Department of Electric & Electronic Engineering,
Boğaziçi University

EE 240 FINAL PROJECT

RICE FARM GAME

Veysel Cemil ÇOLAK-2016401225
Kaan İÇER-2017401072

Project Advisor : Prof. Dr.Şenol MUTLU

27.05.2019

Table of Contents

1 INTRODUCTION

In this report, the process to create a game called RICE FARM is explained as Problem Statement, Background,Design,Definitons, Results and Conclusion .

2 PROBLEM STATEMENT

In the RICE FARM game, the user has 3x3 fields and each field is surrounded by streams. User should turn on the switch [SW(1)] to start or end a game. The red cursors help user to indicate the subfield that the user has chosen currently. The seven segment display on the FPGA shows coordinates with respect to y-x axis, harvest count and count of fields waiting for harvest. User starts to play this game from the field at the top left corner with 0 harvest count and 3x3 empty fields. The empty fields are dark green and planted fields are dashed brown on white. User can move left by pressing BTNL button, can move right by pressing BTNR button, can move up by pressing BTNU button and can move down by pressing BTND button; by this movements red cursors also will move. If the field that user has chosen is not planted, user can plant this field by pressing BTNS button. By this action, number of fields waiting for harvest increases by 1. If the field that user has chosen is planted, user can harvest this field by pressing BTNS button. By this action, number of harvest increases by 1 and number of fields waiting for harvest decreases by 1. When user harvests the 10th time, the game and all related data are reset and user goes back to the initial state.

3 RELATED BACKGROUND

3.1.D flip flop: This flip-flop stores the value that is on the data line.

3.2.J-K flip flop: The JK flip flop is basically a gated SR flip-flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level “1”.

3.3.Debouncing: Debouncing is any kind of hardware device or software that ensures that only a single signal will be acted upon for a single opening or closing of a contact.

3.4.Front porch and back porch: The red, green and blue signals are blanked during the 6.6 μs interval comprised of the front porch, synchronization pulse and back porch for horizontal synchronization signal and are blanked during the 1.534 ms interval comprised of the front porch, synchronization pulse and back porch for vertical synchronization signal.

4 DESIGN

Our design consists of 8 main parts: Frequency divider(freq_dvd), debouncer, horizontal synchronization signal generator(hsync_gen), vertical synchronization signal generator (vsync_gen), game controller, screen manager, BCD to seven segment converter(BCD2SS) and seven segment driver(SSEG_DRI). The names for subcircuits used in Xilinx are declared in paranthesis.

4.1.freq_dvd: We designed frequency divider to generate appropriate frequencies for horizontal signal generator, vertical signal generator and debouncer. We divided board clock's frequency by 2^4 by using 4 J-K flip flops that we designed for signal generators. For debouncer, we divided board clock's frequency by 2^20 by using 20 J-K flip flops.

4.2 Debouncer:Debouncer is designed to expect a single contact to be recorded by buttons of FPGA. For this, 4 D flip flops are connected series for each input buttons of FPGA(BTNS, BTNL, BTND, BTNU, BTNR) and clock having the ½^20 frequency of board clock.

4.3-4: hsync_gnrt & vsync_gnrt: Horizontal and vertical synchronization signals are generated by using the clock having the ½^4 frequency of board clock. In this part, time ranges (back porch, front porch and display time) of signals for 640*480 screen are arranged by using lab manual for LAB 7.

4.5 Game controller: The subcircuit that controls which field to be planted or to be harvested, number of fields which is not planted, harvest number and coordinates with respect to y axis and x axis.

4.6 Screen manager: The subcircuit that reflects the image on the screen with the desired colors using the inputs from signal generators and game controller. Fields' image, cursors' movement and streams between the fields are reflected onto screen by this part.

4.7 BCD to seven segment: The subcircuit that takes the coordinates of user and converts these numbers to seven segment as input for seven segment driver.

4.8 Seven segment driver: The subcircuit that determines which digit on the FPGA works and which number will appear.

These 8 main parts are combined in a VHDL MODULE source.

5 RESULTS

Streams between the fields, cursors and fields are reflected onto screen successfully. Game for user who pressed BTNL, BTNU and BTNS buttons one time appears like in Figure 1.

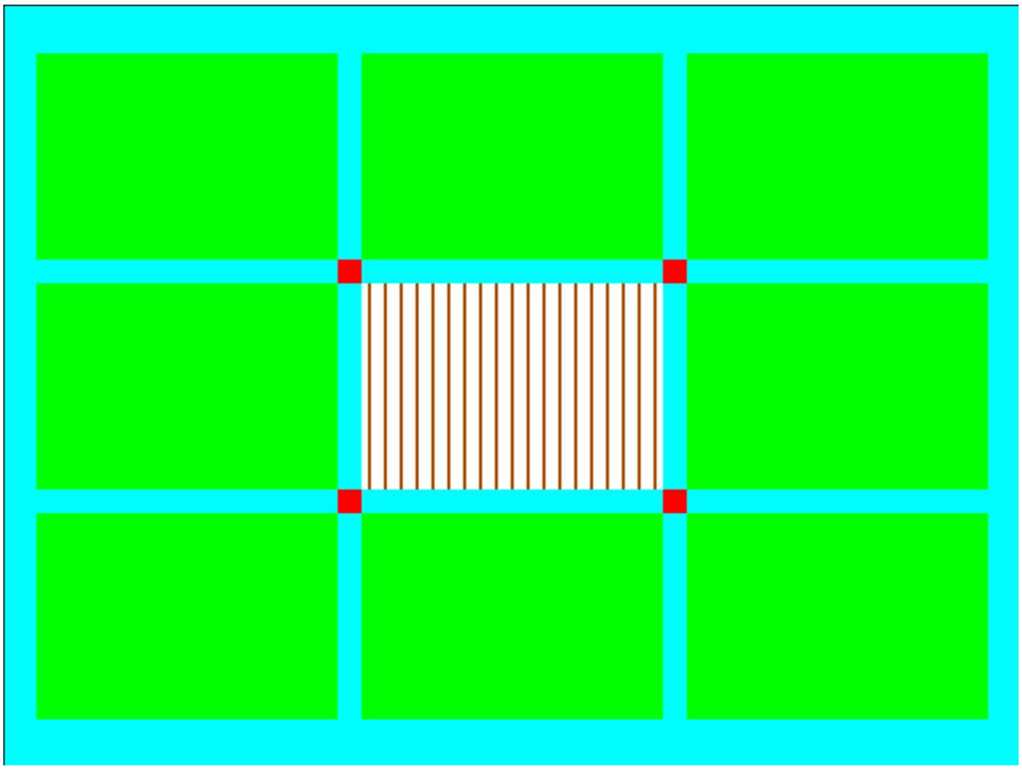

Figure 1: Game scene .

6 CONCLUSION

In proposal, we aimed to act on only a big field consisting of 9 subfields adjacent to each other. But in project we separated the fields by streams to make the game more realistic. Also we reflected brown lines on white color for planted fields, since we wanted to show soil between rice plants. We added cursor movement to show where user is when he/she moves. These things are not mentioned in proposal but we add them to make game more attractive and realistic. Except these things; coordinates, harvest count and number of fields waiting for harvest are shown on seven segment display.
The only problem we may encounter is to get ideal vertical and horizontal dimension ratios since the monitors in lab are 1200x960 whereas our design is prepared for 600x480.

7 REFERENCES

1. Şenol Mutlu and H. Işıl Bozma. Laboratory Manual. Version 7.0. Bogazici University Department of Electrical & Electronics Engineering, 2015.

APPENDIX

A READ ME

This game is driven for 640x480 VGA display.

B USERS' MANUAL

Switches and buttons required to play RICE FARM is given below.

SW(1): When the user turns on SW(1), he/she starts to play RICE FARM game. When he/she turns it off, all data are reset and game is not reflected onto screen.
BTNS: User can plant the empty field and can harvest the field that waiting for harvest by pressing BTNS button.
BTNL: User can move left by pressing BTNL button.
BTNR: User can move right by pressing BTNR button.
BTNU: User can move up by pressing BTNU button.
BTND: User can move down by pressing BTND button.

Cursors and first two digit of seven segment display on FPGA shows where the user is, the last two digit shows harvest count and number of fields that waiting for harvest.