# PELS: A Lightweight and Flexible Peripheral Event Linking System for ULP IoT Processors

Alessandro Ottaviano[1], Robert Balas[1], Philippe Sauter[1], Manuel Eggimann[1], Luca Benini[1,2]

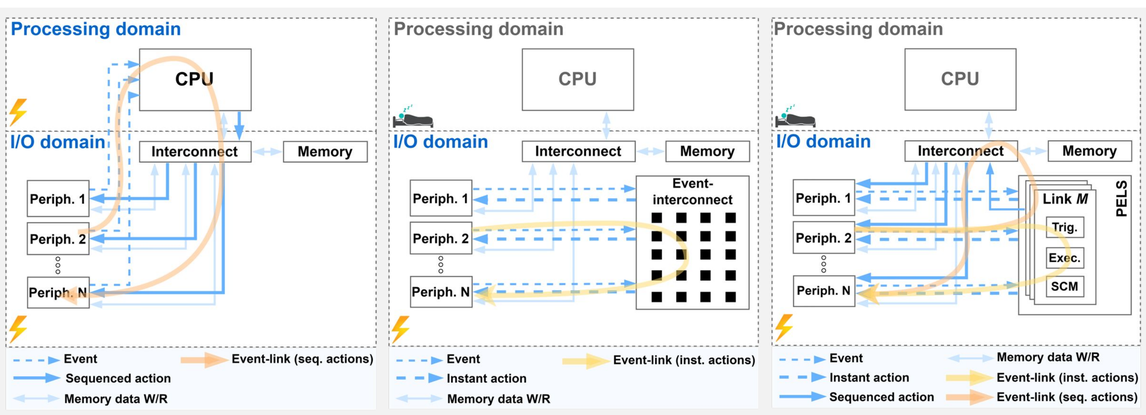
[1]IIS, ETH Zurich; [2]DEI, University Of Bologna;

## 1 Motivation

- **Peripheral event linking (PEL)** is a communication channel that lets peripherals interact directly
- **Main benefit:** reduce wake-up in the processing domain
- **Impact on IoT applications:** battery devices, always-on wearables, real-time/predictable communication

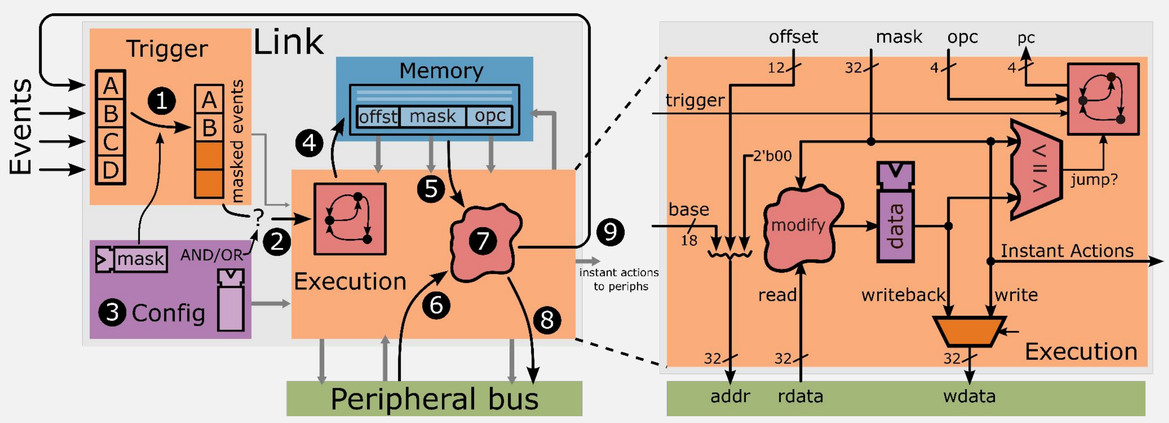
**How to design a PEL system trading off performance, flexibility and deterministic behavior?**

## 2 What's in the market?



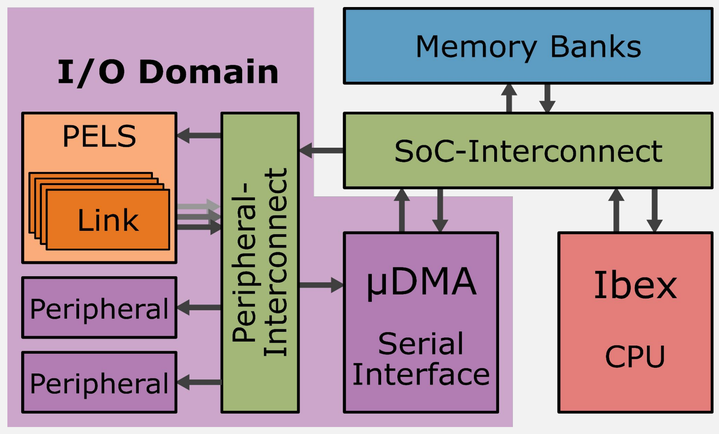

- **Interrupt-based linking: shared bus,** CPU wakes-up frequently
- **Event-interconnect linking:** fixed connections between periphs.
- **FPGA-I/O processor:** high-flexibility, very general-purpose
- **PELS:** hybrid approach (shared bus && fixed connections), private SCM

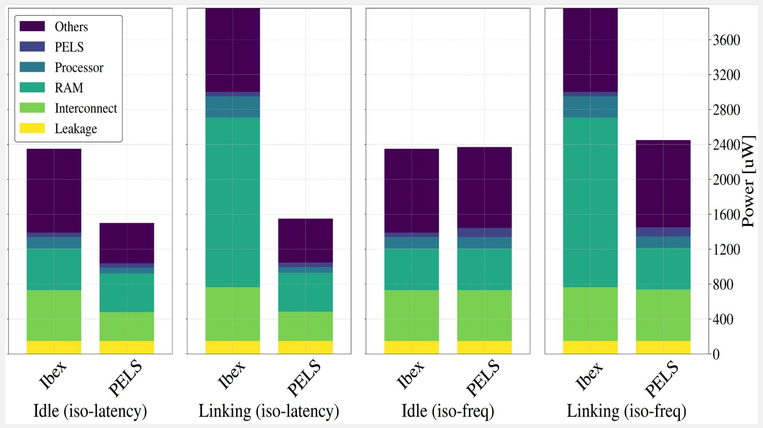
## 4 Architecture




- **Trigger and mask units:** mask events according to user config
- **Private SCM:** per-line command, no contention on shared bus (for predictability), lower power/area overhead than SRAMs
- **Execution unit**
  - **Sequenced actions:** read-modify-write through system bus
  - **Instant actions:** direct wires to peripherals, or loop-back to event lines
- **Commands:**
  - **Encoding:** 4-b opcode, 12-b address offset, 32-b data
  - **`set`**, **`clear`**, **`toggle`**: e.g., for reset/toggle of periphs./GPIOs
  - **`capture`**, **`jump-if`**: e.g., for threshold comparison
  - **`loop`**, **`wait`**: e.g., for watchdog-like functions
  - **`action`**: sets outgoing wires

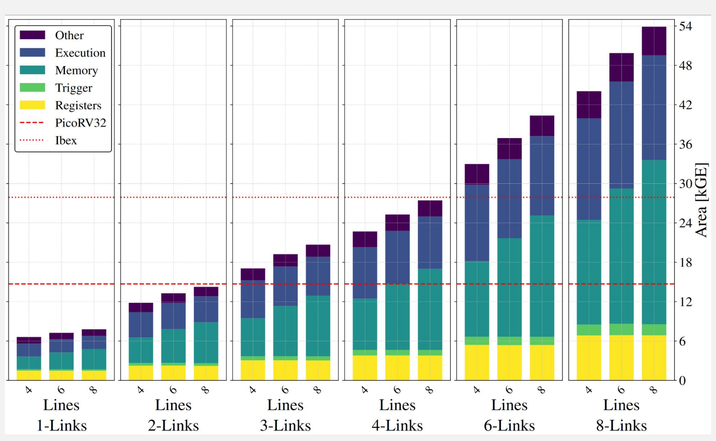
## 3 Integration and evaluation

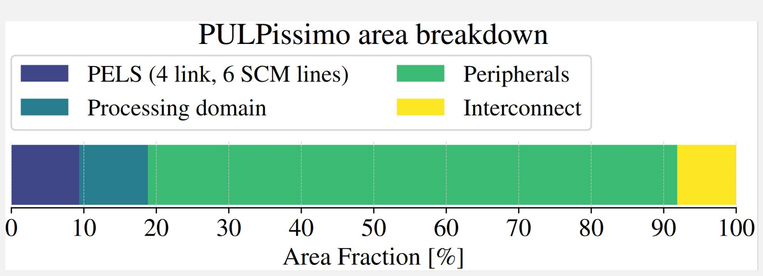



**Functional:**

- **Test case:** Event-linking application with a threshold-crossing check after I/O DMA-managed sensor readout through SPI
- PELS sequenced actions vs. Ibex interrupt
- **2.5x and 1.5x power reduction** due to reduced system memory activity for iso-latency and iso-frequency cases




**Implementation:**

- Synthesis in TSMC65, TT, 25°C
- Only 7kGE in minimal configuration (4 links, 6 SCM lines), **4x and 2x smaller** than **Ibex** and **PicoRV32**, respectively
- Occupies **just 9.5% of PULPissimo logic**, 1% when considering memories

## 5 Conclusion

**We designed PELS, a flexible near-sensor processing systems for fully autonomous multi-sensor and inter-peripheral communication, providing 2.5x power reduction for a linking event compared to traditional interrupt-based approaches, at the cost of just 7kGE in its minimal configuration**

pulp-platform.org  pulp-platform  @pulp_platform  aottaviano@iis.ee.ethz.ch