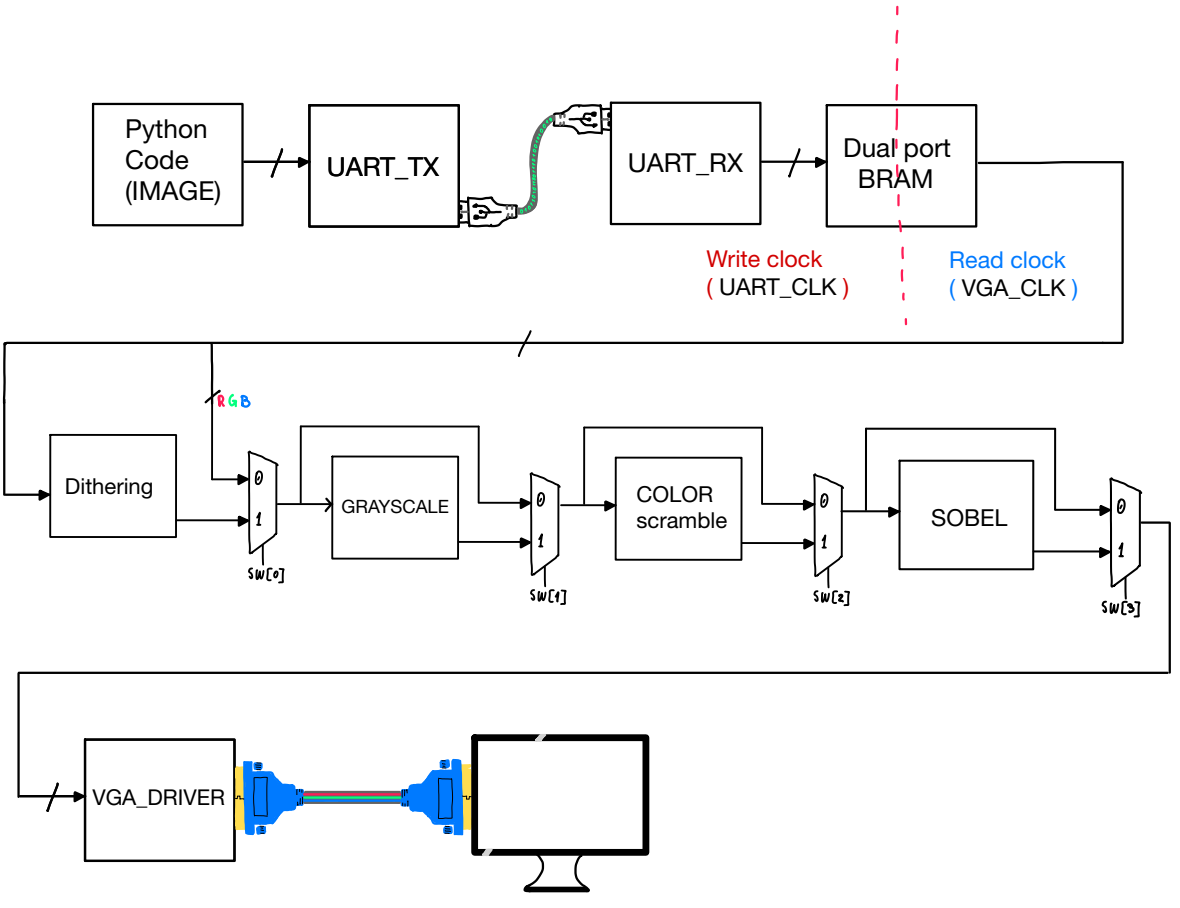
NOTES


Python Code (IMAGE)
UART_TX
UART_RX
Dual port BRAM
Write clock (UART_CLK)
Read clock (VGA_CLK)
RGB
Dithering
0
1
SW[0]
GRAYSCALE
SW[1]
COLOR scramble
SW[2]
SOBEL
SW[3]
VGA_DRIVER

GRAYSCALE

El Algoritmo 'Average' es el MAS Simple
→ Promediar R,G,B y Asignar el Promedio a R,G,B

Ejemplo:

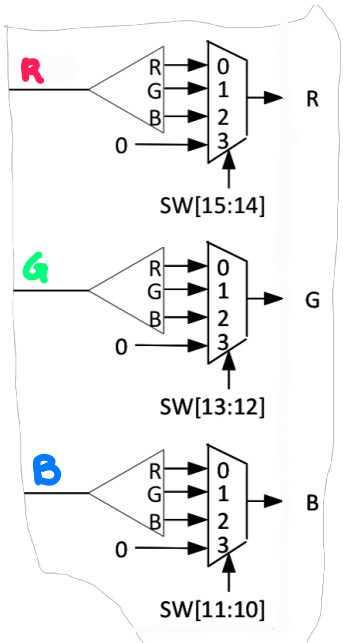
1 Pixel: $(R,G,B) = (25, 32, 125) \xrightarrow{\text{Promedio}} \frac{25+32+125}{3} = 57.6$

1 Pixel Grayscale: $(R,G,B) = (57.6, 57.6, 57.6)$

```
Module Grayscale( Input logic R,G,B
                  Output logic GR,GG,GB);
    Logic GRAY
    Always_comb Begin
        GRAY = (R+G+B)· 0,333
        GR = Gray
        GG = Gray
        GB = GRAY
    END
EndModule
```

Color Scramble

Lo controla El Usuario




```
Module Scrambler (input Logic R,G,B,
                  input logic [3:0] sw
                  output logic sout );
  Always_comb begin
    case (sw)
      0: sout = R;
      1: sout = G;
      2: sout = B;
      3: sout = 0;
  Endcase
EndModule

Module Color_Scramble (
                  input logic R,G,B,
                  input logic [3:0] swR,swG,swB
                  output logic sR,sG,sB);
  Scrambler R(R,G,B,swR,sR)
  Scrambler G(R,G,B,swG,sG)
  Scrambler B(R,G,B,swB,sB)
EndModule
```

Dithering

B & W →

Algoritmo resolución 4x4 →

Se sobre pone esta matriz sobre la matriz de la imagen y se ve si el valor del pixel de la imagen es mayor que el valor del cuadro en la matriz y si es mayor, el pixel se pinta

64	128
192	0

```
MODULE Dithering ( INPUT logic [10:0] nc_visible , vc_visible
                   INPUT logic Pixel_Value
                   OUTPUT logic D_PixVal );

  LOGIC [10:0] nc, vc;


  ALWAYS_comb begin
    nc = vc
```

Sobel (Detección de bordes)

$g_x =$

-1	0	1
-2	0	2
-1	0	1

gradiente Sobel X

$g_y =$

-1	-2	-1
0	0	0
1	2	1

gradiente Sobel Y

Magnitud del borde = $\sqrt{(g_x)^2 + (g_y)^2}$

(Imagen final)