PONG

MATTHEW MICHALEWICZ

STEVEN STEWART

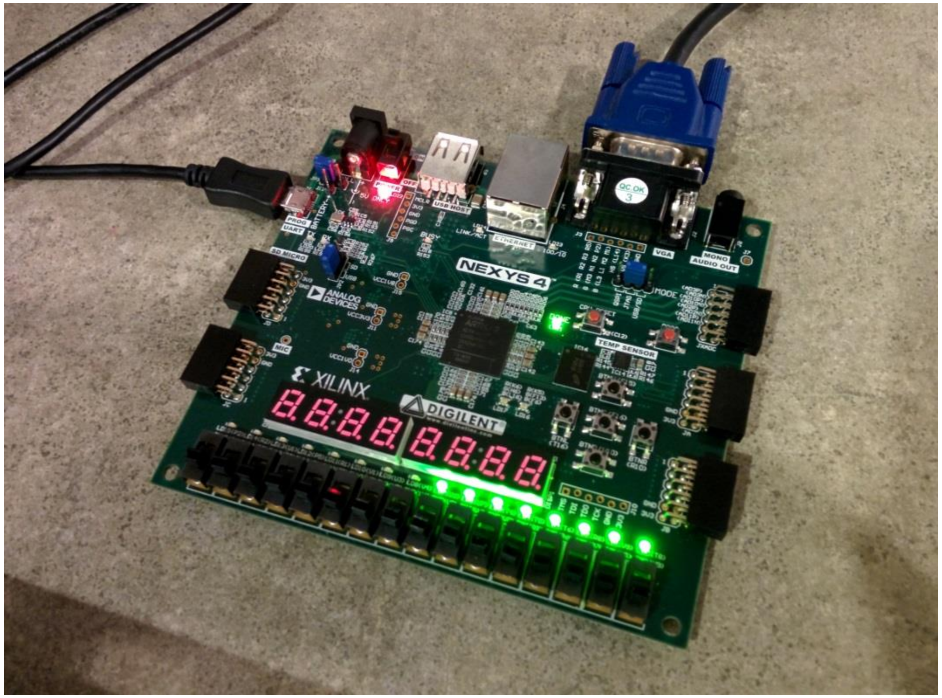
RATIONALE

- A game is the best way to develop and exhibit mastery of machinery since it is the highest form of connection between man and machine

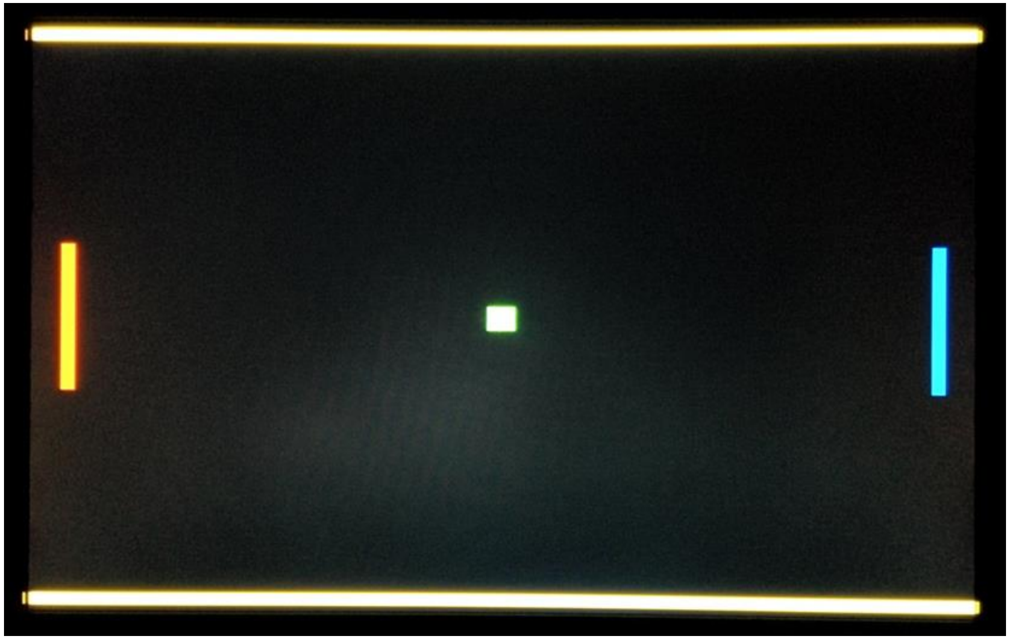
- Challenging but fun application of VHDL

- *For your happy entertainment*

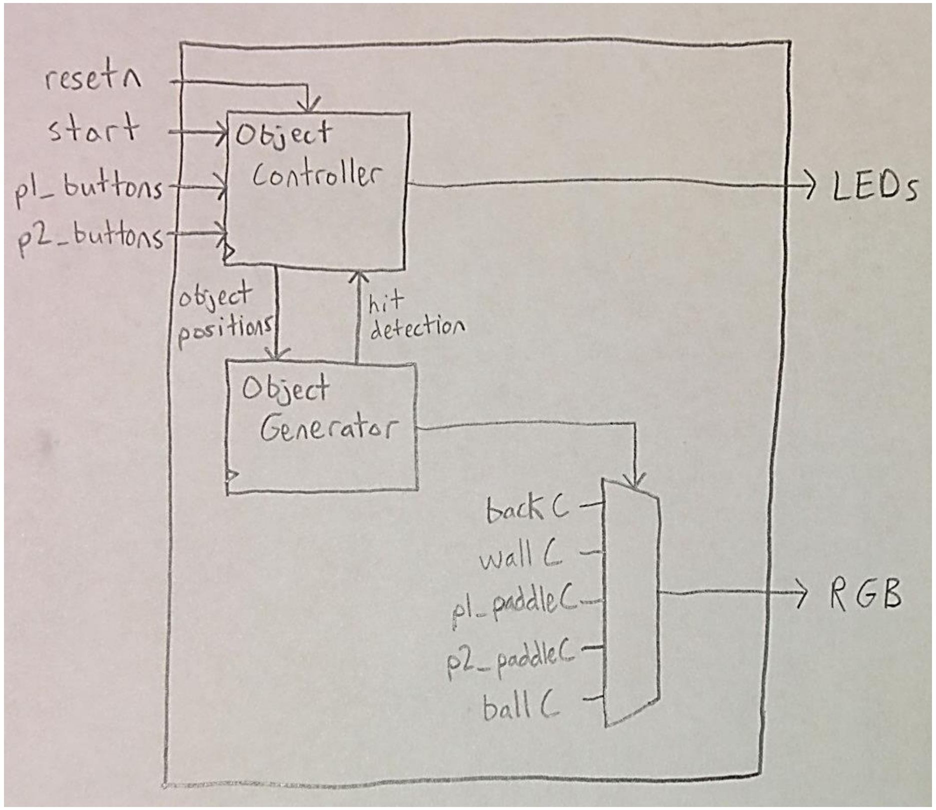
EXTERNAL INTERFACES



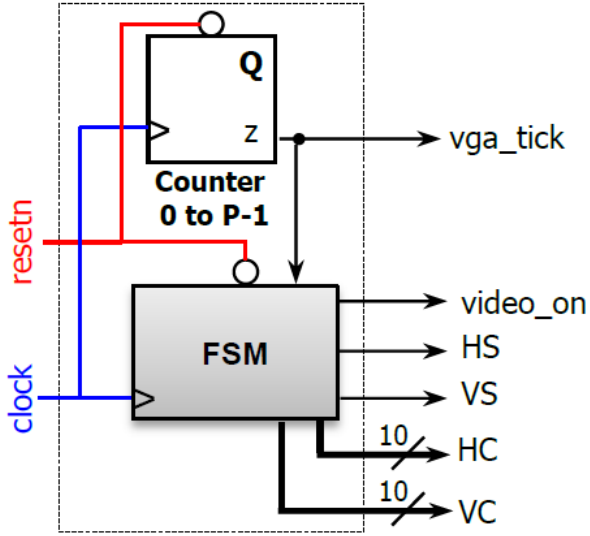
- VGA Display
- Push buttons
- LEDs



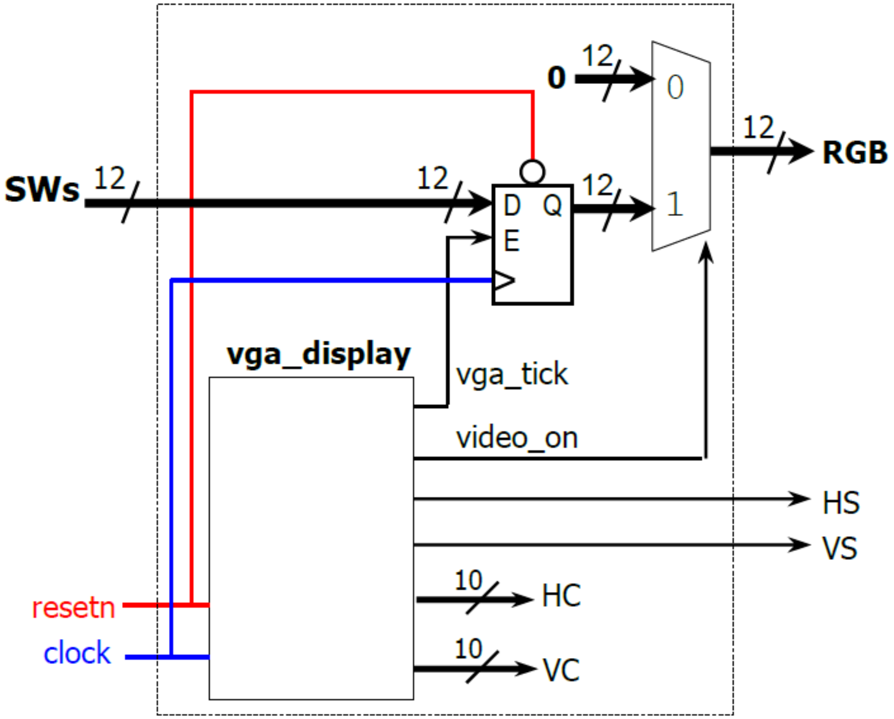
DATAPATH



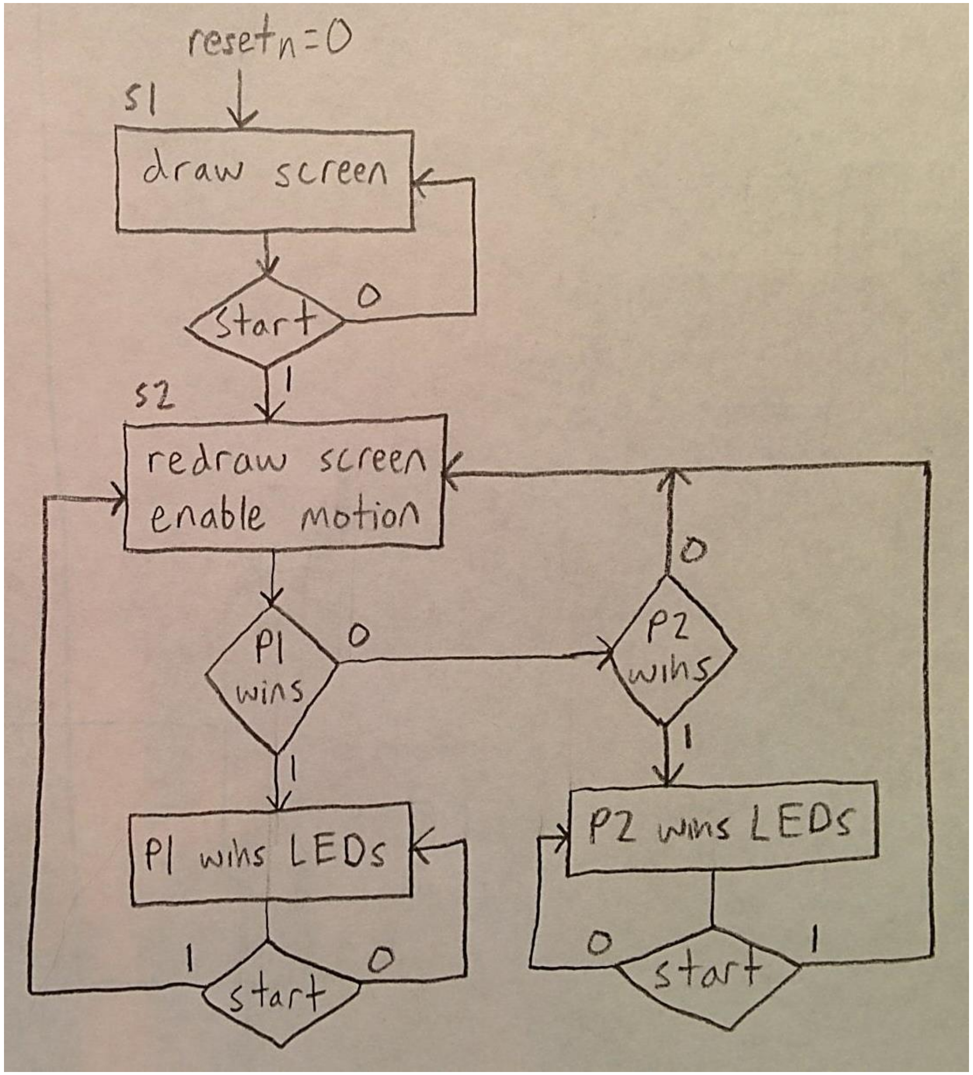
vga_display

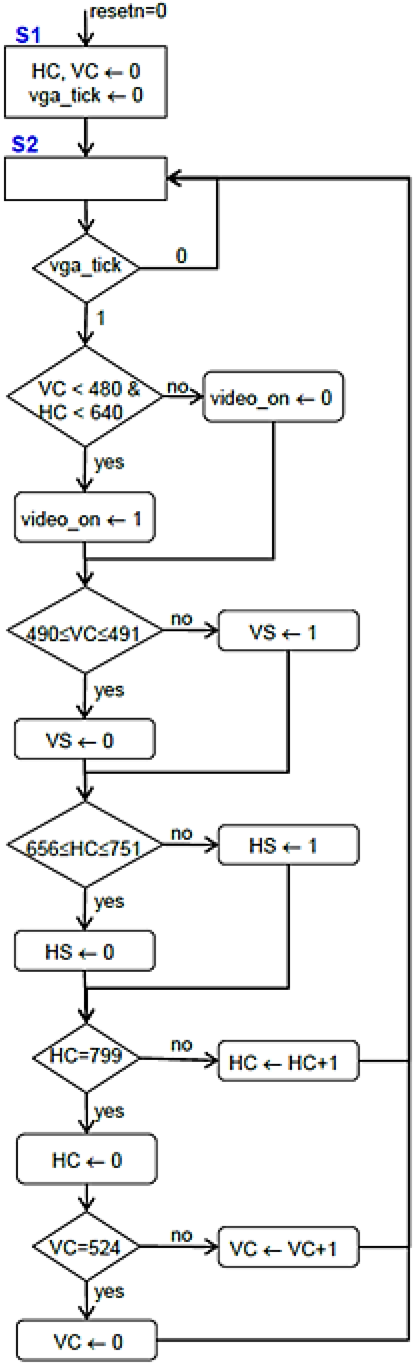


vga_ctrl_simple

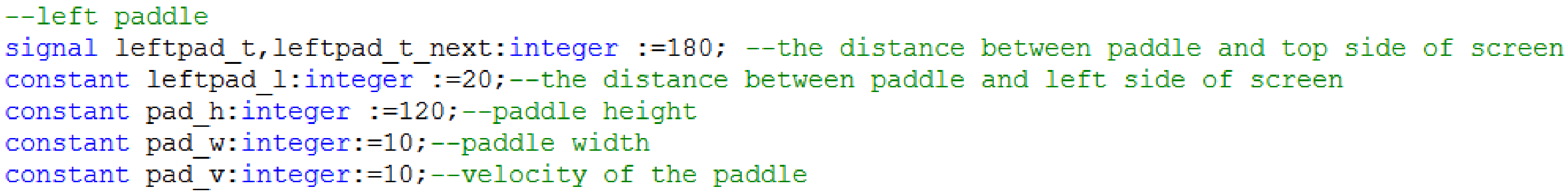


CONTROL CIRCUIT OBJECT CONTROLLER



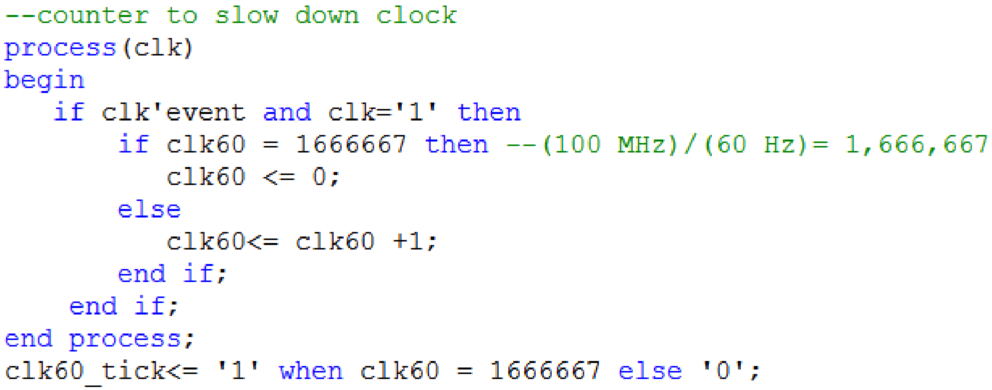


OBJECT GENERATOR



```
--left paddle
signal leftpad_t,leftpad_t_next:integer :=180; --the distance between paddle and top side of screen
constant leftpad_l:integer :=20;--the distance between paddle and left side of screen
constant pad_h:integer :=120;--paddle height
constant pad_w:integer:=10;--paddle width
constant pad_v:integer:=10;--velocity of the paddle
```

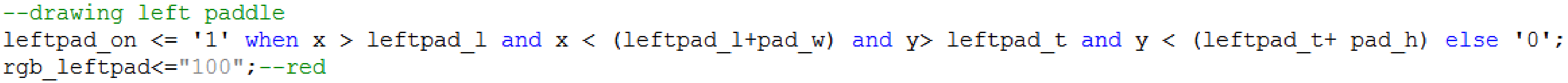
- **Size initialization**



```
--counter to slow down clock
process(clk)
begin
   if clk'event and clk='1' then
       if clk60 = 1666667 then --(100 MHz)/(60 Hz)= 1,666,667
          clk60 <= 0;
       else
          clk60<= clk60 +1;
       end if;
    end if;
end process;
clk60_tick<= '1' when clk60 = 1666667 else '0';
```

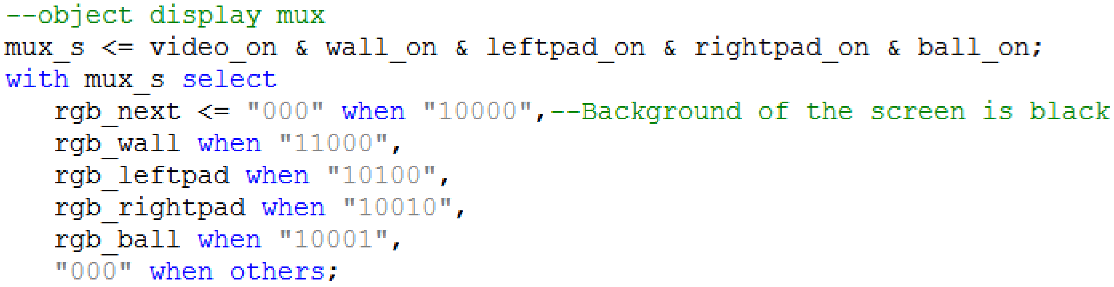
- **Clock divider**

OBJECT GENERATOR



```
--drawing left paddle
leftpad_on <= '1' when x > leftpad_l and x < (leftpad_l+pad_w) and y> leftpad_t and y < (leftpad_t+ pad_h) else '0';
rgb_leftpad<="100";--red
```

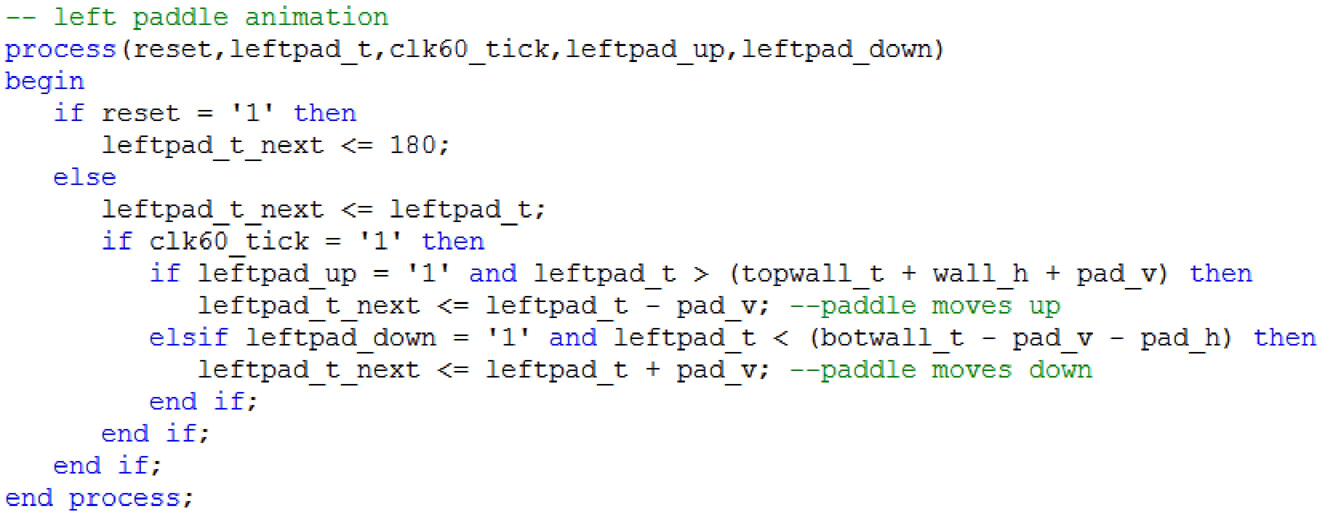
- **Drawing objects**



```
--object display mux
mux_s <= video_on & wall_on & leftpad_on & rightpad_on & ball_on;
with mux_s select
   rgb_next <= "000" when "10000",--Background of the screen is black
   rgb_wall when "11000",
   rgb_leftpad when "10100",
   rgb_rightpad when "10010",
   rgb_ball when "10001",
   "000" when others;
```

- **Object multiplexer**

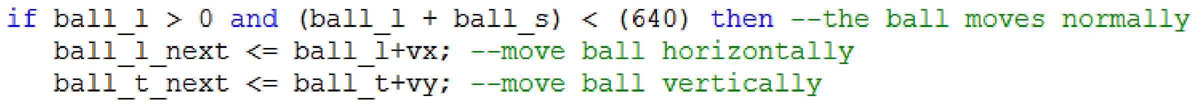
OBJECT GENERATOR



```
-- left paddle animation
process(reset,leftpad_t,clk60_tick,leftpad_up,leftpad_down)
begin
   if reset = '1' then
      leftpad_t_next <= 180;
   else
      leftpad_t_next <= leftpad_t;
      if clk60_tick = '1' then
         if leftpad_up = '1' and leftpad_t > (topwall_t + wall_h + pad_v) then
            leftpad_t_next <= leftpad_t - pad_v; --paddle moves up
         elsif leftpad_down = '1' and leftpad_t < (botwall_t - pad_v - pad_h) then
            leftpad_t_next <= leftpad_t + pad_v; --paddle moves down
         end if;
      end if;
   end if;
end process;
```

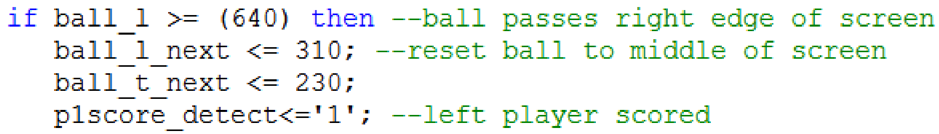
- **Paddle movement**

OBJECT GENERATOR



```
if ball_l > 0 and (ball_l + ball_s) < (640) then --the ball moves normally
   ball_l_next <= ball_l+vx; --move ball horizontally
   ball_t_next <= ball_t+vy; --move ball vertically
```

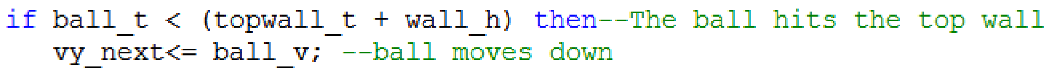
- **Ball movement**



```
if ball_l >= (640) then --ball passes right edge of screen
   ball_l_next <= 310; --reset ball to middle of screen
   ball_t_next <= 230;
   p1score_detect<='1'; --left player scored
```

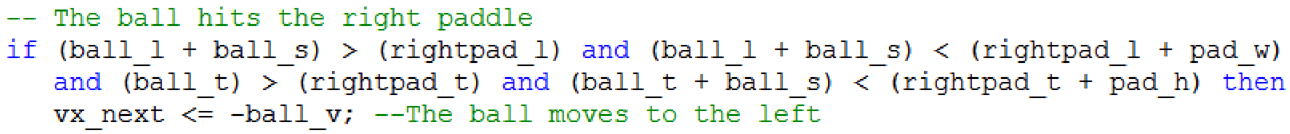
- **Score detection and ball reset**

OBJECT GENERATOR



```
if ball_t < (topwall_t + wall_h) then--The ball hits the top wall
   vy_next<= ball_v; --ball moves down
```

- **Wall hit detection**



```
-- The ball hits the right paddle
if (ball_l + ball_s) > (rightpad_l) and (ball_l + ball_s) < (rightpad_l + pad_w)
   and (ball_t) > (rightpad_t) and (ball_t + ball_s) < (rightpad_t + pad_h) then
   vx_next <= -ball_v; --The ball moves to the left
```

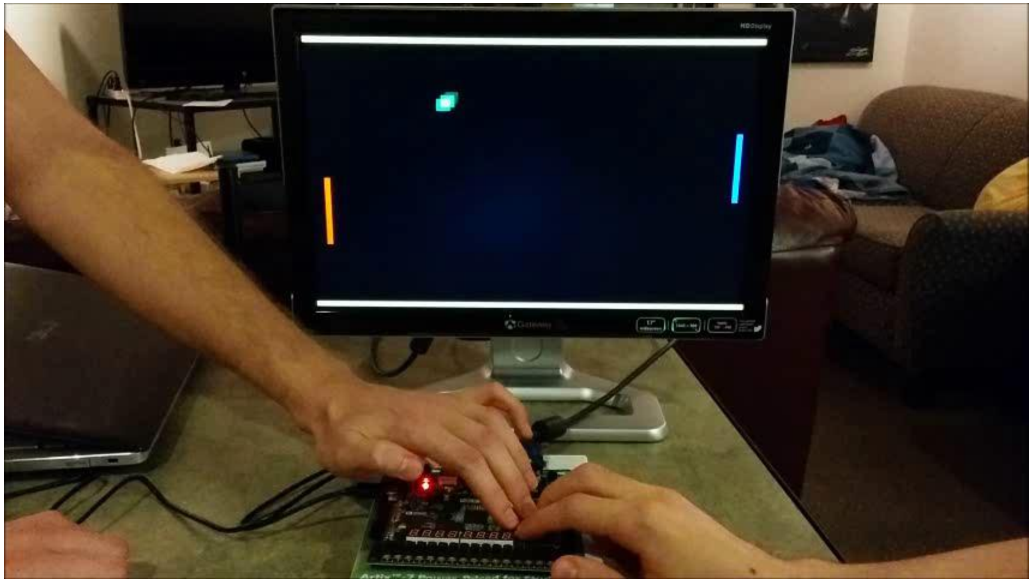
- **Paddle hit detection**

SIGNIFICANT DIFFICULTIES

- Accurate hit detection

- Score keeping

- Match restart and ball reset



POSSIBILITIES FOR ENHANCED FUNCTIONALITY

- **Add score tracking on seven segment displays**
- **Control color scheme via switches**
- **Display splash screen when a player wins**
- **Background picture**
- **External gamepad**
- **Sounds**