DOI: 10.35598/mcfpga.2026.005

Synchronization Methods for FPGA-Based Embedded Digital Twins

Kamelia Cinque
ORCID 0009-0001-7409-0948
Department of Computer Technologies in Publishing and Printing Processes
Institute of Printing Art and Media Technologies in Lviv Polytechnic National University
Lviv, Ukraine
kameliia.chinkue.ki.2023@lpnu.ua

***Abstract*— Digital Twin (DT) technology, when applied to FPGA-based embedded systems, opens a concrete engineering path toward tighter integration between physical sensor hardware and its runtime software representation in IIoT deployments. This paper examines the architectural requirements for building a functional digital twin on Intel/Altera FPGA platforms—specifically the Cyclone and Arria families—where deterministic execution and hardware reconfigurability create both opportunities and constraints absent from microcontroller-based alternatives. Three synchronization challenges receive detailed treatment: hardware-level timestamp acquisition, latency-bounded state propagation across clock domains, and behavioral model consistency under partial reconfiguration. A three-layer architecture separating physical acquisition, transport, and virtual state representation is developed and evaluated against the capabilities of the target hardware.**

Keywords—digital twin, FPGA, embedded systems, state synchronization, Intel Cyclone, Arria, IoT, smart sensors, real-time systems, hardware description language

I. Introduction

The concept of a Digital Twin, originally introduced in the context of product lifecycle management, has progressively gained relevance in the field of embedded systems engineering. A Digital Twin can be defined as a dynamic virtual representation of a physical entity that reflects its state, behavior, and operational context in near real time [1]. Unlike purely static simulation models, a DT continuously receives data from the physical system and updates its internal state accordingly.

FPGA-based embedded systems present both unique opportunities and distinctive challenges for DT implementation. On the one hand, FPGAs offer deterministic execution, reconfigurability, and high-throughput parallel processing—properties that are well aligned with the real-time data acquisition requirements of a DT synchronization pipeline. On the other hand, the absence of a unified abstraction layer between hardware description and software modeling complicates the construction of a consistent behavioral model that can serve as the digital counterpart of the physical device.

In the domain of smart sensor networks and IoT, this problem is particularly relevant. Sensor nodes must report state changes reliably and within bounded time windows, while the DT must remain coherent even under variable communication conditions. Intel/Altera platforms—specifically the Cyclone and Arria families—are widely used in industrial sensor applications due to their low power profiles and rich peripheral interfaces, making them an appropriate focus for this investigation.

This paper addresses three research questions: (1) What architectural layers are required for a viable DT of an FPGA-based sensor node? (2) What synchronization mechanisms are necessary to maintain state consistency between the physical device and its digital representation? (3) What design constraints must be considered when targeting Intel Cyclone and Arria platforms?

II. Background and Related Work

Digital Twin research has developed largely within industrial automation and structural health monitoring. Grieves and Vickers [1] provided an early formal definition, and Tao et al. [2] extended it into a five-dimensional model covering the physical entity, virtual entity, services, data, and their interconnections. Both frameworks target manufacturing systems and do not address the resource constraints characteristic of embedded FPGA hardware.

Embedded DT implementations published to date have generally relied on microcontroller platforms communicating over MQTT or CoAP to cloud-hosted models. FPGA-specific architectures have received considerably less attention. The closest related work involves FPGA-accelerated simulation of physical processes [3] and hardware-in-the-loop (HIL) configurations, which share structural similarities with DT pipelines but serve a different purpose: HIL is a validation method, whereas a DT maintains a live state model that can feed back into the physical system. The directional difference is architecturally significant.

Intel Platform Designer and the NIOS II soft processor provide the infrastructure for communication subsystems within Cyclone and Arria devices, and both are relevant building blocks for the data acquisition and forwarding stages of the proposed architecture.

III. Proposed Three-Layer Synchronization

The architecture separates DT functionality into three layers defined by execution location and timing requirements.

Layer 1: Physical Acquisition Layer

This layer is implemented directly in HDL on the FPGA fabric. It consists of sensor interface logic (SPI, I2C, or LVDS depending on sensor type), a timestamping unit driven by a high-resolution counter, and a lightweight packetization module that formats acquired samples into fixed-size frames. Hardware-level timestamping is essential for synchronization accuracy, as it eliminates the jitter introduced by software interrupt handling.

Layer 2: Communication and Transport Layer

Implemented as a combination of soft-processor firmware (NIOS II or equivalent) and a dedicated DMA controller, this layer is responsible for buffering acquisition frames, performing protocol encapsulation (e.g., lightweight MQTT-SN or custom UDP framing), and transmitting data to the DT host over Ethernet. Latency budgets at this layer must be characterized per deployment target to ensure that the *DT state age*—the elapsed time since the last synchronization event—remains within acceptable bounds for the monitored process.

Layer 3: Virtual State Model Layer

This layer resides outside the FPGA and represents the digital counterpart of the physical node. It receives timestamped frames from Layer 2, updates an internal state vector that mirrors the current register and I/O state of the physical device, and optionally executes a behavioral model to predict future states during communication gaps. The model can be expressed in SystemC for cycle-accurate simulation or in a higher-level language when timing precision requirements are relaxed.

The interface between Layer 2 and Layer 3 defines the synchronization boundary and determines the minimum acceptable communication bandwidth. For slowly varying environmental sensors, state ages of several hundred milliseconds may be tolerable; high-frequency industrial sensors may require sub-millisecond synchronization, ruling out general-purpose network infrastructure in favor of direct FPGA-to-host links.

IV. Synchronization Challenges on Intel/Altera Platforms

Intel Cyclone and Arria FPGAs provide specific capabilities that are relevant to the synchronization problem. The following challenges and corresponding design considerations are identified.

Clock Domain Management

FPGA designs for DT applications typically involve multiple clock domains: a high-frequency fabric clock for sensor interfacing, a lower-frequency processor clock for the soft-processor subsystem, and an external reference for network timing. Crossing these domains without introducing synchronization errors requires proper use of dual-clock FIFOs and metastability-hardened synchronizers, both of which are available as parameterized IP in Intel Quartus Prime.

Timestamp Reference Alignment

Aligning time bases between the FPGA node and a host running a general-purpose OS is complicated by non-deterministic OS scheduling. Hardware support for IEEE 1588 Precision Time Protocol (PTP) in the Ethernet MAC—implemented natively in Intel Cyclone V and Arria 10 devices—enables sub-microsecond clock alignment and provides a reliable basis for state-age computation [5].

Partial Reconfiguration and Model Consistency

Intel FPGAs support partial reconfiguration, whereby regions of the device fabric can be reprogrammed at runtime without disrupting the rest of the design. From a DT perspective, a partial reconfiguration event constitutes a structural change in the physical system, and the virtual model must be updated accordingly. This requires a mechanism for communicating configuration change events to the DT host alongside measurement data — a requirement not present in conventional IoT sensor deployments.

The three-layer architecture covers the core requirements: deterministic acquisition, bounded-latency transport, and a continuously updated virtual state model. Three areas remain open.

The tradeoff between synchronization frequency and communication overhead has not been quantified for Cyclone and Arria configurations under realistic IIoT network conditions; measured data from hardware experiments is needed before bandwidth recommendations can be made. The Layer 3 behavioral model extrapolates state during communication gaps, but its accuracy depends on prior knowledge of physical process dynamics that may not be available in novel deployments. Finally, the integrity of timestamped sensor frames transmitted over IP networks requires explicit treatment in industrial contexts: a tampered or replayed data stream could cause the virtual model to diverge from the physical device without triggering any local fault condition, which undermines the core utility of the DT.

V. Conclusions

This paper presented a three-layer synchronization architecture for Digital Twins of FPGA-based embedded systems targeting smart sensor and IoT applications on Intel/Altera platforms. The proposed model separates concerns across physical acquisition, communication, and virtual state representation, and identifies hardware-level timestamping and PTP-based clock alignment as key enabling mechanisms for maintaining state consistency.

The analysis highlights that FPGA-specific features—including partial reconfiguration, multiple clock domains, and hardware-accelerated protocol support—introduce synchronization challenges absent from microcontroller-centric DT implementations and require dedicated architectural treatment. Future work should focus on experimental validation of the proposed architecture on Cyclone V or Arria 10 hardware, quantification of synchronization latency under realistic network loads, and extension of the virtual state model to handle partial reconfiguration events.

References

[1] [1] M. Grieves and J. Vickers, “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems,” in *Transdisciplinary Perspectives on Complex Systems*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Cham: Springer, 2017, pp. 85–113.

[2] [2] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[3] [3] A. Rasheed, O. San, and T. Kvamsdal, “Digital twin: Values, challenges and enablers from a modeling perspective,” *IEEE Access*, vol. 8, pp. 21980–22012, 2020.

[4] [4] Intel Corporation, “Cyclone V Device Overview,” Intel FPGA Product Documentation, AN-CV-5V2, 2023. [Online]. Available: https://www.intel.com/content/www/us/en/products/details/fpga/cyclone/v.html

[5] [5] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IEEE Std 1588-2019, 2020.

[6] [6] K. Cinque, "Virtual Shadow Model for Increasing Control Accuracy

[7] of Thermal Objects," in Book of Abstracts of the 92nd Int. Sci. Conf.

[8] of Young Scientists and Students "Youth Scientific Achievements to the

[9] 21st Century Nutrition Problem Solution", Kyiv, Ukraine: NUFT,

[10] Apr. 2026, part 2, p. 353.