Ethernet Battleship

Marc, Apoorva, Shihab, Dan

1. Project Overview

   In this project we plan to implement the board game BATTLESHIP digitally using a DE2 board and a laptop connected via Ethernet. This will be a game for two players, with one player operating each machine. The goal of the project is to have a fairly identical experience for each player, despite the different platforms. The main goal of the game is to sink the other players fleet of ships by successfully determining their locations through trial and error "shots fired." The location of the other players ships is secret, but they receive feedback as to whether they have hit or missed with any particular shot. The ships are arranged on a 10x10 grid that is addressed through alphanumeric coordinates from A1 to J10.

2. Design Block

   Our project consists of several key elements: Ethernet (Networking), VGA Control (Display), PS2 Control (Keyboard), CPU, SDRAM Control (Memory). We are using a bus based design to connect these elements.

3. Software and Hardware Implementation
   a. Software Implementation
      i. On the DE2

         The software will mainly control the event handling of the player's shots, and also handle the Ethernet exchange of information. It will also handle the memory storage elements that keep track of the current state of the game.

         There are several events that need to be handled: The beginning of the game (ship placement), and the firing of a shot. The game play consists of initial placement, and then a series of shots until one player wins. There is only one action that a player can take in the game after initial placement, so the software has to process that event repeatedly and determine when the game is over. All other events will happen automatically based on the result of the shot fired.

         The network interface will communicate between the laptop and the DE2 board in order to keep the game state synced between the two. This means that there are several non-user generated events that the software will need to

handle that arrive as packets from the other computer. The main one of these is the signal that a shot has been fired by the other player, which will contain the shot coordinates and prompt the program to return whether the shot was a miss or a hit. Then it should receive an acknowledgement packet that indicates the data has been processed and that it is time for the next turn. All networking will be done through UDP packets.

ii. On the laptop

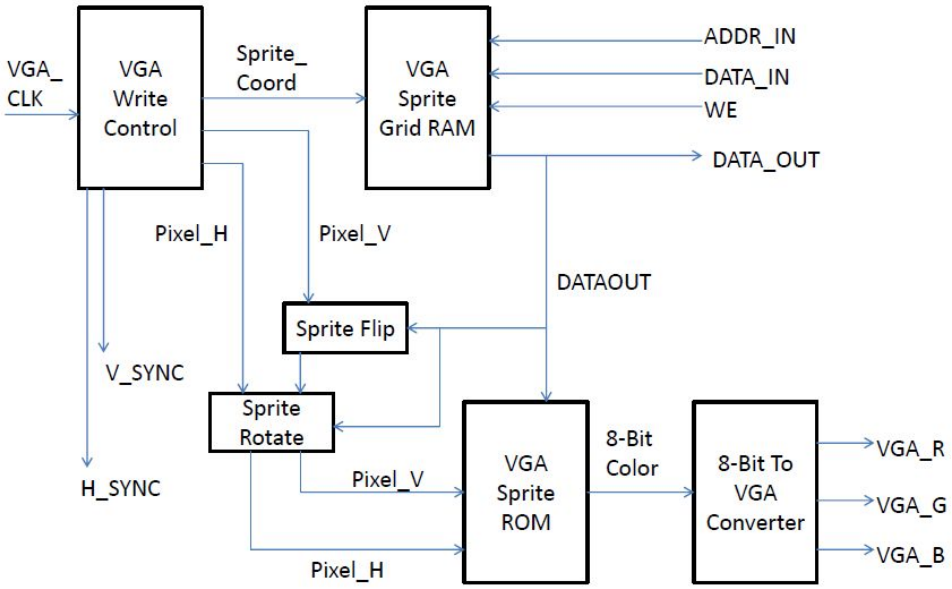
The laptop will have a program written in Java in Linux that should mimic the DE2 board in functionality, with no specific hardware elements other than the laptop screen, keyboard, and Ethernet port.

b. Hardware Implementation

The hardware is responsible for the memory, display, networking, and user input. The display will use sprites displayed upon a 16x16 grid. Each sprite will be a multiple of 16 pixels in each direction. Since the screen is 640x480, the control grid will be 40 x 30 squares. We will also have a mode where the entire screen is taken over by an animation every time a shot is fired to display the result of the shot in a flashy manner.

4. VGA Control

We will use a specialized memory on the board that keeps track of which tile is displaying at each control square. The memory on the board will keep track of 16x16 sprite tiles and display them every frame based on the current state of the memory array. This will be done through the use of hardware components that create a VGA signal out of the array of tiles. Our VGA Control Flow looks like this:



5. PS2 Control
   We use the implementation of the PS2 controller from lab2 and handle keystroke processing from the software side.
6. Ethernet Control
   Ethernet will be our only way of communication between the 2 players, one on the DE2 board and the other on the PC. UDP is going to be used as the transfer protocol for its simplicity. It is unreliable and is based on best effort as it does not ensure packet delivery and handle dropped or lost packets. We will thus implement our own simple handshaking protocol to ensure reliable communication. This can achieve TCP's reliability without the need to implement the advanced TCP controls that can manage bandwidth control, etc. As done in lab 2, we will generate well formed Ethernet packets. We propose connecting the DE2 with PC directly though an Ethernet cable. Since only the 2 participants are connected in this mini-network, we do not have to worry about properly assigning IP address and instead we broadcast our packets.
7. Memory
   We need several chunks of memory for this game:
   a. 10x10x1 memory that stores whether a ship is located at each grid location
   b. 10x10x1 memory that stores whether your opponent has fired at a particular location
   c. 10x10x1 memory that stores whether you have fired at a particular location
   d. 10x10x1 memory that stores whether you hit a particular location
   e. 5x4x1 memory that stores the starting location and direction of each of your ships
   f. 2x4 memory that stores each players score

   All of these will be implemented in software, allocated into the SRAM on the board.

   We also need specialized memory on the board that keeps track of which tile is displaying at each control square, so that it can be accessed every frame regardless of the software's use of memory. This will get updated by the software whenever it decides that he screen needs to change somehow, but does not need input from the software for each frame. The memory on the board will keep track of 16x16 sprite tiles and display them every frame based on the current state of the memory array. Because it needs to store a 40x30 grid of tiles, and we will have around 400 tiles, the memory needs to be 40x30x9.
8. Milestones
   a. Milestone 1 - Integrate PS2, Ethernet, and VGA into one system that we can improve upon in further milestones.
   b. Milestone 2 - Implement VGA display functionality with the tiles and full network functionality with the laptop.
   c. Milestone 3 - Implement full game logic and user input on both laptop and DE2 board. Test and debug.