

Project:

LEARN CHINESE FPGA GAME

Team JCPenny

Osakpolor Evbuomwan

Chris Theriot

Joshua Young

Nick Hermann

ECE 5440

Introduction

Our team's proposal focused on game memorization. The concept of the game begins by showing four different patterns that we've created and associating them somehow with a switch or button. Once the player understands what pattern correlates with which button, then they can press a button to start the game. Every round a different pattern will be displayed on the 7-segment decoder and the player must then append that pattern to the list of patterns they've already seen. After thinking about a game that could use this idea, we incorporated a program dedicated to learning Chinese number characters. This program will display four random Chinese characters where the player will have to press key button that matches a certain character. These Chinese numbers are 1, 2, 3, and 7 which are shown in Figure 1. The player is expected to memorize a button combination. The key combo will be matched to four different Chinese numbers.

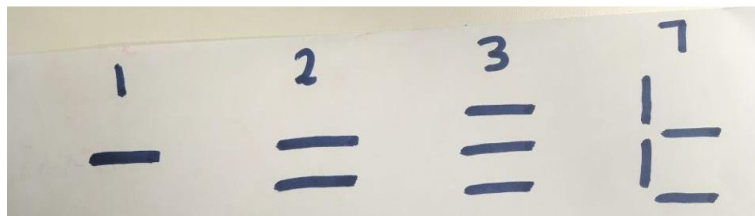


Figure 1. Chinese numbers and their respective symbols.

Each of these Chinese numbers requires a confirm button that must be pressed after each combo. Afterwards the player has 30 seconds to match as many random symbols as possible. If the symbol is matched correctly, then the player gains a point. Towards the end of the game, the highest score is displayed on the segment decoder.

System Architecture Design

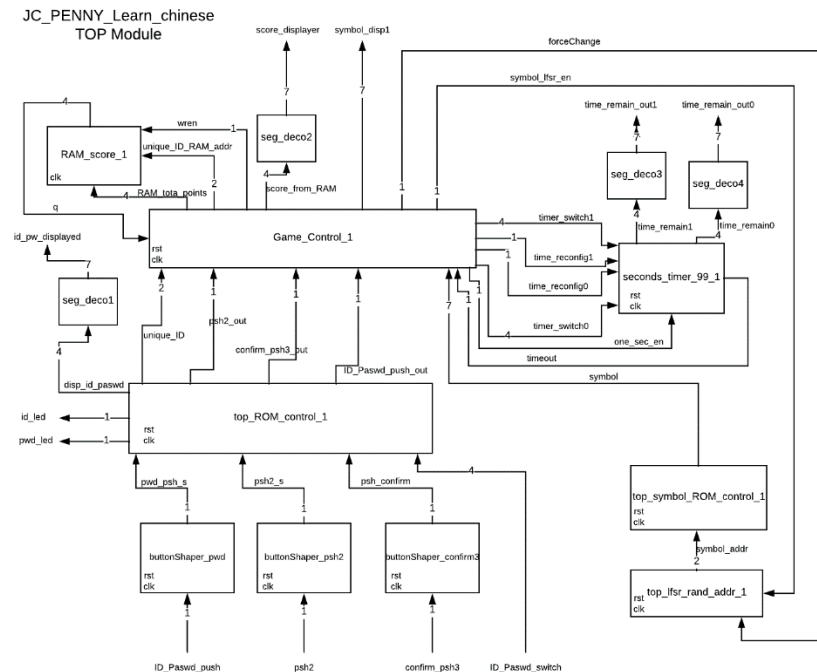


Figure 2. Top-level system architecture of LEARN CHINESE FPGA GAME.

This is our project's top-level design. It first gets authenticated through the ROM, then it reads ID from the ID Rom. Afterwards it then reads the password from password rom. Finally, it sends a unique ID to the game control module. This is where the game have the player will go through a sequence of an ID check and a unique password check using the seven segment displays.

Each team member has a unique ID and password:

Member	ID	Password
Osakpolor Evbuomwan	5273	1234
Chris Theriot	3841	1248
Joshua Young	7752	FFFF
Nick Hermann	3373	1111

When the checking is done, the player will start playing the game and matching the Chinese numbers.

Demo Video Link

<https://drive.google.com/file/d/1LyvDzgzwR0ITvhdleR0gzFyzdAfNi6jG/view>

The link above shows a demonstration of how to play the "Learn Chinese FPGA Game". First the game requires an ID check, so the player must input his or her ID via switch and button input.



Figure 3. Inserting number for ID checking.

Once the ID has been inputted, a green LED will light up to let the player know that the ID was correct. Afterwards the game will require a unique password to input.



Figure 4. Inserting unique password.



Figure 5. Timer and Chinese character displayed on FPGA.

After confirming the password by pressing the confirm button, the timer will countdown on the segment display and a random Chinese character will be displayed as well. The demonstration ends with showing how the game can be played.

Lessons Learned

Our team learned that teamwork is effective. There were a lot of time conflicts during the coding phase of the project. Some of the team members had other ongoing projects with their own teammates and there were others who had to work their daily shift. In the end, we had someone who was willing to find a time that could work for everyone even though if it meant doing the project late at night. Another lesson learned was that a working simulation does not guarantee a working FPGA. Our team focused on making sure the ModelSim could successfully compile every Verilog file, but our FPGA board wouldn't perform the duties that were intended. There were many times when Quartus produce more than 20 warning errors and our team had to debug and research what each warning line meant. During the debugging stage of our project, our team realized that it was better to redraw the architecture and analyze every line of code from the beginning to look for logic bugs. Another lesson that was learned was that it is very important to make a version of every important update and make a dedicated folder

towards that update. Sometimes we would update the code and try to compile using Quartus, but our team mistakenly used the wrong Verilog file. With every update, we made developer notes of every issue and update that occurred. Overall, our team realized that making developer notes were very important because it helped everyone keep track files that needed attention.