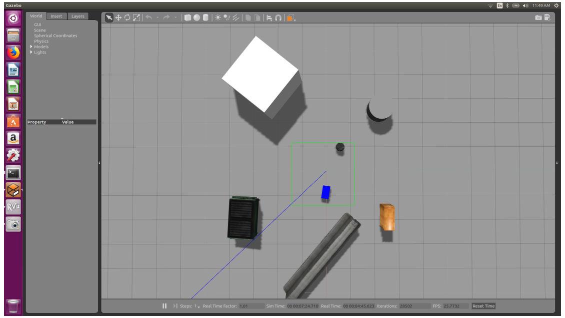

```
GridPtr combineGrids (const vector<nm::OccupancyGrid>& grids, const double resolution)
{
  GridPtr combined_grid(new nm::OccupancyGrid());
  combined_grid->info = getCombinedGridInfo(grids, resolution);
  combined_grid->data.resize(combined_grid->info.width*combined_grid->info.height);
  fill(combined_grid->data.begin(), combined_grid->data.end(), -1);
  ROS_DEBUG_NAMED ("combine_grids", "Combining %zu grids", grids.size());

  BOOST_FOREACH (const nm::OccupancyGrid& grid, grids) {
    for (coord_t x=0; x<(int)grid.info.width; x++) {
      for (coord_t y=0; y<(int)grid.info.height; y++) {
        const Cell cell(x, y);
        const signed char value=grid.data[cellIndex(grid.info, cell)];

        // Only proceed if the value is not unknown
        if ((value>=0) && (value<=100)) {
          BOOST_FOREACH (const Cell& intersecting_cell,
                         intersectingCells(combined_grid->info, grid.info, cell)) {
            const index_t ind = cellIndex(combined_grid->info, intersecting_cell);
            combined_grid->data[ind] = max(combined_grid->data[ind], value);
          }
        }
      }
    }
  }

  ROS_DEBUG_NAMED ("combine_grids", "Done combining grids");
  return combined_grid;
}
```

Option 1: Two-Computer Setup

- One Gazebo instance simulating one single robot/vehicle in each computer
- Over-the-air 802.11p communication (10-MHz OFDM with up to 64-QAM modulation)
- More info: https://github.com/IBM/era/wiki/ERA-in-two-computers

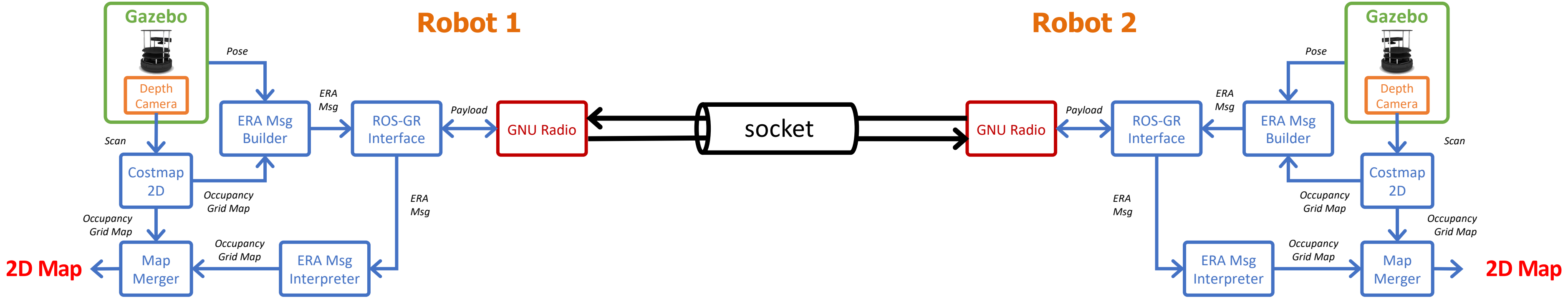


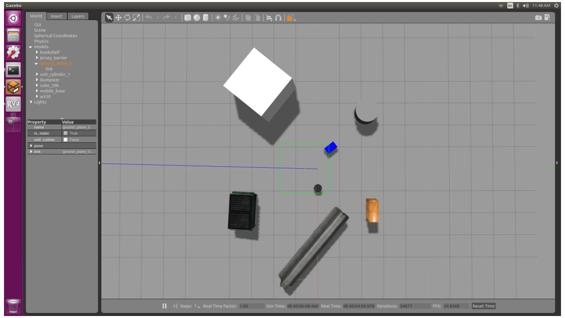


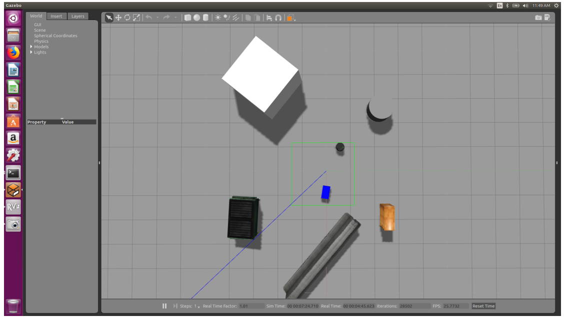


Option 2: Standalone Setup

- Runs on a single computer, replacing over-the-air communication with network sockets
- **Easiest setup to start with ✓**







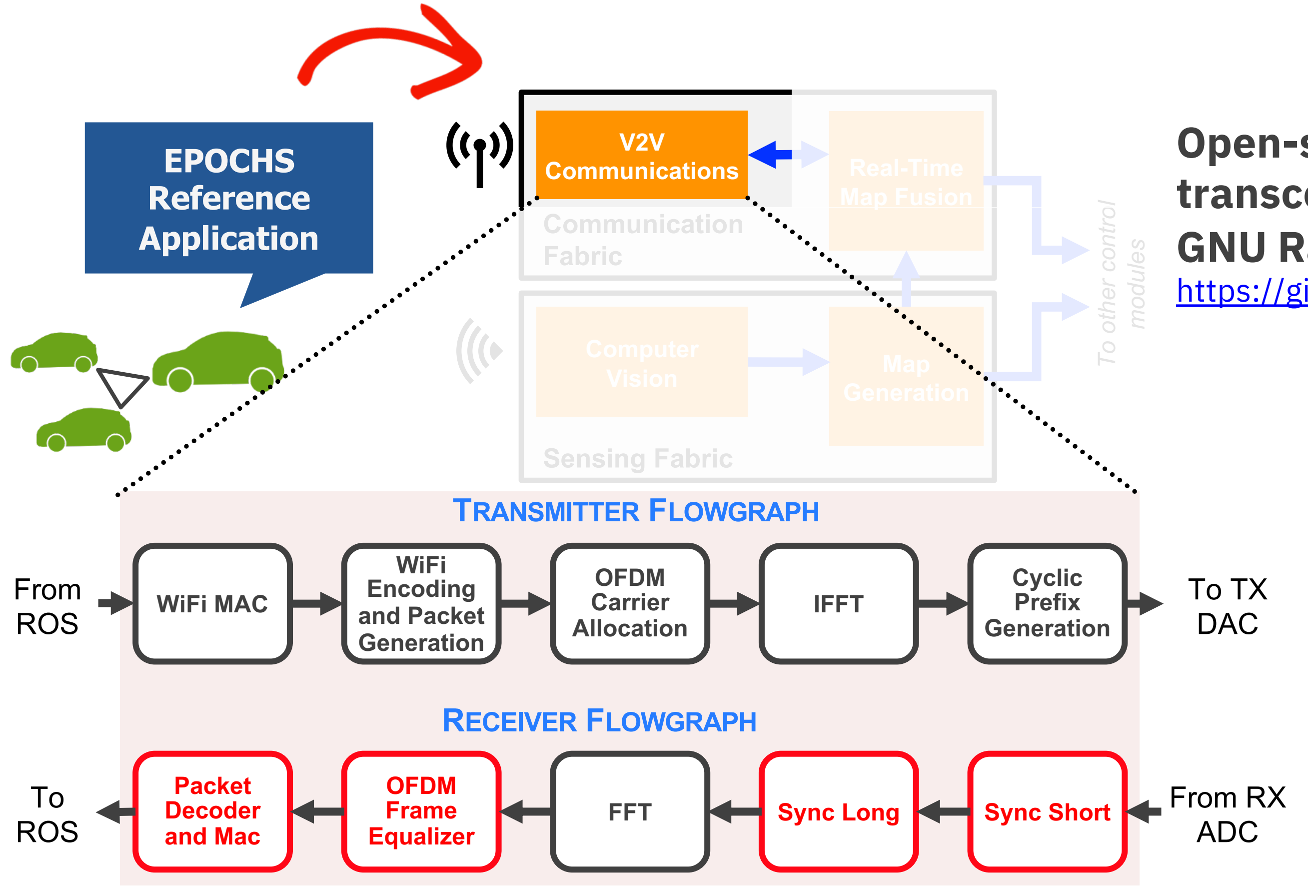
Outline

- Part 1: DARPA-funded EPOCHS project
  - Domain-specific (heterogeneous) SoC development

- Part 2: EPOCHS Reference Application ("ERA")
  - Application domain: *multi-vehicle cooperative perception*

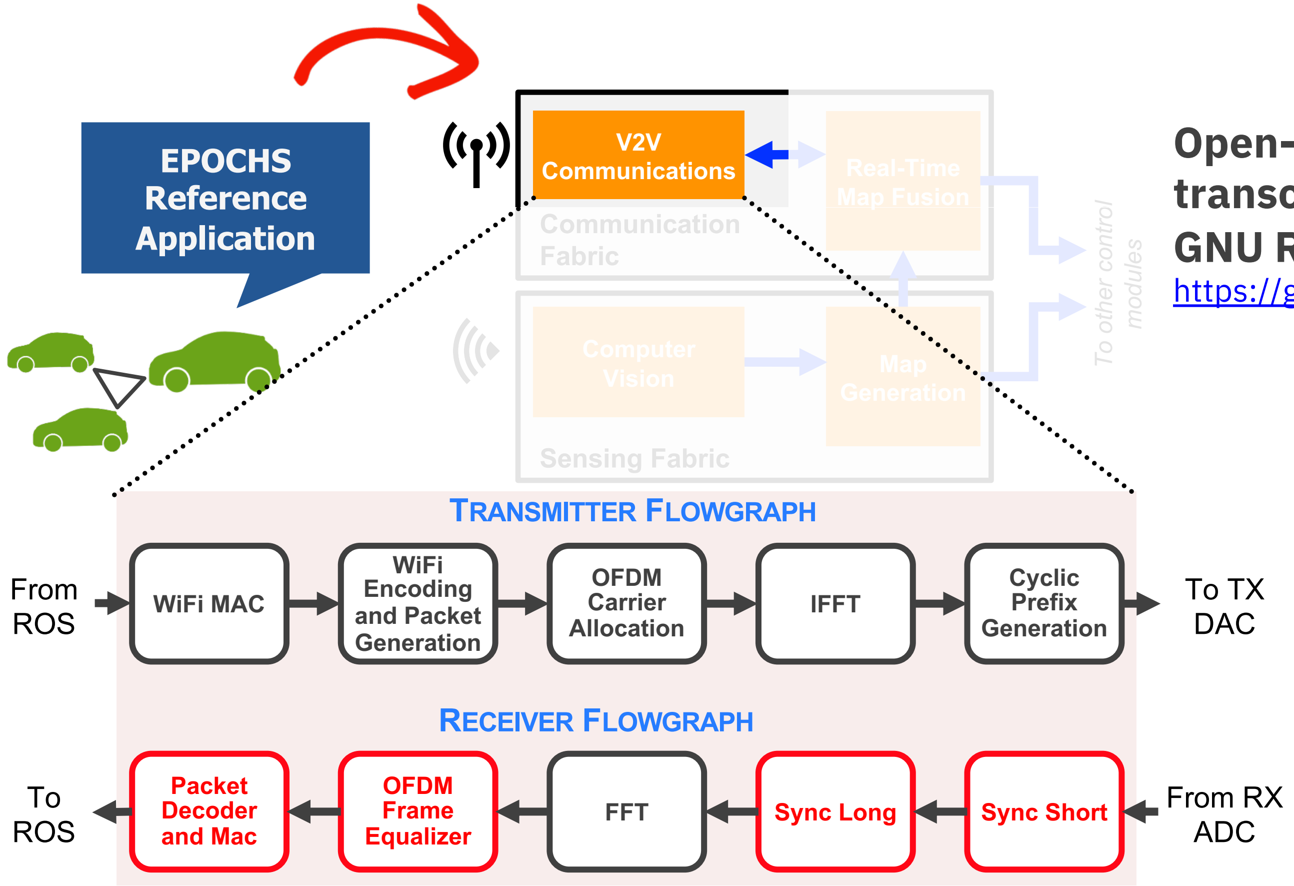
- **Part 3: 802.11p Transceiver**
  - Optimization and acceleration opportunities

802.11p Transceiver within ERA

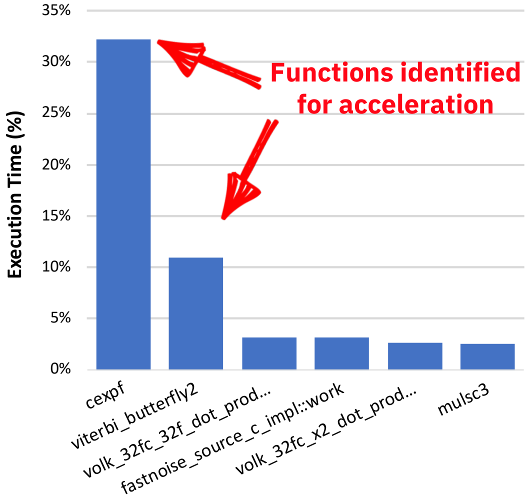


Open-source **IEEE 802.11p** transceiver implemented in GNU Radio by **Bastian Bloessl**
https://github.com/bastibl/gr-ieee802-11

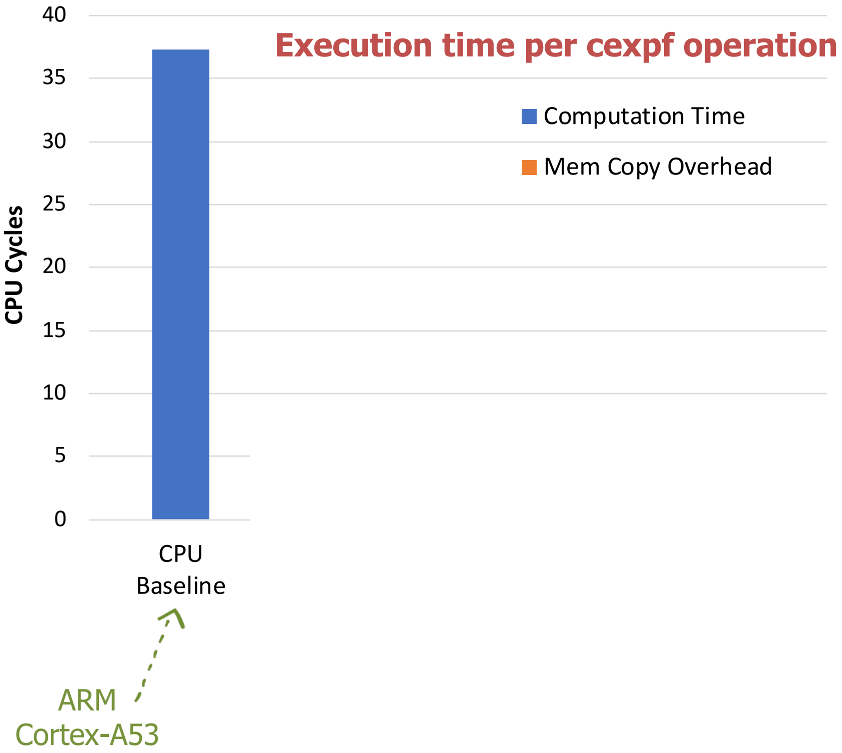
802.11p Transceiver within ERA



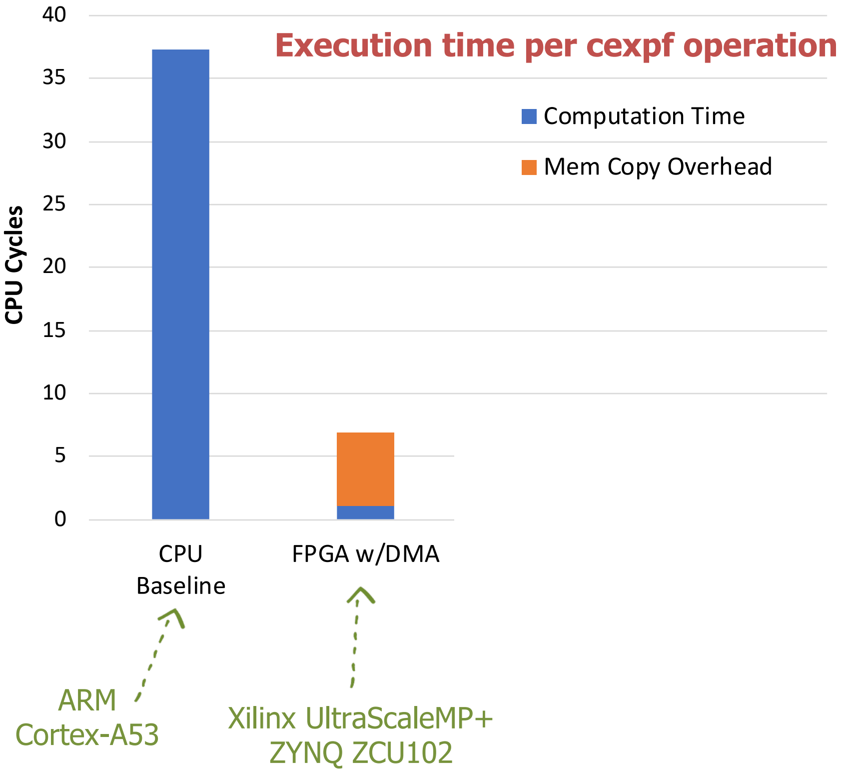
Open-source **IEEE 802.11p** transceiver implemented in GNU Radio by **Bastian Bloessl**
https://github.com/bastibl/gr-ieee802-11



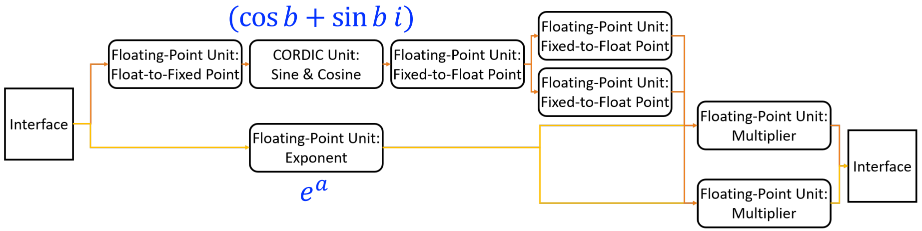
Acceleration Options (for *cexpf*): Preliminary Results



Acceleration Options (for *cexpf*): Preliminary Results

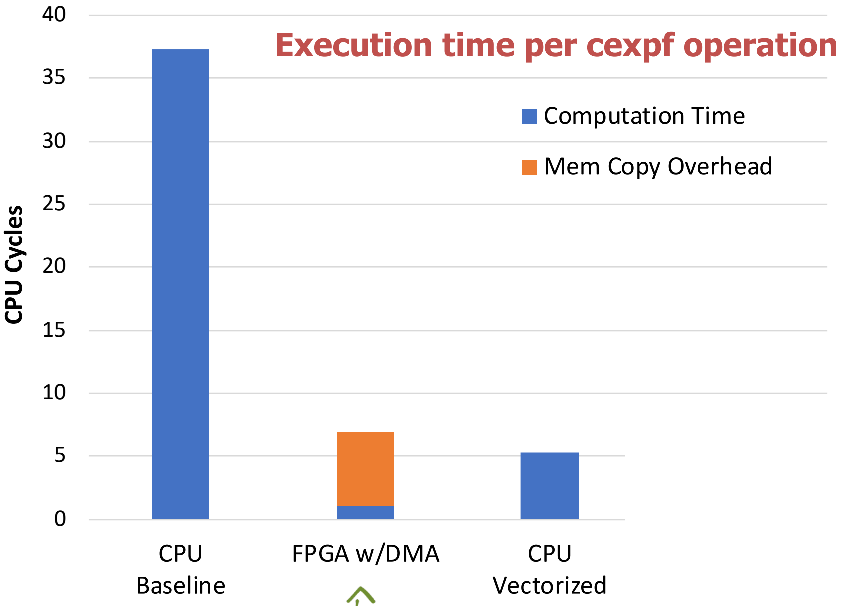


FPGA Implementation
(dual data-path pipeline)



$$e^{a+bi} = e^{a} * (\cos b + \sin b\ i)$$

Acceleration Options (for *cexpf*): Preliminary Results

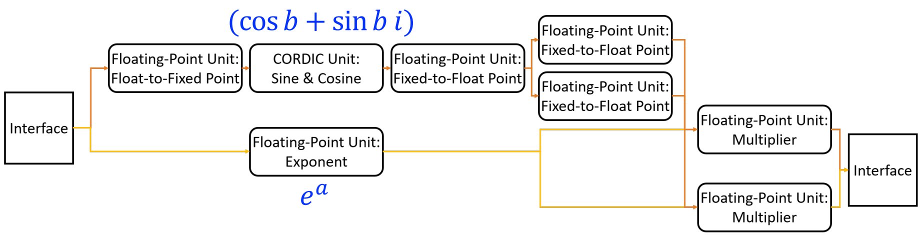


ARM Cortex-A53

Xilinx UltraScaleMP+ ZYNQ ZCU102

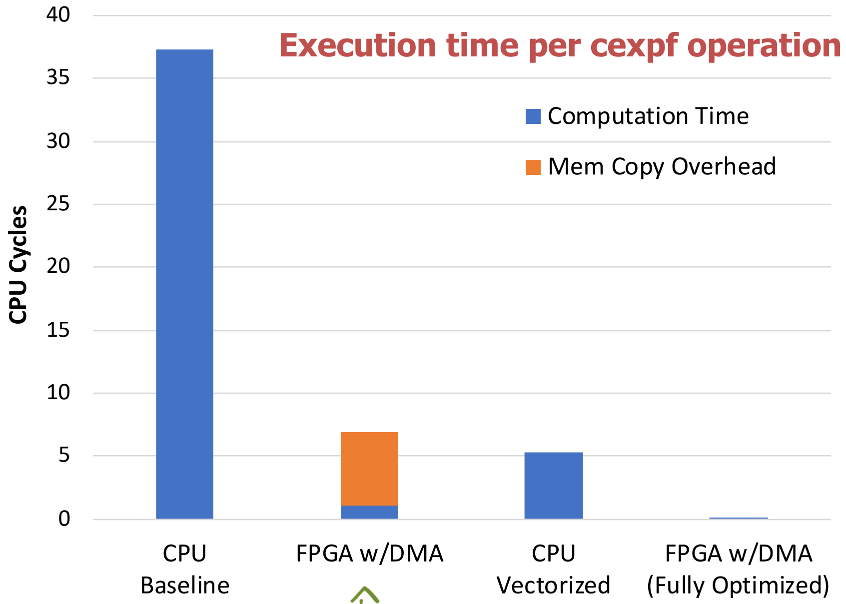
ARM's SIMD extension (NEON)

FPGA Implementation
(dual data-path pipeline)



$$e^{a+bi} = e^{a} * (\cos b + \sin b\, i)$$

Acceleration Options (for *cexpf*): Preliminary Results

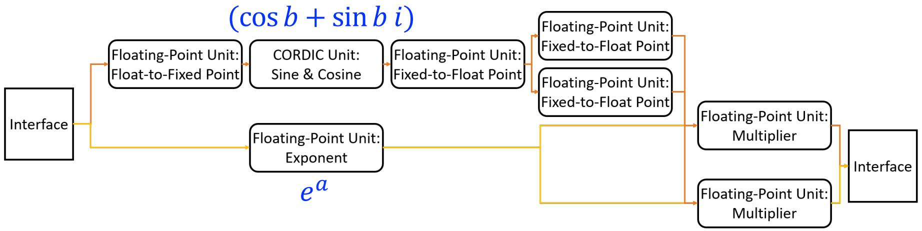


ARM Cortex-A53

Xilinx UltraScaleMP+ ZYNQ ZCU102

ARM's SIMD extension (NEON)

FPGA Implementation
(dual data-path pipeline)

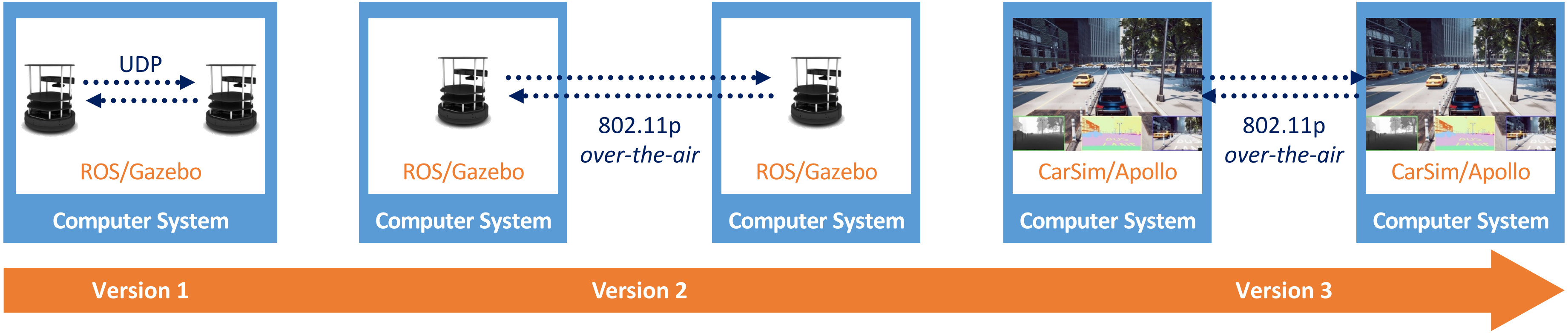


$$e^{a+bi} = e^a * (\cos b + \sin b\ i)$$

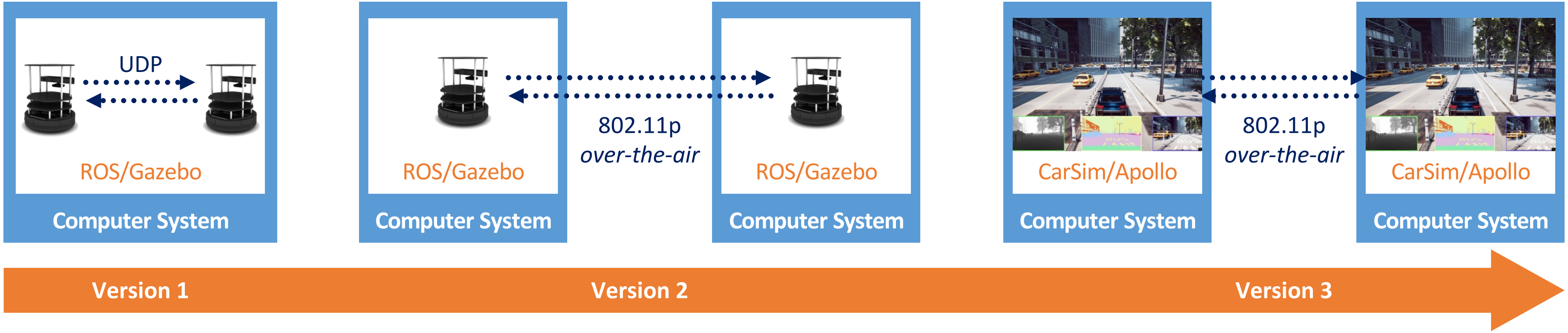
Fully-optimized implementation (idealized bound)

- 300 MHz (instead of 100 MHz)
- Four parallel computation engines
- Memory-copy elimination

ERA Roadmap


UDP
ROS/Gazebo
Computer System
ROS/Gazebo
Computer System
802.11p
over-the-air
ROS/Gazebo
Computer System
CarSim/Apollo
Computer System
802.11p
over-the-air
CarSim/Apollo
Computer System
Version 1
Version 2
Version 3

ERA Roadmap



ERA is only intended to enable **cooperative automotive**, with support for DSRC, and 5G (future)

This makes ERA unique

LAYER 1
World Simulators
(sensor data source)

- CarSim
- Gazebo
- CARLA
- LGSV

LAYER 2
Automotive Platforms
(perception, plan and control)

- Apollo
- Autoware

LAYER 3
Cooperative Vehicles Platform
(swarming and V2X support)

ERA: Apollo API | Autoware API | Multi-Vehicle Cooperation Logic | 802.11p | 5G

ROS | GNU Radio

some raw sensor data is directly fed to ERA

Summary

performance

power efficiency

- The **domain-specific (heterogeneous) SoCs** *era* is here!

- DARPA's Domain-Specific System on Chip (DSSoC) Program
  - Our proposed application domain: **multi-vehicle cooperative perception**
  - Local sensing + V2V communications
  - The **DSRC transceiver plays a critical role** for real-time V2V communications

ROS and GNU Radio "worlds" coexisting

Summary

performance

power efficiency

- The **domain-specific (heterogeneous) SoCs** *era* is here!
- DARPA's Domain-Specific System on Chip (DSSoC) Program
  - Our proposed application domain: **multi-vehicle cooperative perception**
  - Local sensing + V2V communications
  - The **DSRC transceiver plays a critical role** for real-time V2V communications

ROS and GNU Radio "worlds" coexisting

Turn ERA into a benchmark for cooperative mobility that can be easily "plugged" into existing platforms

Do you want to collaborate?

- Contact: ajvega@us.ibm.com
- GitHub: https://github.com/IBM/era

Thank You!

Photo by Balthazar Korab
Source: http://www.shorpy.com/node/15488

ajvega@us.ibm.com

https://github.com/augustojv