

CE339 Assignment 2: VGA based video game

Set by: Dr Xiaojun Zhai (xzhai@essex.ac.uk)
Distributed to students: week 20
Submission deadline: see FASer
Feedback: three weeks from submission deadline
Submission mode: electronic only via FASer

Assignment objectives

This document specifies the second coursework assignment to be submitted by students taking CE339. This assignment is more challenging than the first one and it is meant to provide an opportunity to improve the knowledge of the VHDL language and, more importantly, to design a digital “system”. You will be expected to learn to: a) implement digital system design in VHDL code; b) synthesise and simulate it; c) test, debug, and verify that the design meets the specifications; d) report about your design.

You are required to design code for your target hardware (a Digilent Basys3 board with a Xilinx Artix 7 FPGA) in order to implement a design that meets the specifications (below). Your design will be validated in Vivado simulator. You are required to submit working and correct code and you are strongly encouraged to use a modular coding style (allowing for greater flexibility, maintainability, modularity, and reusability). To show that you master all aspects of the language, your code should prevalently use concurrent statements for combinatorial circuits and sequential code for sequential circuits. Additionally, the use of non-standard packages (e.g. `STD_LOGIC_ARITH`, `STD_LOGIC_UNSIGNED`, `STD_LOGIC_SIGNED`) and `BUFFER` ports is forbidden, while the use of `INOUT` ports is accepted only when strictly necessary.

You are supposed to gain familiarity with VHDL coding during the supporting CE339 lectures and through self-study hours, also with the help of the recommended textbooks or any other book about VHDL. You are expected to work on this assignment mostly during lab hours. Your design project should be stored under the revision control repository that was assigned to you at the beginning of the course. You are supposed to commit often and describe your progress in the commit messages. In order to promote a learning scheme that values the learning process in addition to the submitted final design, your weekly progress (as traced back by the commit logs) will contribute to your assignment mark.

Design specifications

Your task for this assignment is to implement a “bouncing ball” video game. To make the assignment feasible within the time frame available for this module, you are not required to implement the full game, but only a simplification of the phase of the game is required.

The basic layout should include a stick and a ball. The shapes representing these three elements can be very simple, like those in the mock-up in Figure 1. The ball starts, following a linear trajectory, towards the bottom of the screen. If the ball hits the stick a point is awarded. In the interest of time, this condition can be simplified and a point be awarded whenever the ball somehow hits the boundary of the stick. The default screen resolution is 640×480 .

By using the controls available on the Basys3 board, the user determines the speed of the ball. The 4 leftmost switches on the Basys3 board encode the initial horizontal speed of the ball, the

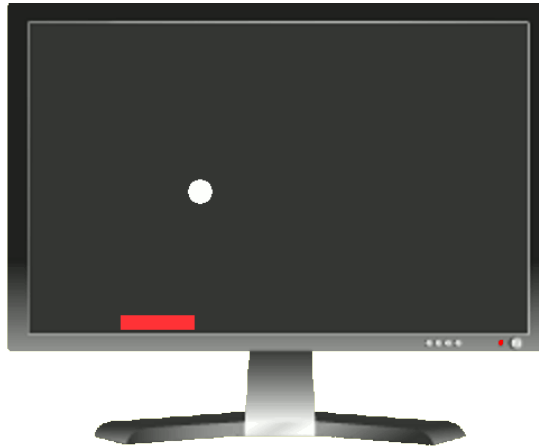


Figure 1: Mock-up for the simplified “bouncing ball” game, final result of this assignment.

next 4 switches encode its vertical speed, and the central button is used to start the ball. The left and right buttons are used to control the movement of stick. The stick ($Width \times Height$ is 60×20) starts from the bottom left corner as the initial position of the game, and it can be controlled by left/right buttons (e.g. each click on the right (left) button will move the stick horizontally in right (left) direction by x , the default value for x is 20). The ball (Radius $R = 20$) starts from the top left corner as the initial position of the game, the initial speeds (pixels per frame) in vertical and horizontal directions can be set by the 8 switches.

These specifications should be interpreted as guidelines and should not constrain you from improving the game by designing better graphics, more realistic criteria for awarding the point, and other modifications that you think would result in a better “product”.

Report

You should write a report introducing the project and then describing your codings and designs for the assignment. It should first describe the design at a higher level and then detail the implementation of each module in a top-down fashion (rather than in chronological order). Also report if your code worked at the first attempt, what was wrong and how you fixed it. The repository log (if you used it properly) should help you considerably in this task. Your report should also include a discussion of design alternatives that you considered and the motivations for your final choice.

Your document should have a title page and be sub-divided into appropriately headed sections. It MUST contain references to material used in your work (e.g., VHDL code or information available in books or on the Internet). All of the VHDL code submitted should be included in the report in the form of code appendices and be typeset in Courier font (or a suitable fixed-width alternative font of your choice). Your report should consist of approximately 1500–2000 words of narrative (i.e. excluding references, code fragments, pictures, diagrams and schematics). Please write registration number and word count on the title page of your report.

Submission

Your work must be submitted to the university’s online FASer submission system at the address <https://faser.essex.ac.uk/> by the deadline given on the system. No other mode of submission is acceptable. You are strongly advised to upload a draft submission before the last lab hours prior to the deadline, and then update it up to the deadline.

You are required to submit one ZIP archive (not RAR or other formats) containing the following files:

1. All source files needed to synthesise your project (but not the temporary files created by Xilinx);

2. A report document submitted in PDF format. DO NOT SUBMIT THIS FILE IN .DOC OR .DOCX OR SIMILAR FORMATS - SUBMIT PDF.
3. Your repository log in .TXT format.
4. If you want (in your own interest, see next section), submit a .TXT or .PDF file with the transcripts of relevant forum discussions you contributed to.

DO NOT WAIT UNTIL CLOSE TO THE DEADLINE TO MAKE YOUR FIRST SUBMISSION. Difficulties with the submission system will not be accepted as an excuse for a missing submission.

Marking criteria

This assignment is worth 40% of the module mark. Marks will be awarded for the VHDL code, including coding style and quality, and for the descriptive document, including content, presentation, and discussions. In addition to the report, each students will be expected to demonstrate and explain her/his design with confidence and competence during a demo lab session. Marks will be assessed based on:

- Implementation
 - Quality of the implementation30%
 - Modular design6%
 - Generic and re-usable code6%
 - Proper use of comments6%
 - Steady progress, adequate use of GIT6%
- Report
 - Correctness and completeness of report12%
 - Clarity of presentation12%
 - Organisation of report12%
 - Quality of diagrams and schematics8%
- Others
 - Compliance with submission instructions2%

Marks below 100% can earn additional credits if the student actively engaged in forum discussions asking pertinent questions and giving competent answers to questions raised by classmates.

Late Submission and Plagiarism

Please refer to the Students' Handbook for details of the Departmental policy regarding submission and University regulations regarding plagiarism.

Revision 1.1
15/02/2021
Xiaojun Zhai