Battle City Game

Team Name: Tank

Group Member: Han Cui hc2737, James Thompson jlt2160, Li Qi lq2156

Instructor: Stephan Edwards

1. Design Overview

The project's purpose is to build the classic video game *"Battle City"* using synthesizable Verilog HDL and C language on a FPGA board. Users should be able to play this 3rd-person multidirectional shooting game using keyboard as controller and monitor for display. To finish this, we need to design hardware units using with Verilog HDL for functions of configuring VGA and total game display, scoring on FPGA board, sound effects and keyboard control; as well as to design software units using C language for tank movement and application of game rules.

1.1 Game Rules:

The basic rule of the game is that players should control a tank to reach the higher level of game by destroying all the enemy tanks while keeping the home safe from enemy tanks. There are 2 enemy tanks in total to be generated at random positions on the edge of the game window in the beginning of each game. And their speed would increase drastically at higher level games, thus harder for players to win. Player's tank would generate at a fixed position right next to the home for every game and should expect 2 chances in total to restart before the end of each round. It's considered players losing the game when they used all the chances of restarting before finishing all the levels of the game when there are totally 3 levels to pass.

1.2 Overall Structure Description:

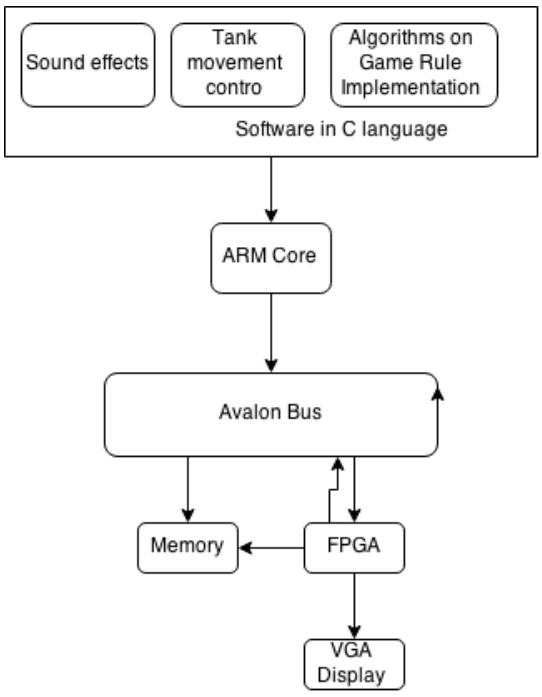

Figure 1. Overall structure

Our design includes software part which is in C language, this part contains code

to depict tank movement, which allows users to control the tanks with keyboard; algorithms that apply the game rule, which would determine if players have died, lost or won, as well as to allow players to pause, resume, start and exit the game; Also this part would generate sound effect according to player's control or the application of game rule like being hit, game over, etc.

Using ARM core and Avalon bus we could bridge the software in C and hardware designed in Verilog HDL to function together. Also the hardware contain VGA display which would generate the everything belongs to the game window. And the current game window should be generated with keyboard command the previous game window stored in the memory. The function of the hardware design would be achieved on the DE2 FPGA board provided.

However the current design is not the final decision on the implementation of the game. The sound effects might be achieved in hardware part, and we expect more stuff to be added to the design.

1.3 Basic Unit Description:

1.3.1 Keyboard:

Keyboard is used to control the total game process, all the functions of the game should be realized with the use of keyboard. These functions include systematic functions such as pause, resume, start and exit; as well as gaming functions such as going forward, going back, making turns and fire. The functions of the keyboard are designed both in Veirlog HDL and C language

1.3.2 VGA Display:

VGA is hardware designed in Verilog HDL to correctly connect the FPGA board with functions loaded to the monitor, as well as to display everything in the gaming window. VGA display would interact with memory which consists of RAM and decoder to generate display at the current moment with respect to the display stored as the previous game window. This is the core part in terms of game display.

1.3.3 Memory

Memory is designed in Verilog HDL with RAM for storage and decoder for distributing address for storage. The memory should be of enough size to store every pixel of the game window.

1.3.4 Avalon Bus and ARM Core

Avalon Bus and ARM Core are used as interface to bridge controlling from C code to display generated by hardware in Verilog HDL.

1.3.5 Tank Movement

This part contains C language code for controlling movements of player's tank in display according to keyboard press by player.

1.3.6 Algorithms on Game Rule Implementation

This part contains the code that implements the total game rule, including enemy tank generation placement, movements, speed control and attacking pattern; as well as the determination of player's death, restart, going into next

level, loss or winner. This is the core unit in terms of game playing.

1.3.7 Sound effects

This part contains C code that generated preloaded sound effects according to the player's control and game algorithm determination. However we might also implement this part with Verilog HDL according to the display. Writing in C language is easier for generating sound while Verilog design would allow the sound effects to be generated accurately.

2. Milestones

Milestone1:

- ✓ VGA Display successfully implemented
- ✓ Keyboard successfully implemented in Verilog HDL
- ✓ Memory successfully implemented in Verilog HDL
- ✓ Start testing hardware units with images

Milestone2:

- ✓ Tank movement keyboard control successfully implemented
- ✓ Algorithm of game rule successfully implemented
- ✓ Sound effects successfully implemented
- ✓ Successfully testing the standalone functions of all software units

Milestone3:

- ✓ Avalon bus and ARM Core successfully configured
- ✓ Clock synchronization successfully implemented
- ✓ Ability to test the functioning of the whole game

Appendix:

1. *Worm Craft report from Spring 2014 CSEE 4840 class"* http://www.cs.columbia.edu/~sedwards/classes/2014/4840/reports/WormCraft.pdf
2. *Avalon memory-mapped interface specifications* *http://www.cs.columbia.edu/~sedwards/classes/2008/4840/mnl_avalon_spec.pdf*
3. *Altera's Avalon communication fabric* http://www.cs.columbia.edu/~sedwards/classes/2015/4840/avalon.pdf