FPGA-based Battery-Assisted Passive (BAP) WISP Tag for EPC Gen2 RFID

Bashar A. Esttaifan
1) Engineering University of Kent
2) Electronics and Communication Eng.
University of Baghdad
Canterbury, Kent, UK
bashar.esttaifan@coeng.uobaghdad.iq
baee2@kent.ac.uk

Robert J. Horne
Engineering
University of Kent
Canterbury, Kent, UK
r.j.horne@kent.ac.uk

John C. Batchelor
Engineering
University of Kent
Canterbury, Kent, UK
j.c.batchelor@kent.ac.uk

***Abstract*— This work presents a Battery-Assisted Passive (BAP) Radiofrequency Identification (RFID) Wireless Identification and Sensing Platform (WISP) tag implemented on an ultra-low-power FPGA that preserves Electronic Product Code Generation-2 (EPC G2) compatibility. To isolate and compare digital-core advantages such as protocol flexibility and precise timing control for the FPGA-based design and the microcontroller-based WISP 6.0 platform, an identical Analog Front End (AFE) of the latter was used. The FPGA implements the EPC G2 tag functions within a modular VHDL architecture, producing a fully configurable, platform- independent digital core that eases sensor integration and enables rapid feature updates without hardware redesign. In compatibility tests, the FPGA-based tag showed real-time adaptation to the reader Queries across different values of the reference time interval for a data-0 in Interrogator-to-Tag signalling (i.e., Tari), Backscatter Link Frequency (BLF), and modulation settings without VHDL changes. The read rate test recorded 1200 reads/second for a 32-bit EPC. The average processing latency was sub-20 µs. A 307 kb/s tag-to-reader throughput was achieved for a 480-bit EPC.**

Keywords*— *RFID, FPGA, WISP, Wireless Identification and Sensing Platform, sensor platforms, energy harvesting, battery-free sensor, VHDL.

I. INTRODUCTION

Remote and wireless sensing increasingly demands compact, low-power platforms that deliver higher data rates and longer read ranges for healthcare, logistics, and smart environments. Wi-Fi and Bluetooth offer high throughput but can consume too much power for battery-free or energy-harvesting devices. Passive Radiofrequency Identification (RFID) supports ultra-low-power operation via harvesting, but tags have limited range and modest data rates [1][2]. Wireless Identification and Sensing Platform (WISP) addresses these limits by combining an Analog Front End (AFE) for energy harvesting with a low-power microcontroller that implements Electronic Product Code Generation-2 (EPC G2) protocol functions [3] and sensor interfaces. WISP has enabled accelerometers, monitors, cameras, and sensors, but microcontroller-based designs struggle with higher throughput, extended range, and rapid adaptation to changing [1][4][5][6].

This paper presents a WISP tag implemented on an ultra-low-power FPGA, preserving EPC G2 compatibility while supporting higher throughput and broader reader interoperability. Using the same AFE as WISP 6.0, the FPGA implements the tag-side EPC G2 functions [7] in a modular VHDL architecture. Jadak M7E-mega reader settings (Tari=7.5 µs, BLF = 640 kHz, M2) in EU3 frequency bands was used to investigate the FPGA based tag read rate, the reader recorded read rates exceeding 1200 reads/second for a 32-bit EPC. Communication The FPGA achieves sub-50 µs response latency, enabling under fast Jadak M6E reader setting (Tari=6.25 µs, BLF = 640 kHz, FM0) for EU3 frequency bands and supports data transfer up to 307 kb/s with a 480-bit EPC. These results demonstrate that an FPGA-based digital core can significantly improve throughput while remaining fully adaptive to different reader configurations without code modification.

The paper is organised as follows: Section II describes the WISP6.0 basic node design; Section III describes FPGA based tag design; Section IV provides Results and Discussion; and Section V presents Conclusions and Future Work.

II. WISP 6.0 BASIC NODE DESIGN

The WISP 6.0 platform is a battery-free RFID sensing node that reads sensor data and transmits it to an RFID reader via backscatter modulation. This version improves earlier WISPs by harvesting more RF power and supporting higher data rates. Fig. 1 shows a WISP 6.0 node: a half-wave dipole antenna captures RF energy and backscatters the reader carrier, while an L-shaped matching network matches the antenna 50 ohm input impedance to downstream circuitry [1][3]. The node comprises power harvesting, AFE, processing, and backscatter control. The power harvester rectifies captured RF, performs voltage boosting as needed, and regulates DC to the operating voltage. The AFE extracts the reader baseband command stream using an envelope detector and comparator to produce a digital receive signal for the processor [1][6].

An ultra-low-power MSP430FR5969 microcontroller serves as the digital core and, together with an RF switch used for backscatter modulation, replaces a dedicated RFID IC. The microcontroller performs sensor interfacing (temperature, humidity, accelerometer, etc.), local processing, protocol framing, and timing control for backscatter responses. The RF switch toggles the antenna load to encode the tag response onto the reader carrier [1].

III. FPGA BASED TAG DESIGN AND IMPLEMENTATION

This paper replaces the WISP 6.0 digital core with a low static power and sufficient resources LCMXO2-7000ZE Lattice FPGA [8] to increase tag throughput from a few kilobits/s [1][2] to hundreds of kilobits/s and to make the tag adaptive to diverse reader settings. The remainder of the WISP 6.0 hardware (antenna, L-match, rectifier, regulator, envelope detector, comparator and RF switch) is retained as in both [1] and [6] so that performance differences reflect the digital-core

advantages. The processing (i.e. digital-core) shown in Fig.1 was developed using Lattice Diamond 3.13 and written in VHDL. No chip-specific primitives or mega-functions were used; the EPC G2 tag functionality is implemented as a modular, reconfigurable core. The block diagram of the VHDL functional blocks programmed onto the FPGA platform is shown in Fig. 2 and can be described as follows:

- Start-of-transmission detector: detect reader carrier onset and delimiters starting.
- Delimiter verifier: identify the 12.5 µs ±5% delimiter and mark frame start.
- Preamble handler: distinguish reader to tag signalling variants, preamble format, or frame-sync pattern.
- Cyclic Redundancy Checker (CRC): compute CRC on received commands and compare with received CRC.
- Command decoder: parse and classify reader commands (e.g., Query, QueryRep, ACK, Read, Write…).
- Memory access controller: read/write operations across EPC memory banks.
- Sensor manager: initialize and configure attached sensors then read sensor data into response buffers.
- Response processor: assemble response payloads, compute response CRC, and encapsulate header, payload, and termination.
- Pseudo-Random Generator (RNG): 16-bit Pseudo-RNG for slot selection and anti-collision.
- Backscatter timing and RF-switch controller: generate precise modulation scheme (FM0 and Millers) for backscatter responses.

Implementation each block as a separate VHDL module or entity with well-defined interfaces can simplify verification plan and future extensions in power management or protocol extensions.

IV. Results and Discussion

A high-performance workstation (Intel i9-12950HX, 64 GB RAM) was used to collect reader output and avoid acquisition latency. Universal Reader Assistant (URA) from Jadak was used for initial verification; subsequently, a C# application based on the Mercury API was used to configure and communicate with the readers, ensuring higher data throughput and integrity. The experimental setup is shown in Fig. 3, the tag and the reader antenna were separated by 1.2 m to ensure adequate received power within the EU3 band. A DIGILENT Digital Discovery logic analyser was used to monitor internal FPGA signals and timing measurement. An AFE board identical to WISP 6.0 circuit design was connected to the MachXO2 -7000HE breakout board via three lines TX, RX, and GND. Tests used Jadak ThingMagic readers (M6E [9] and M7E [10]) configured as in Table I. To avoid any latency introduced by sensor acquisition, mock data was used during testing. All measurements were repeated at least five times per configuration.

A. Experimental methodology

1. Compatibility tests: verify tag correctly detects and adopts reader parameter combinations of Tari, BLF and modulation and responds

2. Read-rate tests: measure successful tag responses per second to Query/QueryRep cycles.

3. Latency measurements: digital analyser measurement of processing latency from the end of the reader command to the tag backscatter start.

4. Throughput calculations: compute usable tag to reader data rates excluding protocol overhead.

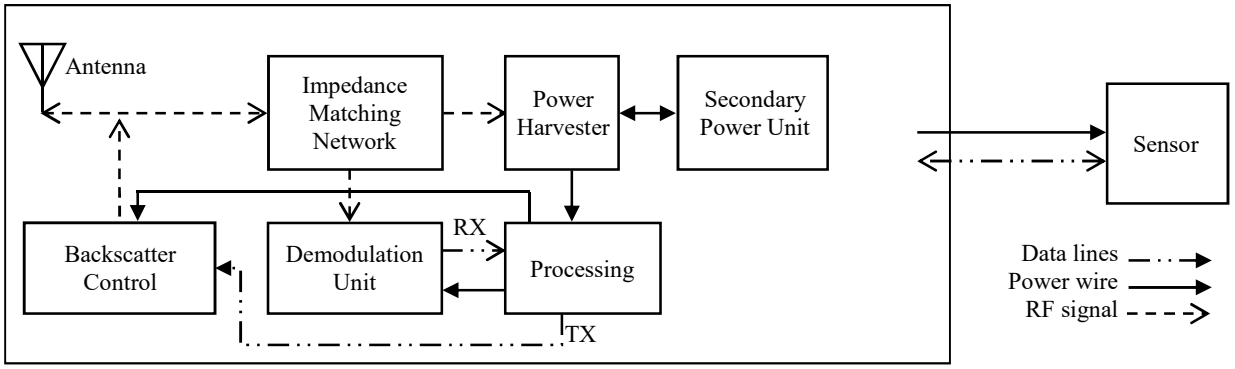

Fig. 1. Block diagram of a basic WISP-6.0 board.

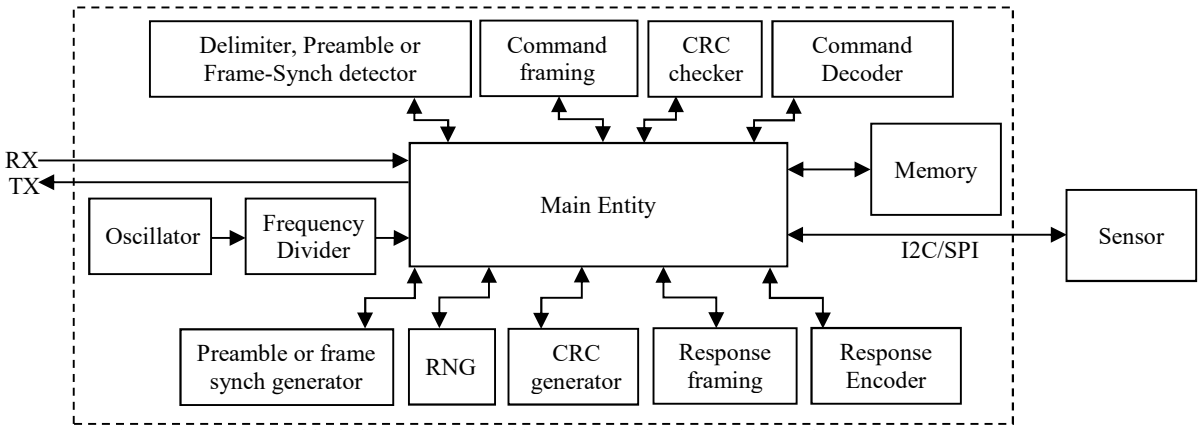

Fig. 2. Processing block diagram developed in VHDL and implemented onto the LCMXO2-7000ZE FPGA.

B. *FPGA tag compatibility with reader settings*

The Jadak M6E reader was selected for compatibility testing because it supports Tari values of 6.25, 12.5, 25 µs, BLF settings of 240, 320, 640 kHz, and modulation schemes FM0, M2, M4, M8. The FPGA tag successfully:

- Detected the reader-issued BLF and set backscattering timing accordingly.
- Identified delimiters, preamble and Tari and aligned frame boundaries within the EPC G2 tolerance given in [7].
- Switched modulation when the reader Query changed modulation schemes.

No VHDL recompilation or reconfiguration was required during these tests, the tag digital core parsed Query commands and adjusted timing and modulation dynamically. The logic analyser waveforms confirmed that the FPGA reliably produced the expected control signals across all tests. This capability is a key advantage over microcontroller-based WISP implemented in [1] that may require firmware updates or cannot meet tight timing windows.

C. *Read-rate investigation*

For read-rate characterization we used the M7E reader with the setting listed as in the second column of Table I. to explore the tag response capacity under higher throughput conditions. Read rate was measured as successful responses per second to Query/QueryRep cycles for varying EPC lengths and Slot-count (Q) values (i.e. anti-collision parameter). Key observations from read-rate curves of Fig. 4 can be summarised as follow:

- Read rate vs EPC length: Read rate decreased with increasing EPC length. EPCs ≤160 bits consistently reached 1200 reads/second with 32bit EPC length, while full-length EPCs (480 bits) caused reader buffer to saturate and limited read rate to ≈400 reads/second in open-lab environment.
- Effect of Slot-count (Q): Increasing Q increased read rate up to a point; large Q values produced no change due to reader buffer. Q=7 provided a practical balance where high read rates achieved without buffer saturation across runs.

The FPGA tag low processing latency and rapid backscatter timing allow it to exploit reader capacity effectively for short to medium EPC lengths. For long EPC lengths, system throughput becomes reader-limited rather than tag-limited; this suggests that application designers should match EPC length to expected reader capabilities and use Q tuning to avoid buffer saturation.

D. *Latency and internal timing*

Logic analyser traces show average processing latency of 20 µs across tested configurations. The low latency is attributable to the FPGA parallel logic and deterministic state machines for the tag functions.

The 20 µs latency explains the observed read-rate gains: the tag can complete response preparation and begin backscatter within a single reader timing slot, enabling higher reads/second than microcontroller-based tags that require longer wake-up and processing times.

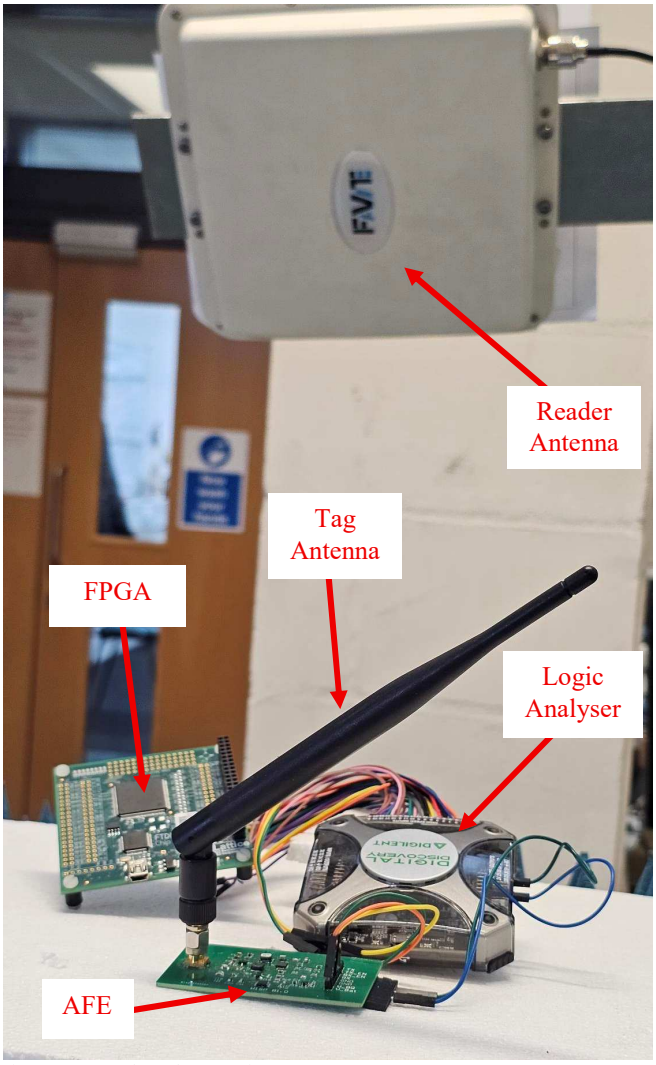

Fig. 3. FPGA-based tag testing.

TABLE I. READERS SETTING IN B AND C.

Reader parameter	Value for read rate test in B	Value for data rate test in C
Model	M7E	M6E
Protocol	EPC G2	
Antenna port	1	
Region	EU3 (865-868 MHz)	
Tari	7.5 µs	6.25 µs
Tag encoding	M2	FM0
BLF	640 kHz	
Q	0 to 15	7
Baud rate	921600 b/s	

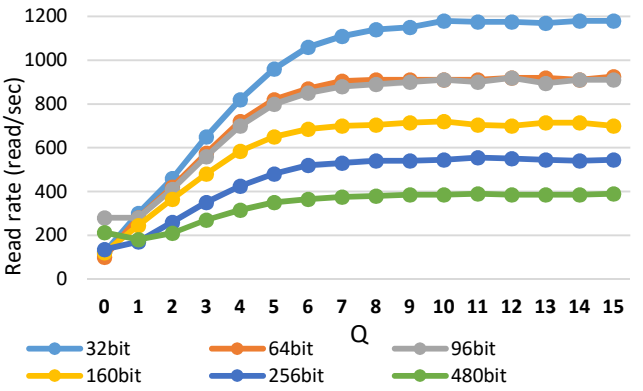

Fig. 4. FPGA tag read rate versus different values of Q and EPC length.

E. Tag-to-reader data-rate measurement

Throughput tests used the M6E configured to the fastest BLF and modulation settings listed in column 3 of Table I. For each EPC length we recorded the steady-state read rate and computed usable throughput according (1).

$$\text{Throughput (or data rate)} = \text{read rate} \times \text{EPC length} \quad (1)$$

Calculated throughput results shown in Fig. 5 point to the following:

- Minimum rate of 20 kb/s at EPC length of 16-bit.
- 480-bit EPC produced a peak measured throughput of 307 kb/s.
- Throughput scaled approximately linearly with EPC payload up to the point where reader buffering and processing limited reads/second.

Fig. 5 confirms the ability of the FPGA implementation to achieve tag-to-reader data rates in the hundreds of kilobits per second for large EPC length when paired with a reader configured for fast settings. This performance represents a substantial improvement over microcontroller-based WISP implementations [5] and EM4325 chip-based RFID [2], which are typically constrained to a few kilobits/s ranges due to processing and timing limitations.

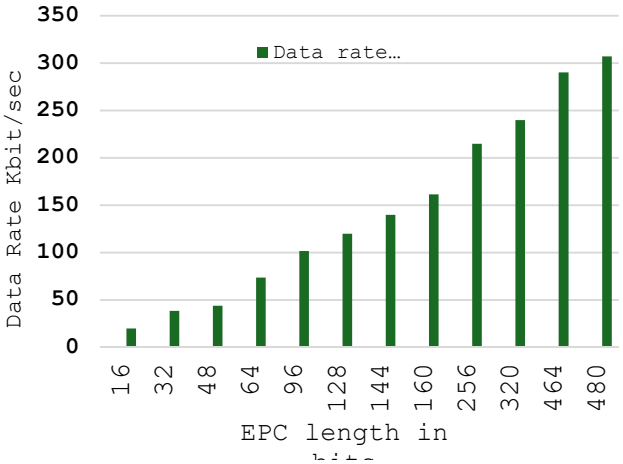

Fig. 5. FPGA tag data rate for different values of EPC length

F. MSP430FR5969 vs FPGA based WISP6.0

The results presented in Sub-section (B) highlight a fundamental difference between the two WISP architectures. The FPGA-based tag adapts automatically to changes in Tari, BLF, and backscatter modulation issued by the reader through standard EPC G2 Query commands. In contrast, the WISP 6.0 based on MSP430FR5969 cannot adjust to these parameters at run time. Any change in reader settings requires modifying the code which is written in both C and assembly [6], recompiling, and reprogramming the microcontroller.

The microcontroller-based implementation is inherently chip-dependent, the timing, counters, communication peripherals, and interrupt structure are tied to the MSP430 hardware. As a result, the WISP 6.0 firmware is only stable under a narrow set of reader configurations specifically Tari = 6.25 µs, BLF = 640 kHz, and FM0 modulation [5][6].

In contrast, the FPGA-based WISP implements EPC G2 in VHDL, without relying on device-specific timers or peripherals. The design uses generic logic blocks and does not depend on fixed pin assignments or microcontroller-specific hardware features. This makes the FPGA core platform-independent, fully deterministic, and capable of adapting to any valid reader configuration without code changes.

V. Conclusions and Future Work

The FPGA-based replacement of the WISP digital core improves performance and adaptability while preserving the WISP 6.0 AFE. Implementing EPC Gen2 functions in modular VHDL yields deterministic timing, rapid response, and seamless adaptation to diverse reader configurations without firmware changes. Experiments confirmed sub-20 µs processing latency, read rates exceeding 1200 reads/second for short EPCs, and a maximum tag-to-reader throughput of 307 kb/s for a 480-bit EPC under fast reader settings. These results demonstrate the advantages of parallel hardware logic for high data-rate, fast sensing cycles and ability to operate without firmware changes across multiple reader settings demonstrates a key benefit of the FPGA tag. This positions the design as a strong candidate for next-generation BAP RFID systems that demand both flexibility and high performance.

Future work will focus on system improvements, particularly the power-management subsystem, to enable battery-free operation. The implemented VHDL will be generalised and tested across multiple platforms. Sensors will be attached to the FPGA tag to evaluate integration and end-to-end performance. Performance investigations and comparative studies will be conducted against WISP 6.0 and other RFID chip-based systems.

References

[1] R. Menon, A. Saffari, and J. R. Smith. 2022. Wireless Identification and Sensing Platform Version 6.0. In Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems (SenSys '22). Association for Computing Machinery, New York, NY, USA, 899–905. https://doi.org/10.1145/3560905.3568109

[2] R. Horne and J. C. Batchelor, "A framework for a low power on body real-time sensor system using UHF RFID," IEEE Journal of Radio Frequency Identification, vol. 4, no. 4, pp. 391-397, Dec. 2020, doi: 10.1109/JRFID.2020.3018405.

[3] J. R. Smith, A. P. Sample, P. S. Powledge, S.t Roy, and A. Mamishev. 2006. A wirelessly-powered platform for sensing and computation. In Proceedings of the 8th international conference on Ubiquitous Computing (UbiComp'06). Springer-Verlag, Berlin, Heidelberg, 495–506. https://doi.org/10.1007/11853565_29

[4] H. Solar, A. Beriain, R. Berenguer, J. Sosa and J. A. Montiel-Nelson, "Semi-Passive UHF RFID Sensor Tags: A Comprehensive Review," in IEEE Access, vol. 11, pp. 135583-135599, 2023, doi: 10.1109/ACCESS.2023.3336761.

[5] A. Sample, D. Yeager, M. Buettner, and J. Smith, 'Development of Sensing and Computing Enhanced Passive RFID Tags Using the Wireless Identification and Sensing Platform', Development and Implementation of RFID Technology. I-Tech Education and Publishing, Jan. 01, 2009. doi: 10.5772/6521.

[6] (2026). WISP Wiki. Accessed: March. 9, 2026. [Online]. Available: *https://sites.google.com/uw.edu/wisp-wiki/wisp6*.

[7] (2026). *GS1.org,* "*Gen2 Release 3.0, Ratified Jan 2024*". Accessed: March. 9, 2026. [Online]. Available: *https://www.gs1.org/standards/rfid/uhf-air-interface-protocol*.

[8] (2026). latticesemi.com. Accessed: March. 9, 2026. [Online]. Available:*https://www.latticesemi.com/Products/FPGAandCPLD/MachXO2*.

[9] (2026). jadaktech.com. Accessed: March. 9, 2026. [Online]. Available: *https://www.jadaktech.com/product/thingmagic-m6e-uhf-rain-rfid/*.

[10] (2026). jadaktech.com. Accessed: March. 9, 2026. [Online]. Available: *https://www.jadaktech.com/product/thingmagicm7e-uhf-rain-rfid-module-se*