

# PRACTICA CAPGEMINI

Prof. Coord.: Feldioreanu George

Student: SIMION BIANCA-GEORGIANA

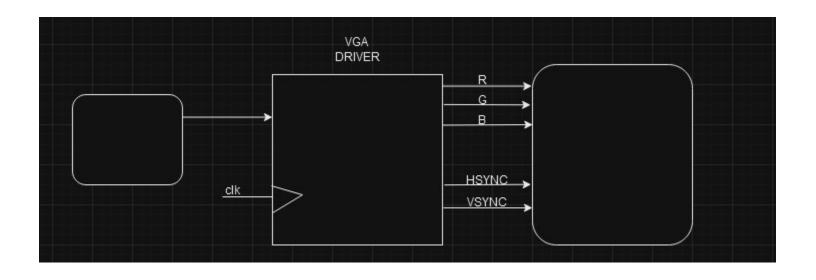
Anul de studii: II

Grupa:4LF621

# **CUPRINS**

•	Schema Bloc	pag 3
•	Tabel cu porturi	pag 3
•	Arhitectura internă	pag 4
•	Descrierea functionalitatii	pag 5
•	Aplicatii	pag 6

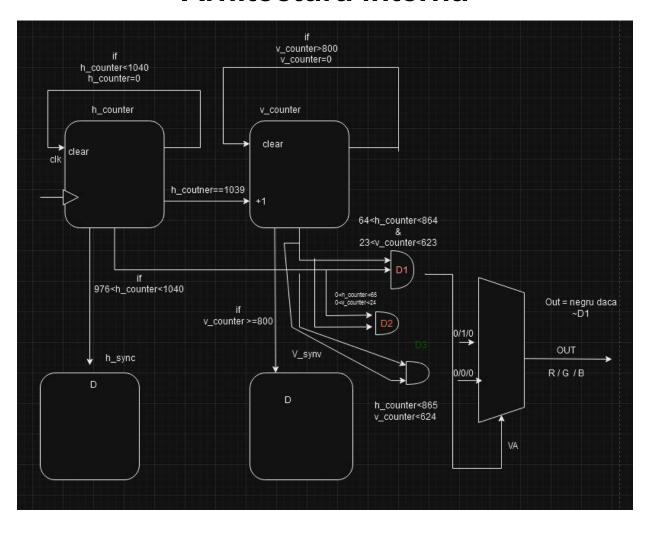
# **Schema Bloc**



# Tabel cu porturi

Signal name	Functionality	I/O	Width[bits]
clk	Clock 50 MHz	I	1
rst	Asynchronous Reset active high	I	1
en	Enable	I	1
h_sync	Horizontal Synchronization	0	1
v_sync	Vertical Synchronization	0	1
red	Red color	0	1
green	Green color	0	1
blue	Blue color	0	1

# Arhitectura internă



## Descrierea functionalitatii

Acest modul Verilog descrie un controler VGA simplu care generează semnalele necesare pentru a afișa imagini pe un monitor VGA. Haideţi să detaliem funcţionalitatea fiecărei secţiuni din acest cod:

### Intrări:

clk: semnalul de ceas de 50 MHz.

rst: semnalul de resetare asincronă activă pe nivel înalt.

**en**: semnalul de activare pentru a controla dacă ieșirile RGB sunt active sau nu.

### leşiri:

**h\_sync**: semnalul de sincronizare orizontală.

**v\_sync**: semnalul de sincronizare verticală.

red, green, blue: semnalele de culoare RGB.

### Contoare pentru sincronizarea orizontală și verticală

**h\_count**: contorul orizontal care rulează de la 0 la 1039 (1040 total de cicluri).

v\_count: contorul vertical care rulează de la 0 la 665 (666 total de cicluri).

### Proces de actualizare a contoarelor

Acest proces este declanșat la fiecare semnal de ceas (clk) sau de resetare (rst). Dacă semnalul de resetare este activ, contoarele orizontal și vertical sunt resetate la 0. În caz contrar:

Contorul orizontal (h\_count) este incrementat până ajunge la 1039.

La atingerea valorii 1039, contorul orizontal este resetat la 0 și contorul vertical (v\_count) este incrementat până ajunge la 665.

La atingerea valorii 665, contorul vertical este resetat la 0.

Generarea semnalelor de sincronizare orizontală și verticală

Acest proces este declanșat la fiecare semnal de ceas (clk) sau de resetare (rst). Dacă semnalul de resetare este activ, semnalele de sincronizare (h\_sync și v\_sync) sunt setate la 1 (nivel inactiv). În caz contrar:

Semnalul h\_sync este activ (0) între valorile h\_count 856 și 975.

Semnalul v\_sync este activ (0) între valorile v\_count 637 și 642.

Generarea semnalelor RGB pentru imagine depinde in functie de imaginea pe care dorim sa o afisam.

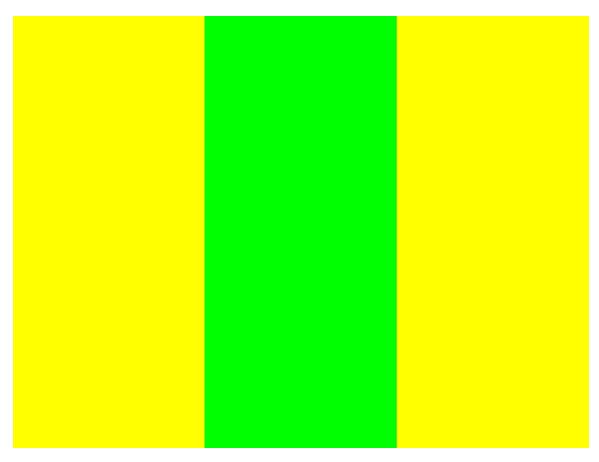
În concluzie, acest modul Verilog definește un controler VGA care generează semnale de sincronizare orizontală și verticală pentru un ecran VGA cu o rezoluție de 800 x 600 pixeli și culori RGB.

# **Aplicatii**

# Display Verde

```
always @(posedge clk or posedge rst) begin
  if (rst) begin
     // Resetare ieşiri RGB la 0
     red \le 0;
     green \leq 0;
     blue <= 0;
  end else if (en) begin
     // Generarea culorii verde dacă semnalul de activare este 1
     if (h_count < 800 && v_count < 600) begin
       red <= 0; // roșu inactiv
       green <= 1; // verde activ
       blue <= 0; // albastru inactiv
     end else begin
       // leşirile RGB sunt 0 în afara zonei vizibile
       red \le 0;
       green \leq 0;
       blue <= 0;
     end
  end else begin
     // Dacă semnalul de activare este 0, ieșirile RGB sunt 0
     red \le 0;
     green <= 0;
     blue <= 0;
  end
end
```

Display Galben Verde Galben



```
always @(posedge clk or posedge rst) begin

if (rst) begin

// Resetare ieşiri RGB la 0

red <= 0;

green <= 0;

blue <= 0;

end else if (en) begin

// Generarea culorii verde dacă semnalul de activare este 1

if (h_count < 800 && v_count < 600) begin

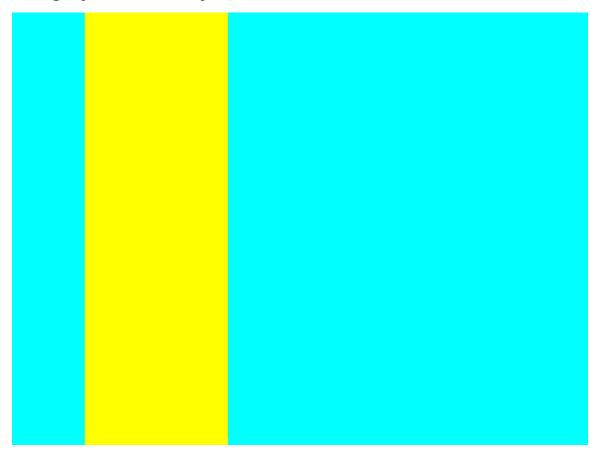
if (h_count < 266) begin

// Banda stângă - Galben

red <= 1;
```

```
green <= 1;
          blue <= 0;
       end else if (h_count < 533) begin
          // Banda mijloc - Verde
          red <= 0;
          green <= 1;
          blue <= 0;
       end else begin
          // Banda dreaptă - Galben
          red <= 1;
          green <= 1;
          blue <= 0;
       end
     end else begin
       // leşirile RGB sunt 0 în afara zonei vizibile
       red \le 0;
       green <= 0;
       blue <= 0;
     end
  end else begin
    // Dacă semnalul de activare este 0, ieșirile RGB sunt 0
     red \le 0;
     green \leq 0;
    blue <= 0;
  end
end
```

### Steag Cyan Galben Cyan



```
always @(posedge clk or posedge rst) begin

if (rst) begin

// Resetare ieşiri RGB la 0

red <= 0;

green <= 0;

blue <= 0;

end else begin

if (h_count < 800 && v_count < 600 && en == 1) begin

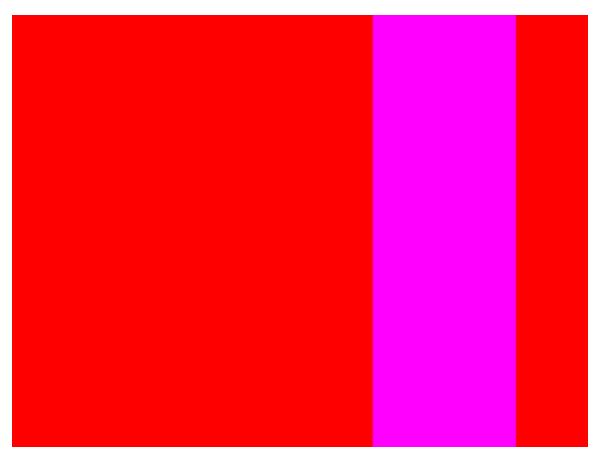
if (h_count <= (800 - 700)) begin

red <= 0; // turcoaz

green <= 1;
```

```
blue <= 1;
       end else if (h_count >= (800 - 500 )) begin
         red <= 0; // turcoaz
         green <= 1;
         blue <= 1;
       end else begin
         red <= 1; // galben
         green <= 1;
         blue <= 0;
       end
    end else begin
       red \leq 0;
       green <= 0;
       blue <= 0;
    end
  end
end
```

### Steag Rosu Roz Rosu



```
always @(posedge clk or posedge rst) begin

if (rst) begin

// Resetare ieşiri RGB la 0

red <= 0;

green <= 0;

blue <= 0;

end else begin

if (h_count < 800 && v_count < 600 && en == 1) begin

if (h_count <= (800 - 300)) begin

red <= 1; // rosu

green <= 0;

blue <= 0;
```

```
end else if (h_count >= 700 ) begin
         red <= 1; // rosu
         green <= 0;
         blue <= 0;
       end else begin
         red <= 1; // magenta
         green <= 0;
         blue <= 1;
       end
    end else begin
       red \leq 0;
       green <= 0;
       blue <= 0;
    end
  end
end
```

### Steag Rosu Cyan Rosu



### Steag Rosu Galben Rosu

```
always @(posedge clk or posedge rst) begin

if (rst) begin

// Resetare ieşiri RGB la 0

red <= 0;

green <= 0;

blue <= 0;

end else begin

if (h_count < 800 && v_count < 600 && en == 1) begin

if (h_count <= (800 - 750)) begin
```

```
red <= 1; // rosu
         green <= 0;
         blue <= 0;
       end else if (h_count >= 750 ) begin
         red <= 1; // rosu
         green <= 0;
         blue <= 0;
       end else begin
         red <= 0; // magenta
         green <= 1;
         blue <= 1;
       end
    end else begin
       red <= 0;
       green <= 0;
       blue <= 0;
    end
  end
end
```

```
always @(posedge clk or posedge rst) begin

if (rst) begin

// Resetare ieşiri RGB la 0

red <= 0;

green <= 0;

blue <= 0;

end else begin

if (h_count < 800 && v_count < 600 && en == 1) begin

if (v_count <= (800 - 750)) begin

red <= 1; // rosu

green <= 0;

blue <= 0;
```

```
end else if (v_count >= 550 ) begin
         red <= 1; // rosu
         green <= 0;
         blue <= 0;
       end else begin
         red <= 1; // magenta
         green <= 1;
         blue <= 0;
       end
    end else begin
       red \leq 0;
       green <= 0;
       blue <= 0;
    end
  end
end
```

### Steag Cyan Verde Cyan

```
always @(posedge clk or posedge rst) begin

if (rst) begin

// Resetare ieşiri RGB la 0

red <= 0;

green <= 0;

blue <= 0;

end else begin

if (h_count < 800 && v_count < 600 && en == 1) begin

if (v_count <= (800 - 300)) begin

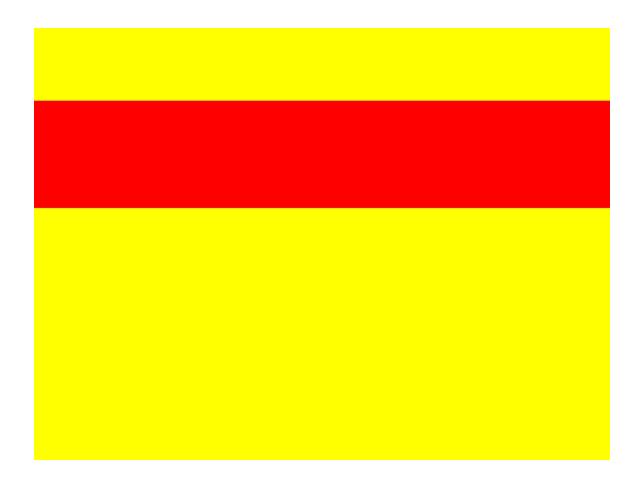
red <= 0; // cyan

green <= 1;

blue <= 1;
```

```
end else if (v_count >= 550 ) begin
         red <= 0; // cyan
         green <= 1;
         blue <= 1;
       end else begin
         red <= 0; // verde
         green <= 1;
         blue <= 0;
       end
    end else begin
       red \leq 0;
       green <= 0;
       blue <= 0;
    end
  end
end
```

### Steag Galben Rosu Galben



```
always @(posedge clk or posedge rst) begin

if (rst) begin

// Resetare ieşiri RGB la 0

red <= 0;

green <= 0;

blue <= 0;

end else begin

if (h_count < 800 && v_count < 600 && en == 1) begin

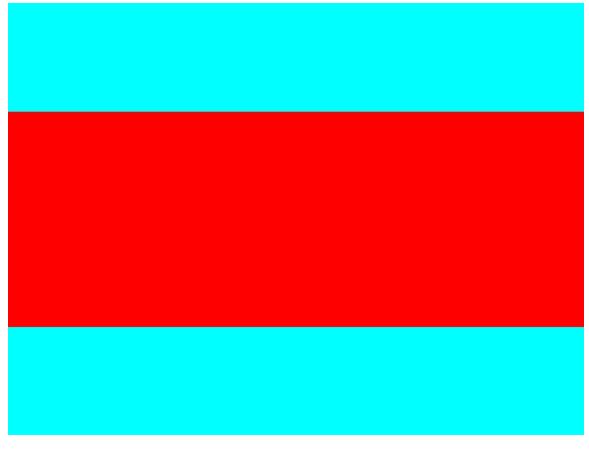
if (v_count <= (800 - 700)) begin

red <= 1; // galben

green <= 1;
```

```
blue <= 0;
       end else if (v_count >= 250 ) begin
         red <= 1; // galben
         green <= 1;
         blue <= 0;
       end else begin
         red <= 1; // rosu
         green <= 0;
         blue <= 0;
       end
    end else begin
       red <= 0;
       green <= 0;
       blue <= 0;
    end
  end
end
```

### Steag Cyan Rosu Cyan



```
always @(posedge clk or posedge rst) begin

if (rst) begin

// Resetare ieşiri RGB la 0

red <= 0;

green <= 0;

blue <= 0;

end else begin

if (h_count < 800 && v_count < 600 && en == 1) begin

if (v_count <= (600 - 450)) begin

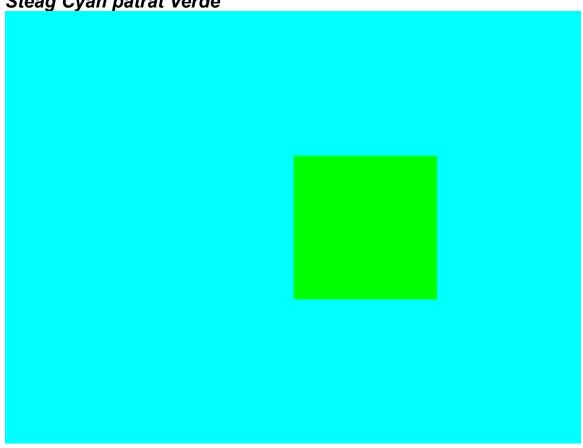
red <= 0; // cyan

green <= 1;

blue <= 1;
```

```
end else if (v_count >= 450 ) begin
         red <= 0; // cyan
         green <= 1;
         blue <= 1;
       end else begin
         red <= 1; // rosu
         green <= 0;
         blue <= 0;
       end
    end else begin
       red <= 0;
      green <= 0;
       blue <= 0;
    end
  end
end
```

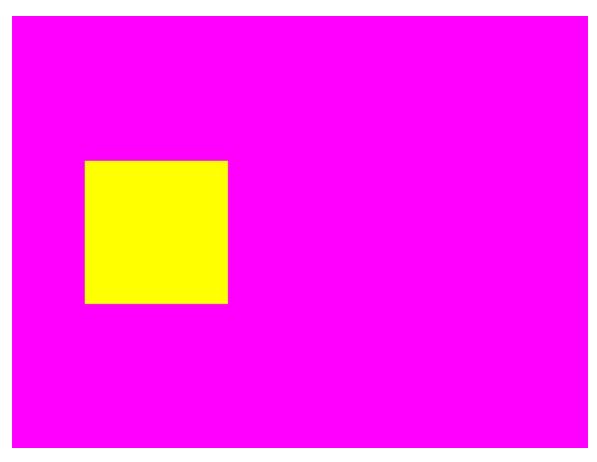
Steag Cyan patrat Verde



```
always @(posedge clk or posedge rst) begin
  if (rst) begin
    // Resetare ieșiri RGB la 0
    red <= 0;
     green \leq 0;
     blue <= 0;
  end else if (en) begin
    // Generarea culorii pe baza poziției curente
     if (h_count < 800 && v_count < 600) begin
       if (h_count > 400 && h_count < 600 && v_count > 200 && v_count
< 400) begin
          // Dreptunghi mijloc - verde
```

```
red \le 0;
          green <= 1;
          blue <= 0;
       end else begin
          // Fundal - cyan
          red \le 0;
          green <= 1;
          blue <= 1;
       end
     end else begin
       // leşirile RGB sunt 0 în afara zonei vizibile
       red \le 0;
       green <= 0;
       blue <= 0;
     end
  end else begin
    // Dacă semnalul de activare este 0, ieșirile RGB sunt 0
    red <= 0;
     green <= 0;
     blue <= 0;
  end
end
```

### Steag Magenta patrat Galben



```
always @(posedge clk or posedge rst) begin

if (rst) begin

// Resetare ieşiri RGB la 0

red <= 0;

green <= 0;

blue <= 0;

end else if (en) begin

// Generarea culorii pe baza poziției curente

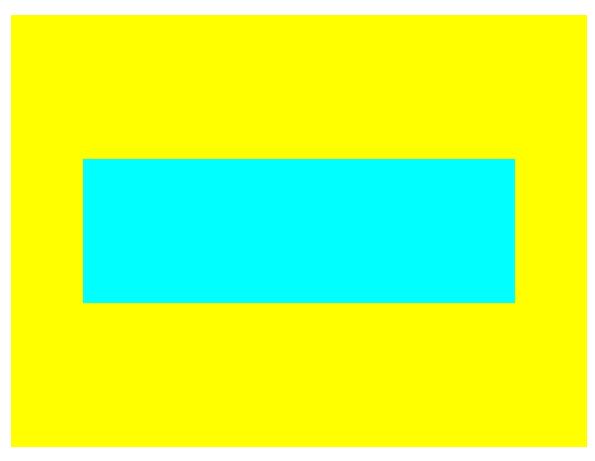
if (h_count < 800 && v_count < 600) begin

if (h_count > 100 && h_count < 300 && v_count > 200 && v_count < 400) begin

// Dreptunghi mijloc - galben
```

```
red <= 1;
          green <= 1;
          blue <= 0;
       end else begin
          // Fundal - magenta
          red <= 1;
          green <= 0;
          blue <= 1;
       end
     end else begin
       // leşirile RGB sunt 0 în afara zonei vizibile
       red \le 0;
       green <= 0;
       blue <= 0;
     end
  end else begin
    // Dacă semnalul de activare este 0, ieșirile RGB sunt 0
    red <= 0;
     green <= 0;
     blue <= 0;
  end
end
```

### Steag Galben dreptunghi Cyan



```
always @(posedge clk or posedge rst) begin

if (rst) begin

// Resetare ieşiri RGB la 0

red <= 0;

green <= 0;

blue <= 0;

end else if (en) begin

// Generarea culorii pe baza poziției curente

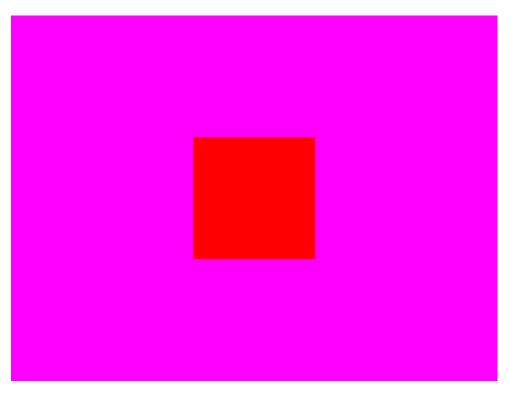
if (h_count < 800 && v_count < 600) begin

if (h_count >= 100 && h_count < 700 && v_count >= 200 && v_count < 400) begin

// Dreptunghi mijloc - Cyan
```

```
red \le 0;
          green <= 1;
          blue <= 1;
       end else begin
          // Fundal - Galben
          red <= 1;
          green <= 1;
          blue <= 0;
       end
     end else begin
       // leşirile RGB sunt 0 în afara zonei vizibile
       red \le 0;
       green <= 0;
       blue <= 0;
     end
  end else begin
    // Dacă semnalul de activare este 0, ieșirile RGB sunt 0
    red <= 0;
     green <= 0;
     blue <= 0;
  end
end
```

### Steag Magenta patrat Rosu



```
always @(posedge clk or posedge rst) begin

if (rst) begin

// Resetare ieşiri RGB la 0

red <= 0;

green <= 0;

blue <= 0;

end else if (en) begin

// Generarea culorii pe baza poziției curente

if (h_count < 800 && v_count < 600) begin

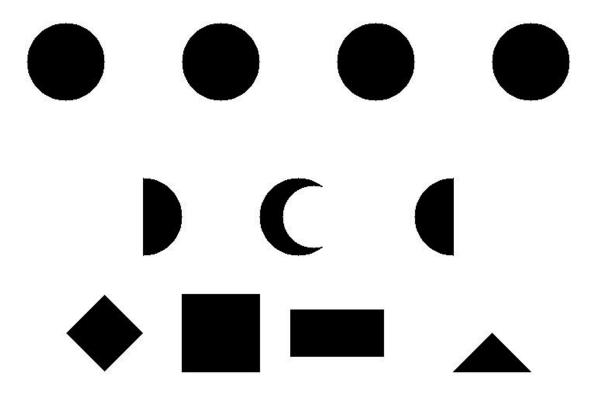
if (h_count >= 300 && h_count < 500 && v_count >= 200 && v_count < 400) begin

// Pătrat mijloc - Roșu

red <= 1;

green <= 0;
```

```
blue <= 0;
       end else begin
         // Fundal - Roz
         red <= 1;
          green <= 0;
          blue <= 1;
       end
     end else begin
       // leşirile RGB sunt 0 în afara zonei vizibile
       red \le 0;
       green <= 0;
       blue <= 0;
     end
  end else begin
    // Dacă semnalul de activare este 0, ieșirile RGB sunt 0
    red <= 0;
    green <= 0;
    blue <= 0;
  end
end
```



Cod: module vgadriver(
input clk,
input rst,
input enable,

output reg h\_sync, output reg v\_sync, output reg red, output reg green,

```
output reg blue
);
reg [11:0] v counter;
reg [11:0] h_counter;
always @(posedge clk or posedge rst) begin
  if (rst)
    h counter <= 0;
  else if (~enable)
    h counter <= 0;
  else if (h counter == 1039)
    h counter <= 0;
  else
    h counter <= h counter + 1;
end
always @(posedge clk or posedge rst) begin
  if (rst)
    v counter <= 0;
  else if (~enable)
    v_counter <= 0;
  else if ((h counter == 1039) && (v counter == 665))
    v_counter <= 0;
  else if (h counter == 1039)
    v counter <= v counter + 1;
end
```

```
always @(posedge clk or posedge rst) begin
  if (rst)
    h sync <= 1;
  else if (h counter >= 856 && h counter < 976 && enable == 1)
    h sync \leq 0;
  else
    h sync <= 1;
end
always @(posedge clk or posedge rst) begin
  if (rst)
    v sync <= 1;
  else if (v_counter >= 637 && v_counter < 643 && enable == 1)
    v sync \leq 0;
  else
    v sync <= 1;
end
always @(posedge clk or posedge rst) begin
  if (rst) begin
    red \le 0;
    green \leq 0;
     blue <= 0;
  end else if (h counter < 800 && v counter < 600 && enable == 1) begin
```

```
if (
       // Randul 1
       // Cercurile Primul Rand
       ((h counter - 100) * (h counter - 100) + (v counter - 100) *
(v counter - 100) <= 50 * 50) ||
       ((h counter - 300) * (h counter - 300) + (v counter - 100) *
(v counter - 100) <= 50 * 50) ||
       ((h_counter - 500) * (h_counter - 500) + (v_counter - 100) *
(v counter - 100) <= 50 * 50) ||
       ((h_counter - 700) * (h_counter - 700) + (v_counter - 100) *
(v counter - 100) <= 50 * 50) ||
       // Randul 2
       // Semicerc indreptat spre dreapta
       (((h counter - 200) * (h counter - 200) + (v counter - 300) *
(v counter - 300) \le 50 * 50) & h counter \ge 200) | |
       // Semicerc indreptat spre stanga
       (((h counter - 600) * (h counter - 600) + (v counter - 300) *
(v counter - 300) <= 50 * 50) && h counter <= 600) ||
       // Semiluna
       ((((h_counter - 400) * (h_counter - 400) + (v_counter - 300) *
(v counter - 300) \le 50 * 50) &&
       ~((h counter - 420) * (h counter - 420) + (v counter - 300) *
(v counter - 300) <= 40 * 40))) ||
       // Randul 3
```

```
// Patrat
       ((h counter >= 300 - 50 && h counter <= 300 + 50) && (v counter
>= 450 - 50 && v counter <= 450 + 50)) ||
       // Dreptunghi
       ((h counter \geq 450 - 60 && h counter \leq 450 + 60) && (v counter
>= 450 - 30 && v counter <= 450 + 30)) ||
       // Romb
       ((h counter >= 150 ? h counter - 150 : 150 - h counter) +
       (v counter >= 450 ? v counter - 450 : 450 - v counter) < 50) ||
       // Triunghi echilateral cu varful in sus
       (v counter >= 400 \&\&
        v counter <= 500 &&
        h counter >= 650 - (v counter - 450) &&
        h counter <= 650 + (v counter - 450))
     ) begin
       red \le 0;
       green \leq 0;
       blue <= 0;
    end else begin
       red <= 1;
       green <= 1;
       blue <= 1;
     end
```

```
end else begin
red <= 0;
green <= 0;
blue <= 0;
end
end
endmodule
```