**ECE241 - Digital Systems Final Project Report**

Margaret Riese 1001199151

Shiyao Chen 1001792474

December 7th, 2015

**Introduction – description of goals of project; motivation**

The idea behind our project was to create a game that would demonstrate what we had learned in Digital Systems. The game prompts the player to rearrange six animals printed on the screen in an arbitrary order given and times how long the user takes to complete the task accurately. The game can also take in a display name, of the player’s choice, to go with their time. The motivation behind the project idea was that we both love playing games and we are interested in game development and design as a possible career path in the future.

**The Design**

****

**Major Design Modules**

* *VGA adapter*
  + The VGA adapter controlled the image output on the screen. Depending on which state the game was in, a different screen image would be printed.
* *Sequence Detector*
  + The sequence detector included a state machine that controlled the entire game. It utilized many smaller modules to judge when the state should be moved from the ‘current’ state to the ‘next’ state depending on our state table and called the game’s functionality modules accordingly. For example, in the ‘play’ state the ‘Game Answer’ and ‘Timer’ modules were utilized as they controlled the main gameplay. In the ‘name’ state, the player was able to input a 3 character username, using switches zero through four, to accompany their time. The state was primarily moved forward by the player pressing KEY[0]. While in the ‘play’ state, depending on whether the players game answer was correct or not, the pressing of KEY[0] would lead the player to either the ‘lose’ state or the ‘win’ state accordingly.
* *Control Path*
  + The control path dealt with counting the individual 160x120 pixels of each image readying each colour input to be sent to the VGA adapter. It also contained an FSM to control when the screens were changed. It could also have been used to print and redraw sprites in the event that we had extra time to implement that particular component.
* *Data Path*
  + The data path organized outputs to be sent to the VGA adapter. Originally it was able to print an entirely black screen which we then modified to print images stored in RAM files.
* *Game Answer*
  + The game answer module accepted signals from switches one through six and stored their values consecutively in separate RAM modules, using KEY[3] to load. The values were then sent to a small branching module to judge whether the values stored in the RAM were the same as a preset sequence, to determine whether the player had won the game or not.
* *Timer*
  + The timer was modified from one of our previous adder labs. The clock was set to a certain speed corresponding to counting in seconds and the clock controlling the timer was enabled only when in the ‘play’ state, meaning it would stop only when the player pressed KEY[0] to proceed to either the ‘win’ or ‘lose’ states, therein recording the time taken for the animals to be rearranged. A new HEX display module was also built so that the counter would not include letters ‘A’ through ‘F’ while counting.

**Did it work**

Our project was very successful. We utilized and modified many of our previous labs and combined the separate pieces into one whole new project. Although we did not meet our second milestone in time, in the end the game performed as intended; it asked for the player to enter a three char username, a clock timed how long it took for the player to rearrange the six animals, and depending the player’s input either the ‘win’ screen or the ‘lose’ screen was displayed.





**Parts that didn’t work - speculate as to why**

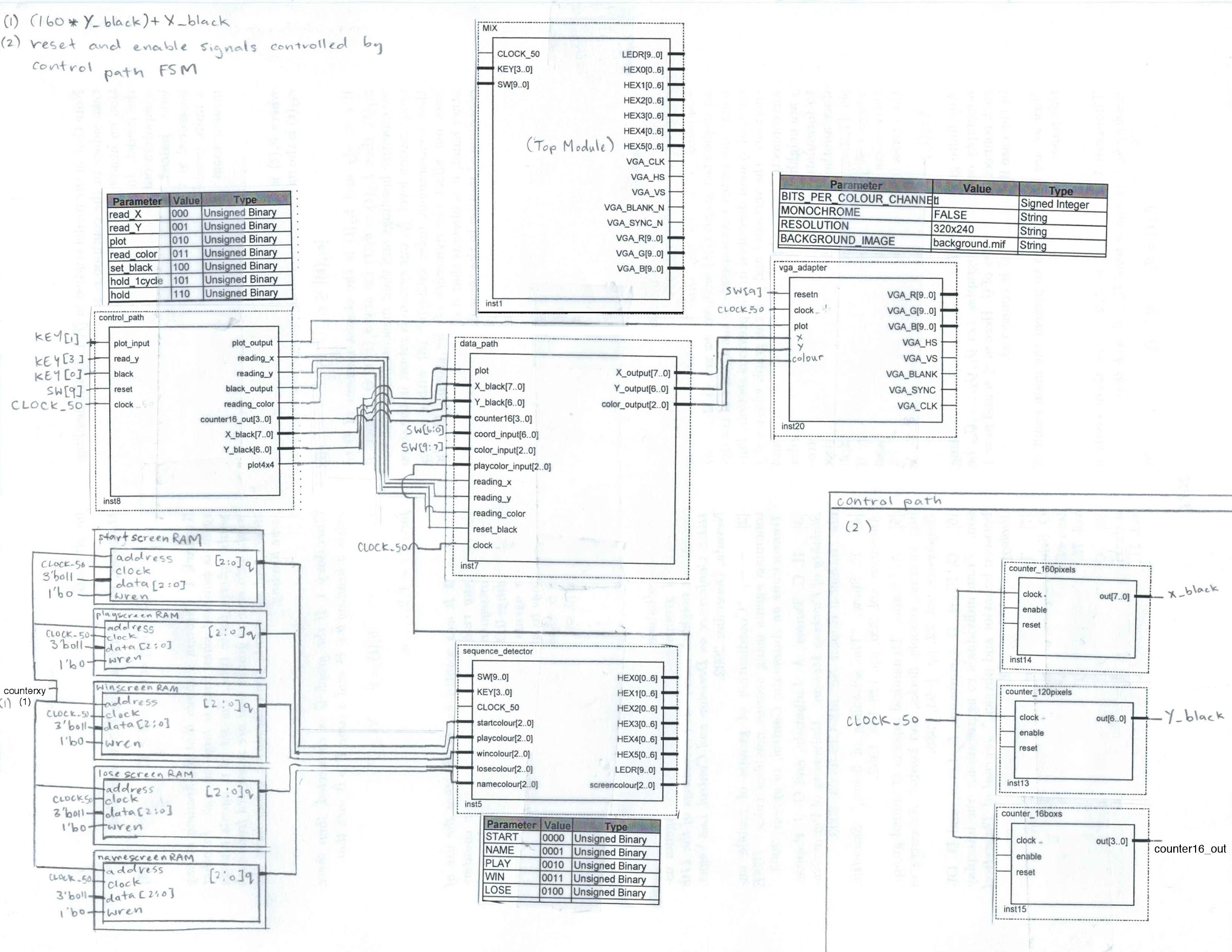
We were not able to reset the clock and the username without restarting the program. We had a major setback half way through the project which put us behind schedule. We had not completed the previous lab which illustrated how to use the VGA adapter module and figuring out how to do it took longer than expected. We had initially hoped to expand the project but were unable to realize these ideas due to time constraints caused by said setback.

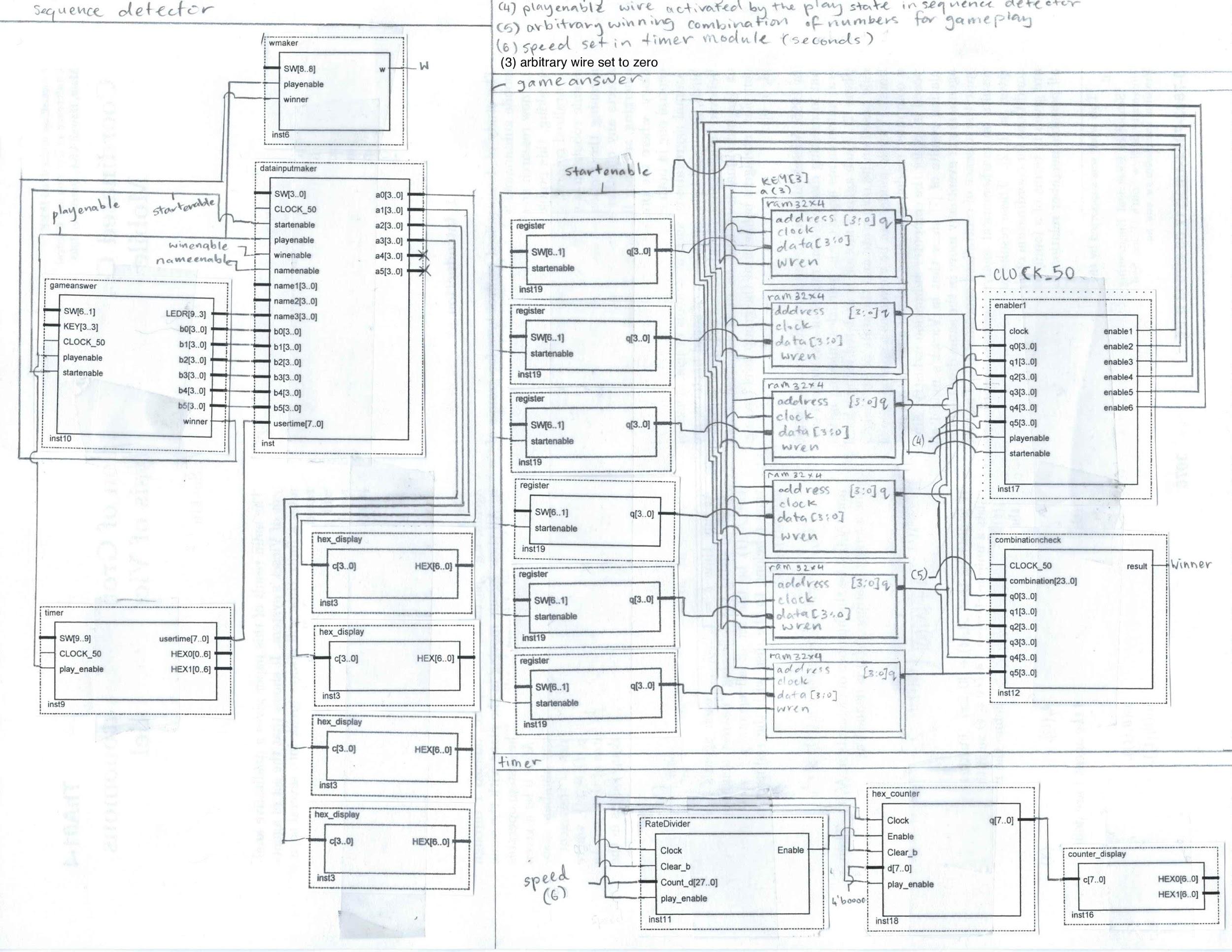
**What would you do differently – If you were going to start all over again**

Instead of writing the verilog for the non-essential modules of our game during the first week, we would have allotted more time to figuring out how the VGA adapter works as fast as possible. This way, we would have completed the essential components of the game on time and used any extra time to work on additional components and make overall improvements rather than leaving the main functionality of the design to the last minute. We would also try to expand upon the ideas we did not have time for and try to implement them in our design. These ideas included: exploring the use of the keyboard for inputs rather than the DE1SoC, integrating the use of the mouse to rearrange the animals with rather than comparing sequences of numbers, printing out a high score board with the fastest winning times, enabling a reset function so that the program would not need to be restarted after each play through, and possibly including moving sprites and animations.

**Appendix**

**Schematics**

****

****