

VEŽBA 4 i 5

XPS Video adapter (HW and SW)

Zadatak

Realizovati AXI perifernu jedinicu za ispis teksta i grafike na VGA ekran (*vga_ctrl*). Tekst za ispis na ekranu, kao i veličina fonta i boje teksta pozadine kao i karakteristike objekta grafike (veličina, oblik, boja, ...) se definišu iz programske podrške. Zadatak obuhvata izradu:

1. Realizacija fizičke podrške periferne jedinice za sprezanje VGA_TOP modula na AXI magistralu
2. Prateće programske podrške (drajver + demo aplikacija)
3. Prateće dokumentacije

Periferna jedinica

Na slici je dat blok dijagram periferne jedinice koju treba realizovati:

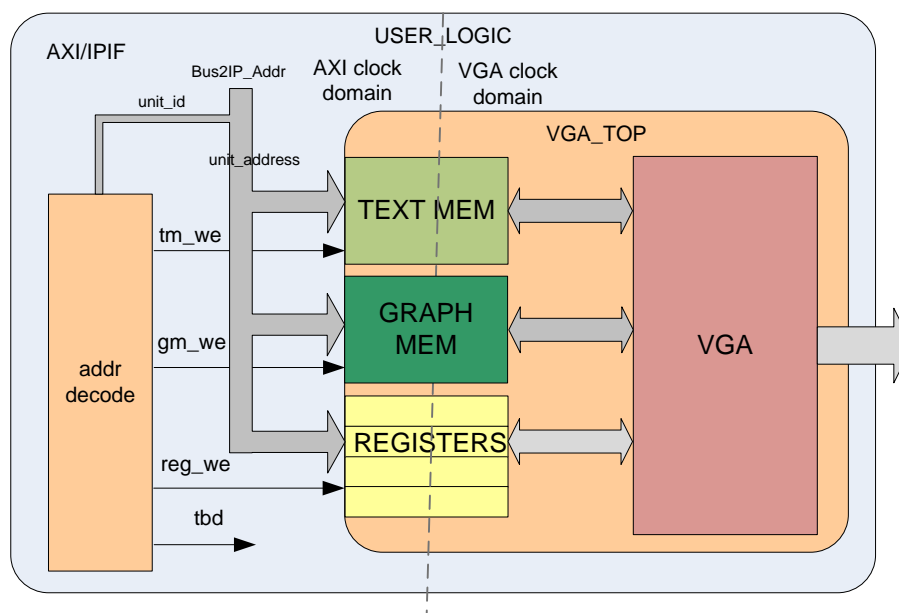


Figure 1: Blok dijagram *vga_ctrl*

- Registrima se podešavaju parametri i vraćaju informacije ka programskoj podršci:

Adresa	Ime registra	Širina registra	Opis
0x00000000	direct_mode	1	Direktni mod
0x00000004	display_mode	2	Displaj mod
0x00000008	show_frame	1	Prikaži ovir
0x0000000C	font_size	4	veličina fonta
0x00000010	foreground_color	24	boja teksta
0x00000014	background_color	24	boja pozadine
0x00000018	frame_color	24	Boja okvira

- VGA i CHAR_ROM moduli se preuzimaju iz prethodnih vežbi.
- TEXT_MEM je potrebno izmeniti kako bi napravili dvoprístupnu memoriju sa sinhronim čitanjem i nezavisnim taktovima (eng: *dual-port RAM with synchronous read (read through) and two clocks*) koja omogućuje sinhrono čitanje i pisanje na različitim takt domenima tj. da se piše sa AXI magistrale (na AXI takt domenu) a čita iz VGA_TOP modula (na vga takt domenu). Memorija sadrži adrese karaktera iz char_rom-a koji se ispisuju na ekranu.
- GRAPH_MEM je potrebno izmeniti kao i TXT_MEM tako da se piše sa AXI magistrale (na AXI takt domenu) a čita iz VGA_TOP modula na vga takt domenu). Memorija sadrži definiciju slike za prikaz na nivou pojedinačnih piksela. U nastavku je dat primer implementacije memorije za GRAPH_MEM modul.

```

DP_WR_GRAPHICS_MEM : process (wr_clk_i) begin
    if (rising_edge(wr_clk_i)) then
        if (we_i = '1') then
            graphics_mem(wr_addr_i) <= wr_data_i;
        end if;
    end if;
end process;

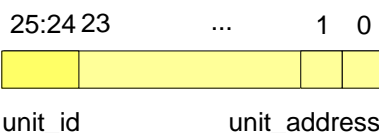
DP_RD_GRAPHICS_MEM : process (rd_clk_i) begin
    if (rising_edge(rd_clk_i)) then
        rd_addr <= mem_up_addr;
    end if;
end process;
rd_value <= graphics_mem(conv_integer(rd_addr));
rd_data_o <= rd_value(conv_integer(mem_lo_addr));

```

- Adresni prostor periferne jedinice je potrebno povećati na 64 M u skladu sa potrebama periferne jedinice.
- Memorijsku mapu periferne jedinice formiramo tako što sa 2 MSB bita [25:24] adresiramo jedan od modula (GRAPH_MEM, TEXT_MEM, REG_MAP, ...) a prestali deo LSB [23:0] koristimo kao adresu unutar modula kao sto prikazuje Figure 1.

Table 1: Peripheral Memory Space

Address space	Module	Read/Write
0x00000000 - 0x00FFFFFF	REG_MAP	R/W
0x01000000 - 0x01FFFFFF	TEXT_MEM	W
0x02000000 - 0x02FFFFFF	GRAPH_MEM	W
0x03000000 - 0x03FFFFFF	tbd	



unit_id	modul
00	REG_MAP
01	TEXT_MEM
10	GRAPH_MEM
11	reserved

Programska podrška

Programska podrška se sastoji od izrade drajvera i demo aplikacije. Drajver treba da omogući funkcije iz sledeće tabele:

unit_id	modul
print_char	Ispis znaka na ekranu
print_string	Ispis stringa na ekranu
clear_text	Brisanje teksta na ekranu
clear_graphics	Brisanje grafike na ekranu
clear_screen	Brisanje ekrana
set_foreground_color	Postavi boju teksta/objekta
set_background_color	Postavi boju pozadine
set_font_size	Postavi veličinu teksta
draw_rectangle	Nacrtaj pravougaonik
draw_circle	Nacrtaj krug
get_resolution	Pročitaj trenutnu rezoluciju

Dokumentacija

Dokument koji opisuje sve realizovano, po ugledu na laboratorijske vežbe.