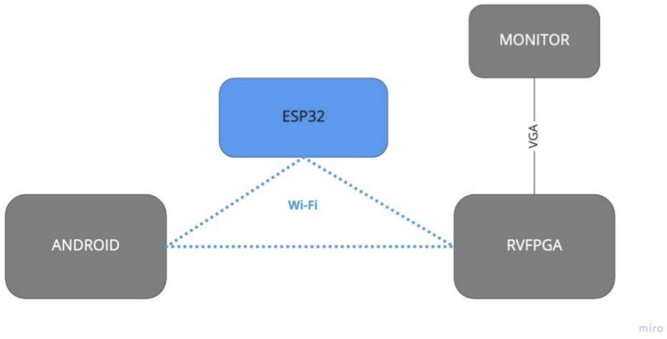
ECE 540
Alexander Maso
Rutuja Muttha

Final Project Proposal
Space Invaders & Android Controller

General Description: We are creating a space invaders game that is displayed on the 640x480 pixel monitor through the vga port of the rvfpga board. The game controller will be implemented through a custom made android app that will attach to the rvfpga board via wifi through the use of an ESP32 microcontroller.

Block Diagram



RVfpga Design: The fpga board will be used to control all of the image data as well as most of the functionality of the game. All sprites will be generated through the use of pixels (or with a coe file and block memory generator to add details once the higher priority deliverables are completed) and each will have their own module to hold position and movement data. The RVfpga will contain the SystemVerilog code and logic for the movement of the aliens, the movement of the missiles (both alien and player missiles), and the changes to interactions between the different sprites.

Android Design: The Android app will serve as the controller for the game itself and it will connect to the fpga board via wifi through the use of the ESP32 microcontroller. The android app will show the HUD for the game and display, remaining lives, time elapsed and kill count. The sound effects for the game will be controlled by the android app as well and will come from the phone itself. The pause and start menus for the game will be displayed on the android app and will include start, resume, restart and exit. Lastly, high scores and the players initials will be saved and can be displayed on the android app.

Project Division: The project division will be primarily along the lines of the SoC design and that of the peripheral android controller. Alex is responsible for using the RVfgpa board to display game images through the VGA port, control movements and changes to sprites, and respond to input from the user. This will all be done in Verilog and

SystemVerilog and implemented with Vivado. Rutuja is responsible for the game controller and connection to the fpga board through the esp32 microcontroller. Coding for the android app will be done in Kotlin and the coding for the esp32 microcontroller will be in Python. Finally, there will likely need to be a small amount of assembly or C code to help facilitate the transfer of data between the SoC and the peripherals or maybe even the player's movement and game data. Whoever completes their heir half of the design can tackle this or we can complete it together. This code will be simple and should primarily just transfer values between addresses/registers.

Details, Milestones and Goals for Alexander Maso

1. First milestone will be completed by March 9th and will include both the sprite for the player's spaceship and one alien sprite replicated into 15 aliens (3 rows of 5) each with their own module. Spaceship and alien modules will contain their location in terms of row and column and possibly the logic for their movement (logic for movement of all aliens may be in a separate module). This initial milestone includes movement of the spaceship left and right through the pushbuttons and correct movement of the alien ships: Horizontal movement left to right until they reach the edge of the screen and then down one row and right to left until they reach the other edge. Does NOT include the increasing speed of the aliens in this milestone
2. Milestone 2 will be completed by March 14th and will include the ability to shoot missiles from the player's spaceship (via the center push button for now) and will automate the firing of missiles from the alien ships. This milestone will include interactions between the different sprites. A missile hitting the spaceship will cause both to disappear (one life for now, no animation of death) and the same goes for an alien.
3. Milestone 3 will be completed by March 17th and will add the barriers the player is able to hide behind, update alien sprites to have 3 distinct species. This milestone will also include a few other sprites to emulate an animation of alien and spaceship explosions as well as the crumbling of the defensive barriers from alien missiles.
4. Milestone 4 will be completed by March 20st and will include connections with the android controller through the esp32 as well as a little bit of additional coding. It will include adding logic to increase the speed of aliens, an initial title screen, a game over screen and, if memory and space constraints permit it, any images I think could improve "animation" and want to include via coe file and block mem.

Milestones 1 and 2 as well as most of 3 are my core goals or deliverables. My “stretch” goals include the use of my own images, coe files and block memory generator to improve “animation” of movements of the different sprites as well as their interactions, the background and title screen images etc. In addition, my stretch goals also include a variable difficulty setting that can either be implemented by starting the aliens at a faster speed or could change the pattern of their movement.

Details, Milestones and Goals for Rutuja Muttha

Milestones

1. March 9th - Figuring out the connection between esp32, android app and rfpga board using mqtt protocol or any other suitable way.
2. March 12th - Creating layout or general outline for the android app and its interaction (how it passes data back and forth) with the esp32 and rvfpga board.
3. March 15th - Finishing the development of android app using Android Studio and checking whether it meets the requirements listed below (other than the HUD which requires data to come from the rvfpga board)
4. March 18th - Complete the communication with rvfpga board using the wifi module (ESP32). Add HUD for the player displaying the number of lives remaining, time elapsed and kill count). Begin to add sounds now that communication is working. Start with sounds for an explosion and for firing a missile.
5. March 20th - Debug and resolve any remaining errors (if any occur). Run through a test game. Then begin to add more complex sounds as well as high score menu/function

Details of Implementation

How the android app will work with rvfpga game:

1. Configuring the ESP32
   A. Configured with appropriate settings to establish a wifi connection
   B. Using Arduino IDE platform
2. Integrating ESP32 module with rvfpga board
   A. ESP32 module connected to rvfpga using appropriate communication interface such as UART or SPI
   B. Rvfpga board programmed to send and receive data to and from ESP32 using appropriate protocol
3. Development of Android App:
   A. Creating layout for the control buttons for controlling the spacecraft’s movement. For example, having buttons for moving the spacecraft up, down, left, and right

B. Defining the button click handlers. When the button is clicked, the app should send a command over the wifi network to the rvfpga board indicating the desired movement of the spacecraft.

C. Send commands to the rvfpga board using socket connection. The board should be programmed to receive the commands and update the spacecraft's position accordingly.

D. Update the UI as the spacecraft moves in the game. Using a timer or other mechanism to periodically retrieve the spacecraft's position from the rvfpga board and update the UI accordingly.

4. Testing and debugging the app with rvfpga board

Short term goal:
Sketching an outline for the android app and getting the ESP, the APP and the rvfpga board all connected.

Long term goal:
Develop controller app for game that includes controls to move the player's spaceship, shoot missles, and pause the game. Pausing will display a menu with options to resume, exit, or restart. The app will also display game information like lives remaining, length of current game and kill count.

Detailed list of stretch goals:
Adding audio effects for explosions as well as when the player fires a missile.
Add a simple melody using a handful of different notes. Increase speed of the melody as the aliens increase their speed