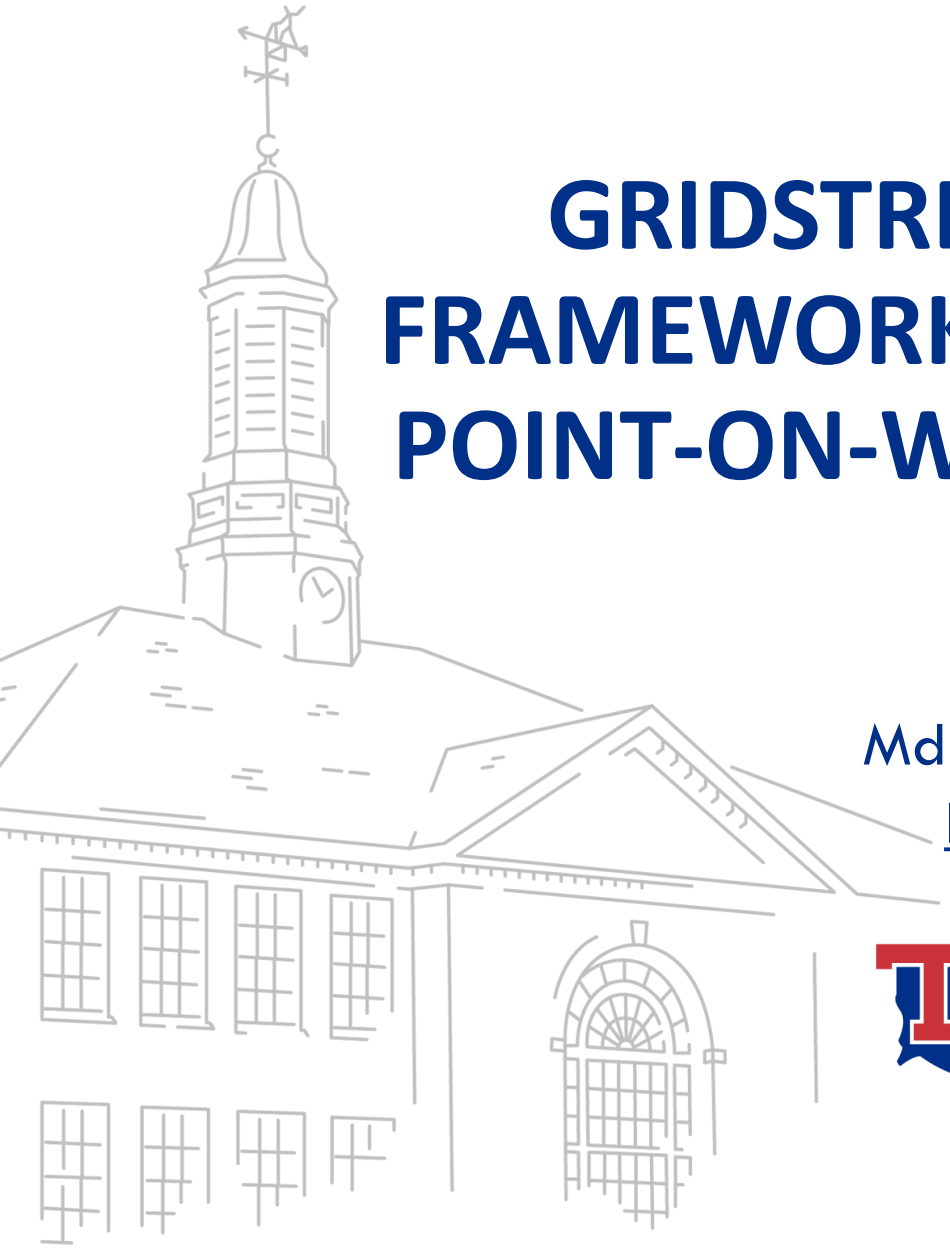


GRIDSTREAM: A HARDWARE-EFFICIENT FRAMEWORK FOR BANDWIDTH CONSTRAINED POINT-ON-WAVE DISTURBANCE MONITORING

Md Kibria Saroare, Md Abul Hasnat, Md Rubel Ahmed
Louisiana Tech University, University of Wyoming

LOUISIANA TECH UNIVERSITY

UNIVERSITY OF WYOMING

Motivation: The New Grid Reality

➢ **Faster Dynamics & Uncertainty**

Modern power grids have faster dynamics and enhanced uncertainty due to:

- Inverter-based resources (IBRs).
- Electric Vehicles (EVs).
- Large time-varying loads (e.g., Data Centers).

Implication: Grid monitoring requires increased **temporal granularity** to capture stresses and anomalies.

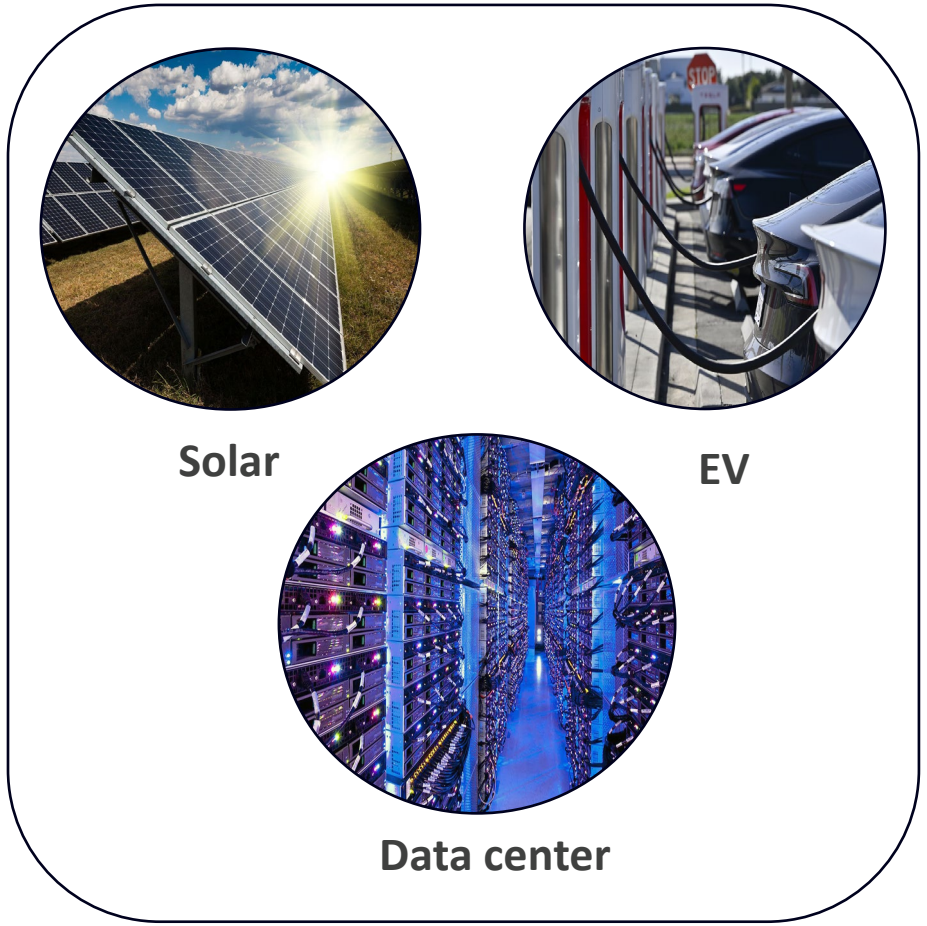


Figure 1: Sources of **fast dynamics & uncertainty**

The Edge-Intelligence Gap

- To observe **harmonics** and **sub-synchronous components**, we must capture raw **Point-on-Wave (PoW)** waveforms.
- A single distribution feeder sensor generates **terabytes** of raw data daily.
- Utility communication networks are designed for **low-rate traffic** and cannot support continuous high-resolution streaming.
- This creates a bottleneck called **Edge-Intelligence Gap.**

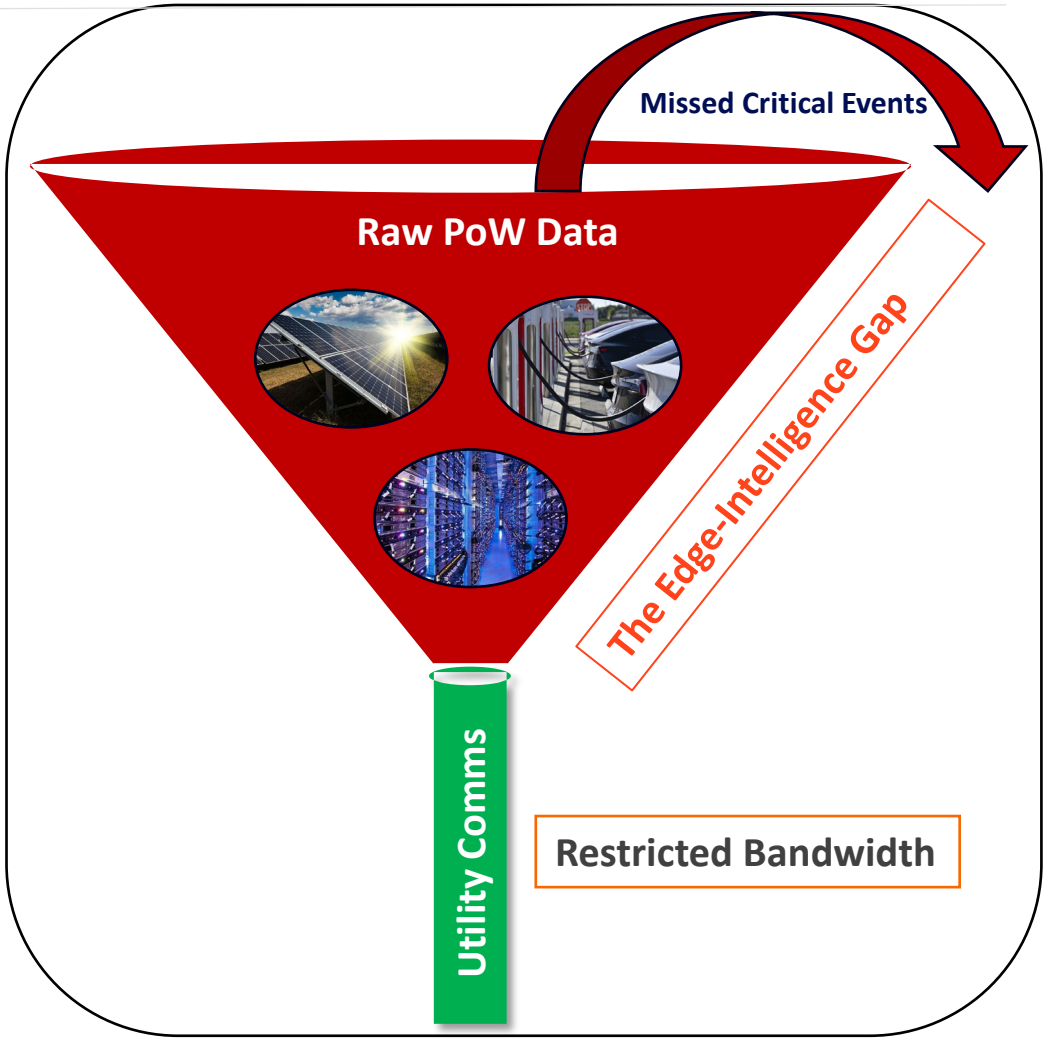
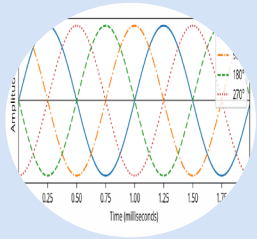


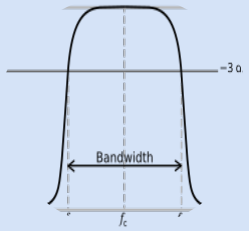
Figure 2: The Edge-Intelligence Gap

Why Standard Monitoring Fails?



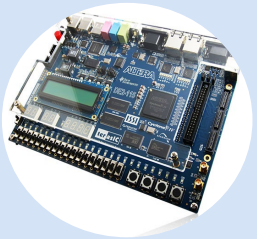
PMU Model Limitations

Traditional PMUs rely on a sinusoidal assumption, unable to capture the harmonics and sub-synchronous components of grids.



Bandwidth Constraints

Utility communication networks are designed for low-rate traffic and cannot support continuous high-resolution streaming.



Hardware Gap

Existing event detection algorithms are mostly developed on simulation platforms and not optimized for low-power edge hardware.

Research Question & Proposed Solution

Q1: The Bandwidth Gap

How can we capture and transmit high-fidelity disturbances without overwhelming the communication network?

A1: Streaming Event-Triggered Framework.

Q2: The Hardware Gap

How do we implement this on low-cost FPGAs without sacrificing accuracy?

A2: Hysteresis & Targeted FPGA Optimization.

Our Solution: GridStream Framework

Streaming Event-Triggered with Two-Threshold Hysteresis

Methodology Overview: GridStream Architecture

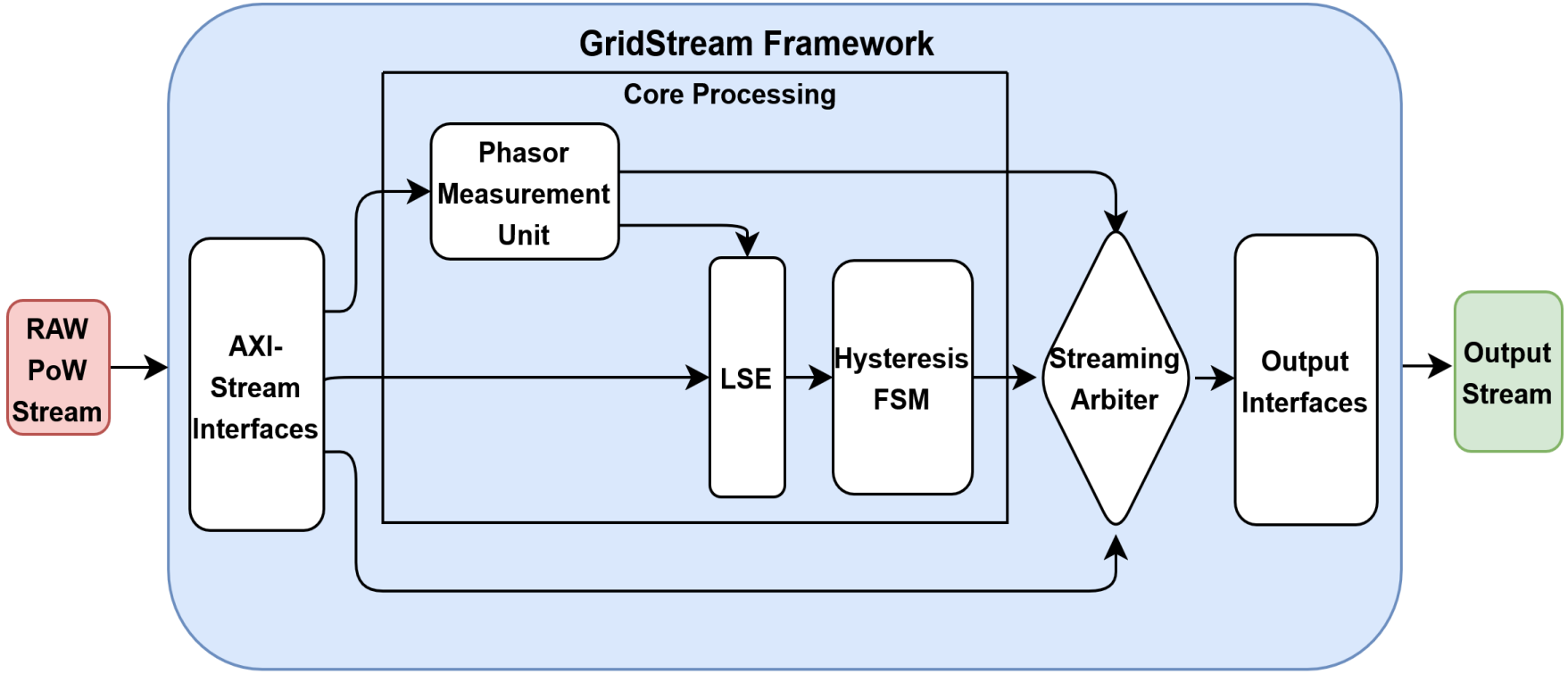


Figure 3: GridStream Architecture

- **Phasor Path:** Continuous situational awareness (NCO, FIR, CORDIC).
- **Event Path:** Calculates Least Squares Error (LSE) recursively.

Single-Threshold: Noise-Induced Chattering

- Standard detection uses a single threshold (T).
- In noisy distribution grids, the error signal fluctuates around the threshold.
- Rapid on/off switching creates fragmented, unusable data packets.

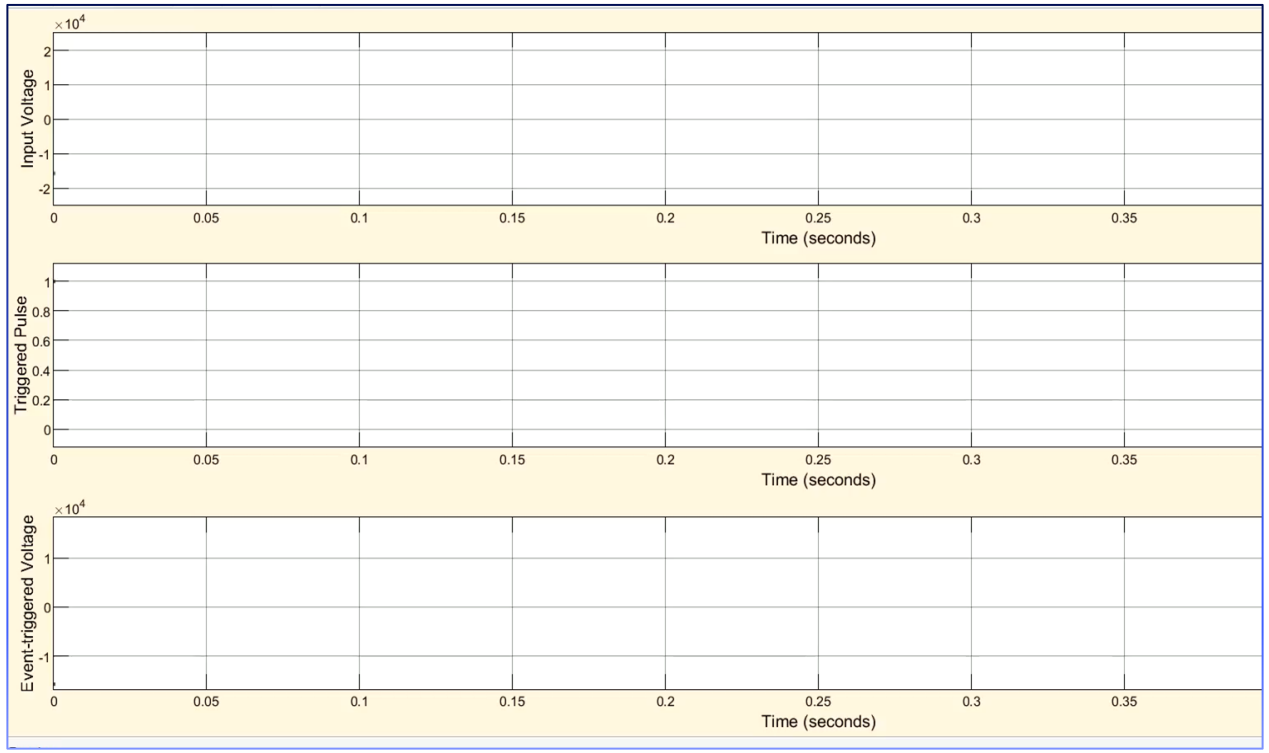


Figure 4: Single-Threshold chattering trigger state.

Key Innovation: Two-Threshold Hysteresis

- We introduce a novel two thresholds hysteresis.
  - **High Threshold (T_H):** Triggers the event start.
  - **Low Threshold (T_L):** Determines the event end.
- A clean, contiguous event window with zero fragmentation.

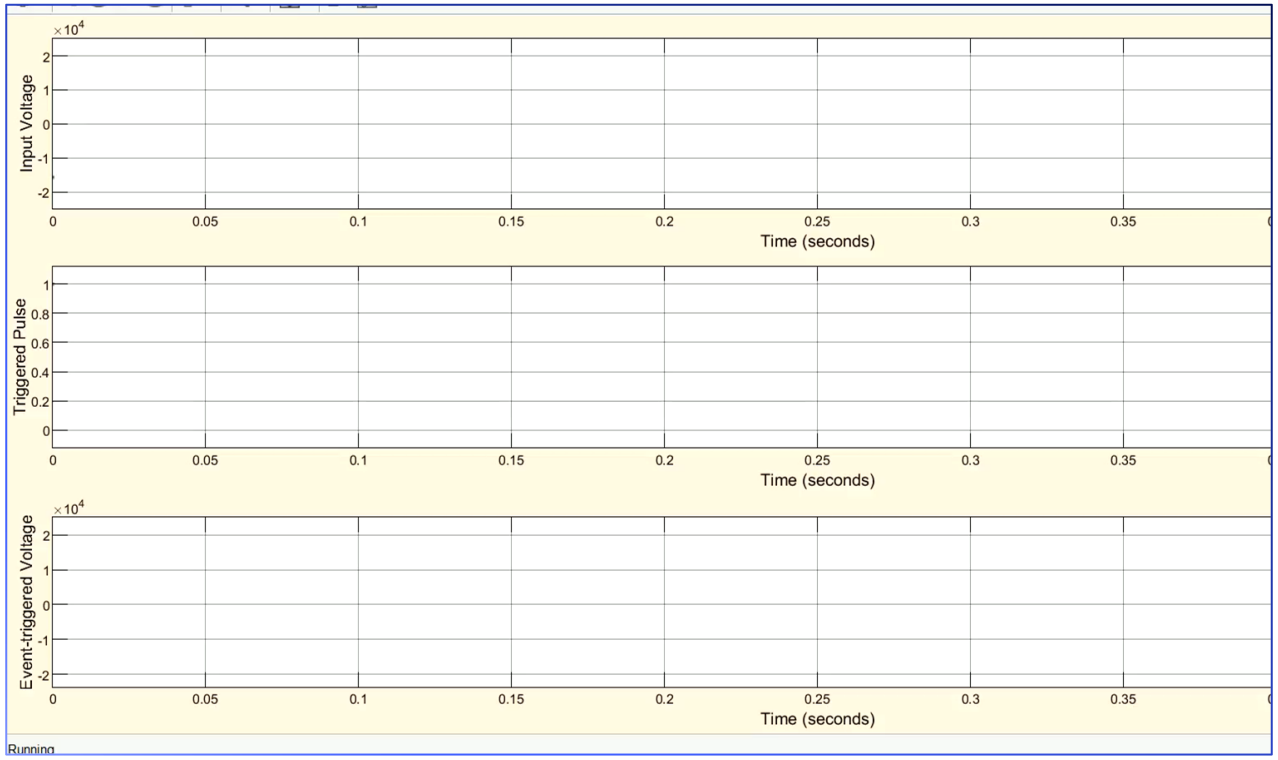


Figure 5: Two-Threshold continuous trigger state.

Automated DSE: Sampling & Findings

➢ The Sampling Space & Objectives

We utilized a Python-based automation script to sweep three key design axes:

- **Arithmetic Families:** Minimal vs. Safe (Dynamic Range).
- **Quantization:** Fixed-Point (16–32 bits) vs. IEEE-754 Floating-Point.
- **Resource Allocation:** Varied Loop Unrolling (1, 2, 8) and Pipeline Initiation Intervals (II=1, 2, 4).

Objective: To identify architectures that maximize throughput (F_max) while fitting within the strict logic density constraints of Edge-Class FPGAs.

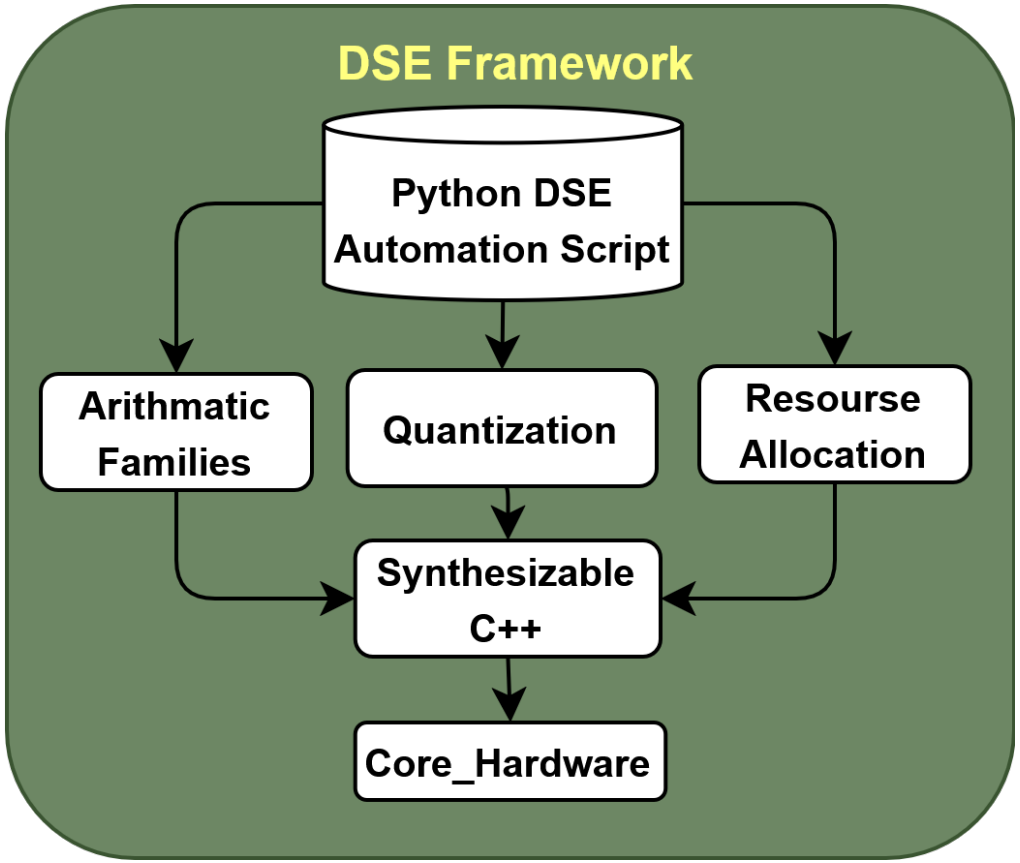


Figure 6: Automated Design Space Exploration Framework

Key Findings

- Contrary to conventional fixed-point wisdom, Floating-Point architectures provided the optimal balance.
- Fixed-point designs required massive bit-widths (>48 bits) to handle the LSE dynamic range compared to floating-point.

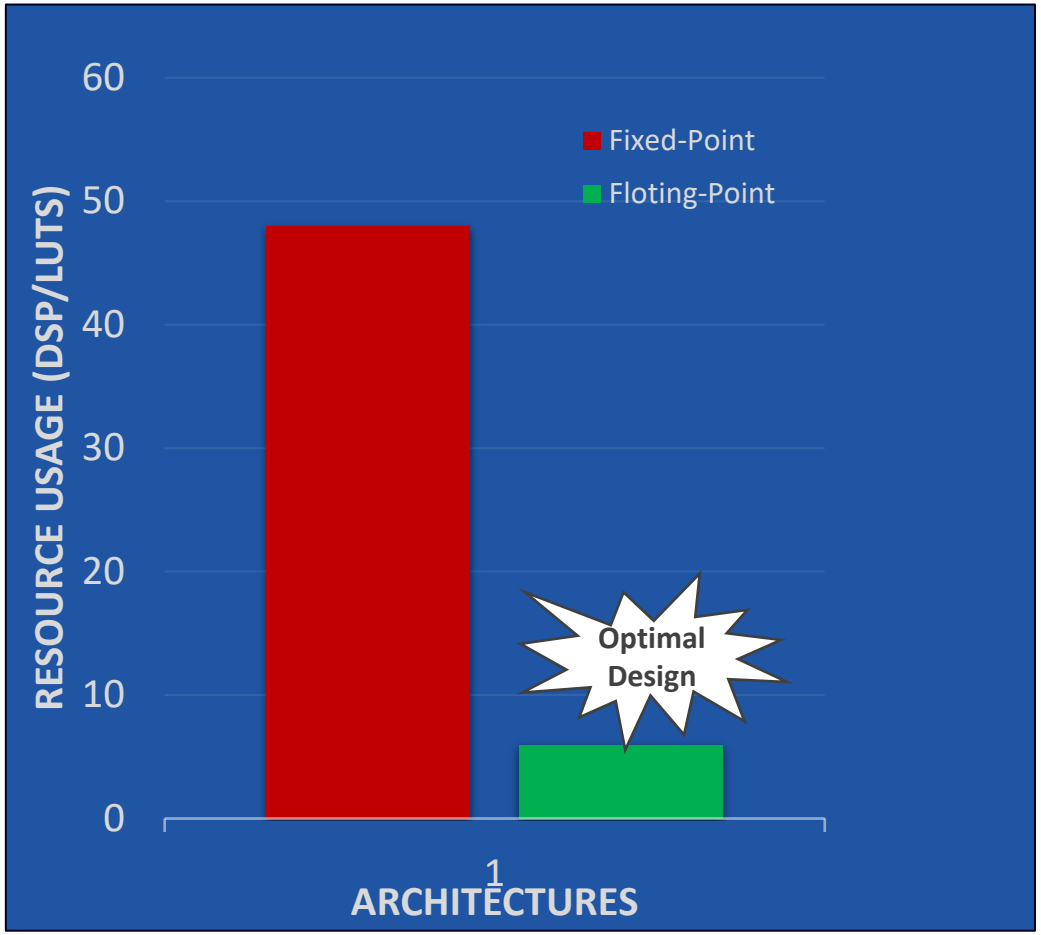


Figure 7: Floating vs. Fixed Point Comparison

Finding: Pareto Frontier

- Achieved 407 MHz highest throughput incurring a 53-fold penalty in DSP usage compared to floating-point on the Zynq UltraScale+.
- The optimal floating-point design achieved >300 MHz operation using only 8,417 LUTs on Edge-Class FPGA Spartan-7.

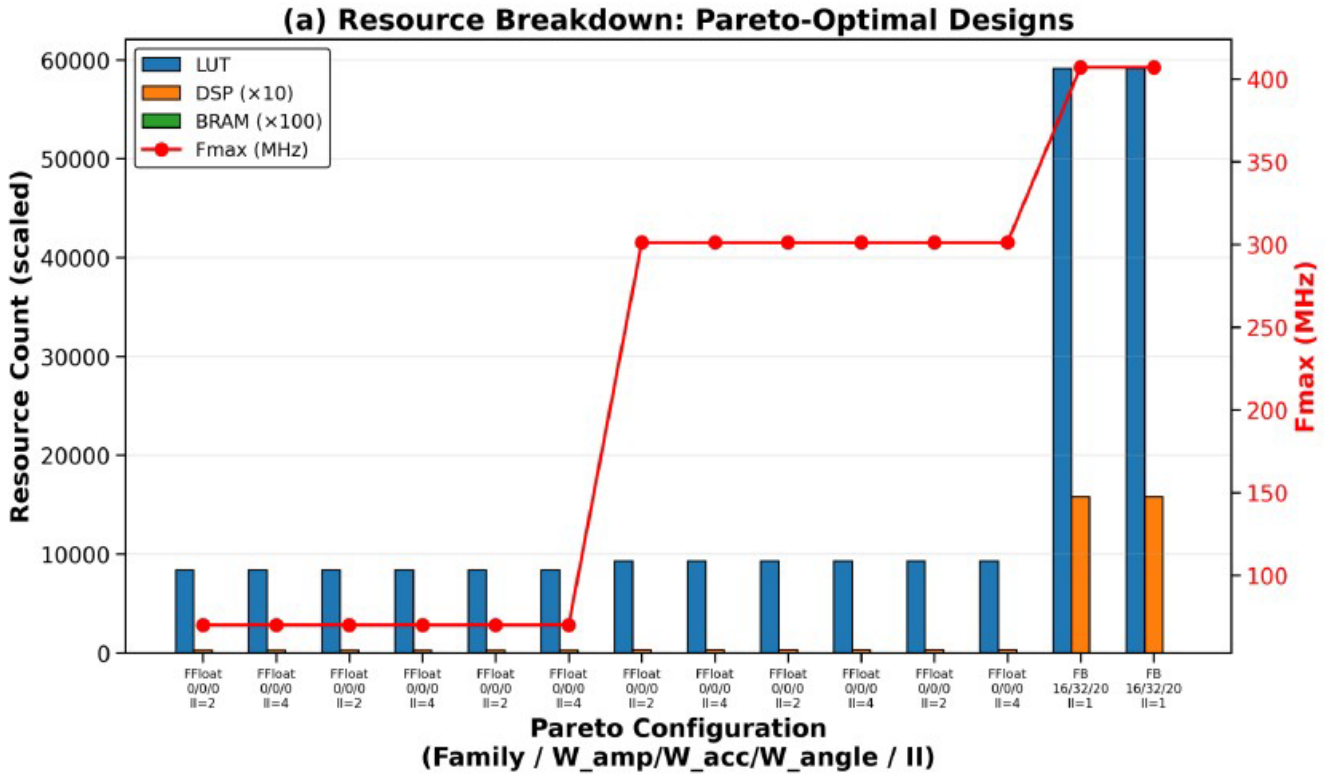


Figure 8: Resource breakdown for Pareto-optimal configurations.

Summary

High Accuracy

Novel Two-Threshold Hysteresis approach solved detection chatter problem in noisy environments.

8,417 LUTs

Floating-point implementations consuming as few as 8,417 LUTs on low-cost Spartan-7 FPGAs.

407 MHz

High-performance fixed-point variants achieving 407.2 MHz on Zynq UltraScale+ devices.

Future Directions

Adaptive DSE

Extending the framework to implement online tuning of hysteresis thresholds and bit-widths that dynamically minimize frag. and false alarms as background grid noise changes.

Multi-Channel Scaling

Utilize time-multiplexing to support 3-phase voltage & current on a single FPGA core by leveraging high *F_max* to process multiple channels within one sampling period.

Event-Aware Streaming

Jointly optimize data delivery time and power consumption Create *Event-Aware* streaming that prioritizes critical fault data when power or bandwidth is scarce.

Current focus: Can Lingua Franca Improve High-Resolution PoW Monitoring?

Current Limitations

- **Non-deterministic event ordering**
  Callbacks and scripts race across channels; event replay is inconsistent.
- **Imprecise timing semantics**
  PoW sampling at high rates (e.g., 256 samples/cycle) requires precise coordination; jitter corrupts sub-synchronous detection.
- **Opaque multi-domain coordination**
  FPGA event path, phasor path, and network streaming are coupled through ad hoc glue logic that is hard to inspect or verify.
- **Threshold adaptation brittleness**
  Online tuning of T_H/T_L has no safe sequencing model; concurrent updates can cause transient chattering.

How LF Can Help

→ **Deterministic reaction ordering**
LF reactors enforce a fixed priority at each logical time step, making event replay reproducible across runs.

→ **Explicit logical-time model**
Sampling ticks are first-class timing primitives; delays between event detection and streaming are expressed and verified explicitly.

→ **Transparent multi-domain composition**
FPGA event path, phasor path, and UDP streamer become explicit LF reactors with visible ports and connections.

→ **Safe modal threshold adaptation**
T_H/T_L updates are guarded by LF modes; transitions complete at a logical time boundary before new thresholds take effect.

Acknowledgment

This work is jointly funded by the NSF-Louisiana EPSCoR and COES at Louisiana Tech University.

Open-Source

The code and dataset of GridStream are open source and available on GitHub.
https://github.com/KIBRIA-SAROARE/GridStream

LOUISIANA TECH UNIVERSITY

Questions..?

Email for additional queries: msa078@latech.edu