Laboratorij za načrtovanje integriranih vezij

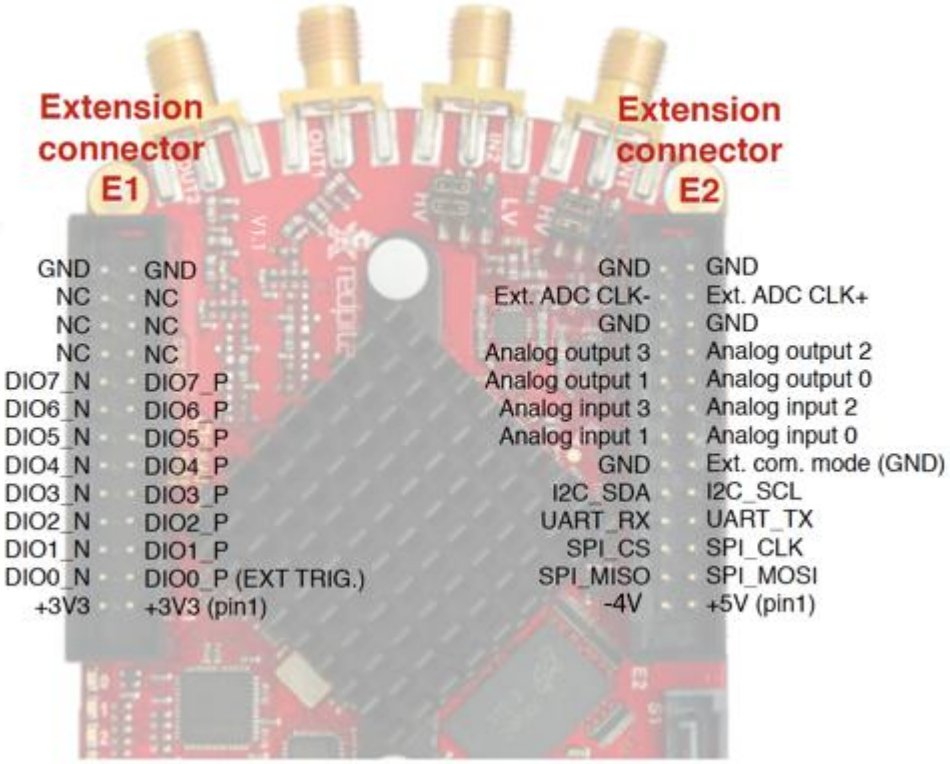
Univerza *v Ljubljani*
Fakulteta *za elektrotehniko*

Red Pitaya – vmesnik in VGA izhod

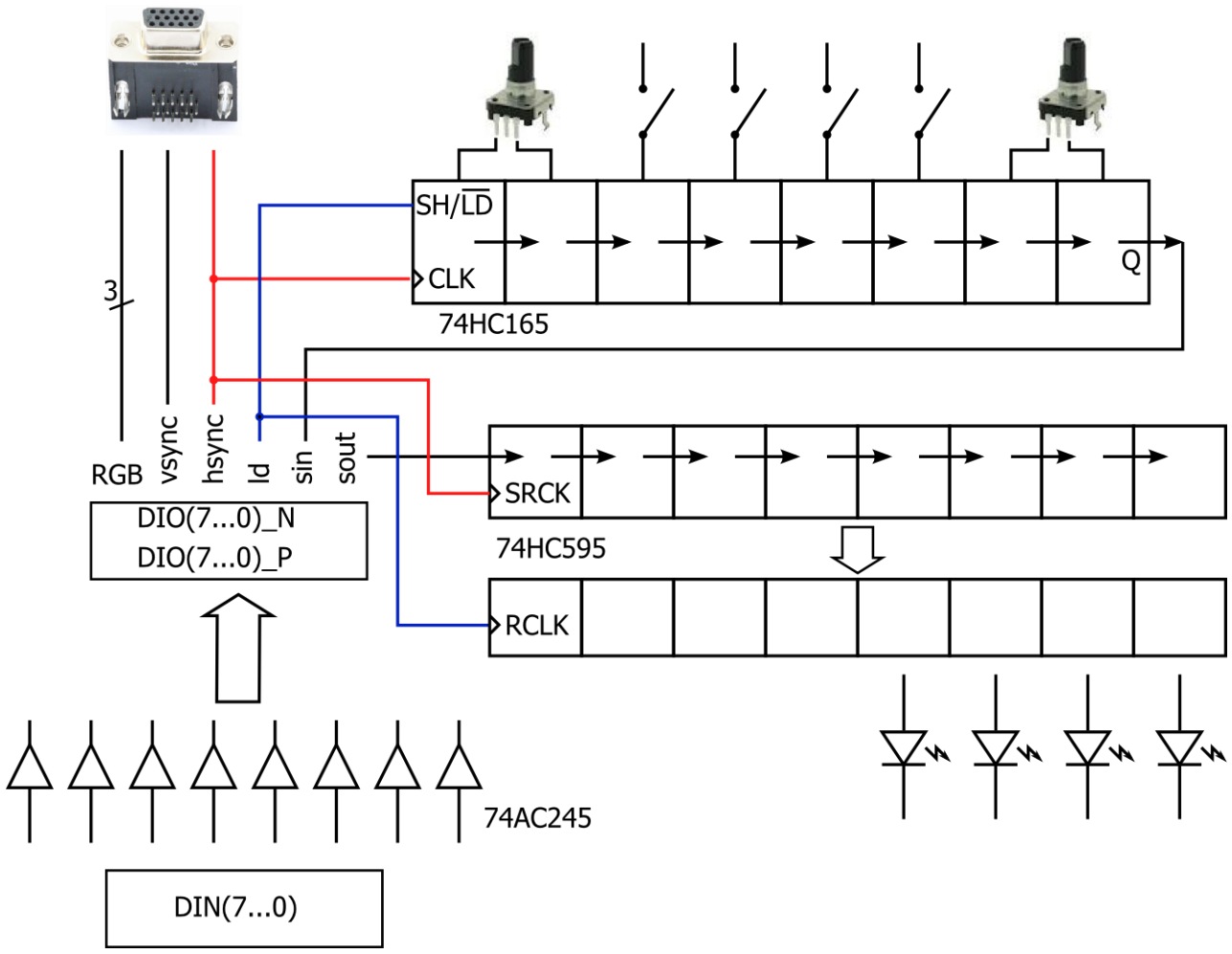
DES 2014/15 - razvoj vgrajenega sistema

Red Pitaya modul

- izdelava VGA vmesnika
  - grafični izhod preko signalov na razširitvenem konektorju
- tipke in rotacijski kodirnik
- dodatni digitalni vhodi
  - 8-bitni logični analizator



Red Pitaya vmesniški modul



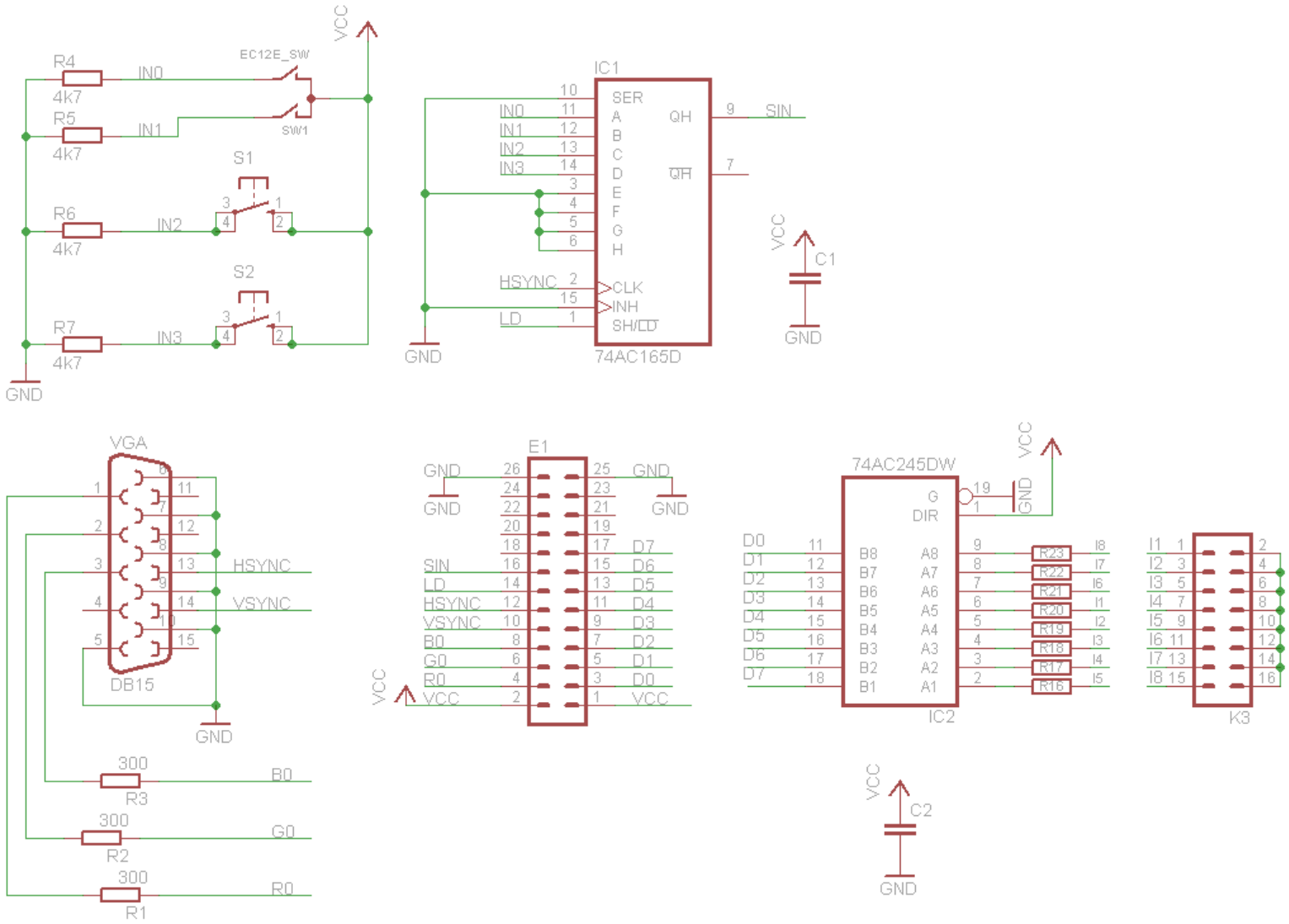
Delo po skupinah

- Arhitektura
  - načrt vezja v Eaglu, prototip vezja
  - meritve
- Grafika
  - prikazovanje sličice in znakov
  - prikazovanje grafaikona
- Vmesnik
  - branje tipk in povezava gradnikov
  - testiranje v Linux konzoli
- Skupna naloga
  - sinhronizacija in prikaz VGA slike

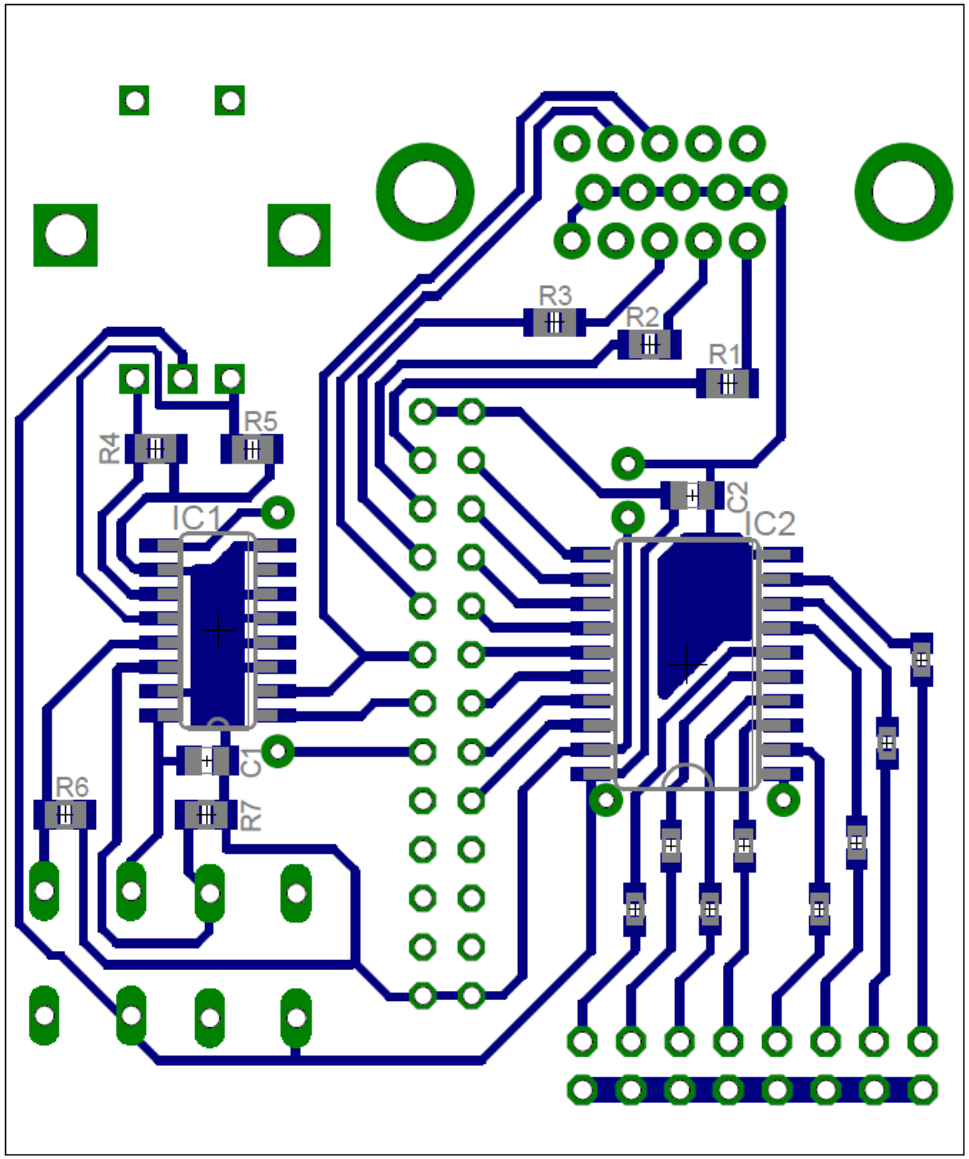
Arhitekturna skupina – razširitveni modul

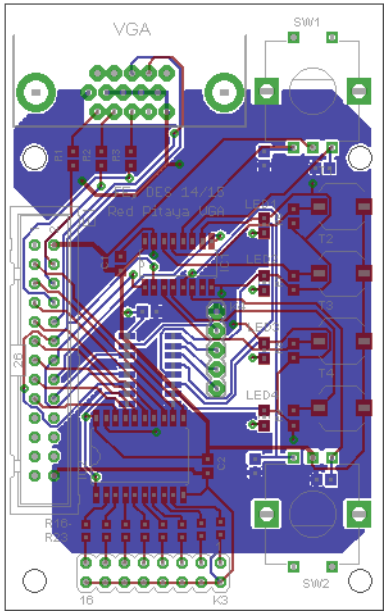
1. shema vezja z VGA, serijska vezava tipk (PISO), vhodno vezje za logični analizator
2. risanje načrta tiskanega vezja
   - razmestitev elementov, napajanje in ostale povezave
3. izdelava prototipa modula
   - prototipna izdelava (spajkanje) vezja
4. meritve na vezju
   - meritev lastnosti logičnih vhodov

Prototipno vezje – poenostavljena shema



Prototipno vezje – načrt tiskanega vezja





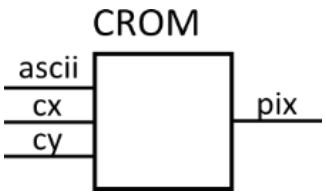
Grafika

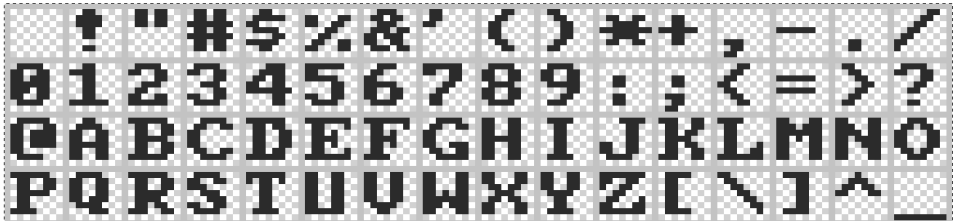
- VGA grafika: 800 x 600, 72 Hz

1. logika za prikaz sličice
   - logotip in testna struktura
2. prikaz vrstice znakov
   - branje CROM in prenos na izbrane koordinate
3. prikaz grafikona
   - risanje okvirja in osi
   - branje podatkov iz BRAM in prikaz točk
4. oblikovanje grafikona in preizkus na vezju

Grafika – prikaz vrstice znakov

- 64 slik (8x8) znakov od ASCII kode 32 naprej

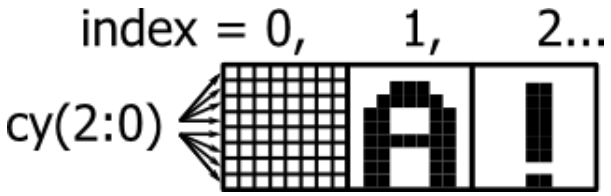




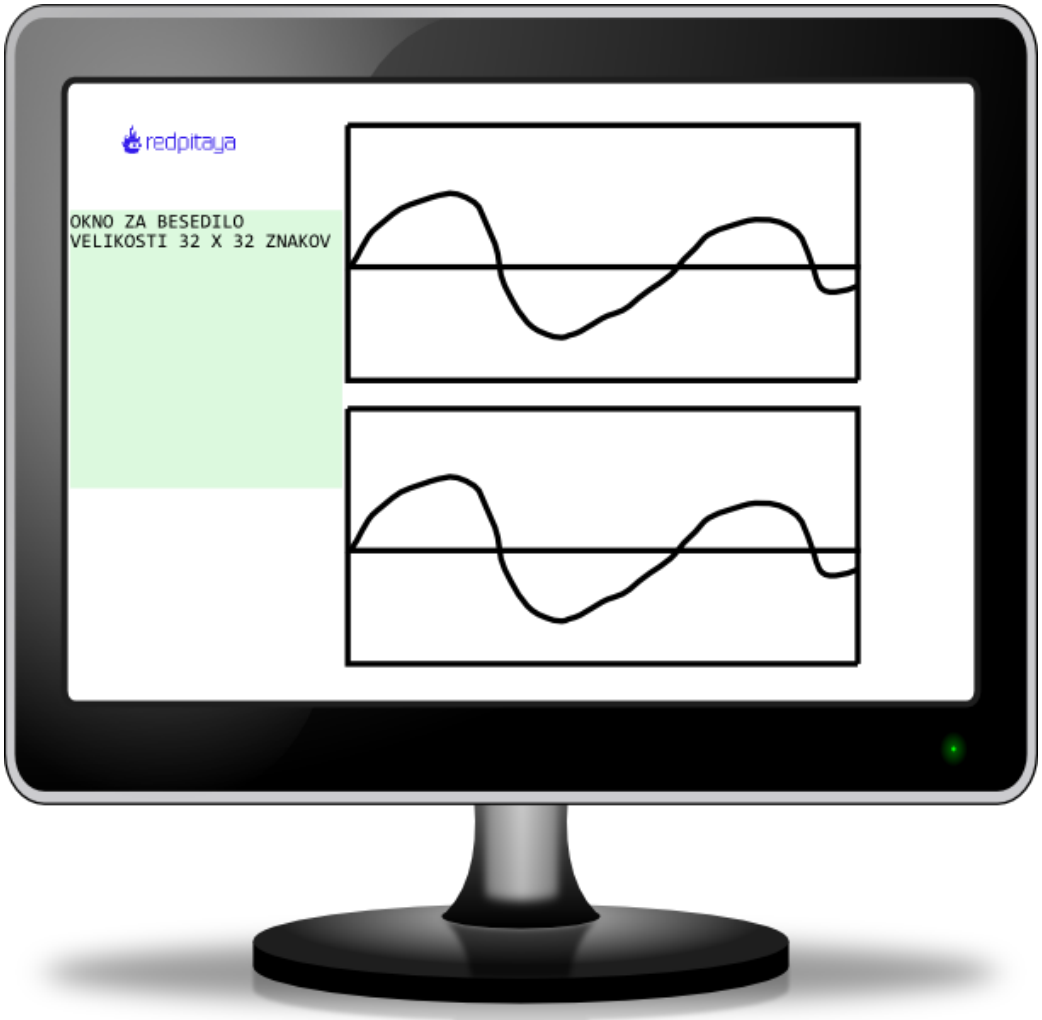
- branje točk iz zbirke chrom(), vsak znak zasede 8 vrstic:
  - zgornji del naslova: ASCII-32, spodnji del: 3-bitni cy
  - iz 8-bitnega podatka izluščimo točko: naslov 3-bitni cx

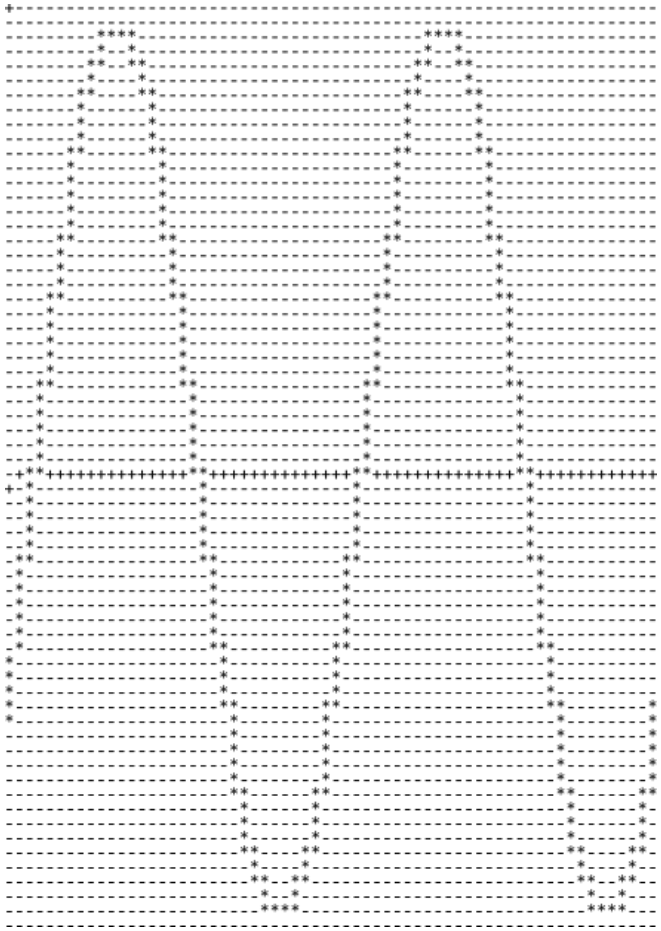
```
adr <= resize( unsigned( ascii ) - 32, 6 );
data <= chrom( to_integer( adr & cy(2 downto 0) ) );
pix <= data( to_integer( cx(2 downto 0) ) );
```

- branje znakov iz vrstice (zbirka 8 znakov)



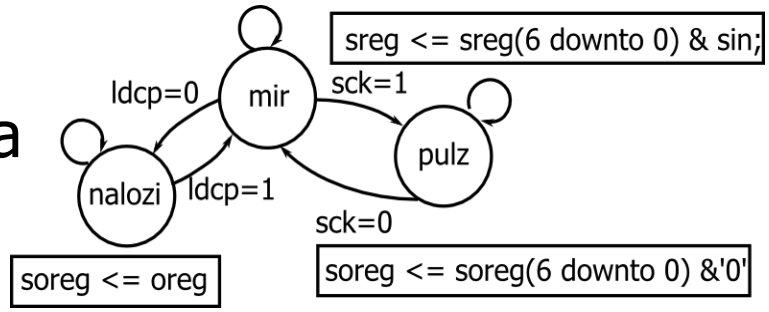
Grafika – prikaz grafikona

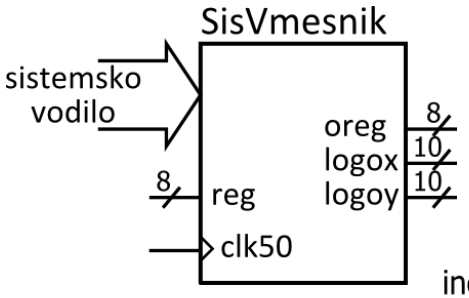


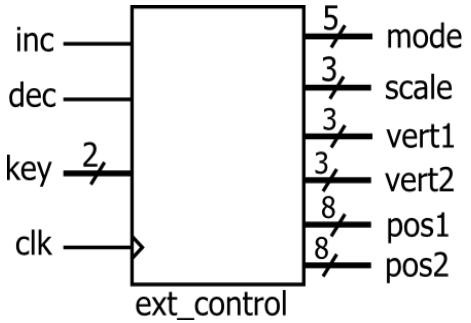


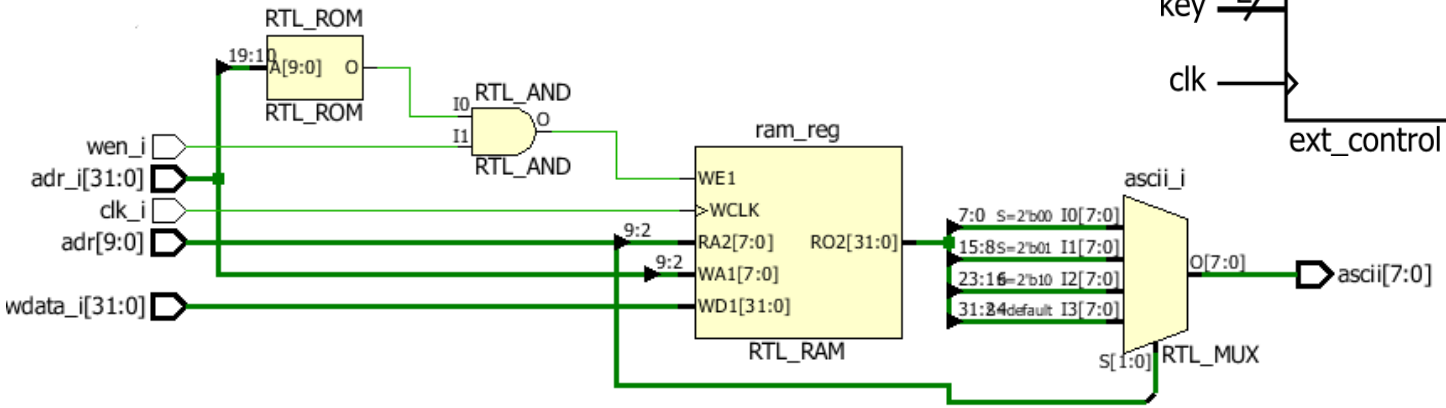
Vmesnik med HW in SW

1. krmiljenje PISO in branje tipk
2. dekodiranje rotacijskega kodirnika
   - nastavljanje registrov
3. povezava gradnikov
   - sistemsko vodilo in registri
4. celotno vezje
   - izdelava kontrolne komponente
   - prevajanje celotne kode za FPGA

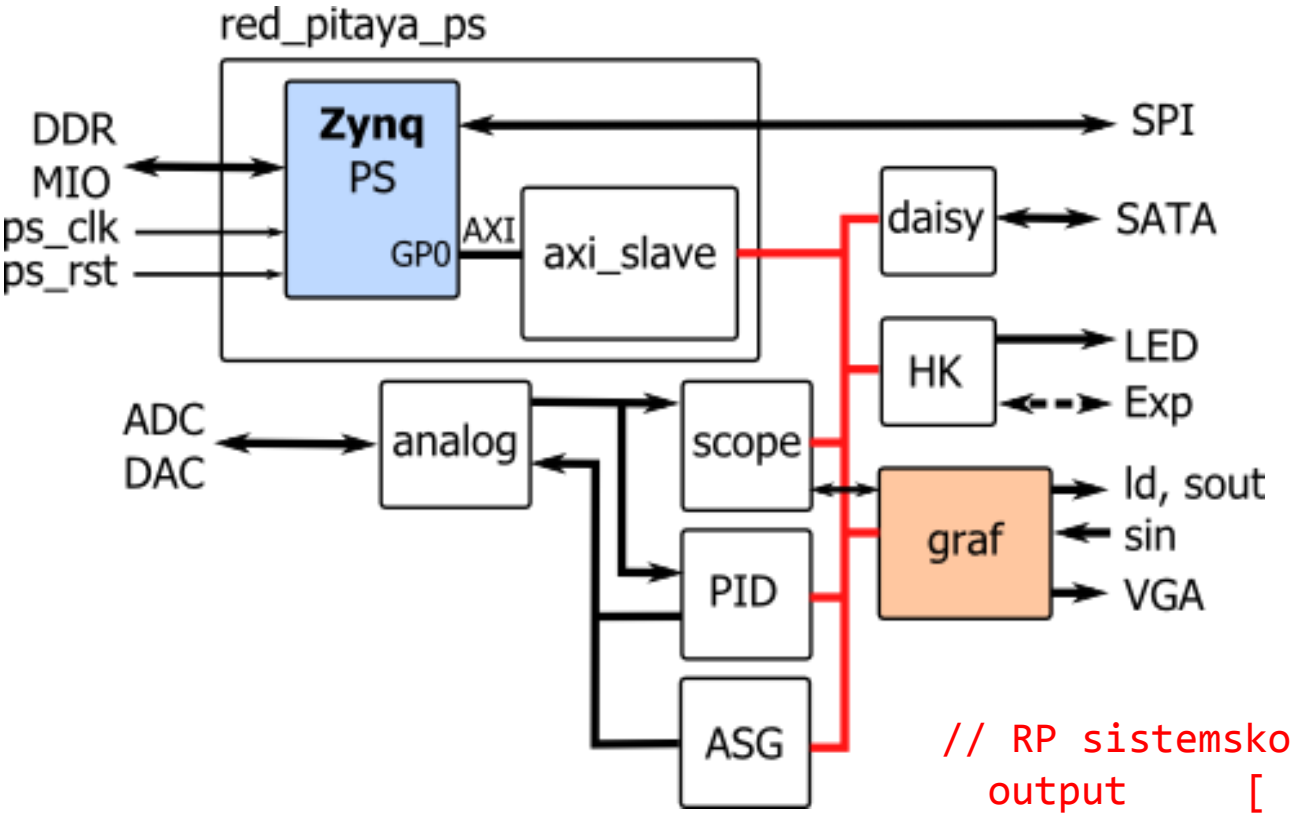






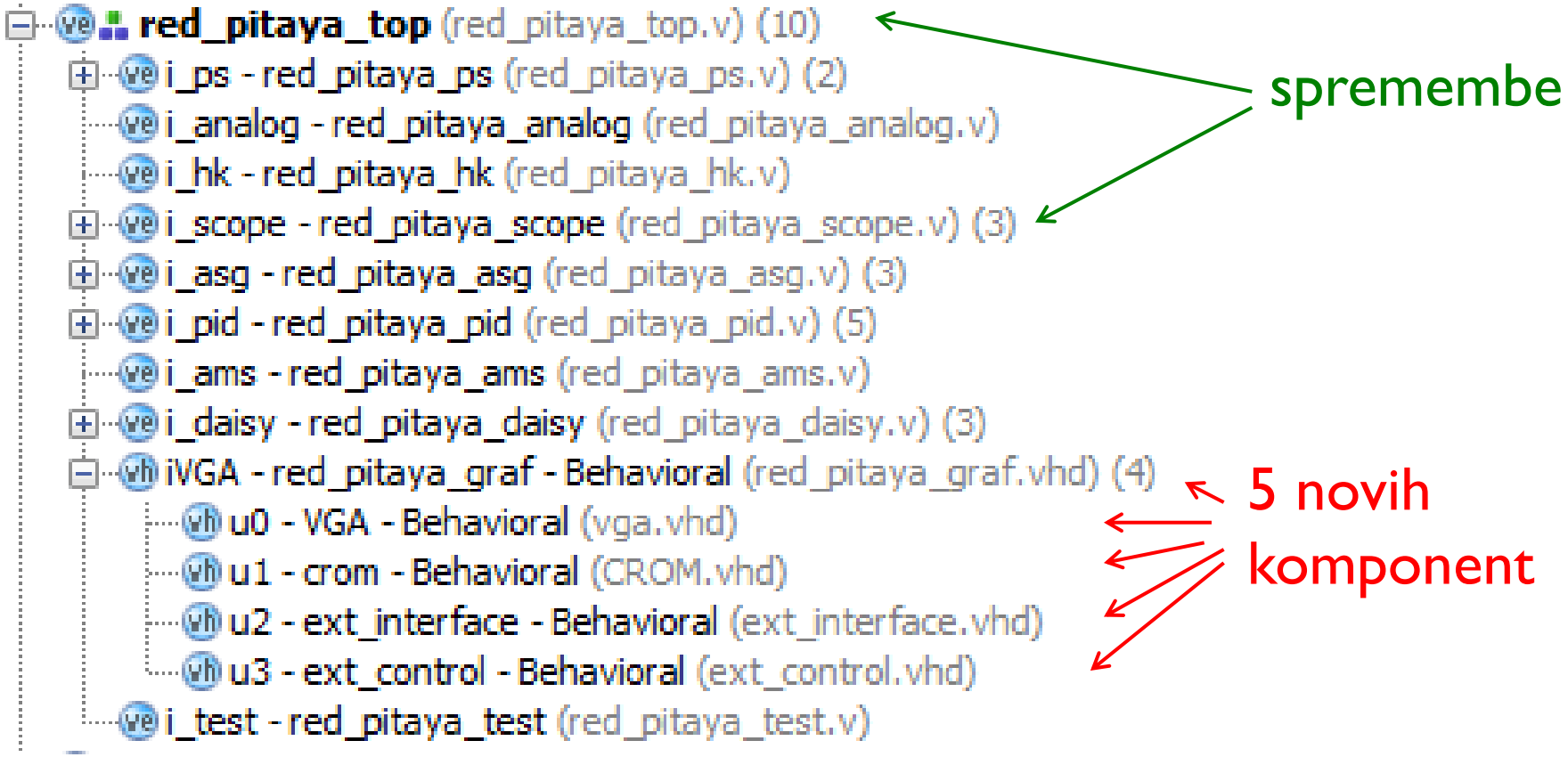


Sistem v vezju RedPitaya



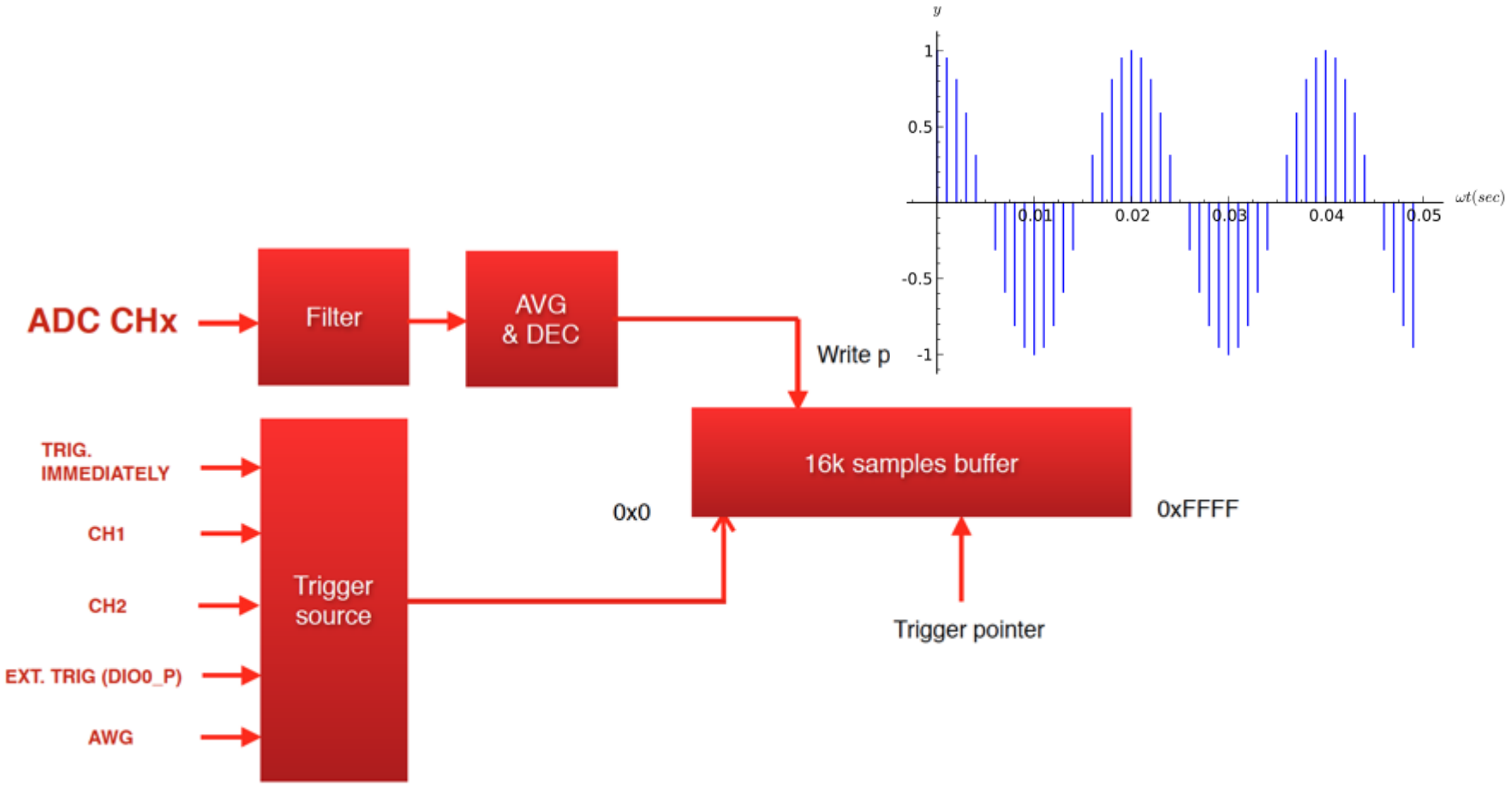

```
// RP sistemsko vodilo
  output      [ 31: 0] sys_addr_o
  output      [ 31: 0] sys_wdata_o
  output reg  [  3: 0] sys_sel_o
  output reg           sys_wen_o
  output reg           sys_ren_o
  input       [ 31: 0] sys_rdata_i
  input                sys_err_i
  input                sys_ack_i
```

Komponente prototipnega sistema



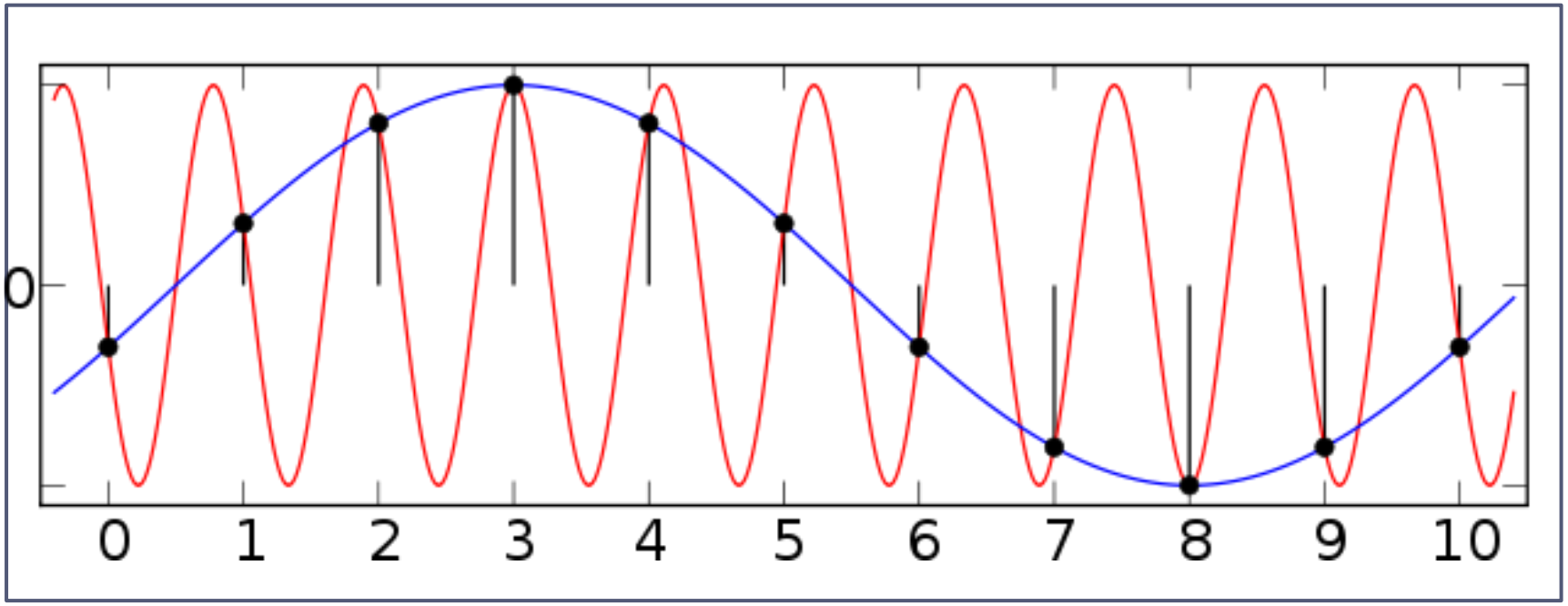
Vhodni del komponente red_pitaya_scope

- vzorčenje signalov iz ADC
- dodali smo vzporedni pomnilnik, ki ga bere graf komponenta



Obdelava vhodnih podatkov

- branje točke iz pomnilnika in skaliranje po časovni osi
  - pomnilnik 16k vzorcev, beremo 512 vrednosti (decimacija)
  - Pred decimacijo bi bilo potrebno filtriranje signala, sicer lahko opazujemo navidezni signal!



Obdelava podatkov komponente Scope

- vertikalno skaliranje (14 > 8 bitov) z nasičenjem:

```
constant sat1: signed(7 downto 0) := "01111111";
constant sat0: signed(7 downto 0) := "10000000";
```

```
psat: process (ack_data, ack_datb, vert1, vert2)
begin
  case vert1 is
    when "000" =>
      g1dat <= signed(ack_data(13 downto 6));

    when "001" =>
      if ack_data(13 downto 12)="01" then g1dat <= sat1;
      elsif ack_data(13 downto 12)="10" then g1dat <= sat0;
      else g1dat <= signed(ack_data(12 downto 5)); end if;
    ...
```

Določanje barve izhodnih točk

- risanje logotipa, besedila, črt, in točk grafikonov

```
elsif cx >= 256 and cx < 768 and cy >= 32 and cy < 512+32 then
   rgb <= "000";
   if cx=512 or cy=288 then   -- glavne osi bele barve
      rgb <= "111";
   elsif oe='1' and (cx=262 or cx=387 or cx=637 or cx=762 or
                              cy=51  or cy=169 or cy=407 or cy=525) then
      rgb <= "111";
   end if;
   if (vert1/="101") and cx(2)='0' and cy=g1abs then   -- črtkana abscisa
      rgb <= "100";
   end if;
...
```

Vmesnik in registri prototipnega sistema

- dekodiranje naslova: 0x40600000 + odmik

odmik	register	pomen
0x0000	mode (R, 5bit)	način rotacijskega kodirnika
0x0008	logox (W, 8 bit)	X položaj logotipa
0x000C	pos0 (W, 8 bit)	Pozicija grafikona 0
0x0010	scale (R, 3 bit)	Horizontalna skala
0x0014	vert1 (R, 3 bit)	Verikalna skala za graf1
0x0018	vert2 (R, 3 bit)	Verikalna skala za graf2
0x0020	graf0 (W, 8 bit)	Podatki za grafikon 0
0x0400 - 0x07FC	txt (W, 32 bit)	Besedilo (4 znaki / naslov)

Prevajanje prototipnega sistema

- zasedenost vezja Xilinx Zynq xc7z010clg400

	Red Pitaya V0.90	Red Pitaya in VGA grafika
Slice LUT	3641 / 21%	4223 / 24%
Slice Registers	3998 / 11%	4033 / 11%
Slice	1357 / 30%	1576 / 36%
Block RAM	28 / 47%	40 / 66%
DSP	26 / 80%	26 / 80%
IOB	95 / 95%	94 / 94%

Programski del osciloskopa z VGA izhodom

- program v zanki bere način delovanja rotacijskega kodirnika in nastavljene vrednosti skaliranja in izpisuje besedilo

- dostop do HW registrov v Linuxu :

```
fd = open("/dev/mem", O_RDWR | O_SYNC);

addr = 0x40600000;  /* Map one page */
map_base = mmap(0, MAP_SIZE, PROT_READ | PROT_WRITE, MAP_SHARED, fd, addr & ~MAP_MASK);
if(map_base == (void *) -1) FATAL;

virt_addr = map_base + (addr + 0x10 & MAP_MASK);
scale = *((unsigned long *) virt_addr);

if (scale==0) sprintf(str, " Scale: 1 us/div");
```

Poročilo: 3-5 strani, do **15.6.**, andrej.trost@fe.uni-lj.si

- Arhitektura
  - predstavi shemo končnega vezja, izseki in razlaga prenosa podatkov
  - VGA vmesnik, izračun uporov pri 3-bitnem in 1-bitnem D/A
  - seznam uporabljenih elementov
  - napetostni nivoji, karakteristika logičnih vhodov
- Grafika
  - razloži VGA komponento in sinhronizacijske signale
  - predstavi delovanje logike za prikaz slike in vrstice znakov
  - opiši vezje za prikaz grafikona, nekaj izsekov iz kode z razlago
  - nariši blokovno shemo z vsemi komponentami za prikaz slike
- Vmesnik
  - razloži delovanje zaporednega vmesnika in rotacijskega dekoderja
  - predstavi vmesnik za sistemsko vodilo z izseki VHDL kode
  - blokovna shema: povezava na sist. vodilo in struktura vezja
  - predstavi rezultate prevajanja celotnega vezja (zasedenost vezja)