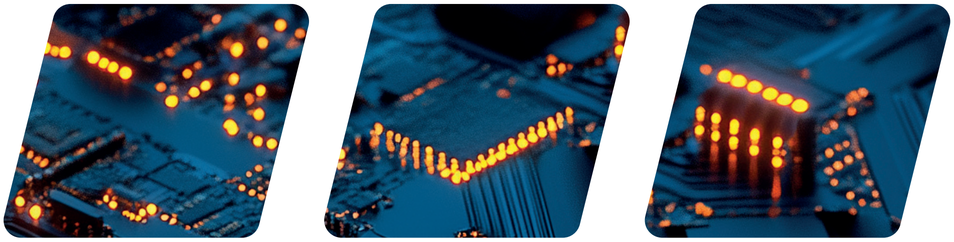
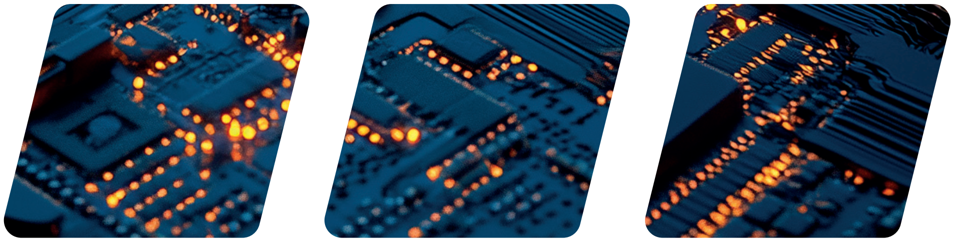


Sensor Hub For Near-Sensor Low-Latency Data Fusion In AI Systems



White Paper

Author:
Hoon Choi, Sr. Fellow, R&D, Lattice Semiconductor

DISCLAIMERS

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. Lattice products and services are not designed, manufactured, or tested for use in life or safety critical systems, hazardous environments, or any other environments requiring fail-safe performance, including any application in which the failure of the product or service could lead to death, personal injury, severe property damage or environmental harm (collectively, "high-risk uses"). Further, buyer must take prudent steps to protect against product and service failures, including providing appropriate redundancies, fail-safe features, and/or shut-down mechanisms. Lattice expressly disclaims any express or implied warranty of fitness of the products or services for high-risk uses. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

INCLUSIVE LANGUAGE

This document was created consistent with Lattice Semiconductor's inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice's inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

ABSTRACT

The FPGA-based sensor hub provided by Lattice Semiconductor benefits the development of smart robots. Its flexible Input/Output (I/O) and parallel computing near the sensors enable the connection with multiple sensors and actuators with low-latency and low-power computing. It offloads sensor-specific, low-level, and real-time computation from the CPU and GPU, allowing them to focus on high-level intelligence. The small form factor, low power consumption, and lack of need for a cooling system (e.g., fan) make Lattice solutions most suitable for robots in factories.

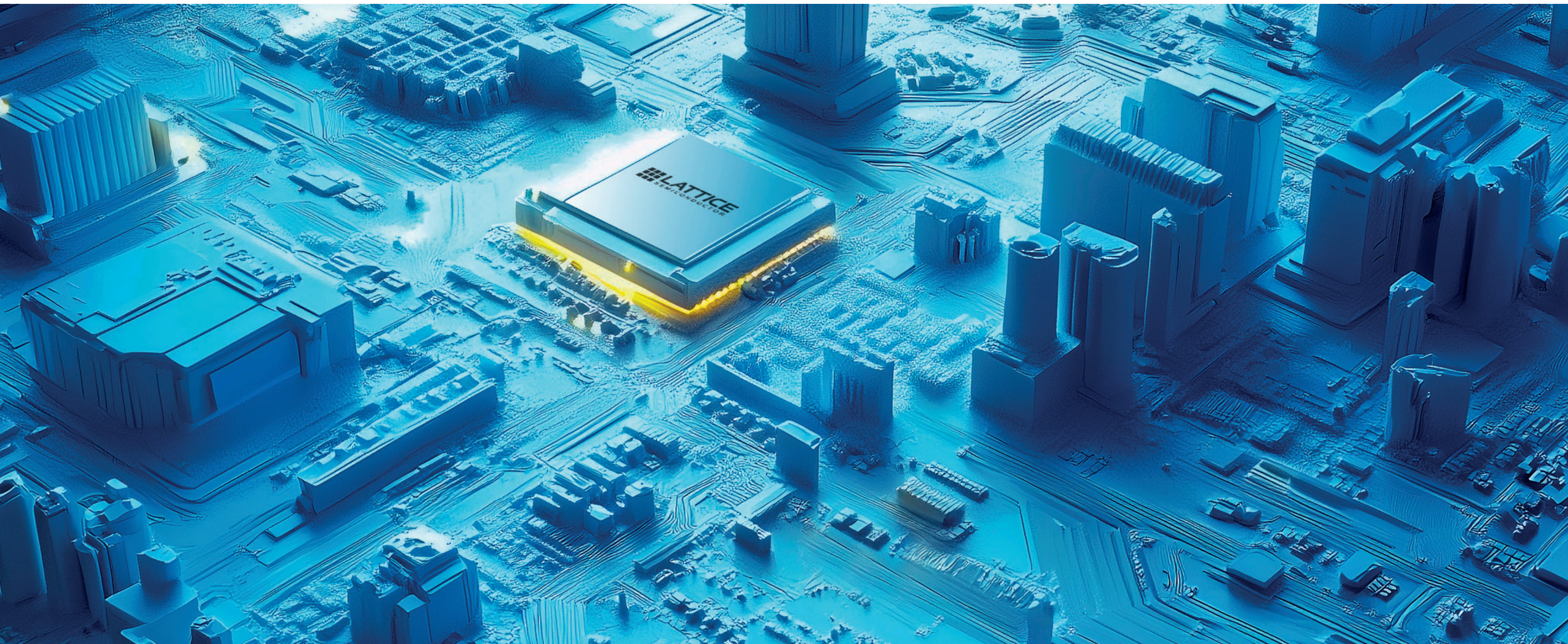


TABLE OF CONTENTS

1. Introduction

Autonomous robots have been working alongside humans in industrial manufacturing for over half a century. Ever since the world's first industrial robot was developed and deployed in the 1950s, businesses have been offloading tedious or dangerous tasks to these machines to free up their workers for more nuanced responsibilities. Today, the use of these robots is no longer limited to the industrial sector; it has expanded into many other verticals, including healthcare, retail, and agriculture.

Moreover, technological advancements — especially in the field of AI and machine learning — have spurred a new generation of even smarter robots that go beyond repetitive tasks to perform far more sophisticated functions. For example, through capabilities like computer vision and autonomous mobility, they can perform everything from product assembly to quality control to advanced threat detection and response.

In short, they have become an indispensable asset to bolster workforces, offering a high level of precision and an unlimited capacity for productivity. However, as businesses demand more from their robotic assistants, these systems are getting exponentially harder to design.

2. Growing Design Challenges

Smart, AI-powered robots require significantly more sensors and actuators than their predecessors, including components like cameras, lidar, radar, inertial measurement units (IMUs), motor encoders, and pressure sensors. They also need to perform much more complex computing tasks — like 3D vision processing, Simultaneous Localization and Mapping (SLAM), and pick point calculations — in real time.

As a result, building these systems requires the use of more I/O to accommodate more sensors, as well as higher performance to perform more advanced computing functions in the processing module (e.g., CPUs, GPUs, or NPUs). The challenge is that these two requirements conflict with each other. To get higher performance, processing modules need to be fabricated using an advanced processing node that makes each transistor even smaller, thus more transistors can be integrated. However, in such an advanced processing node, the design of I/O becomes more challenging. For example, the size of I/O is not shrinking even in the advanced processing node, thus it becomes relatively expensive. In addition, supporting multiple voltages in the I/O becomes more challenging in the advanced node. As a result, I/O in the high-performance computing modules becomes consolidated to support only the standard high bandwidth I/O such as PCIe, Gigabit Ethernet, etc. instead of supporting low-bandwidth, general, non-standard I/O. This makes it often unfeasible for designers to rely solely on processing module to interface with the multitudes of sensors required in these systems.

Additionally, even if a computing module does provide enough or the right I/O for smart robot connectivity, transmitting high volumes of raw sensor data directly to the computing module is not energy efficient. Furthermore, the computing module like CPUs and GPUs are not optimized for the real-time processing that smart robots require. For example, delegating critical tasks like sensor fusion to CPUs would introduce latency to these systems and significantly slow down their operations.

Thankfully, hardware designers and developers are focused on innovating products that help bridge these gaps.

3. FPGAs: A Valuable Hardware Solution

Field Programmable Gate Arrays (FPGAs) — flexible semiconductors that act as a “bridge” between sensors, actuators, and CPUs — are one such product. Thanks to their real-time, near-sensor computing capabilities, FPGAs can take on low-level, sensor-specific tasks to support the smarter, more responsive robots that businesses need.

Once the data has gone through a first layer of processing, it's then transferred to the CPU via a standard, high bandwidth channel. By partitioning the processing tasks of smart robots in this way, FPGAs take some of the processing load away from the CPU to save energy for higher-level computations like trajectory planning or clustering and object detection. This allows the CPU to focus on the types of optimization and decision tasks that are hard to perform at the hardware level.

For example, this setup also helps developers overcome challenges related to:

- **Connectivity:** FPGAs offers more I/O that are highly customizable in protocols and voltage levels. This allows developers to connect and control more sensors and actuators using I/O like Ethernet, SPI, HDMI, MIPI, I2C, UART, and even GPIOs to support non-standard protocols with cycle-by-cycle accuracy. This expands developers' options to meet the needs of different applications.

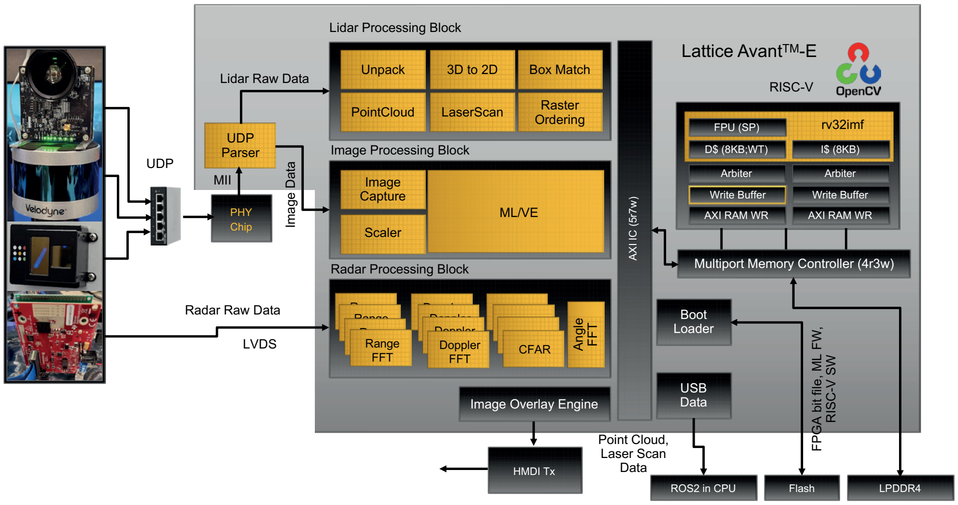
- **Power:** FPGAs provide hardware-based parallel processing close to robotic sensors. By performing real-time local computing to process sensor data and then sending the processed data to the computing module (like a CPU), they help reduce the system's energy consumption.
- **Latency:** FPGAs significantly speed up critical tasks like sensor fusion, which combines data from various sensors (e.g., cameras and lidar) to create a comprehensive view. This enhances the accuracy and decision-making of robots. For instance, imagine 384 pieces of distance data from a VLP16 lidar sensor being transmitted over the network every 1.32 milliseconds. An FPGA processes this data in just 0.32 milliseconds, operating at a speed of 100 million operations per second.

These benefits give designers the flexibility to integrate numerous and diverse sensors into intelligent robots, pushing their capabilities while managing power and latency constraints.

4. The Lattice Solution

To demonstrate the capabilities of the sensor HUB using a Lattice FPGA, we created a proof-of-concept (PoC) demo. This demo processes data from a camera, lidar, and radar simultaneously within a single Lattice Avant™ FPGA. Figure 1 shows the block diagram of the PoC.

Figure 1: Block Diagram: Proof-of-Concept Demo for Sensor HUB Capabilities in a Lattice Avant FPGA



The camera data and lidar data come over Ethernet as UDP packets to the FPGA. The lidar data is processed by the lidar processing block where the UDP packet is unpacked to extract the distance information for each laser. The distance information is converted into point cloud information using mathematical calculation. This mathematical calculation uses FP32 floating point numbers, and we generated RTL from the reference C code using HLS (High Level Synthesis) tool. The point cloud is mapped into a 2D plane with a given input viewpoint to show on the output screen. This portion is also generated from the reference C code using the HLS tool. The box matching block is for the sensor fusion with the machine learning result from the camera sensor.

The image data from the camera is captured and scaled down for the machine learning-based human detection processing. The ML/VE block executes the neural network that detects the human upper torso and emits the bounding boxes. This was implemented using the sensAI solution stack.

For the radar data, the ADC output from the radar comes through the LVDS bus. In this PoC, the radar was set to 4 channels, and each channel's data goes through range FFT, doppler FFT, CFAR, and angle FFT. The output is the list of coordinates and speed of multiple objects. In the development, we used Matlab to tune the radar parameters, and we used real data captured from the radar to validate the algorithms and parameters in Matlab. After verifying the parameters, we used Simulink HDL flow to generate the FFT and other processing blocks.

The bounding box output from the human detection neural network is fused with the lidar and radar output for better accuracy and decision. We calculate the section of the lidar point cloud that corresponds to each of the detected bounding boxes (i.e., humans). From this, we can calculate the actual distance to each human. Similarly, we identify the radar dots corresponding to the human, and it gives us the speed of the human and more confidence that it's a real human, not a human-scaled picture. All this information goes to the RISC-V CPU for post-processing and user interface using OpenCV. See Figure 2.

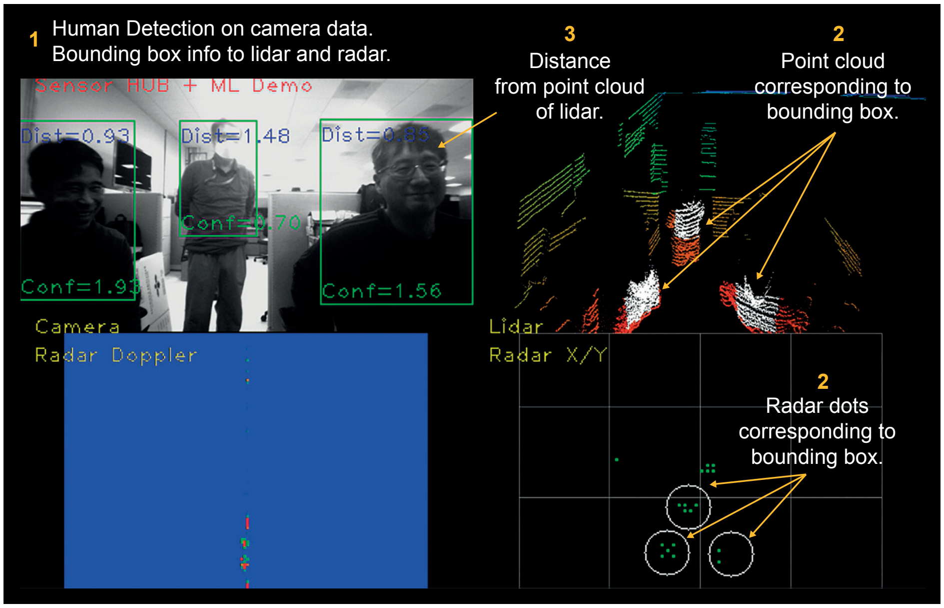
This sensor fusion enables many smart applications. For example, in a surveillance system, instead of running the camera and AI/ML all the time, radar first detects any moving objects, and only after that do we run the AI/ML for the camera data. This reduces the power consumption coming from AI/ML computation. Another example is the virtual safety fence in the factory. The conventional radar-based safety fence sensor needs to understand the environment to filter out the reflection noise coming from the structure around the sensor. Otherwise, it's hard to identify the signals from the human among the reflection noises from other structures. Here, the use of AI/ML-based human detection using the camera helps. The output from the human detection neural network gives the region that is occupied by a human (i.e., region of interest), and we can focus on the radar data only in that region. This eliminates the cumbersome process of static reflection noise measurement and filtering. Even if the environment changes, the AI/ML-based ROI detection can handle the cases.

The processing time of lidar and radar is as follows:

- For VLP16 lidar that sends 384 distance information in one UDP at every 1.32 ms, we used a single processing engine (generated using HLS), and it took 0.32 ms for each packet at a 100 MHz clock. If we need more throughput, we can use multiple processing engines in parallel.

- For TI FMCW 2243 radar, we set it to send 1 frame of data every 40 ms for 4 channels. The processing time when we used one processing engine (i.e., 4 channels were processed one by one in serial) was 6.5 ms. If we add more processing engines running parallel for each channel, it can be reduced to 1.6 ms at a 100 MHz clock.

Figure 2: Fusion of Bounding Box Output from Human Detection Neural Network with Lidar and Radar Data



Link to video: https://www.youtube.com/watch?v=Lq9fiztXBAo

5. Conclusion

The FPGA-based sensor hub from Lattice provides benefits for the development of smart robots. Its flexible I/O enable the connection and aggregation of multiple sensors and the control of multiple actuators. The hardware-based parallel computing near the sensors provides low-latency and low-power computing and offloads sensor-specific, low-level, real-time computation from CPUs/GPUs/NPUs. This makes it possible to reduce the BOM by using low-end computing modules or enhance functionality by putting more high-level functions into the computing modules instead of doing low-level computing. The development flow can use HLS, Matlab/Simulink, and Hardware-in-the-loop along with optimized RTLs. Last but not least, the small form factor, low power consumption, and no need for a cooling system (e.g., fan) make the Lattice solution most suitable for robots in dusty factories and smart sensors inside plastic enclosures.

READY TO LEARN MORE?

To learn more about Lattice low power FPGA-based solutions for industrial, automotive, communications, computing, and consumer applications, visit **www.latticesemi.com** or contact us at **www.latticesemi.com/contact** or **www.latticesemi.com/buy**.

TECHNICAL SUPPORT ASSISTANCE

Submit a technical support case through **www.latticesemi.com/techsupport**.
For frequently asked questions, please refer to the Lattice Answer Database at **www.latticesemi.com/Support/AnswerDatabase.**

V.1 WP0044