Robotic Computing on FPGAs

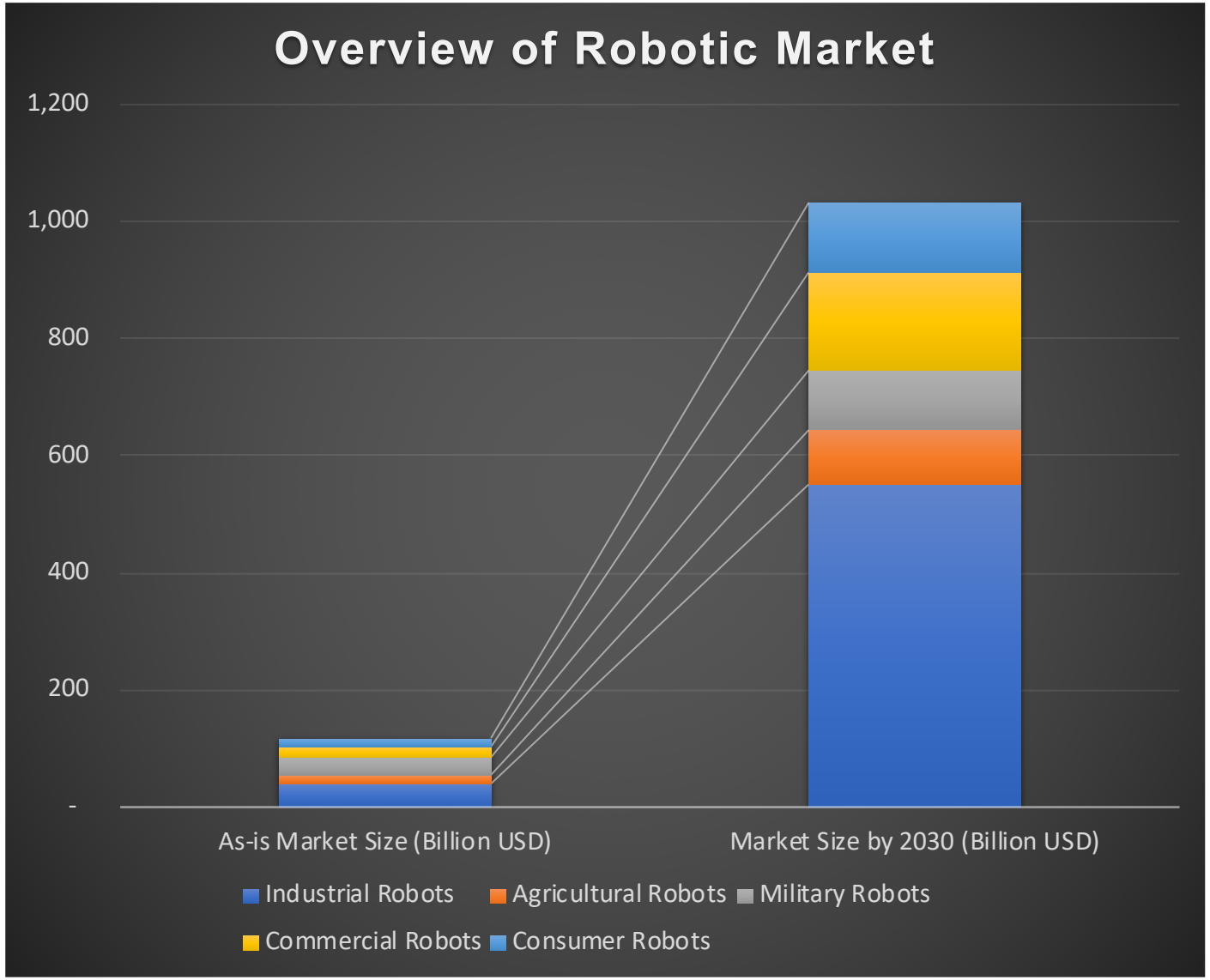
Shaoshan Liu
shaoshan.liu@perceptin.io

ROS Hardware Acceleration WG, meeting #4

Overview

- Why Autonomous Machines?
- Experiences of Building a Computing System for Autonomous Machines.
- Using FPGAs in Autonomous Machine Computing Systems.
- Partial Reconfiguration is So Underrated.
- Summary

Underestimated Market

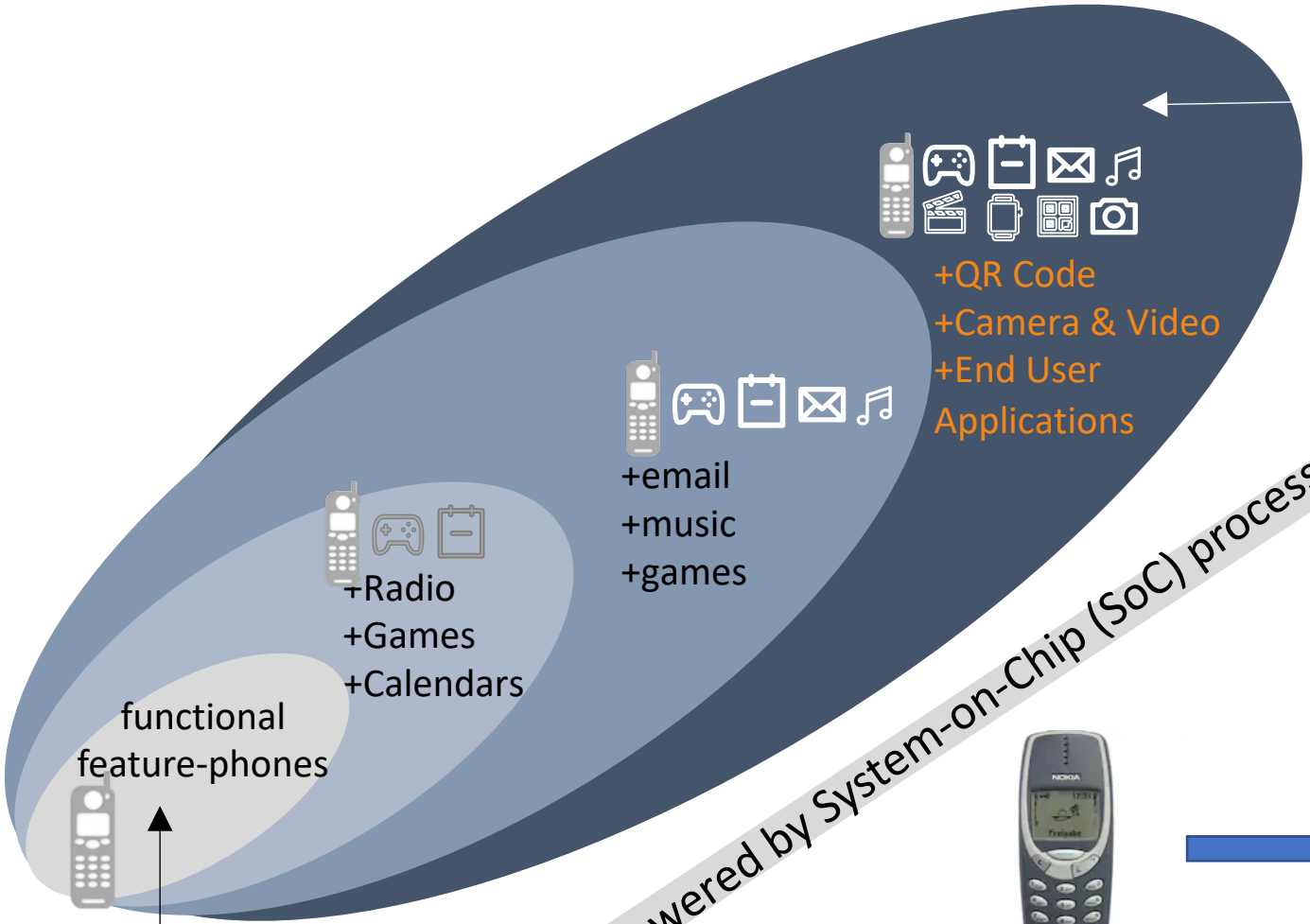


- **Market Potential**: the projected average compound annual growth rate (CAGR) of 22%, and by 2030 this sector will have a market size of $1 trillion

- **Problem:** sadly today autonomous vehicles and robots are still not able to perform interesting tasks, as these autonomous vehicles and robots are spending most of their "brain" power struggling to complete the basic chores, such as moving, recognizing obstacles, *etc*

- **Solution:** We aim to implement a computer system that handles all the basic functions in hardware, so we can reserve most of the "brain" power for interesting applications, and thus unlocking the imagination and power of software developers to create world-changing robotic applications.

A Historical Analogy: Mobile Phone Market



$10B Market Size

90% basic functions

10% end-user applications

Powered by System-on-Chip (SoC) processor

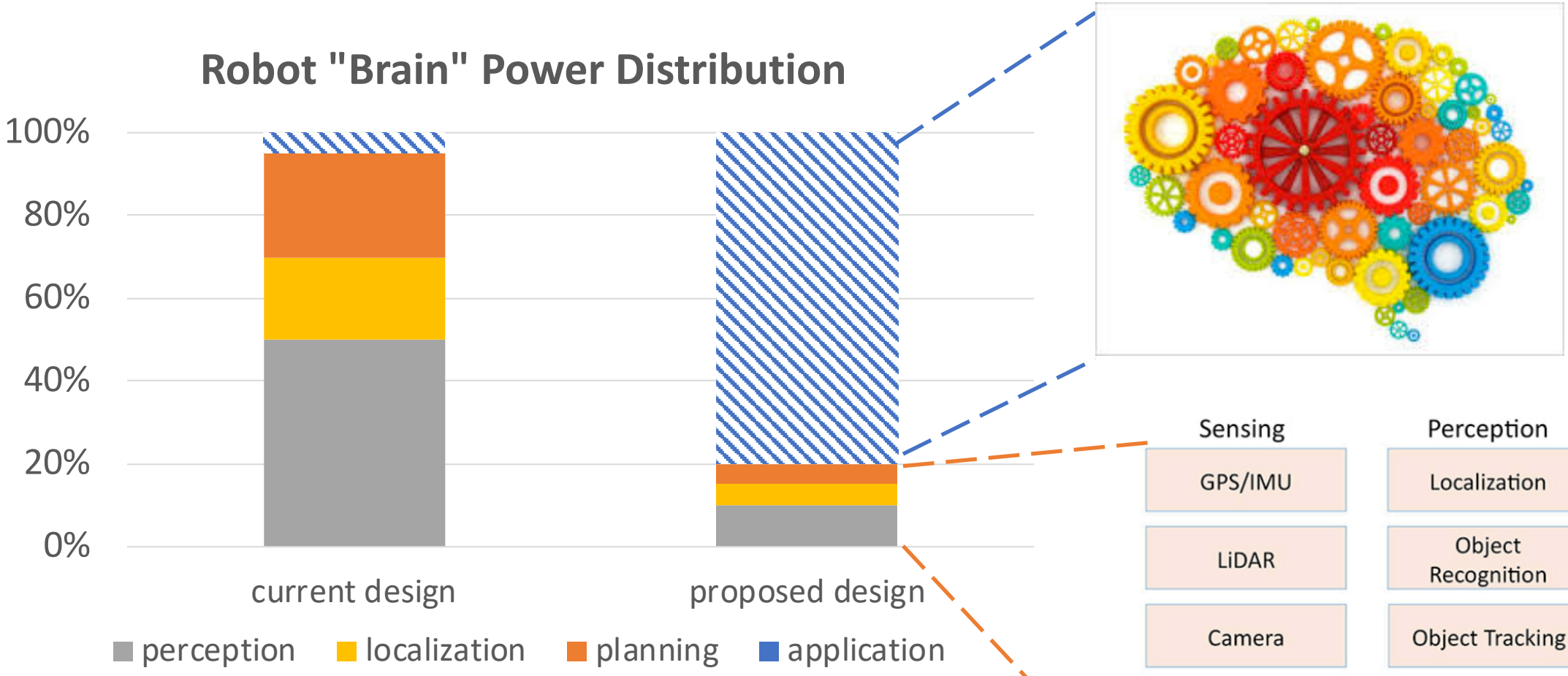
$800B Market Size

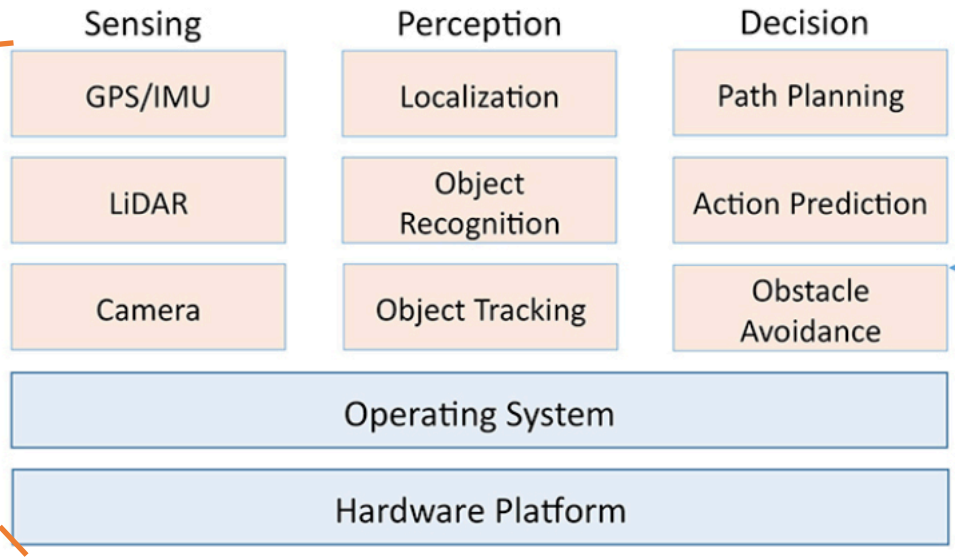
90% end-user applications, such as YouTube, Instagram, Facebook, etc

10% basic functions

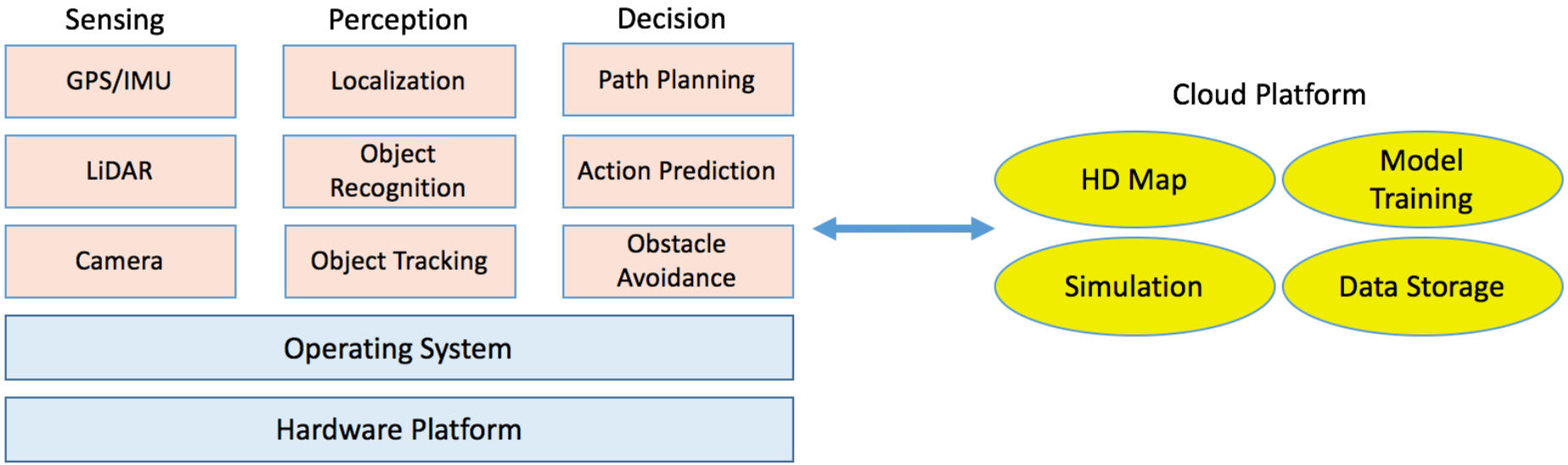
Apps!

More Room for Applications





Autonomous Machine Computing Systems



Liu, S., Li, L., Tang, J., Wu, S. and Gaudiot, J.L., 2017. Creating autonomous vehicle systems. Synthesis Lectures on Computer Science, 6(1)

Real Business Story Behind

- **Option 1**: Optimization of commercial mobile SoC
- **Option 2**: Procurement of specialized autonomous driving computing systems
- **Option 3**: Development of proprietary autonomous driving computing systems

Journals & Magazines > IEEE Engineering Management R... > Early Access

Critical Business Decision Making for Technology Startups: A PerceptIn Case Study

Publisher: IEEE | Cite This | PDF

Shaoshan Liu **All Authors**

24 Full Text Views

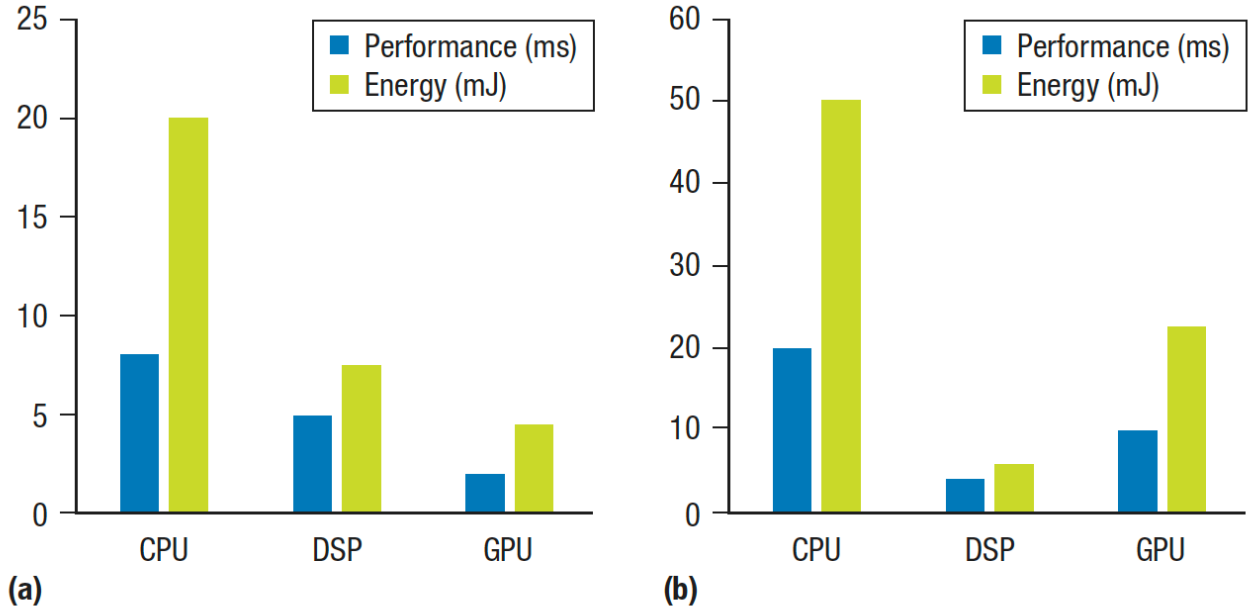
Abstract
Authors
Keywords
Metrics

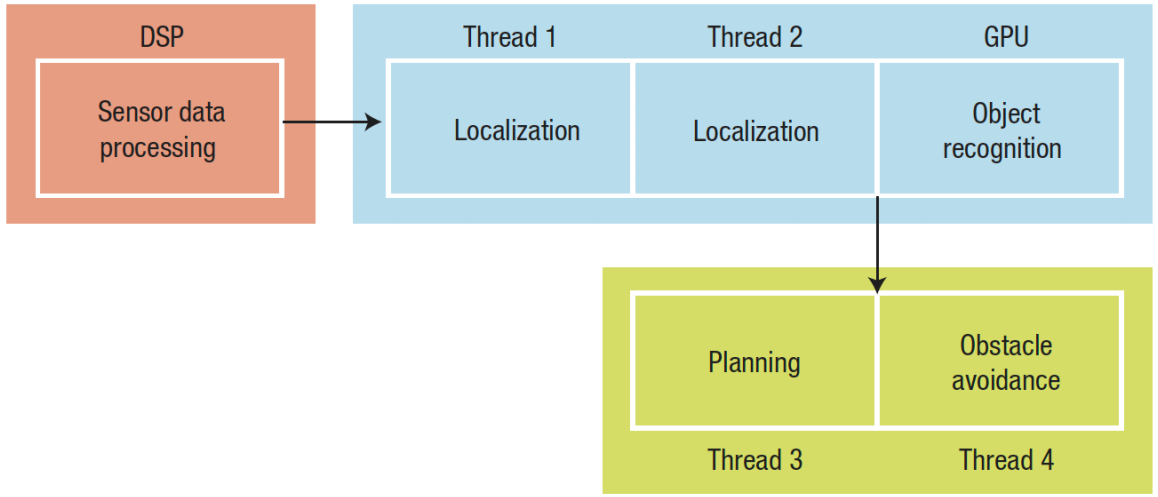
Abstract:
most business decisions are made with analysis, but some are judgment calls not susceptible to analysis due to time or information constraints. In this article, we present a real-life case study of critical business decision making of PerceptIn, an autonomous driving technology startup: in early years of PerceptIn, PerceptIn had to make a decision on the design of computing systems for its autonomous vehicle products. By providing details on PerceptIn's decision process and the results of the decision, we hope to provide some insights that can be beneficial to entrepreneurs and engineering managers in technology startups.

Published in: IEEE Engineering Management Review (Early Access)

Liu, S., 2020. Critical Business Decision Making for Technology Startups: A PerceptIn Case Study. IEEE Engineering Management Review, 48(4)

Optimization of Commercial Mobile SoCs





Liu, S., Tang, J., Zhang, Z. and Gaudiot, J.L., 2017. Computer architectures for autonomous driving. Computer, 50(8), pp.18-25.

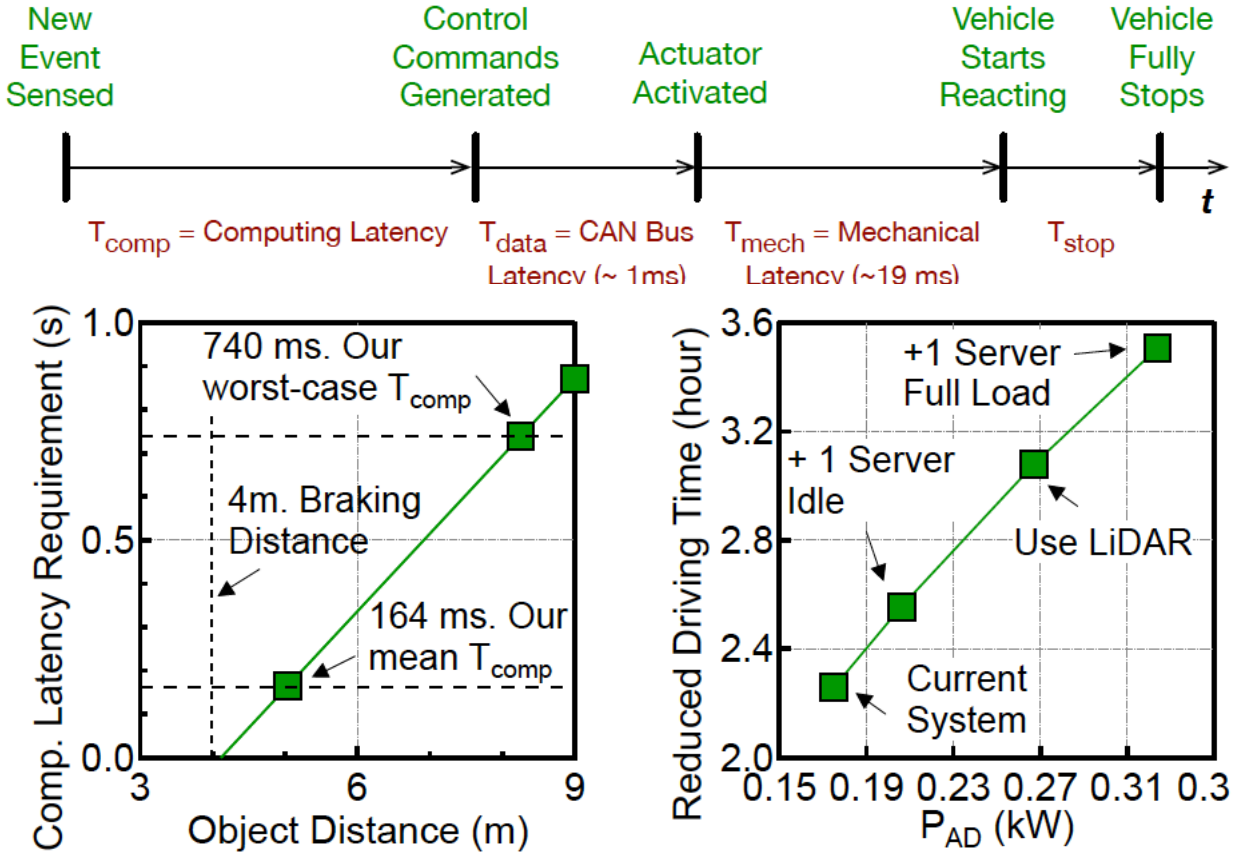
Liu, L., Tang, J., Liu, S., Yu, B., Xie, Y. and Gaudiot, J.L., 2021. π-RT: A Runtime Framework to Enable Energy-Efficient Real-Time Robotic Vision Applications on Heterogeneous Architectures. Computer, 54(4), pp.14-25.

Procurement of Specialized Computing Systems

- There were commercial computing platforms specialized for autonomous driving:
  - from NXP, MobilEye, and Nvidia.
- ASICs that provide high performance at a much higher cost.
- Nvidia PX2 system costs over $10,000.
- They mostly accelerate only the perception function in autonomous driving
- PerceptIn required a system that optimizes the end-to-end performance.

Development of a Proprietary System

1. Latency and Throughput
   1. computing and physical control latencies
   2. throughput 10 Hz
2. Energy and Thermal
   1. computing and sensing drop operation time from 10 to 7.7 hrs
   2. conventional cooling for thermal
3. Cost and Safety
   1. low cost to sustain $1 per trip cost
   2. proactive and reactive paths

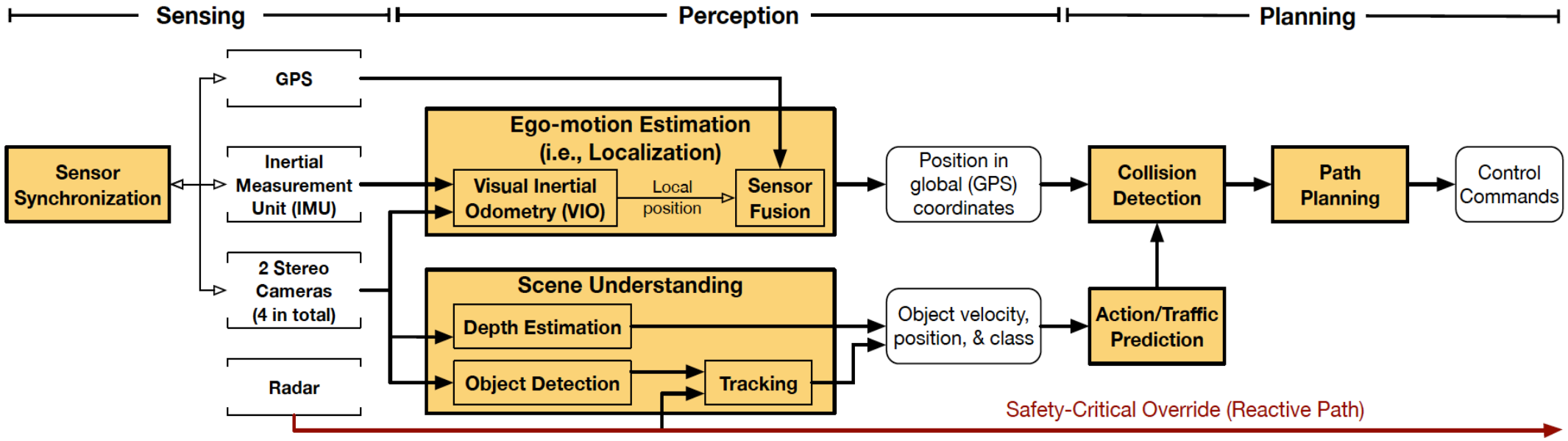


(a) The computing latency requirement becomes tighter when the object to be avoided is closer.

(b) Driving time reduces as the power of the autonomous driving system increases (P_{AD}).

Yu, B., Hu, W., Xu, L., Tang, J., Liu, S. and Zhu, Y., 2020, October. Building the computing system for autonomous micromobility vehicles: Design constraints and architectural optimizations. In 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)

Development of a Proprietary System

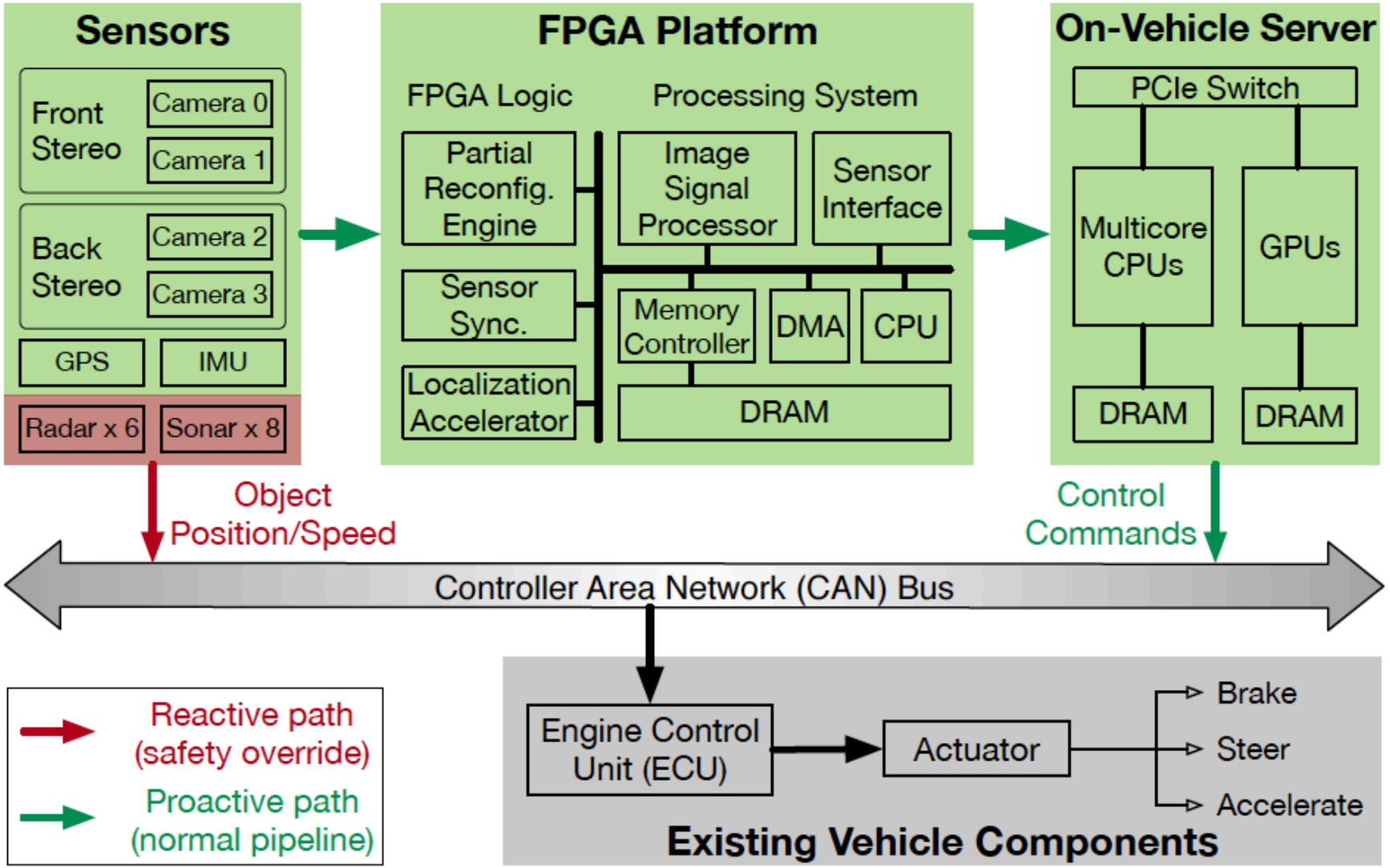


Task	Algorithm(s)	Sensors(s)
Depth Estimation	ELAS [44]	Cameras
Object Detection	YOLO [48]/ Mask R-CNN [49]	Camera
Object Tracking	KCF [46]	Camera, Radar
Localization	VIO [41]	Cameras, IMU, GPS
Planning	MPC [47]	–

- Task-Level Parallelism
  - sensing, perception, and planning are serialized
  - different sensor processing are independent

Yu, B., Hu, W., Xu, L., Tang, J., Liu, S. and Zhu, Y., 2020, October. Building the computing system for autonomous micromobility vehicles: Design constraints and architectural optimizations. In 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)

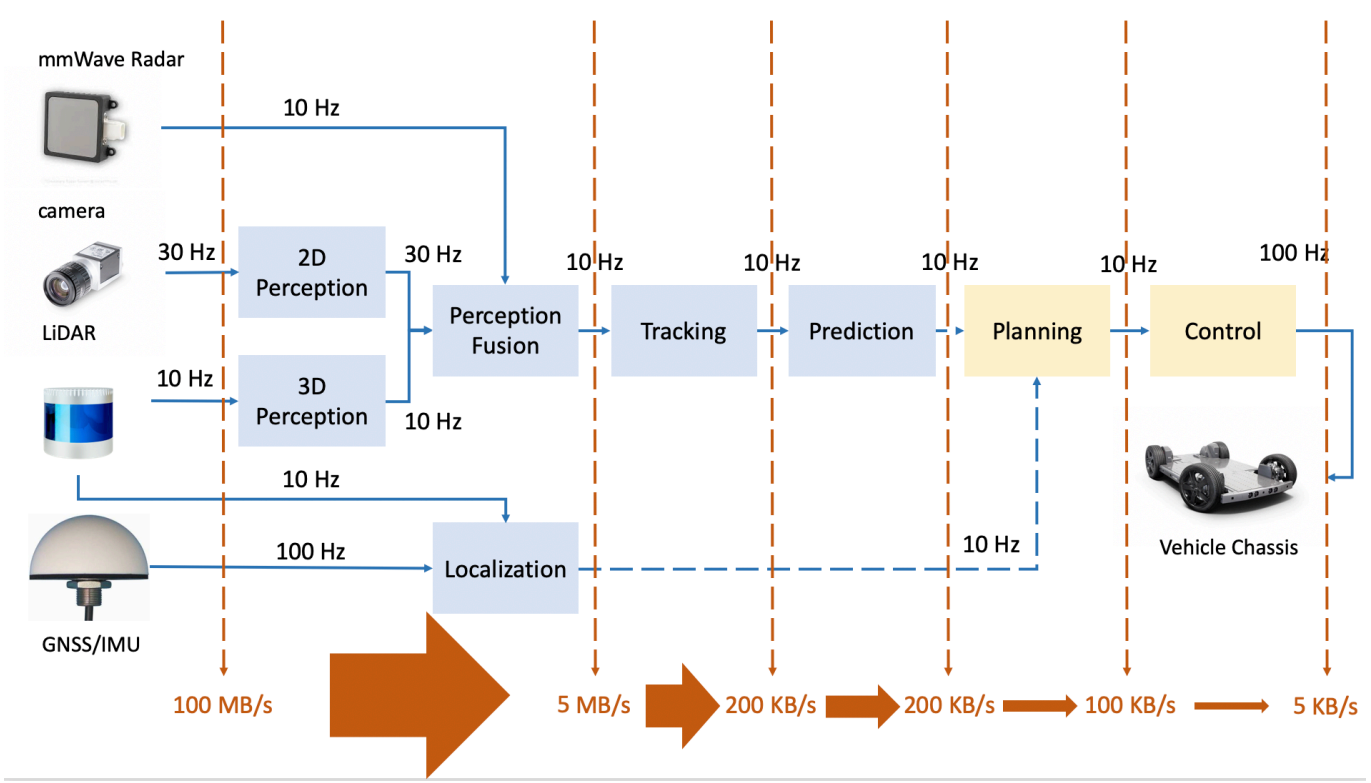
Development of a Proprietary System



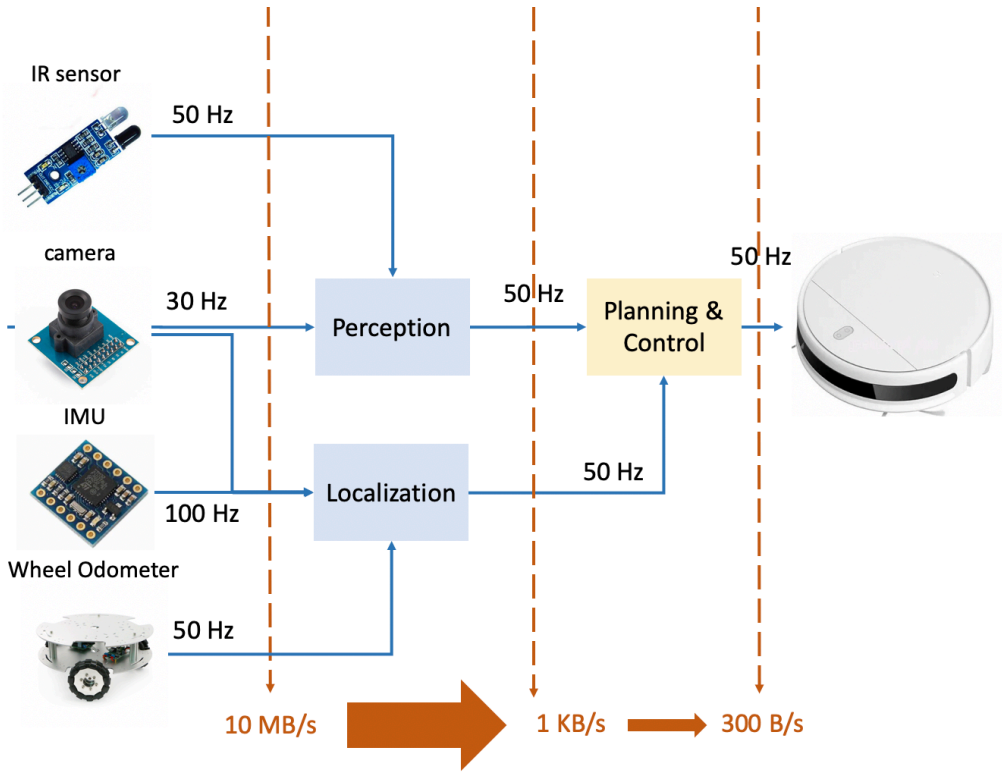
Yu, B., Hu, W., Xu, L., Tang, J., Liu, S. and Zhu, Y., 2020, October. Building the computing system for autonomous micromobility vehicles: Design constraints and architectural optimizations. In 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)

Computation Graphs of Autonomous Machines

a.) Computation Graph of a Level 4 Autonomous Vehicle



b.) Computation Graph of a Robot Vacuum



Liu, S., Zhu, Y., Yu, B., Gaudiot, J.L. and Gao, G.R., 2021. The Promise of Dataflow Architectures in the Design of Processing Systems for Autonomous Machines. arXiv preprint arXiv:2109.07047. https://arxiv.org/abs/2109.07047

Traits of Autonomous Machine Computing

- **Continuous Inputs Provide Abundant Parallelisms**: the sensing data continuously rushes in and exercises the various sensor processing nodes, which in turn generate continuous outputs that further exercise later node.

- **Flexible Dependencies** : a consumer node, while dependent on a producer node, does not have to consume every single piece of producer's output.

- **Deterministic Data Communication**: the data communication volume across nodes is largely deterministic, because each node has a fixed output size and a fixed output frequency.

- **No Loops**: computing graphs are DAGs and the nodes within the DAGs are stateless.

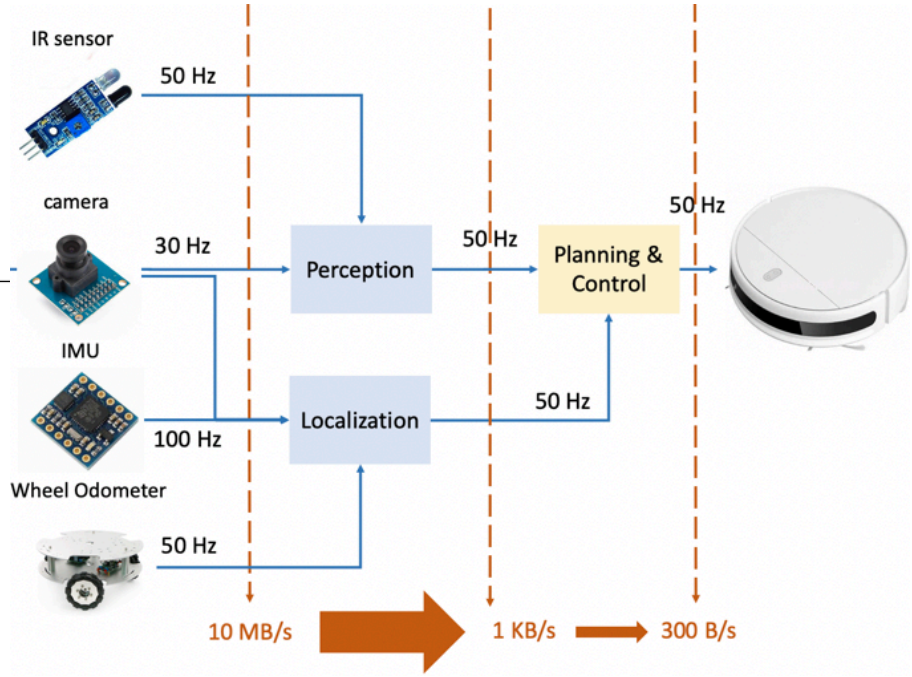
Liu, S., Zhu, Y., Yu, B., Gaudiot, J.L. and Gao, G.R., 2021. The Promise of Dataflow Architectures in the Design of Processing Systems for Autonomous Machines. arXiv preprint arXiv:2109.07047. https://arxiv.org/abs/2109.07047

Dataflow Architecture for Autonomous Machines

Algorithm 1 λ-calculus expression of the computation graph shown in Fig. 1b.)

Require: $IR(frequency >= 50Hz)$ ▷ infrared sensor
Require: $Camera(resolution = 320X240; frequency >= 30Hz)$
Require: $IMU(frequency >= 100Hz)$
Require: $WO(frequency >= 50Hz)$ ▷ wheel odometry sensor
Require: $2DPerception(frequency >= 50Hz)$
Require: $Localization(frequency >= 50Hz)$
Require: $Control(frequency >= 50Hz)$

1: $perc \leftarrow \lambda(IR)\lambda(Camera)\lambda(2DPerception)[2DPerception(IR, Camera)]$
2: $loc \leftarrow \lambda(Camera)\lambda(IMU)\lambda(WO)\lambda(Localization)[Localization(Camera, IMU, WO)]$
3: $cmd \leftarrow \lambda(perc)\lambda(loc)\lambda(Control)[Control(perc, loc)]$



Dataflow + Domain Specific Accelerators

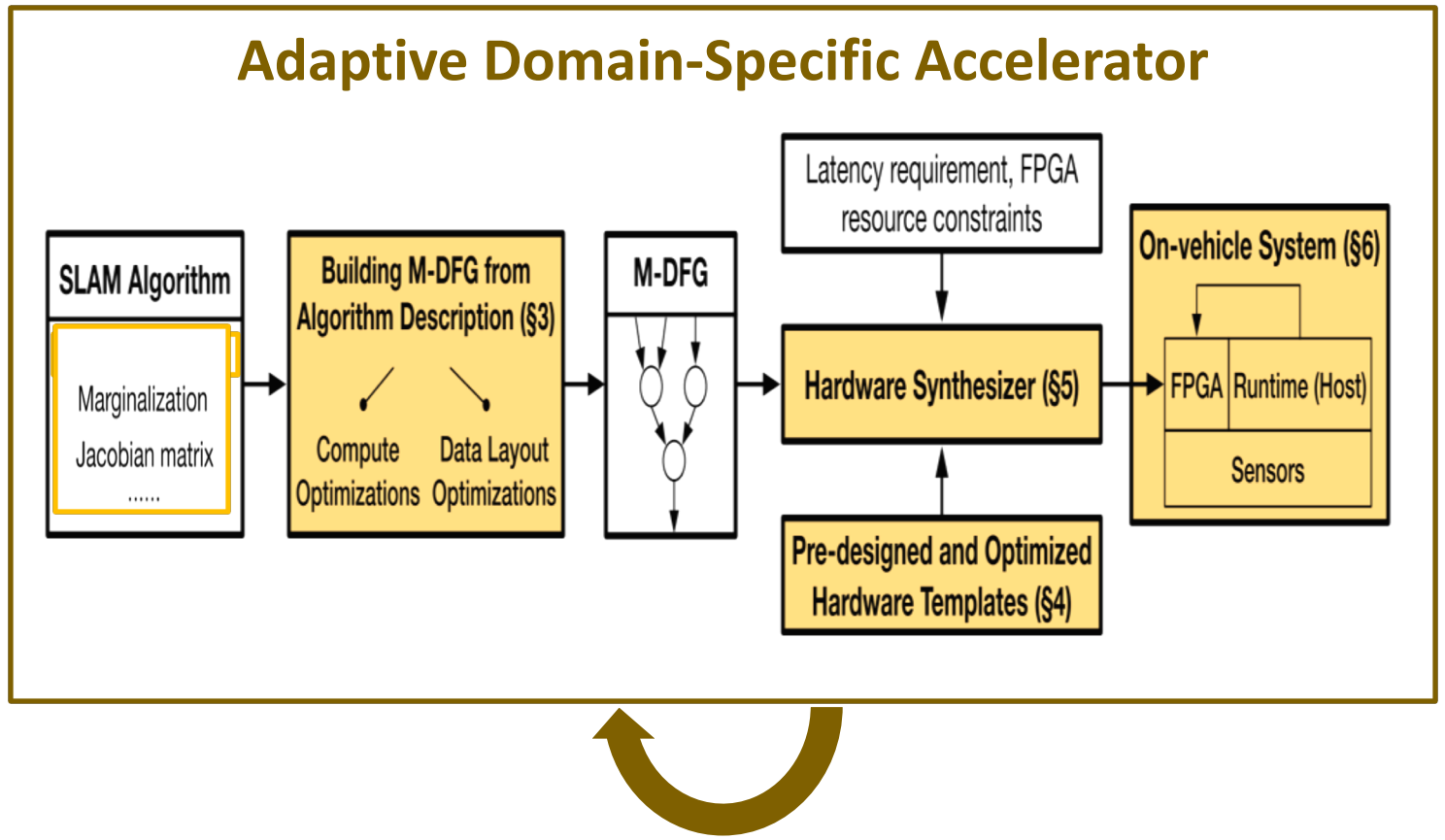
Liu, S., Zhu, Y., Yu, B., Gaudiot, J.L. and Gao, G.R., 2021. The Promise of Dataflow Architectures in the Design of Processing Systems for Autonomous Machines. arXiv preprint arXiv:2109.07047. https://arxiv.org/abs/2109.07047

Dataflow Architecture for Autonomous Machines

1. Software Specifications

2. Hardware Configuration

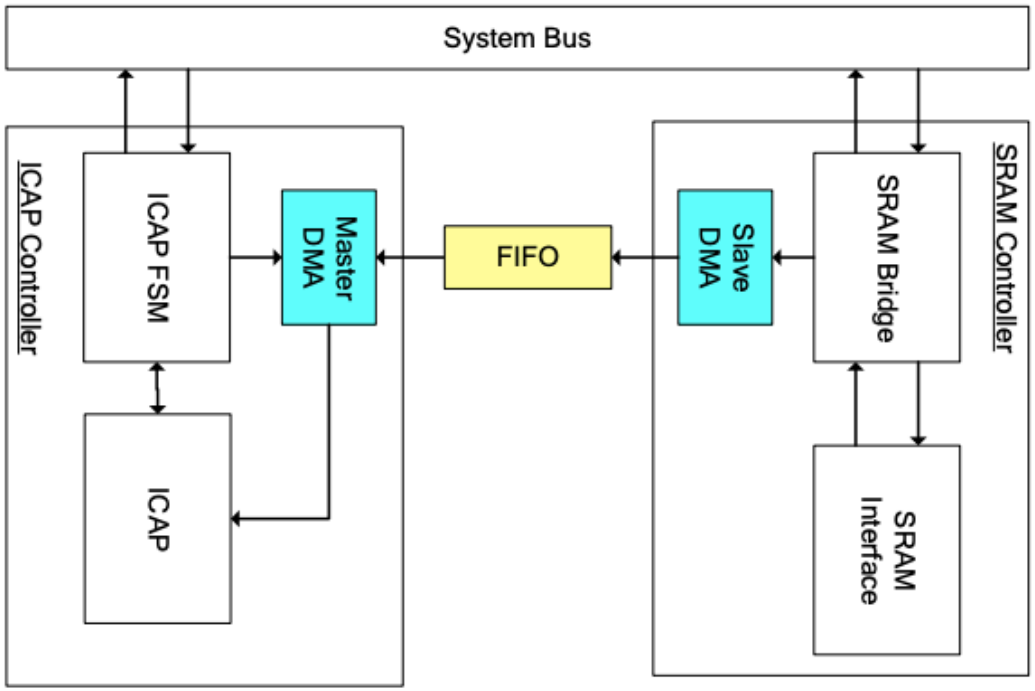
3. Buffer Allocation and Graph Forming

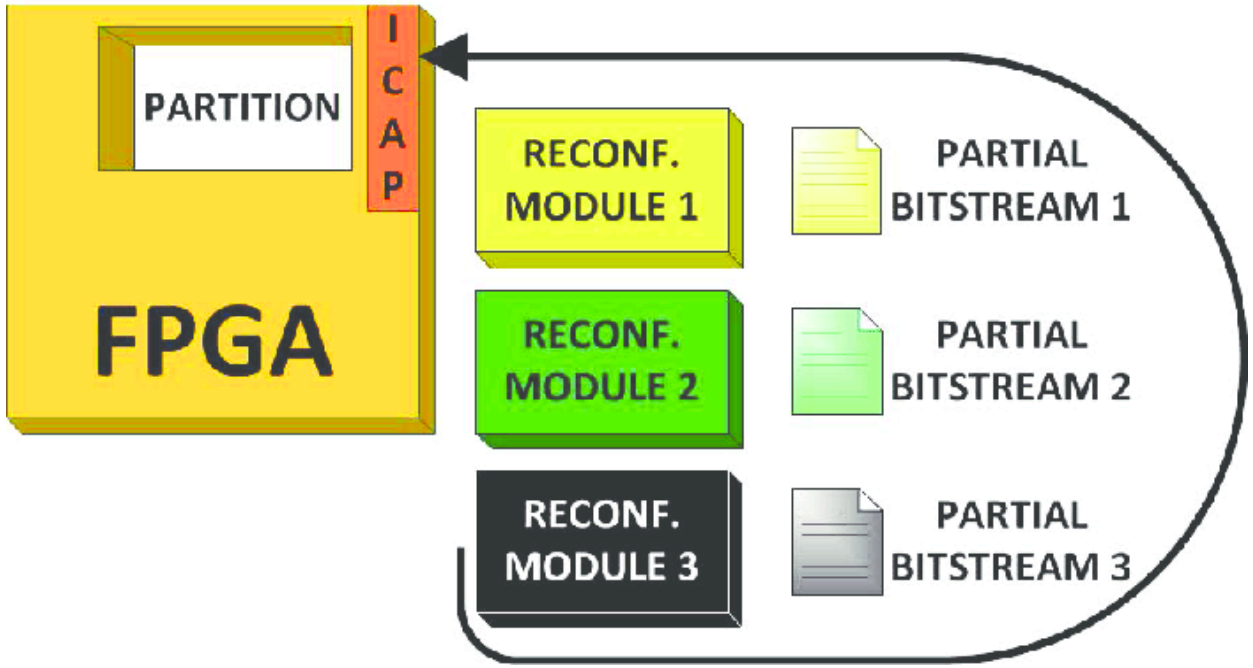


4. Self-Optimization at Runtime

Liu, W., Yu, B., Gan, Y., Liu, Q., Tang, J., Liu, S. and Zhu, Y., 2021. Archytas: A Framework for Synthesizing and Dynamically Optimizing Accelerators for Robotic Localization. In 2021 54th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE.

Partial Reconfiguration





Liu, S., Pittman, R.N., Forin, A. and Gaudiot, J.L., 2013. Achieving energy efficiency through runtime partial reconfiguration on reconfigurable systems. ACM Transactions on Embedded Computing Systems (TECS), 12(3), pp.1-21.

Summary

- FPGAs are the best platform for robotic computing:
  - Robotic workloads are dynamic.
  - Unlike ASICs, FPGAs offer the scalability of field programming and reprogramming without the need to remanufacture by modifying the design.
  - Partial reconfiguration (PR) technology takes this scalability a step further by allowing the running FPGA design to be modified by loading a partial configuration file.
- Software support for FPGAs is too weak now, a software ecosystem like NVIDIA CUDA is required for FPGAs to take off.
- ROS support for heterogeneous computing is also weak, which gives a great opportunity to the latecomers (robotics computing operating systems).
- The new generation of Robotic OS needs to support multiple algorithms and multiple computing platforms. In this regard, the opportunities for Robotic OS are unlimited.

The End