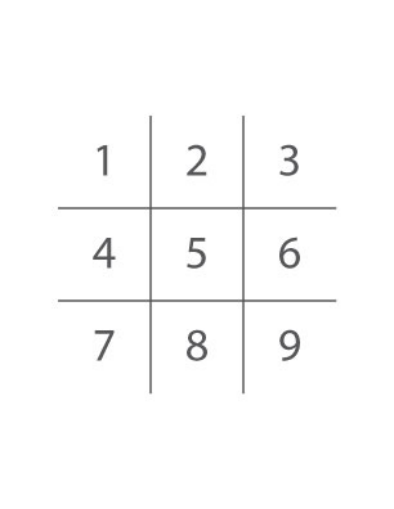
Tic Tac Toe with VGA Display

By: John Akroush, Nicholas Wheelis, Tao Wang



Introduction

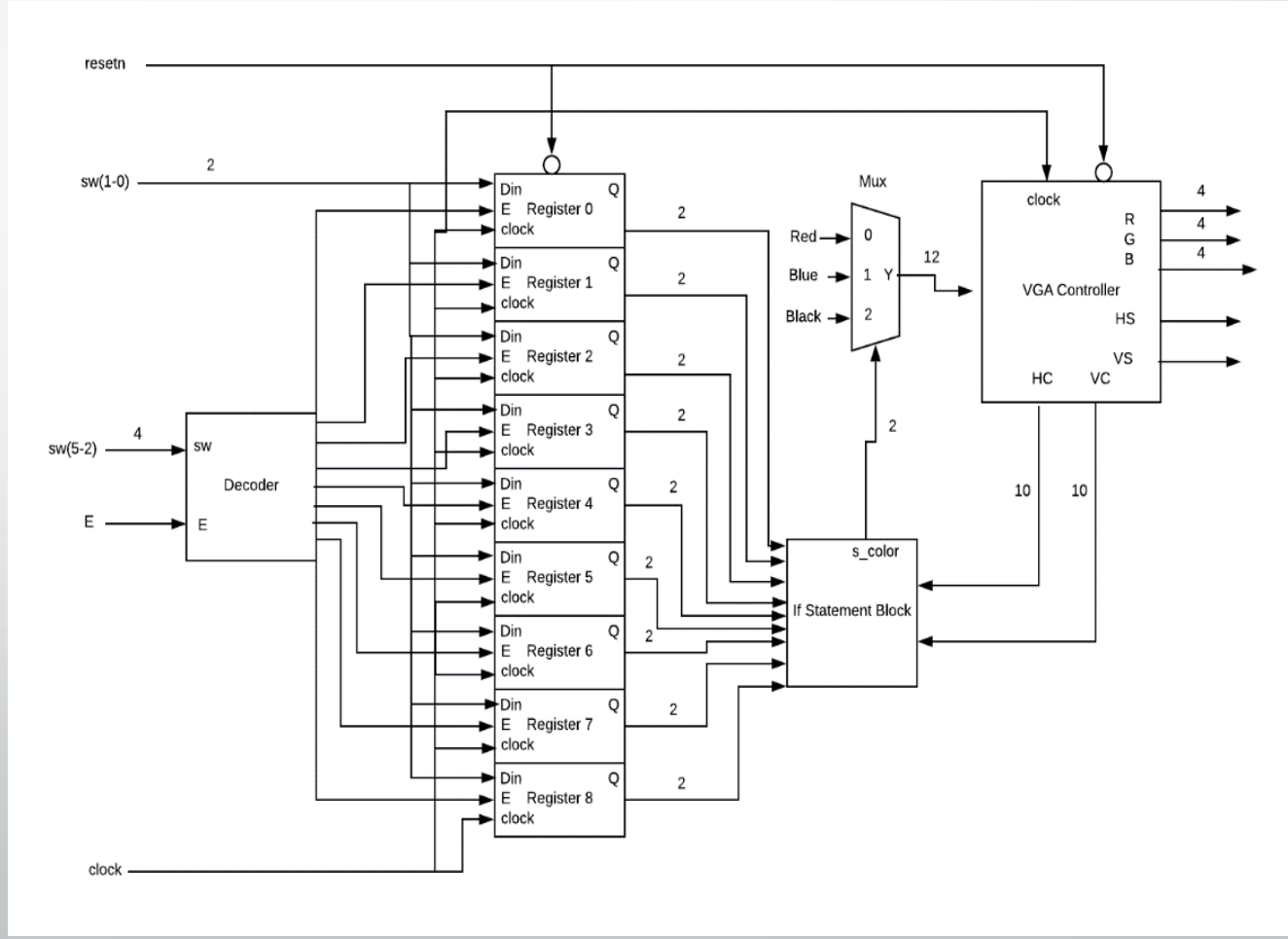
- Tic tac toe is a staple pastime for people to play with friends when they are bored. Instead of using X's and O's as the player marker, red and green rectangles are used. This project covers registers, decoders, multiplexers, addresses, and time with clocks. The project taught us how to display code to a VGA display.

How to Play

- To play Tic Tac Toe on this FPGA device, we start by using SW0 and SW1 to select the player. Next, the player will use SW2 through SW5 to select the rectangle the player wants to take. The player will then enable the system using SW6 to store their selection into the registers which will also display there color in the selected rectangle on the VGA display. After a player has won, they should press the resetn button to restart the game.

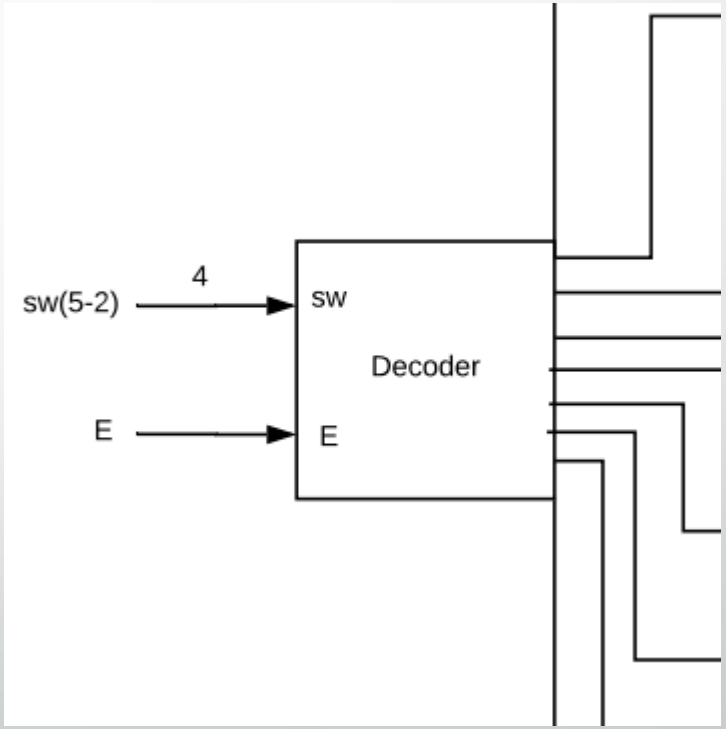
Main Design

- The main parts of this design are a decoder with enable, nine registers, a circuit with a list of if statements, a multiplexer, and a VGA display controller.



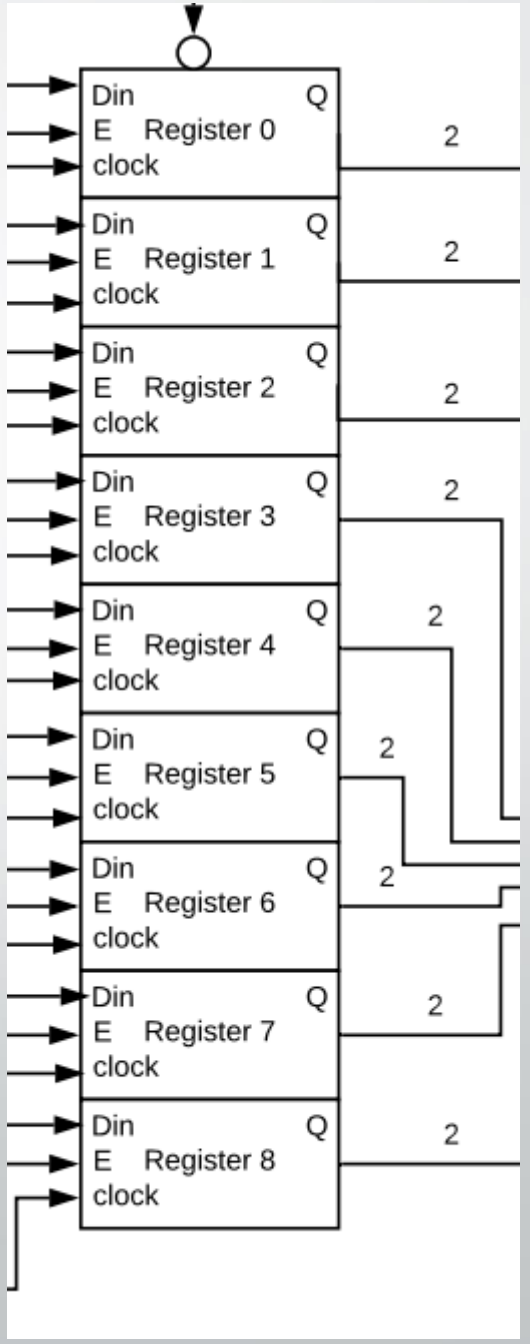
Decoder

- The decoder is used to enable a certain register that the player chooses which corresponds to one rectangle location on the tic tac toe board and is based on their address input. For example: an address of 0110 would enable register 6, which corresponds to the bottom-left rectangle on the board. SW2-SW5 are used for the address values, and SW6 is used for the enable.



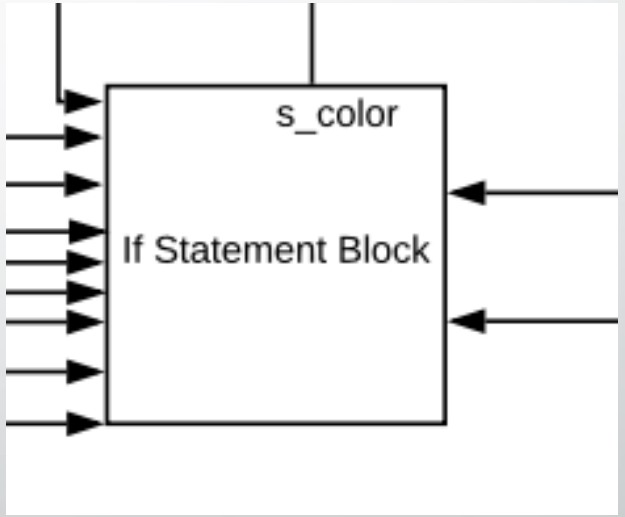
Registers

- The registers in this design act as the “memory” of the system so when the player ends their turn, their selection is saved. The input of the register is based on SW0 and SW1, which is the player number. The register that is used is based on the address put into the decoder.



If Statement Block

- This part of the design is a collection of If statements to decide, based on the H-count and V-count locations, which is from the VGA controller, and all the outputs of the registers to conclude what color goes to which Tic Tac Toe tile.

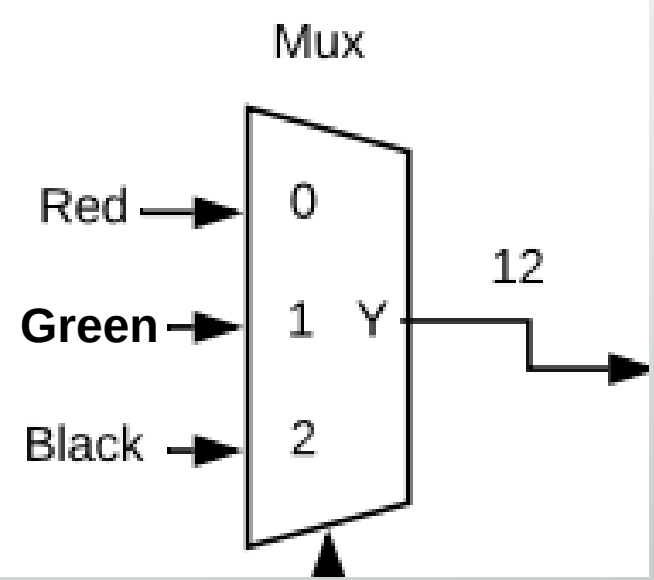


- If statements for first 5 rectangle box locations on display.

```
70  if (HC >= "0000000000") AND (HC <= "0011010100") AND (VC >= "0000000000") AND (VC <= "0010011111") then
71      if r0 = "01" then
72          s_color <= "01";
73      elsif r0 = "10" then
74          s_color <= "10";
75      else
76          s_color <= "11";
77      end if;
78  end if;
79
80  --if "0100000110" <= HC <= "0111011011" AND "0000100001" <= VC <= "0011000000" then
81  if (HC >= "0011010101") AND (HC <= "0110101001") AND (VC >= "0000000000") AND (VC <= "0010011111") then
82      if r1 = "01" then
83          s_color <= "01";
84      elsif r1 = "10" then
85          s_color <= "10";
86      else
87          s_color <= "11";
88      end if;
89  end if;
90
91  --if "0111011100" <= HC <= "1010101111" AND "0000100001" <= VC <= "0011000000" then
92  if (HC >= "0110101010") AND (HC <= "1001111111") AND (VC >= "0000000000") AND (VC <= "0010011111") then
93      if r2 = "01" then
94          s_color <= "01";
95      elsif r2 = "10" then
96          s_color <= "10";
97      else
98          s_color <= "11";
99      end if;
100 end if;
101
102 --if "0000110000" <= HC <= "0100000101" AND "0011000001" <= VC <= "0101100000" then
103 if (HC >= "0000000000") AND (HC <= "0011010100") AND (VC >= "0010100000") AND (VC <= "0100111111") then
104     if r3 = "01" then
105         s_color <= "01";
106     elsif r3 = "10" then
107         s_color <= "10";
108     else
109         s_color <= "11";
110     end if;
111 end if;
112
113 --if "0100000110" <= HC <= "0111011011" AND "0011000001" <= VC <= "0101100000" then
114 if (HC >= "0011010101") AND (HC <= "0110101001") AND (VC >= "0010100000") AND (VC <= "0100111111") then
115     if r4 = "01" then
116         s_color <= "01";
117     elsif r4 = "10" then
118         s_color <= "10";
119     else
120         s_color <= "11";
121     end if;
122 end if;
```

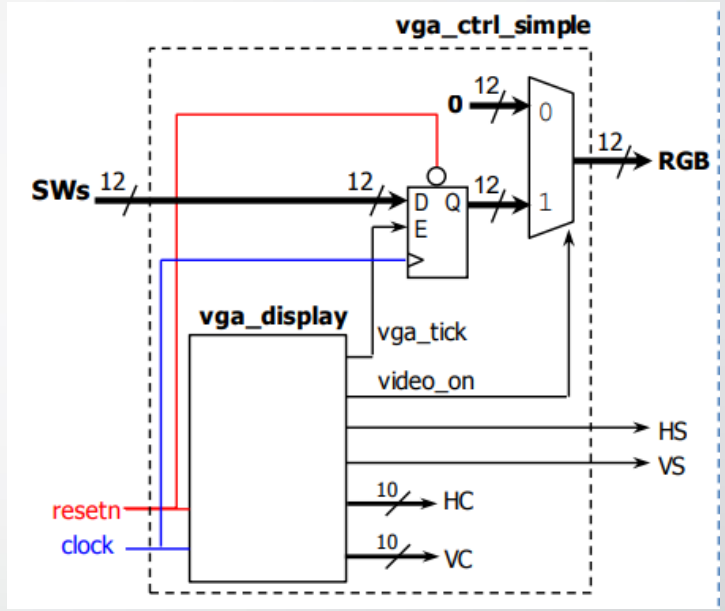
Multiplexer

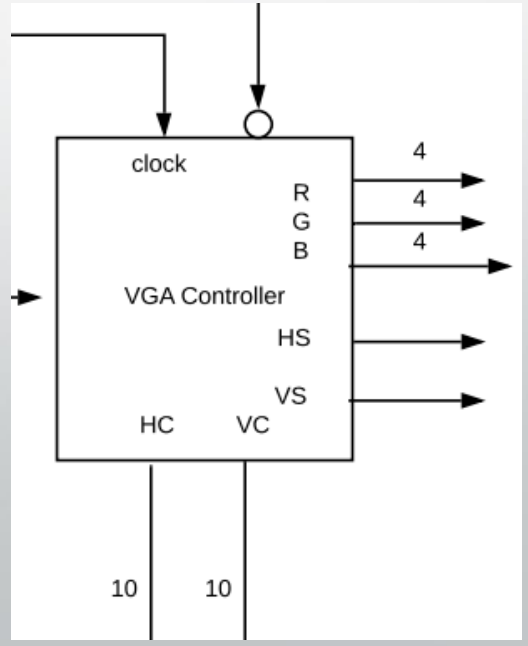
- The mux in this design is simple. Its input is the s_color that the If Statement Block outputs. This input selects one of the 3 color inputs of the mux and outputs the 12-bit value for that color.



VGA Controller

- The vga_ctrl_simple component is used to control the VGA display. It uses a 12-bit input to control the color of the pixels.





Demonstration