ScreamJump

Embedded Systems Final Project Proposal

Team Members:

- Ananya Maan Singh - am6542
- Sharwari Bhosale - sb5032
- Kamala Vennela Vasireddy - kv2446

OVERVIEW

Our idea for the final project is ScreamJump, an interactive game in which the audio levels of a player's voice control the movement of the on-screen character. The FPGA detects audio levels using a microphone and processes the signal to determine jump height and length. The louder the player screams or speaks, the higher and longer the character will jump.

The game will be displayed on a VGA monitor, with sprite-based graphics for the characters and platforms. The goal is to reach the endpoint by jumping from platform to platform, avoiding obstacles, and avoiding falling off the screen.

HARDWARE VS SOFTWARE IMPLEMENTATION

Hardware (FPGA)

- Microphone Input Processing
  - Capture real-time audio signals from a microphone.
  - Implement sound intensity detection using peak amplitude analysis.
  - Classify the sound into different intensity levels (soft, medium, loud).

- Game Logic in SystemVerilog
    - Control the character's jump height and length based on detected voice intensity.
    - Implement collision detection for platforms and falling mechanics.
    - Manage state transitions for different game phases (jumping, landing, falling).
- Display Output (VGA monitor)
    - Render 2D sprites for the game environment and character.
    - Update real-time character movements and interactions.

Software (C++/Python)

- UI & Game Enhancements
    - Implement game scoring, difficulty scaling, and animations.
    - Keeping track of game state (ex: time, score)
    - Pause and restart functionality

INTEGRATION OF HARDWARE AND SOFTWARE

1. Microphone captures voice input → FPGA processes sound intensity.
2. FPGA determines jump height and length → Updates character movement in real-time.
3. Game logic & physics handled in FPGA → Displays updates on VGA.
4. Software (C++/Python) handles UI & additional features → Sends data to FPGA if needed.

SPEECH & MICROPHONE PROCESSING:

We will use a simple amplitude-based method to classify sound intensity:

- Capture raw sound amplitude from the microphone.
- Apply a sliding window average to smooth variations.
- Compare to predefined thresholds to determine jump height:
    - Low Volume (Soft Talking) → Small jump
    - Medium Volume (Normal Speaking) → Medium jump
    - High Volume (Loud Screaming) → High jump

GRAPHICS: SPRITE-BASED GAME DISPLAY

- Sprite-based character movement will be used for a more interactive visual experience.
- The VGA monitor will render real-time character jumps.
- Platforms will be generated dynamically, requiring collision detection logic.

POSSIBLE MILESTONES

- Set up the microphone interface.
- Test sound intensity detection.
- Develop the VGA-based game environment with sprites.

- Implement jump physics and collision detection on FPGA.
- Integrate audio processing, finalize UI, and playtesting.

CONCLUSION

This project combines real-time audio processing, FPGA-based physics simulation, and graphics rendering to create an engaging voice-controlled game. By utilizing Verilog for game logic and display control while incorporating software for advanced features, we aim to build an immersive and technically challenging embedded systems project.