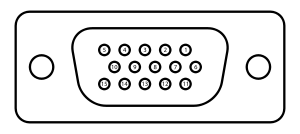
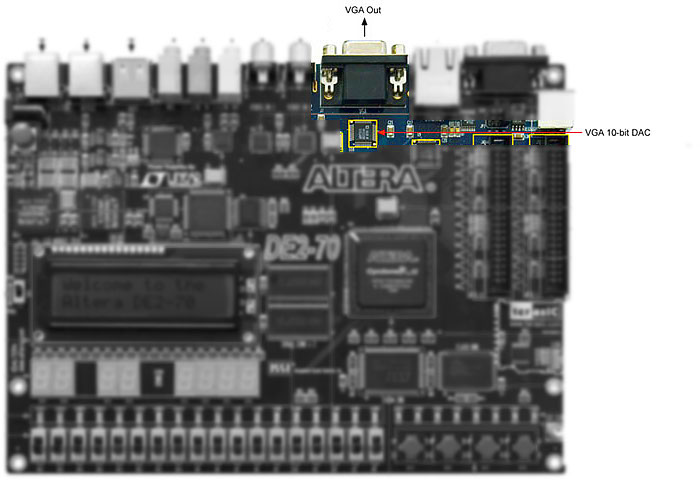
Opis interfejsu VGA

Rysunek przedstawia złącze damskie DE15

Najbardziej rozpowszechnionym rozwiązaniem interfejsu fizycznego dla VGA jest 3-rzędowe, 15 pinowe złącze DE15, nazywane inaczej D-Sub lub HD-15. Obsługuje ono duży zasięg rozdzielczości obrazu od 640×350px przy 70 Hz (24 MHz) przez 1280×1024px (SXGA) przy 85 Hz (160 MHz) aż do 2048×1536px (QXGA) przy 85 Hz (388 MHz).

**Interfejs fizyczny realizowany na złączu DE15:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pin 1** | RED | Red video | **Pin 9** | KEY/PWR | formerly key, now +5V DC |
| **Pin 2** | GREEN | Green video | **Pin 10** | GND | Ground (VSync, DDC) |
| **Pin 3** | BLUE | Blue video | **Pin 11** | ID0/RES | formerly Monitor ID bit 0, reserved since E-DDC |
| **Pin 4** | ID2/RES | formerly Monitor ID bit 2, reserved since E-DDC | **Pin 12** | ID1/SDA | formerly Monitor ID bit 1, [I²C](https://en.wikipedia.org/wiki/I²C) data since DDC2 |
| **Pin 5** | GND | Ground (HSync) | **Pin 13** | HSync | Horizontal sync |
| **Pin 6** | RED\_RTN | Red return | **Pin 14** | VSync | Vertical sync |
| **Pin 7** | GREEN\_RTN | Green return | **Pin 15** | ID3/SCL | formerly Monitor ID bit 3, [I²C](https://en.wikipedia.org/wiki/I²C) clock since DDC2 |
| **Pin 8** | BLUE\_RTN | Blue return |  |  |  |

**Układ DE2\_70 jest wyposażony w układ obsługujący te wyjścia – naszą rolą było stworzenie logiki obsługującej ten układ.**

|  |  |
| --- | --- |
| Interfejs w głównym module | Opis |
| oVGA\_CLOCK | zegar synchronizujący wysyłanie poszczególnych klatek |
| oVGA\_HS | sygnał synchronizacji poziomej |
| oVGA\_VS | sygnał synchronizacji pionowej |
| oVGA\_BLANK\_N |  |
| oVGA\_SYNC\_N |  |
| oVGA\_R | dane koloru czerwonego (10 bitów) |
| oVGA\_G | dane koloru zielonego (10 bitów) |
| oVGA\_B | dane koloru niebieskiego (10 bitów) |

|  |  |
| --- | --- |
| Interfejs w głównym module | Deklaracja typu wewnątrz modułu |
| oVGA\_CLOCK | output oVGA\_CLOCK; |
| oVGA\_HS | output oVGA\_HS; |
| oVGA\_VS | output oVGA\_VS; |
| oVGA\_BLANK\_N | output oVGA\_BLANK\_N; |
| oVGA\_SYNC\_N | output oVGA\_SYNC\_N; |
| oVGA\_R | output [9:0] oVGA\_R; |
| oVGA\_G | output [9:0] oVGA\_G; |
| oVGA\_B | output [9:0] oVGA\_B; |

Logika sterowania tym układem zostaje przeniesiona z głównego modułu *GreenScreen* do modułu *vga\_controller*.

Opis modułu *vga\_controller*

|  |  |
| --- | --- |
| Definicja modułu | Zadeklarowanie instancji tego modułu w module głównym |
| module vga\_controller(  inRed,  inGreen,  inBlue,  outRequest,  outVGA\_R,  outVGA\_G,  outVGA\_B,  outVGA\_H\_SYNC,  outVGA\_V\_SYNC,  outVGA\_SYNC,  outVGA\_BLANK,  iCLK,  iRST\_N,  ); | vga\_controller vga ( .outRequest(Read), .inRed(outVideo[9:0]), .inGreen(outVideo[19:10]  .inBlue(outVideo[28:10]),  .outVGA\_R(oVGA\_R), .outVGA\_G(oVGA\_G), .outVGA\_B(oVGA\_B), .outVGA\_H\_SYNC(oVGA\_HS), .outVGA\_V\_SYNC(oVGA\_VS), .outVGA\_SYNC(oVGA\_SYNC\_N), .outVGA\_BLANK(oVGA\_BLANK\_N), .iCLK(VGA\_CTRL\_CLK), .iRST\_N(DLY\_RST\_2)  ); |

|  |  |
| --- | --- |
| Interfejs | Deklaracja typu |
| inRed | input [9:0] inRed; |
| inGreen | input [9:0] inGreen; |
| inBlue | input [9:0] inBlue; |
| outRequest | output reg outRequest; |
| outVGA\_R | output reg [9:0] outVGA\_R; |
| outVGA\_G | output reg [9:0] outVGA\_G; |
| outVGA\_B | output reg [9:0] outVGA\_B; |
| outVGA\_H\_SYNC | output reg outVGA\_H\_SYNC; |
| outVGA\_V\_SYNC | output reg outVGA\_V\_SYNC; |
| outVGA\_SYNC | output reg outVGA\_SYNC; |
| outVGA\_BLANK | output reg outVGA\_BLANK; |
| iCLK | input iCLK; |
| iRST\_N | input iRST\_N; |

Moduł używa pliku konfiguracyjnego vga\_config.txt. Deklaruje on parametry obrazu. Jego treść to:

// Horizontal Parameter ( Pixel )

parameter H\_SYNC\_CYC = 112;

parameter H\_SYNC\_BACK = 248;

parameter H\_SYNC\_ACT = 1280;

parameter H\_SYNC\_FRONT = 48;

parameter H\_SYNC\_TOTAL = 1688;

// Vertical Parameter ( Line )

parameter V\_SYNC\_CYC = 3;

parameter V\_SYNC\_BACK = 38;

parameter V\_SYNC\_ACT = 1024;

parameter V\_SYNC\_FRONT = 1;

parameter V\_SYNC\_TOTAL = 1066;

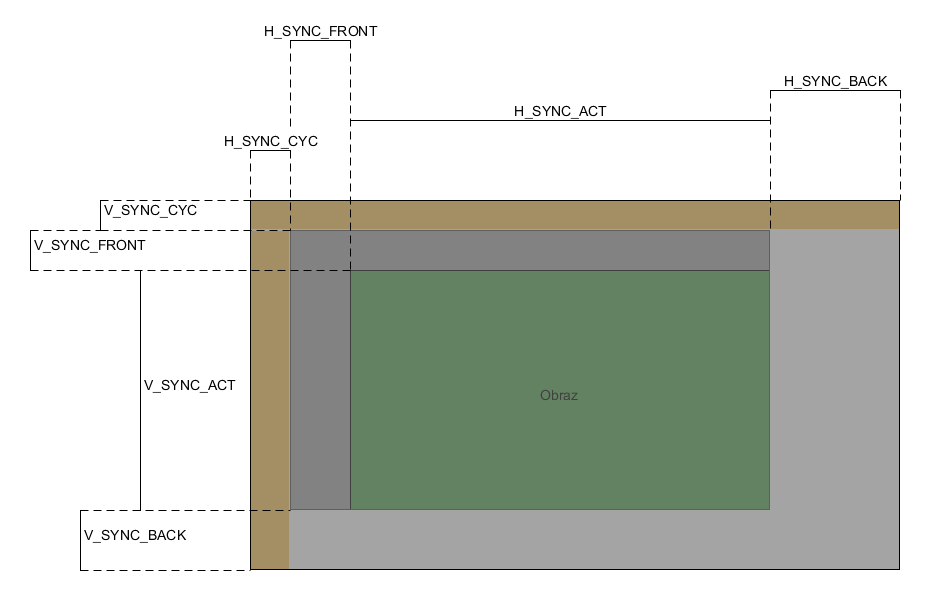
// Start Offset

parameter X\_START = H\_SYNC\_CYC+H\_SYNC\_BACK;

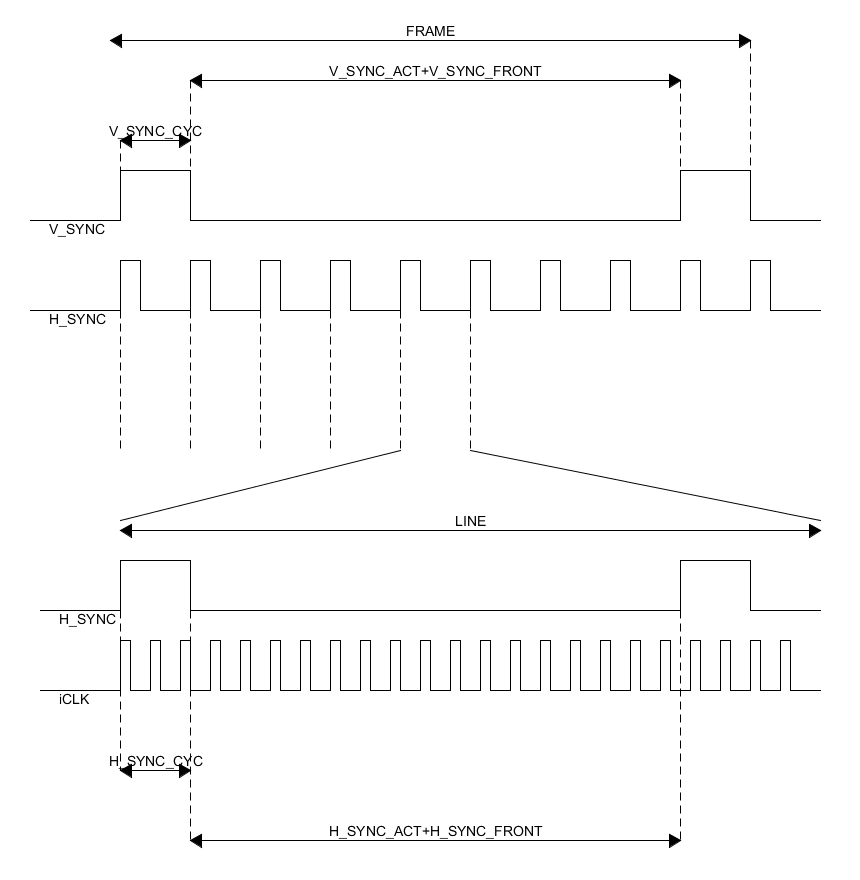
parameter Y\_START = V\_SYNC\_CYC+V\_SYNC\_BACK;

Jak widać moduł kontrolujący wyjście VGA jest dosyć prosty, obsługuje generowanie poprawnego sygnału na pinach oraz wysyłanie szeregowe wartości kolorów.

Zatem nasz moduł musi przejąć odpowiedzialność za prawidłową synchronizację pionową i poziomą obrazu.



Rysunek prezentujący poszczególne szczeliny czasowe służące do synchronizacji obrazu, proporcje zostały zmienione w celu poprawienia czytelności.

Przebiegi sygnałów sterujących

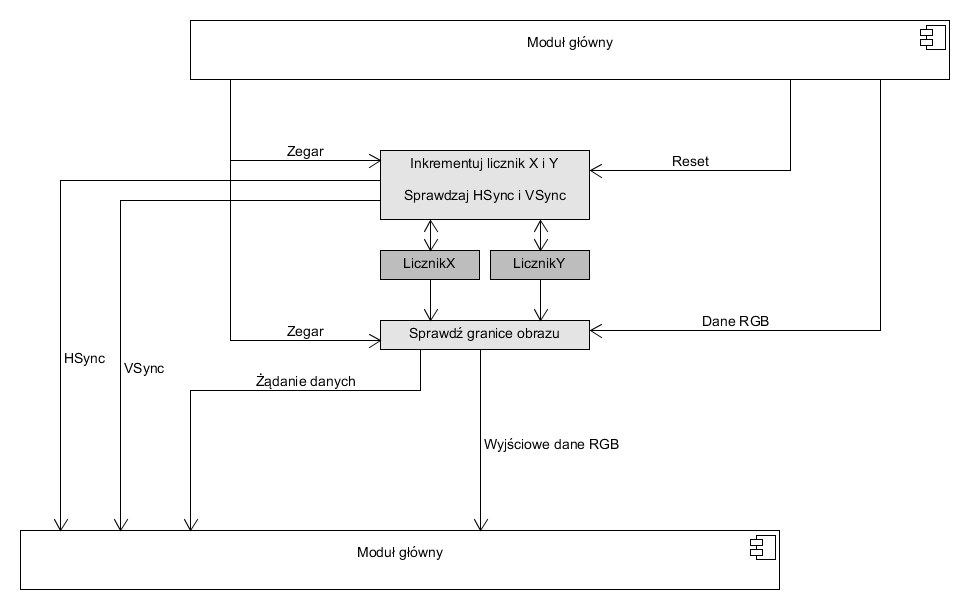


Diagram pseudo-DFD pokazujący działanie modułu.

Aby zapewnić prawidłowe działania skorzystamy z dwóch liczników.

// Counters for X and Y

reg [12:0] H\_Cont;

reg [12:0] V\_Cont;

**Licznik jest inkrementowany H\_Cont co interwał.**

mVGA\_H\_SYNC jest ustawiany na stan niski w początkowej fazie

linii.

always@(posedge iCLK or negedge iRST\_N)

begin

if(!iRST\_N) // Do not draw when reset

begin

H\_Cont <= 0;

mVGA\_H\_SYNC <= 0;

end

else

begin

// Increments while in correct horizontal area

if( H\_Cont < H\_SYNC\_TOTAL )

H\_Cont <= H\_Cont + 1;

else

H\_Cont <= 0;

//

if( H\_Cont < H\_SYNC\_CYC )

mVGA\_H\_SYNC <= 0;

else

mVGA\_H\_SYNC <= 1;

end

end

**Licznik V\_Cont gdy H\_Cont jest równy 0.**

mVGA\_V\_SYNC jest ustawiany na stan niski w początkowej

fazie klatki.

always@(posedge iCLK or negedge iRST\_N)

begin

if(!iRST\_N) // Do not draw when reset

begin

V\_Cont <= 0;

mVGA\_V\_SYNC <= 0;

end

else

begin

// When we are at the start of the line

if(H\_Cont==0)

begin

// Increments while we are in correct vertical area

if( V\_Cont < V\_SYNC\_TOTAL )

V\_Cont <= V\_Cont + 1;

else

V\_Cont <= 0;

// Checking if we are in the area in which sync should be low

if( V\_Cont < V\_SYNC\_CYC )

mVGA\_V\_SYNC <= 0;

else

mVGA\_V\_SYNC <= 1;

end

end

end

Ponadto zapewnione utrzymanie sygnałów kolorów na poziomie niskim poza obszarem obrazu.

assign mVGA\_R = ( H\_Cont >= X\_START && H\_Cont<X\_START + H\_SYNC\_ACT &&

V\_Cont >= Y\_START && V\_Cont<Y\_START + V\_SYNC\_ACT )

? inRed : 0;

assign mVGA\_G = ( H\_Cont >= X\_START && H\_Cont<X\_START + H\_SYNC\_ACT &&

V\_Cont >= Y\_START && V\_Cont<Y\_START + V\_SYNC\_ACT )

? inGreen : 0;

assign mVGA\_B = ( H\_Cont >= X\_START && H\_Cont<X\_START + H\_SYNC\_ACT &&

V\_Cont >= Y\_START && V\_Cont<Y\_START + V\_SYNC\_ACT )

? inBlue : 0;

Zmienne, którym powyżej przypisywane są wartości to tymczasowe, 10 bitowe rejestry.

Poniżej wykonywana co zegar, która przypisuje zmiennym wyjściowym wartości z rejestrów tymczasowych:

always@(posedge iCLK or negedge iRST\_N)

begin

if (!iRST\_N) // Checking if reset is high

begin

outVGA\_R <= 0;

outVGA\_G <= 0;

outVGA\_B <= 0;

outVGA\_BLANK <= 0;

outVGA\_SYNC <= 0;

outVGA\_H\_SYNC <= 0;

outVGA\_V\_SYNC <= 0;

end

else

begin

outVGA\_R <= mVGA\_R;

outVGA\_G <= mVGA\_G;

outVGA\_B <= mVGA\_B;

outVGA\_BLANK <= mVGA\_BLANK;

outVGA\_SYNC <= mVGA\_SYNC;

outVGA\_H\_SYNC <= mVGA\_H\_SYNC;

outVGA\_V\_SYNC <= mVGA\_V\_SYNC;

end

end