3 Body Collision Simulation

Amey Gaikwad (15D260002)
Rahul Dandwate (15D260006)

April 4, 2017

1 Abstract

The objective of this 1 month long project was to design and simulate the movement of 3 independent balls subject to the constraint of collisions among the three of them and the walls of the VGA Display delimited by numbers given by the user. The simulation has been done with the help of an FPGA board in the VHDL language. VHDL is not the most ideal way to simulate the 3 body problem, however the parallelism of the language was used for refreshing the frame at the same time as the motion of the three bodies is being animated. Non triviality of the language and inexperience in using the parallelism in the most optimum way possible presented difficulties. 12% of the logic elements of the board have been used and further building up on the present logic lines can extend the present problem to N bodies easily, albeit the fact that the processor significantly reduces in its accuracy with increase in complexity. Minor glitches due to the limitation of the frame refresh rate and maybe the algorithm can be corrected by a detailed look into the way the sync mod module generates the output on the VGA. Only the integer library of VHDL has been used in our simulation.

2 Implementation

2.1 VGA Display

The output has been displayed on a VGA display of resolution 640 by 480. The refresh rate of the VGA display 60Hz. The Horizontal and Vertical resolution attributes for a resolution of this nature can be found on the web. A module was designed for feeding the video into the VGA. This is the sync_mod module. A separate module img_gen was developed was developed for generating the three balls and animating them with given and user defined initial velocities and position on the VGA monitor. A final module was developed for combining the code and giving the final output. A detailed working in the form of block diagram is given in the following pages.

2.2 Ball Animation

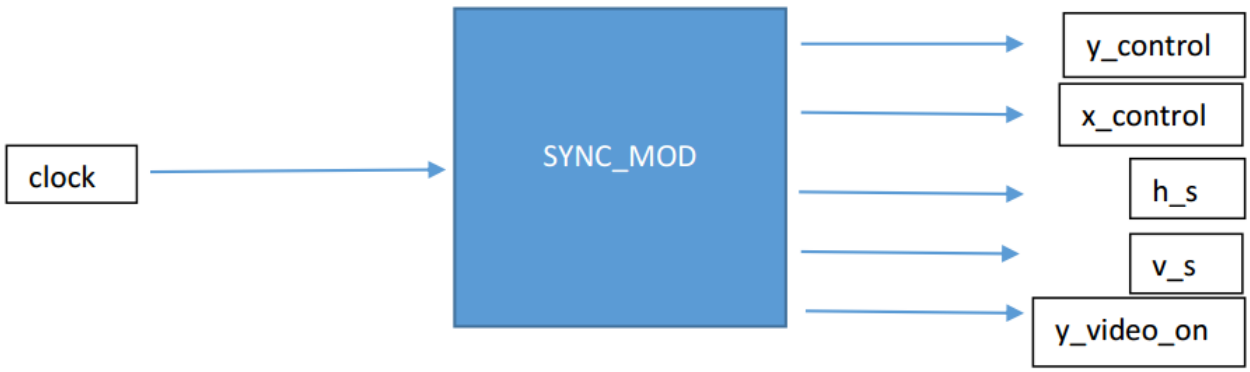
The conditions for the collisions of the balls with each other are explained in this section. On collision, depending from which side the balls collide, the velocities in that direction of the two balls is exchanged with the other component remaining the same for both the balls. Boundary conditions have been placed in the form of virtual walls at the edges of the VGA so that the balls do not disappear out of the display. On collision with the wall, specific velocities predefined by the user are imparted to the balls, as implementing a wall with coefficient of restitution 1 leads to the formation of combinational cycles leading to errors in the compilation. Because of this reason the above approximation has been made for ease of logic and implementation for the first project using FPGA. This problem can be supposedly overcome by creating some new variables.

2.3 Ball Acceleration

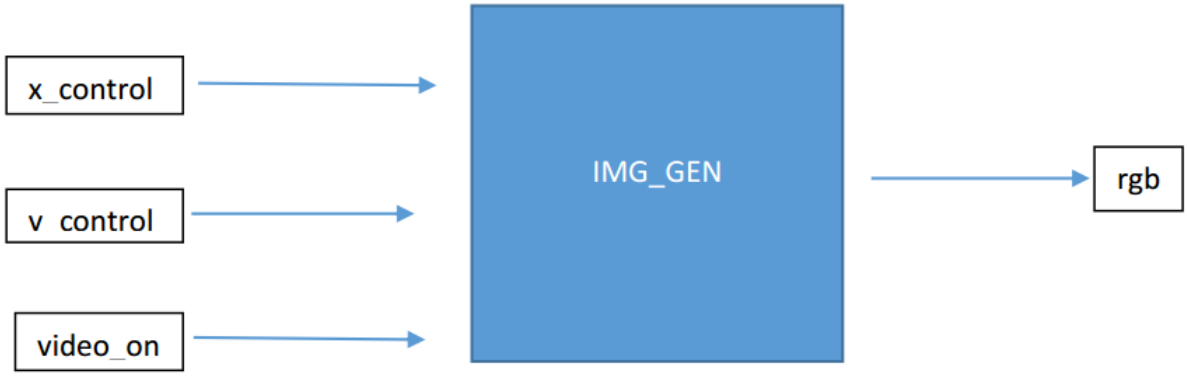
Attempts were made to add acceleration to the balls. Because of using only the integer library, without invoking the fixed point library, adding acceleration was a challenge. A counter was developed which updated a variable to 2^{25} to slow down the counter to add acceleration, but without success. Preliminary trials at using the fixed point library led to changing the code a lot. So it was sacrificed for an additional ball to be displayed.

3 The Modules

3.1 Sync_Mod



3.2 Img_Gen

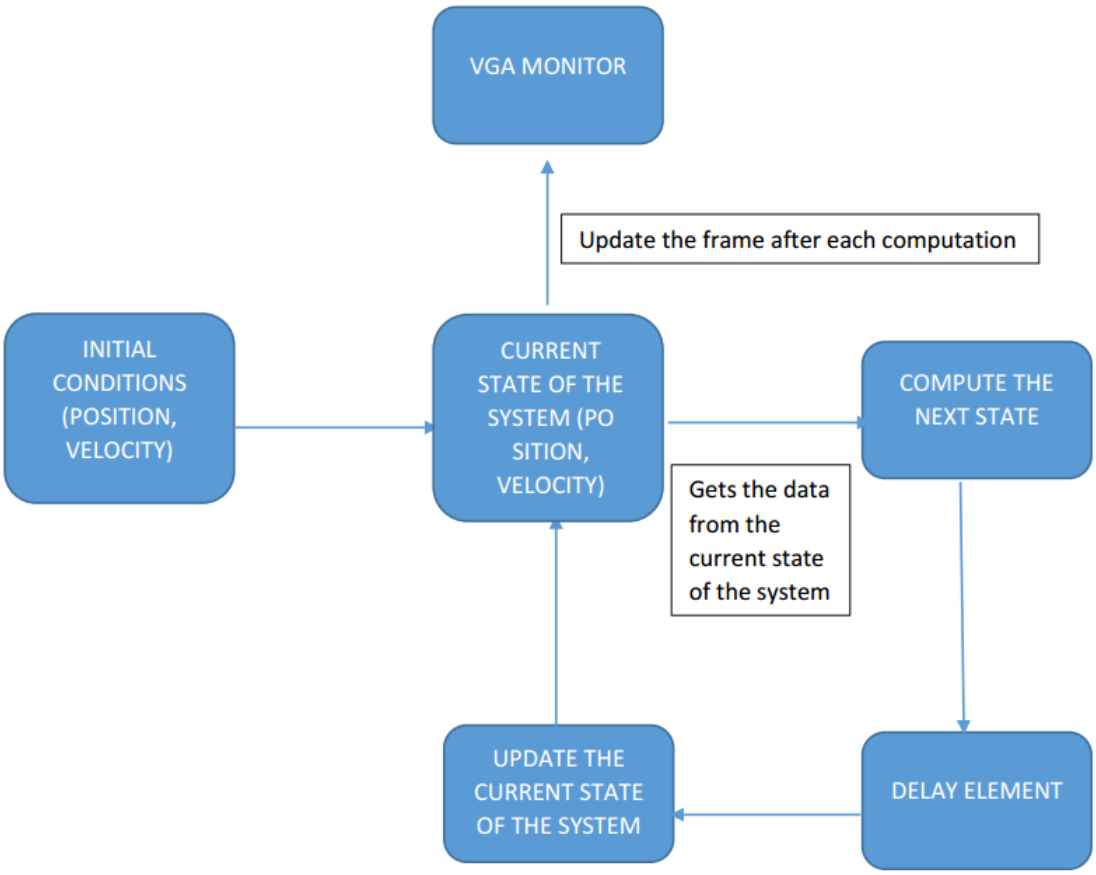


Note: The rgb signal output shown in the Img gen module above refers to three outputs corresponding to the three balls.

3.3 VGA_Control

Combines the above two modules. Both the modules are run by the same clock.

4 Block Diagram



5 Weekly Distribution

5.1 Week 1

- Developed the hardware required for the correct working of the VGA Display.
- Learned VHDL.
- Showed something on the VGA.

5.2 Week 2

- Showed the 2 balls on the screen.
- Added motion to the balls.
- Added some rough constraints.

5.3 Week 3

- Developed the algorithm to show two body collisions.
- Tried adding acceleration to the balls using fix point library and using the counter to slow down the iteration, else the balls disappear from the screen.

5.4 Week 4

- Added final touches to the correct working of the two body collider.
- Successfully extended the algorithm to make a three body collider.
- Tried adding acceleration with success. One dimensional acceleration was successful but with some glitches.

6 Distribution of work

6.1 Amey

1. The two body problem
2. Extension of the algorithm and the code to 3 bodies
3. Fix point library.
4. Debugging

6.2 Rahul

1. Setting up the hardware for VGA Display.
2. Two body problem.
3. Designing the counter for acceleration problem
4. Debugging

7 Challenges on the way

- The clock on the FPGA board works on 50 MHz clock cycle. However for the resolution we are operating the VGA on the ideal clock frequency is 25 MHz. For overcoming this, a modulo counter is used to counter the issue.
- With absolutely no prior knowledge of VHDL, developing the 3 body simulator, albeit a few problems was a huge achievement for us as the learning curve was pretty steep.

- Adding the acceleration to the balls as a secondary project was tough, but was successfully done for a single direction. Developing the counter to slow down the iteration as substitute for using the fix point library was tough.
- Implementing and trying the fix point library in its minimal stature. The library was ultimately not used by us at all, but a lot of time was given to understand it and trying to implement it into the code.
- Initial trial with the code for the 2 body collisions showed many errors. 3 new algorithms were developed over the course of 2 weeks trying to make the bodies collide well and with as less errors as possible. The final algorithm is the best of the three, albeit the fact that it does come with its own set of glitches.

8 Testing and Verification

- Change the initial velocities of the 3 balls.
- Change the relative position of the three balls on the VGA monitor.
- Change the color of the three balls.

9 Conclusion

The 3 body simulator we have developed can be easily extended to the case of N bodies without much trouble. The counter developed also neared its completion and if given enough time, can be developed into a viable alternative for the fix point library for acceleration simulation. It was a huge task to make this project and we are very proud to finally see our hard work bear fruition. We hope you enjoy our project as much as we did making it.

10 Acknowledgements

We would like to thank Prof. Sarin, Mr. Pawar, Mr. Amol, the lab staff and the TAs Himanshu and Vaishakh for their immense help in this project. None of this would have been possible without them.

11 Bibliography

1. FPGA Centre: http://fpgacenter.com/examples/vga/index.php
2. EdaBoard : http://www.edaboard.com/group3.html
3. Cornell University : https://people.ece.cornell.edu/land/courses/ece5760/FinalProjects/