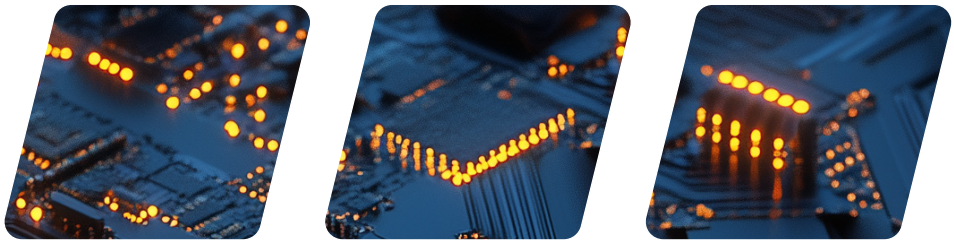
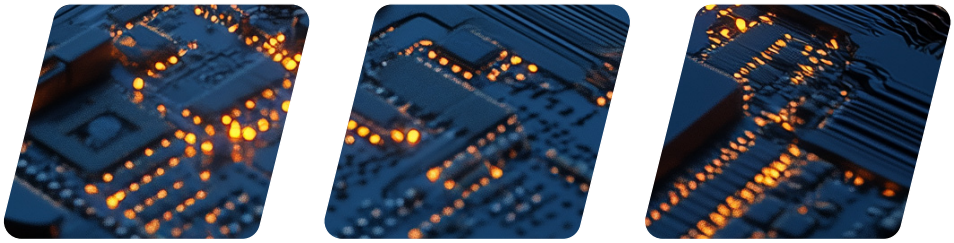


Hardware-Synchronized VSLAM with ADI IMUs and the Lattice Holoscan Sensor Bridge Solution



White Paper

Authors:

Deepak Boppana, Senior Director, Segment Marketing, Lattice Semiconductor
Jay Shu, Director, Systems Engineering, Analog Devices, Inc.

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

Visual Simultaneous Localization and Mapping (VSLAM) algorithms require microsecond-level synchronization between imaging sensors and Inertial Measurement Units (IMUs) to help prevent trajectory drift. This paper details a hardware-in-the-loop synchronization architecture that utilizes the Lattice Semiconductor Holoscan Sensor Bridge (HSB) solution and Analog Devices, Inc. (ADI) precision IMUs. By leveraging the IEEE 1588 Precision Time Protocol (PTP) and an FPGA-based Sync Pin Generator, the system synchronizes both the camera's exposure window and the IMU's Scaled Sync Mode to a unified, phase-aligned hardware pulse. Furthermore, the paper outlines a two-stage software pipeline that is designed to preserve these hardware-level UNIX epoch timestamps through the ROS 2 middleware, helping to bypass OS-level software jitter and deliver highly synchronized, time-aligned sensor data to the SLAM algorithm. This synchronization architecture is well suited for robotics that depend on precise visual-inertial sensor fusion for perception, localization, mapping, and navigation.

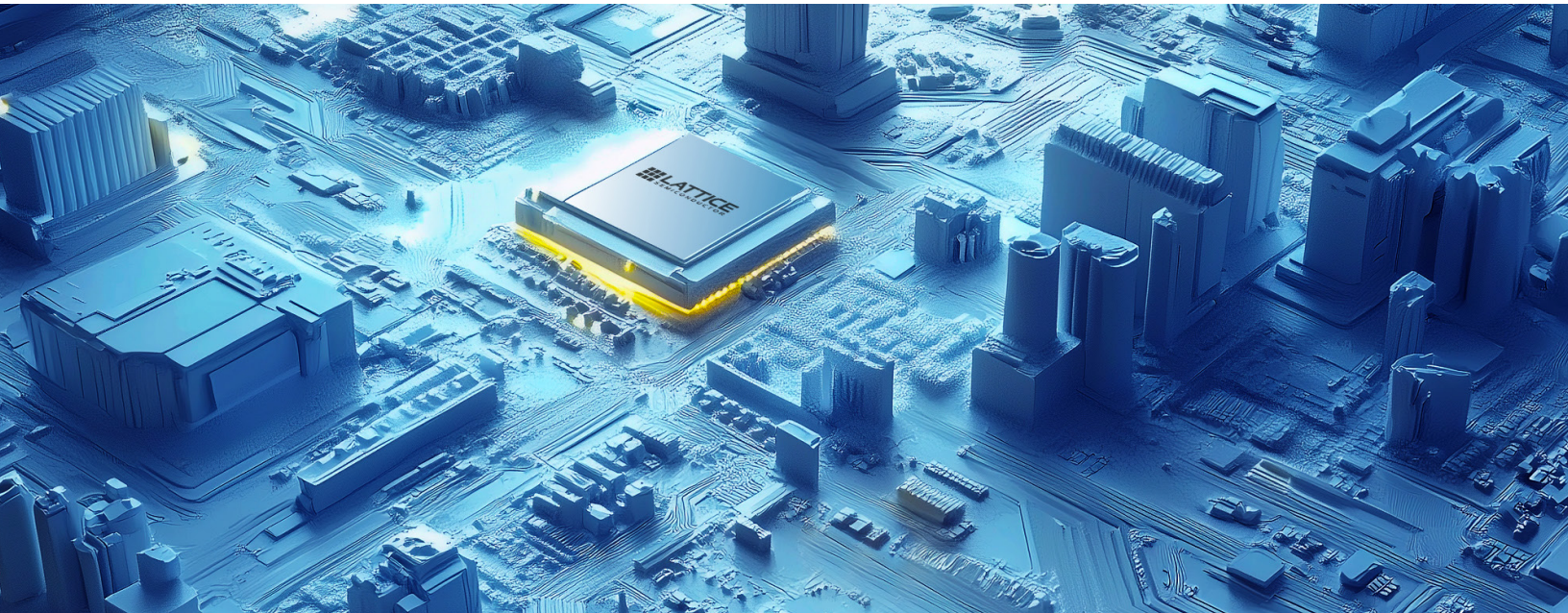


TABLE OF CONTENTS

The Visual-inertial Timing Challenge

In VSLAM systems, the mathematical fusion of optical data from cameras and inertial data from IMUs largely determines the accuracy of the spatial graph. A fundamental source of error in these systems is temporal misalignment. Standard cameras operate at discrete, low-frequency intervals (for example, 30 FPS or 60 FPS), while precision IMUs sample at much higher frequencies. If these two systems rely on independent, free-running oscillators, their clocks can drift over time. Even a temporal offset of a few milliseconds between a camera's exposure window and the corresponding IMU acceleration vector can result in significant visual-inertial drift, causing the SLAM algorithm to map straight lines as curves or potentially fail to accurately estimate trajectories during highly dynamic motion. To achieve tightly coupled VSLAM, both the imaging sensor and the IMU should be synchronized to a unified, phase-aligned external hardware trigger rather than relying solely on their internal clocks.

The Absolute Timing Foundation: PTP and 1PPS

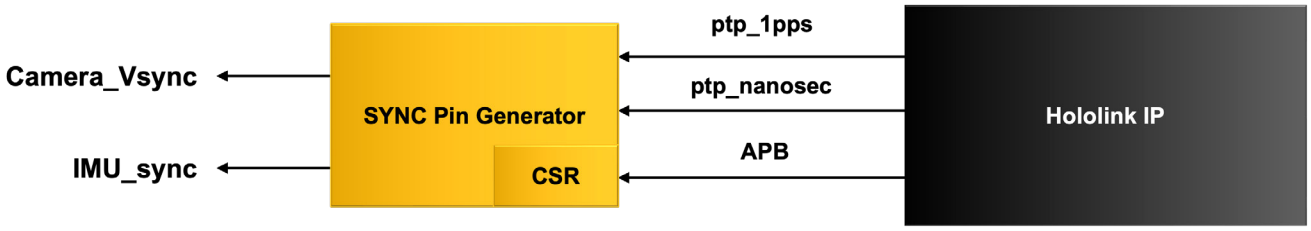
In distributed edge AI systems such as NVIDIA Holoscan implementations, sensor synchronization is established over the network using the Precision Time Protocol (PTP, IEEE 1588v2). PTP is designed to align the clocks of the host compute module and the Lattice HSB solution with sub-microsecond synchronization accuracy. Once network time is synchronized, the HSB derives a highly stable 1 Pulse-Per-Second (1PPS) hardware signal. This 1PPS signal serves as the absolute phase reference for all subsequent high-frequency triggers generated within the system.

Lattice FPGA-based SYNC Pin Generator

The Lattice Holoscan Sensor Bridge solution employs a dedicated FPGA-implemented Sync Pin Generator that is designed to produce deterministic, phase-aligned trigger pulses to unify the timing domains of all connected sensors. Unlike software timers, OS-scheduled events, or general-purpose PWM peripherals, which inherently suffer from jitter and non-deterministic execution, the FPGA derives all trigger outputs directly from a PTP-locked hardware time base. This hardware-driven architecture is designed to provide sub-microsecond alignment between imaging devices and ADI precision IMUs, even under highly dynamic operating conditions.

The synchronization sequence begins with the FPGA operating from a PTP-synchronized system clock. The first trigger pulse is explicitly aligned to the rising edge of the 1PPS signal. Subsequent pulses are generated using a user-configured synchronization period derived from this reference. By locking its internal counters to the PTP nanosecond-level timestamping domain, the FPGA is designed to help ensure that every emitted pulse, whether for a camera frame trigger or an IMU sampling event, maintains a consistent and repeatable phase relationship with the global timing infrastructure. See Figure 1.

Figure 1: FPGA Sync Pin Generator block diagram with PTP timing inputs and camera/IMU synchronization outputs



Within the FPGA, frequency-division logic and phase-coherent pulse-synthesis blocks translate the 1PPS-derived time-base into two synchronized hardware trigger signals:

Camera VSYNC Pulse Train

A low-frequency pulse sequence (for example, 30 Hz or 60 Hz) is generated to match the desired camera frame rate. This pulse is routed to the camera's external trigger input, where it serves as the primary Vertical Synchronization (VSYNC) signal. Each rising edge is designed to define the start of the global-shutter exposure interval, helping to eliminate inter-frame drift and maintain timing consistency throughout long-duration captures.

IMU External SYNC Pulse Train

In parallel, the FPGA synthesizes a higher-frequency pulse train that is typically an integer multiple of the camera frame rate and matched to the target IMU output rate.

Because both pulse trains originate from a common hardware counter and share the same 1PPS and PTP nanosecond reference, the camera and IMU remain aligned to the same global epoch. This eliminates timing drift, even during extended runtime or under aggressive motion, where software-synchronized architectures may fail to maintain sensor coherence. The FPGA Sync Pin Generator exposes a set of configuration registers that allow users to tailor trigger behavior to specific sensor and system requirements. These controls include:

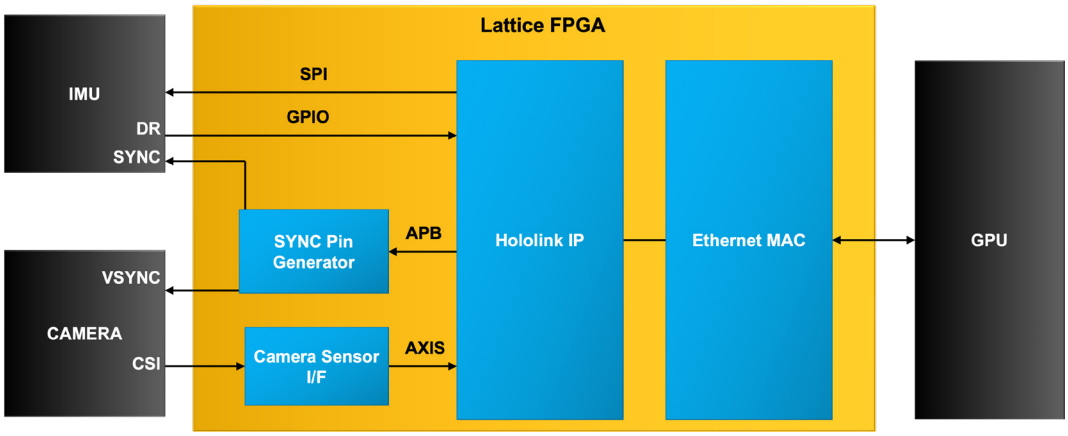
- Sync pulse timing configuration
- Programmable start delay for initial pulse alignment
- Sync clock multiplier for IMU trigger rate generation

These settings provide precise control over trigger cadence, alignment, and sensor-specific timing constraints while retaining the deterministic behavior inherent in the hardware-based synchronization pipeline.

ADI IMU Scaled Sync Mode & Lattice FPGA Hardware Polling

ADI precision IMUs, including the ADIS16505 and ADIS16607, feature a dedicated synchronization architecture designed for this topology. See Figure 2.

Figure 2: System-level architecture of the Lattice FPGA–based sensor interface, illustrating synchronized IMU and camera connectivity.



When the IMU receives an external clock through its SYNC pin, it enters Scaled Sync Mode and locks to the pulse train generated by the HSB FPGA. Each rising edge on the SYNC input triggers the IMU to capture the current inertial state, process the measurement, and assert its Data Ready (DR) signal.

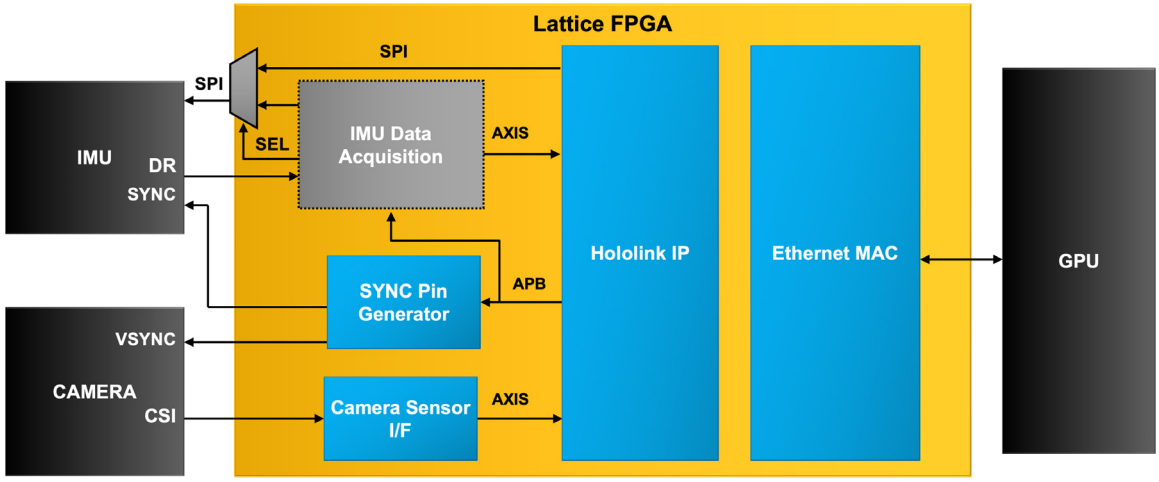
The IMU subsequently captures inertial samples on the rising edges of the PLL-generated scaled clock, helping ensure that the resulting data remains phase-locked to the FPGA's reference clock.

A key feature of this architecture is the hardware burst-read capability available in the ADIS16505 and ADIS16607 IMUs. Using a single optimized SPI transaction, the complete multi-axis inertial and temperature data set is transferred as one contiguous frame. This burst-read architecture minimizes SPI bus utilization and provides the foundation for moving data acquisition entirely into the FPGA fabric.

Future Architectural Optimization

A planned enhancement to the synchronization architecture is the migration of IMU DR handling from software to FPGA hardware. In the current implementation, a high-frequency CPU thread continuously polls the IMU's DR signal to determine when a synchronized inertial sample is available and then initiates the SPI burst read. While effective, this software-driven approach consumes CPU resources and can introduce small but measurable variations in read latency. See Figure 3.

Figure 3: System Architecture of the Lattice FPGA-based Sensor Interface for Synchronized IMU and Camera Operation



In the enhanced architecture, the FPGA assumes full responsibility for the DR handling pipeline, enabling a fully hardware-driven data acquisition process. Under this approach, the FPGA autonomously:

- Detects DR assertions using dedicated hardware edge-detection logic
- Initiates the SPI burst read immediately after DR activation, eliminating software-induced delays
- Buffers a user-configurable block of samples in on-chip memory to support efficient batch transfers
- Streams the buffered data to the host through the HSB's AXI-Stream (AXIS) interface
- Generates a host interrupt after the data block has been transferred to system memory

By removing CPU polling from the critical timing path, this hardware-accelerated architecture reduces latency variation and is designed to approach the minimum achievable sensor-to-software transfer time. The result is a more deterministic timing model that helps support the demanding synchronization requirements of high-precision VSLAM and real-time sensor fusion applications.

Software Architecture: Preserving Hardware Determinism

To bring highly deterministic hardware data into the robotics software stack without introducing latency or execution jitter, the system employs a two-stage Holoscan application graph.

Stage 1: C++ Holoscan Hardware Operator

At the hardware interface layer, a custom C++ Holoscan operator (adi_imu_op) manages communication with the HSB and processes raw IMU data before it enters the software pipeline.

- Architecture-Specific SPI Bursting: The operator automatically selects the appropriate driver based on the target IMU. For the ADIS16505, it performs a standard 32-bit full-duplex SPI burst transaction. For the ADIS16607, it uses an optimized 34-byte half-duplex transfer sequence designed for that device's architecture.

- Data Alignment and Checksum Validation: Because the ADIS16607 combines pairs of 16-bit registers to create 32-bit words containing 24-bit precision measurements, the C++ operator performs several processing tasks before forwarding the data. These include validating the device's 16-bit checksums and applying a 32-to-24-bit arithmetic right shift ton preserve the two's-complement sign information.
- Hardware Timestamping: The most critical function of the operator is preserving timing integrity. At the moment the hardware read is initiated, the operator queries the system's PTP-synchronized clock and appends a UNIX epoch timestamp to the sensor payload. This enables downstream applications to evaluate measurements based on when they were physically captured rather than when they were processed by software. As a result, subsequent software latency has minimal impact on time-correlation accuracy.

Stage 2: The Python ROS 2 Publisher Node

The C++ operator passes the timestamped, properly aligned sensor data to a native Python Holoscan operator, which serves as the bridge between the Holoscan framework and the ROS 2 ecosystem.

- Data Scaling and Calibration: With the C++ layer handling low-level processing such as bit shifting and sign extension, the Python operator can focus on data conversion and calibration. It applies datasheet-defined scaling factors to convert raw IMU measurements into physical units, including angular velocity (rad/s) and acceleration (m/s^2). The node can also apply user-defined calibration offsets and local gravity constants as needed.
- ROS 2 Integration: The Python operator populates standard sensor_msgs/Imu and sensor_msgs/Temperature messages for use within ROS 2 applications. To preserve timing accuracy, it replaces the standard ROS now() timestamp with the hardware-level timestamp generated by the C++ operator, ensuring the measurements reflect the actual acquisition time.
- CycloneDDS Transport: To support high-frequency sensor streams without unnecessarily increasing network traffic, the node uses a Best Effort Quality of Service (QoS) policy with the rmw_cyclonedds_cpp middleware. This configuration is designed to provide low-latency data delivery to downstream applications, including SLAM algorithms, sensor-fusion pipelines, RViz2, and Madgwick filter implementations.

High-frequency Transport and ROS 2 Compute Boundaries

Although ADI precision IMUs support extremely high sampling rates, up to 2,000 Hz for the ADIS16505 and 8,000 Hz for the ADIS16607, processing and transmitting individual messages at these rates can place significant demands on a robotics software stack. Publishing thousands of ROS 2 messages per second increases the overhead associated with serialization, deserialization, middleware processing, and context switching, potentially reducing the compute resources available for SLAM and other real-time applications.

To address this challenge, the system leverages the IMU's Digital Phase-Locked Loop (DPLL). The IMU continuously samples the physical environment at its maximum native rate, helping preserve signal fidelity and reduce the risk of aliasing. It then applies hardware-based decimation filtering to produce a lower output rate that is better suited for VSLAM workloads, typically in the 200 Hz to 400 Hz range. The HSB FPGA is configured to drive the IMU's SYNC pin at this optimized output frequency, ensuring that the inertial data remains synchronized with the camera while keeping data rates manageable for the ROS 2 ecosystem. As a result, SLAM applications receive accurate, phase-aligned inertial measurements without overwhelming system resources.

Conclusion

Accurate and reliable VSLAM performance depends on sensor data that remains precisely synchronized over time. By combining the Lattice Holoscan Sensor Bridge solution with ADI precision IMUs, this architecture delivers a deterministic, hardware-based timing foundation designed to minimize drift and improve system reliability. PTP-locked FPGA triggers provide sub-microsecond synchronization between camera exposures and IMU sampling, while the custom Holoscan software pipeline preserves precise timing information through ROS 2 to support more accurate perception, localization, and navigation. As IMU data acquisition is further integrated into the FPGA fabric, the architecture is expected to deliver even lower jitter and greater system determinism, helping developers build scalable autonomous platforms with the timing precision required for next-generation robotics and edge AI applications.

READY TO LEARN MORE?

To evaluate the FPGA solution, visit the Lattice Holoscan Bridge Sensor Interfaces Reference Design webpage or contact your Lattice representative. For more information on ADI IMU technology, visit the IMU sensors webpage or contact holo.scan@analog.com.

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.**

© 2026 Lattice Semiconductor Corporation and affiliates. All rights reserved. Lattice Semiconductor Corporation and the Lattice Semiconductor logo are trademarks and/or registered trademarks of Lattice Semiconductor and affiliates in the U.S. and other countries. Analog Devices, Inc., A²B, and the ADI logo are trademarks and/or registered trademarks of Analog Devices, Inc. in the U.S. and other countries. Other company and product names may be trademarks of the respective owners with which they are associated.

V.1 WP0056