# FAST-AR Proposal

Ananya Virmani, Harshul Jalan, Henry Wysong-Grass, Andrew John Cadiena

# Problem Statement

What is the specific challenge you are solving?

- Software-based AR causes motion sickness due to high latency and burns through battery life

Why is an FPGA better for this than a standard CPU or microcontroller? (Think

concurrency, latency, or custom IO)

- We need microsecond latencies and massive concurrency that standard CPUs simply can't provide.
- GEMM operations are often memory bound - live FPGA processing eliminates this.

# Proposed Solution & Features

High-level overview of your system.

Key Features: List 3–4 "Must-haves" and 1–2 "Stretch goals" (features you'll add if you finish early).

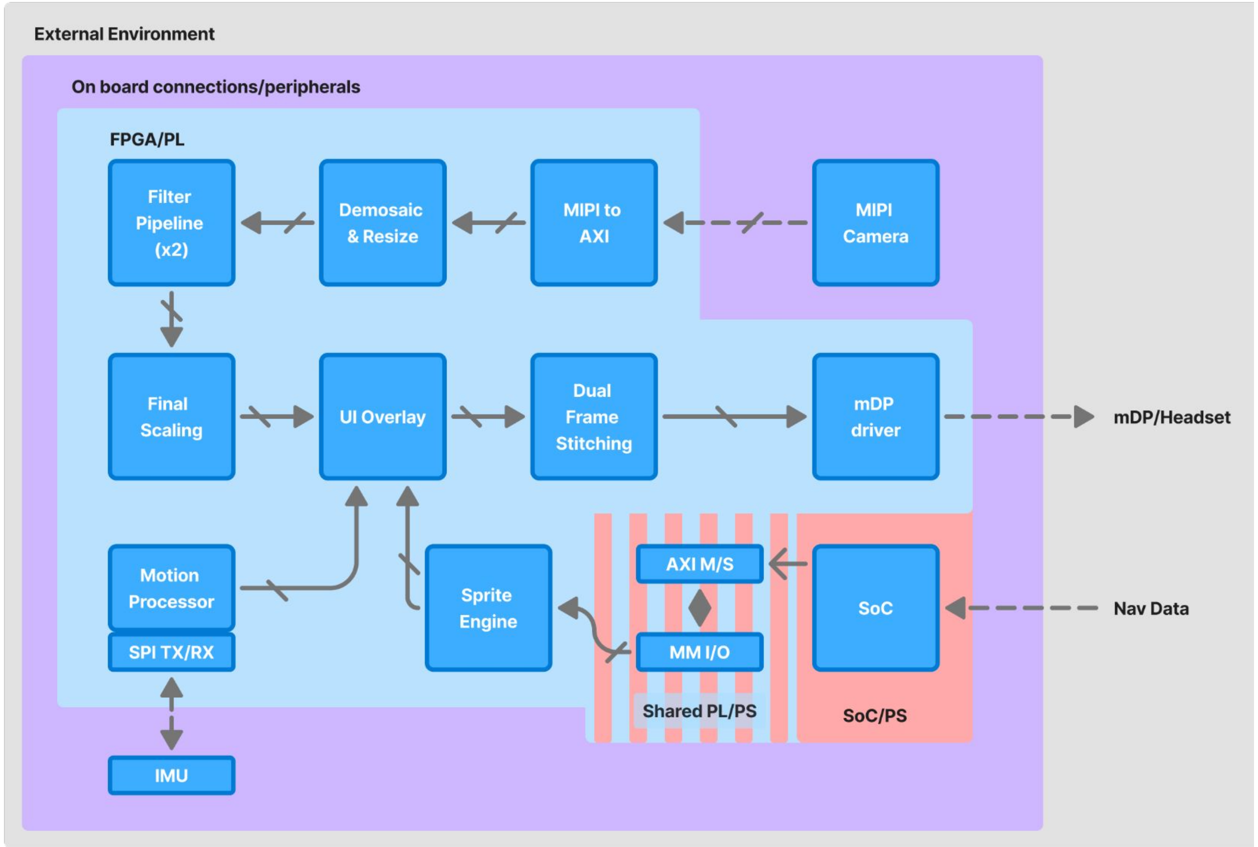
Must-haves:

- Direct MIPI-to-mDP video passthrough
- Real-time hardware filters (Edge-detect, Sharpening)
- Hardware Sprite Engine for navigation overlays

Stretch Goals:

- Live Barrel Distortion algorithm to fix biconvex lens warping

# System Architecture



**FPGA/PL** handles…

- Core video pipeline
- IMU interface + processing
- Sprite rendering
- UI/Sprite overlay

**SoC/PS** handles…

- Bluetooth interfacing
- Collecting nav data

**Shared PS/PL…**

- MM I/O: SoC □ FPGA via AXI

# Detailed Hardware Modules

Break down your custom RTL modules at a high-level.

• Mention specific protocols you'll use: I2C, SPI, UART, or high-speed interfaces like HDMI/VGA.

# Resource Estimate

I/O Devices: MIPI Camera, mini-DisplayPort, and IMU sensors.

Memory: Several MB of on-chip BRAM needed for image buffering and sprite storage.

Timing: 30 FPS minimum target to ensure a tolerable AR experience (hopefully)

# Verification Plan

- RTL Unit Testing (Simulation) using QuestaSim for all key modules
- SignalTap or ILA (Integrated Logic Analyzer) allow for probing internal signals while the board is running.
- I/O Verification: Ensuring the UART stack correctly parses incoming navigation strings and updates the hardware sprites accordingly
- Final Demo Verification - Ensuring that the final headset works as expected without any issues and if yes, debugging those.

# Project Timeline

# Potential Challenges & Risks

- The initial challenge for us would be to make use of the Xilinx board because we have no prior experience with it and the camera module do

# Conclusion and Expected Outcome

What does the final demo look like? (e.g., "A live video feed with a 60fps filter applied

via hardware").

- Expected takeaway from this project.