

# Kolloquium – Aufgabe 9

## Hinweise zur Implementierung

- Testbench ist nicht nötig, auch wenn dies in der Aufgabenstellung steht.
- Farbwerte dürfen nur an den Ausgang weitergegeben werden, wenn `in_active_region` anliegt. Sonst müssen 0en anliegen!
- Änderungen am Modul von Aufgabe 9 kann man später auch in ISE vornehmen.
- Wichtig ist, bei der Erstellung des ISE-Projekts auf die Konfiguration zu achten. (Spartan6, ...)

## Fehlersymptome:

Standbild mit Flackern am oberen und unterem Rand (blau).

- Grund: Die Farbausgabe ist fehlerhaft. `in_active_region` ist nicht berücksichtigt und/oder es werden keine 0en herausgegeben wenn nicht in active region.

## 9a/b) Graphic Output/ Synthesis of `graphic_output`

Erläutert euren Code und zeigt das richtige Ergebnis als Ausgabe des FPGAs auf dem Bildschirm.

Was ist die Aufgabe des packages hier?

## Relevant Code (9a.2 - graphic\_output\_package):

package graphic\_output\_package is

component sync\_pulse\_generator

generic (G\_H\_PIXEL\_NUMBER : integer := 800;

G\_H\_RESOLUTION : integer := 640;

G\_H\_FRONT\_PORCH : integer := 8;

G\_H\_BACK\_PORCH : integer := 48;

G\_H\_SYNC\_LENGTH : integer := 96;

G\_H\_SYNC\_ACTIVE : std\_logic := '0';

G\_V\_PIXEL\_NUMBER : integer := 525;

G\_V\_RESOLUTION : integer := 480;

G\_V\_FRONT\_PORCH : integer := 2;

G\_V\_BACK\_PORCH : integer := 33;

G\_V\_SYNC\_LENGTH : integer := 2;

G\_V\_SYNC\_ACTIVE : std\_logic := '0');

port (clk : in std\_logic := '0';

reset : in std\_logic := '0';

o\_h\_sync : out std\_logic := G\_H\_SYNC\_ACTIVE;

o\_v\_sync : out std\_logic := G\_V\_SYNC\_ACTIVE;

o\_in\_active\_region : out std\_logic := '0');

end component;

component graphic\_buffer is

port (clk : in std\_logic := '0';

reset : in std\_logic := '0';

i\_select : in std\_logic := '0';

i\_shift\_enable : in std\_logic := '0';

i\_load : in std\_logic := '0';

i\_rgb\_data : in std\_logic\_vector (32\*8-1 downto 0) := (others => '0');

o\_data\_req : out std\_logic := '1';

o\_rgb : out std\_logic\_vector (7 downto 0) := (others => '0');

end component;

component graphic\_buffer\_controller

generic (G\_H\_RESOLUTION : integer := 640;

G\_V\_RESOLUTION : integer := 480);

port (clk : in std\_logic := '0';

reset : in std\_logic := '0';

i\_data\_req\_reg\_0 : in std\_logic := '0';

i\_data\_req\_reg\_1 : in std\_logic := '0';

i\_read\_done : in std\_logic := '0';

i\_new\_frame\_ready : in std\_logic := '0';

```
    o_reg_select : out std_logic := '0';
    o_load_reg : out std_logic := '0';
    o_read_req : out std_logic := '0';
    o_read_address : out std_logic_vector (22 downto 0) := (others => '0');
    o_page_switched : out std_logic := '0');
end component;

end graphic_output_package;
```