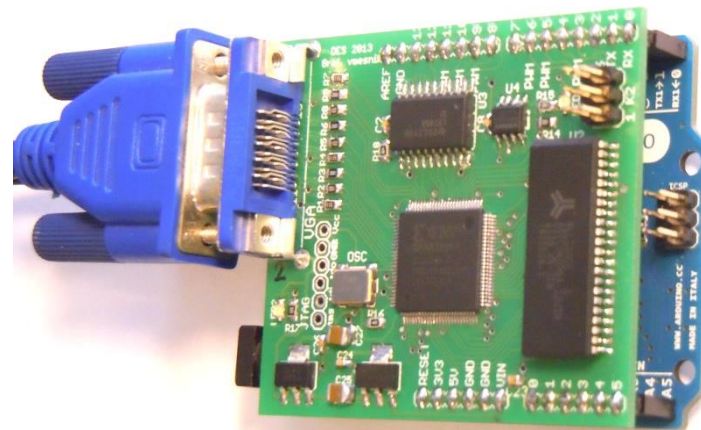




Laboratorij za načrtovanje integriranih vezij

Univerza *v Ljubljani*
Fakulteta *za elektrotehniko*

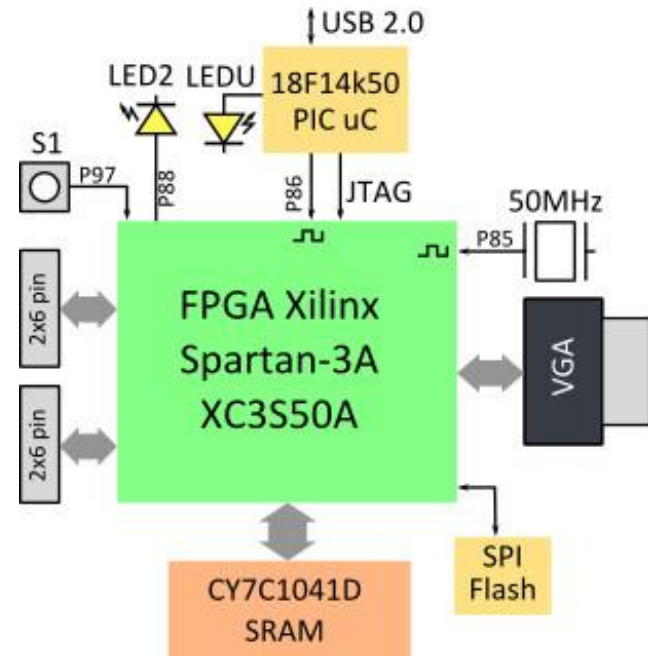
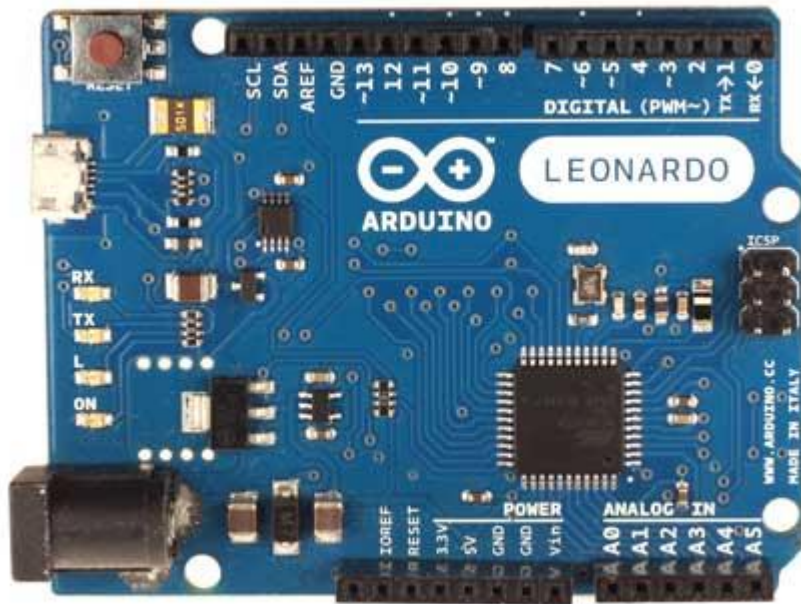


Arduino grafični vmesnik

DES 2012/13 - razvoj vgrajenega sistema

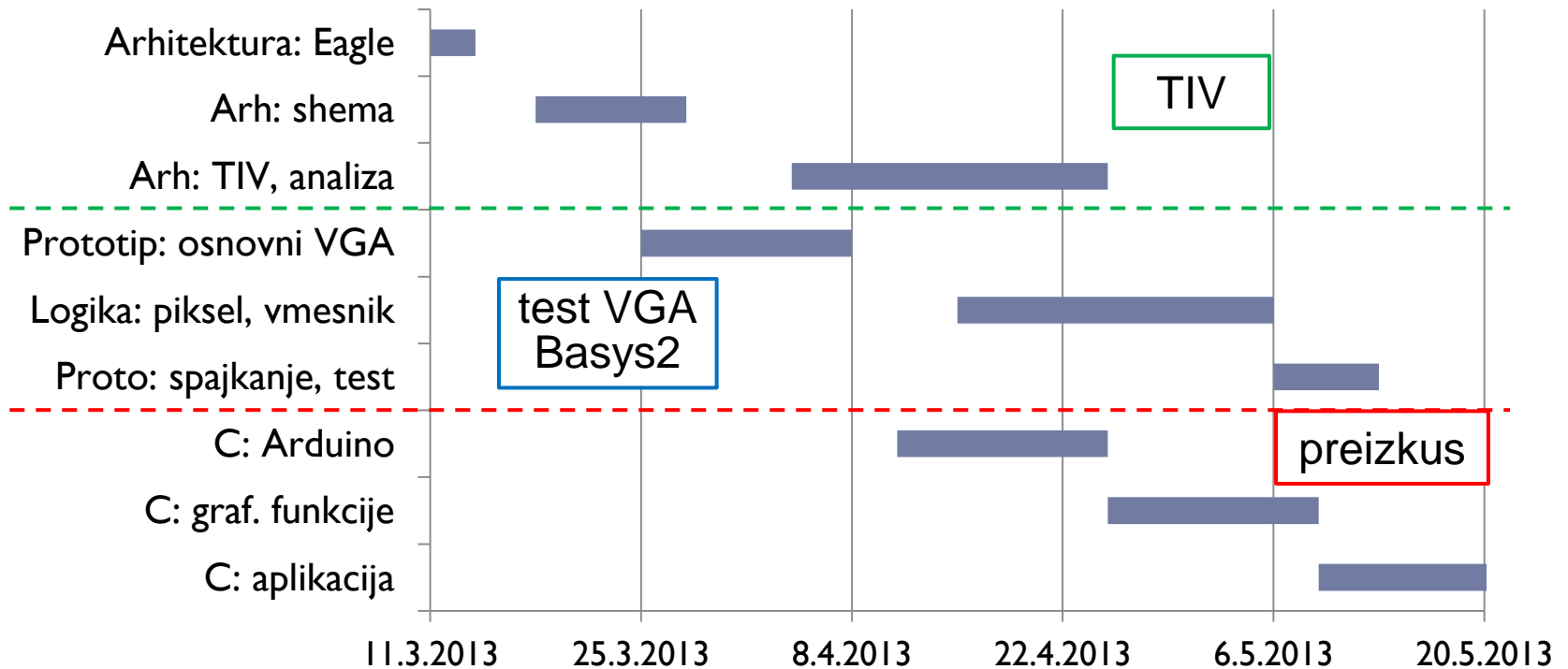
Arduino grafični vmesnik

- ▶ Arduino Leonardo
 - ▶ mikroprocesorska plošča z Atmel AVR
- ▶ Grafični vmesnik z analognim izhodom VGA
 - ▶ modificirana razvojna plošča S3A



Terminski plan - Gantogram

► Naloge za skupine: Arhitektura, Prototipna logika in C

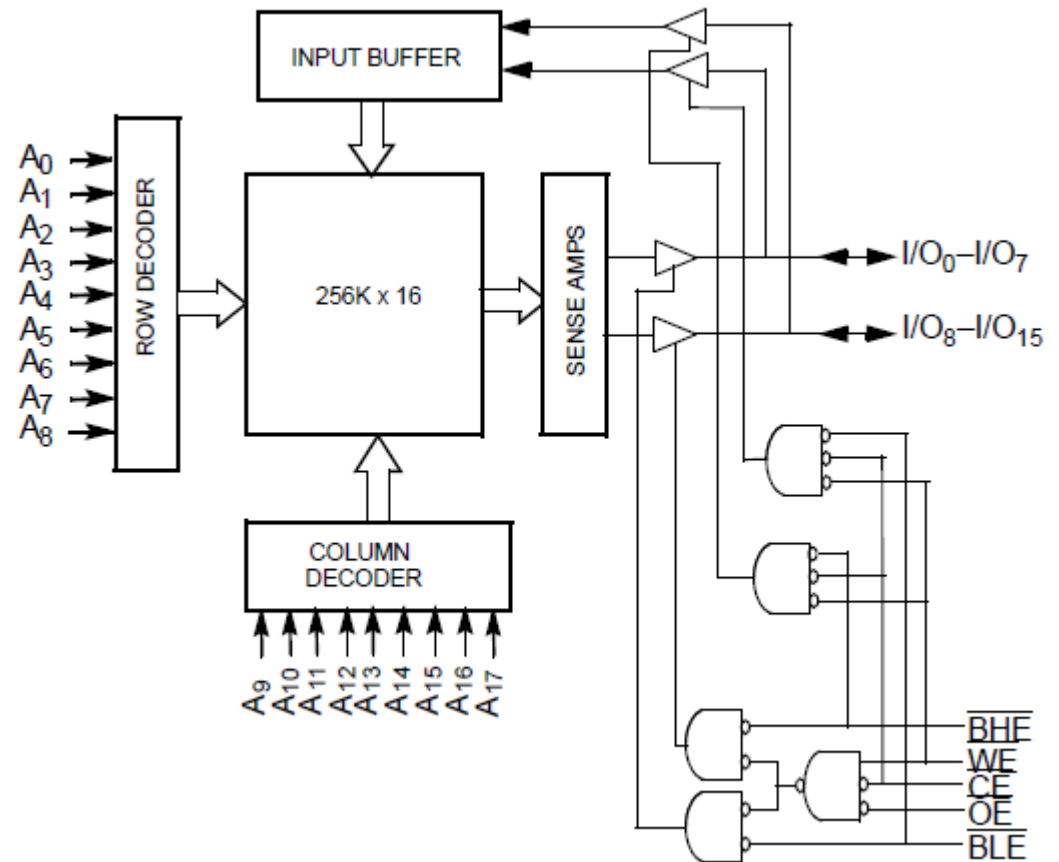


Lastnosti grafičnega vmesnika

- ▶ Način VGA: 640x480, 60Hz, 256 barv
 - ▶ prikazujemo okno: 512 x 480 (256k statični RAM)
- ▶ Logika v FPGA XC3s50A (1.5k LC, 3x 18k RAMB)
 - ▶ branje iz RAM in prikaz na VGA (RAMDAC)
 - ▶ vmesnik za vpisovanje točk v RAM
 - ▶ vmesnik za prikaz znakov
 - ▶ prikazovanje sličic (sprite)
- ▶ Osnovne funkcije za Arduino
 - ▶ prikazovanje točke z določeno barvo
 - ▶ izpis besedila
 - ▶ risanje črt in pravokotnikov
 - ▶ risanje in pomikanje sličic

Statični pomnilnik

- ▶ 256k x 16 bitov
 - ▶ BHE=0, BLE=0, CE=0

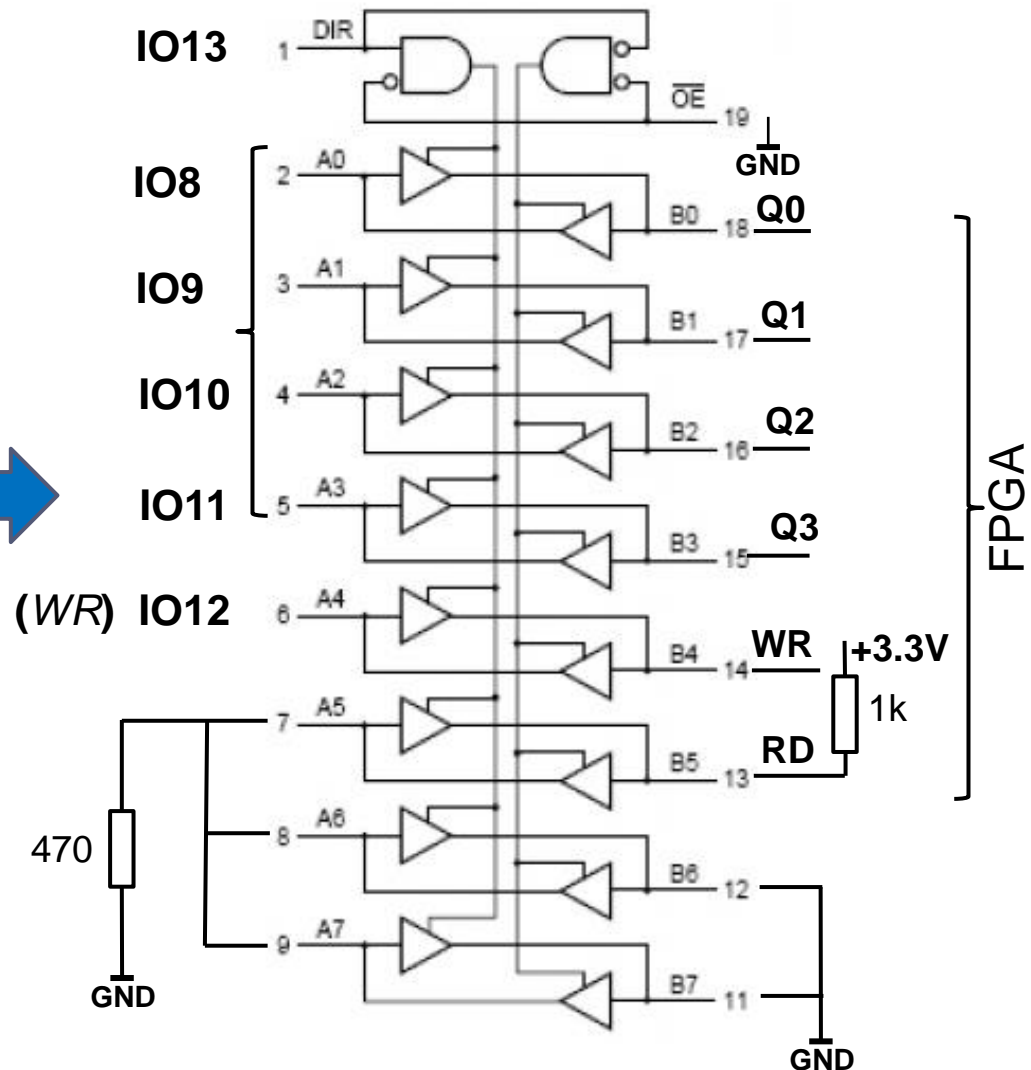
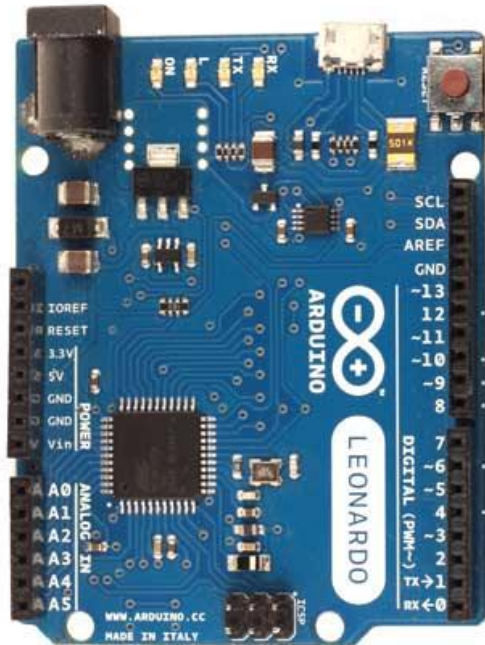


Arhitektura: vmesnik za Arduino

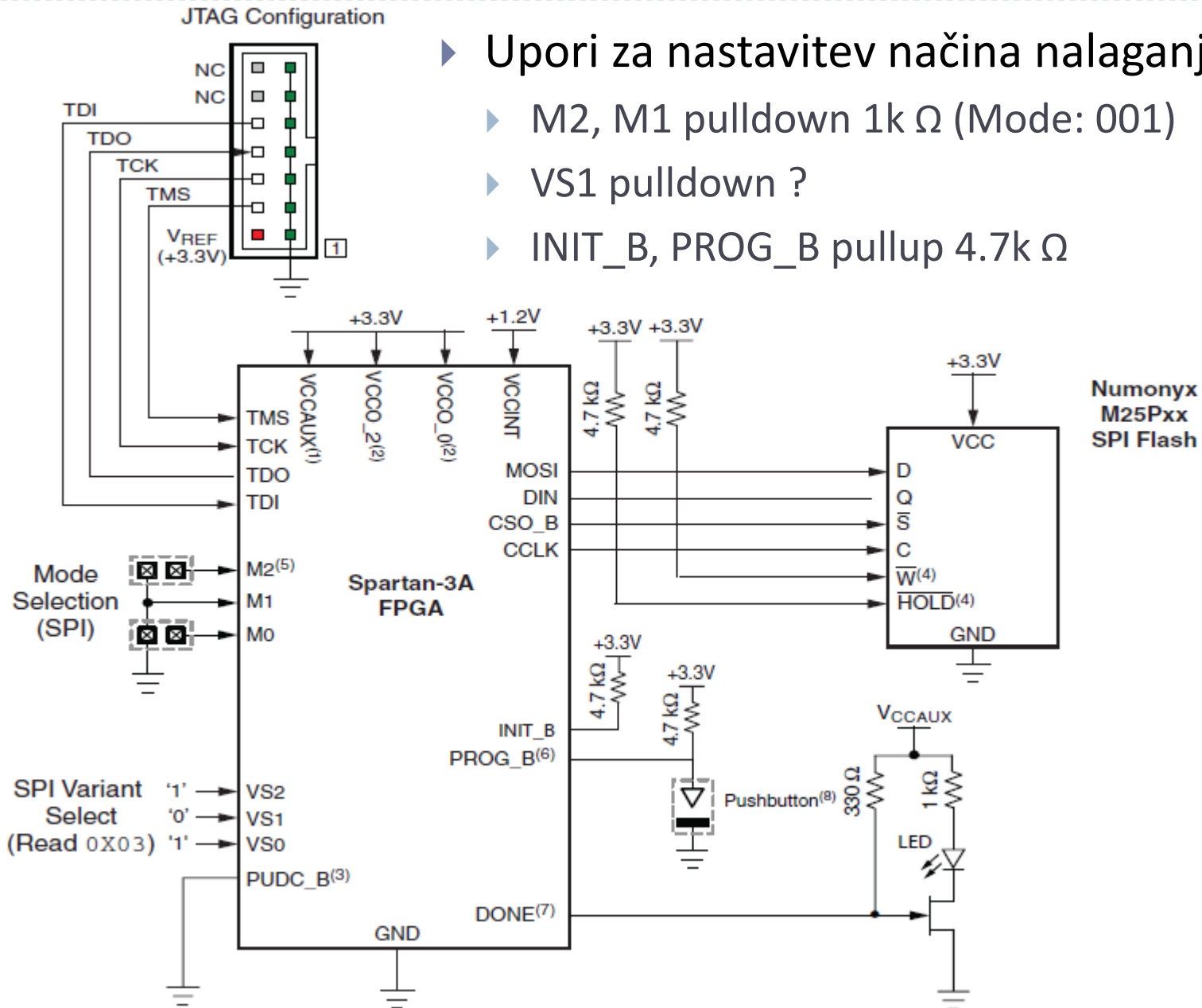
- ▶ Izenačevalnik nivojev (**transceiver**) 74245, napajanje 3.3V

A: AVR 5V, B: FPGA 3.3V

- ▶ vpis: DIR=1, WR impulz
- ▶ branje: DIR=0



Arhitektura: signali za nalaganje FPGA

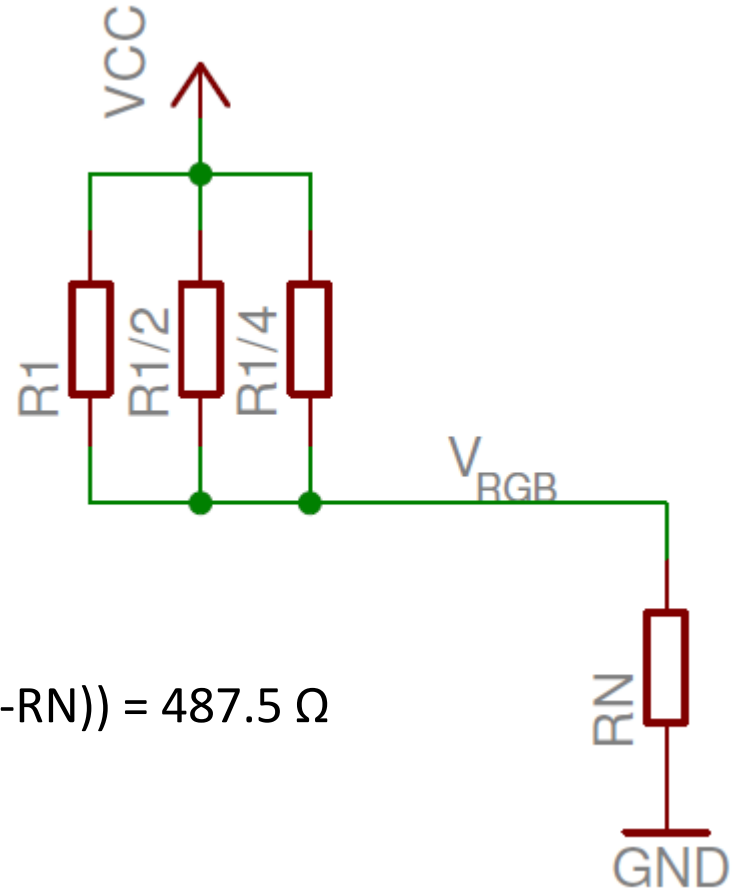


Arhitektura: izračun uporov za VGA D/A

- ▶ $R_N = 75\Omega$, $V_{CC} = 3.3V$, $V_{RGB} = 0.7V$
 - ▶ ko so vsi biti na V_{CC} naj bo na izhodu V_{RGB}
- ▶ Napiši enačbe vezja in izračunaj R
- ▶ Izberi iz nabora vrednosti npr. E24

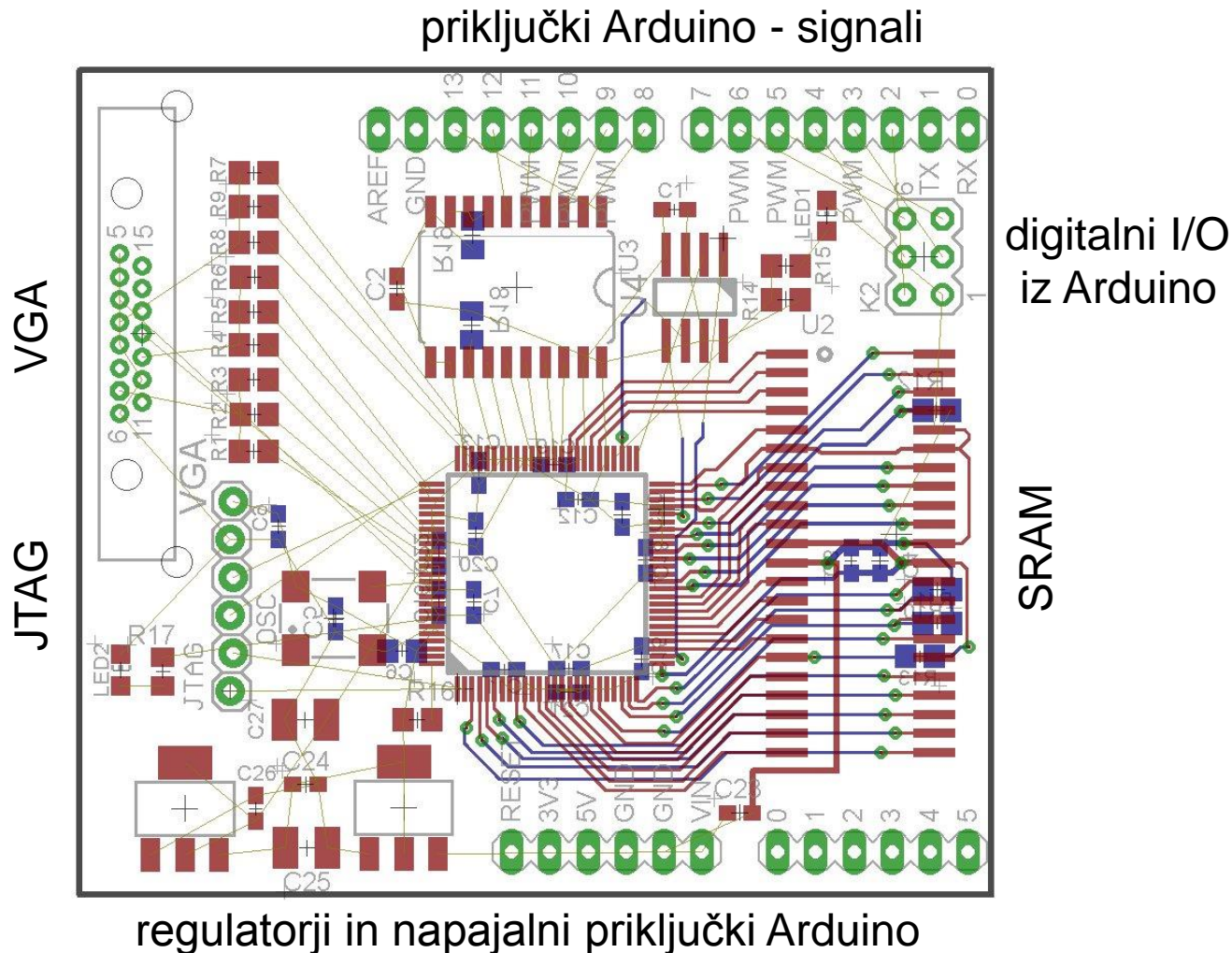
10 12 15 18 22 27 33 39 47 56 68 82
11 13 16 20 24 30 36 43 51 62 75 91

$$R_{1/4} = (1/4 + 1/2 + 1) / (1 / ((R_N * V_{CC}) / V_{RGB} - R_N)) = 487.5 \Omega$$



Arhitektura: tiskano vezje

- ▶ priključki Arduino, modifikacija VGA, pretvorba logičnih nivojev, signali za SRAM



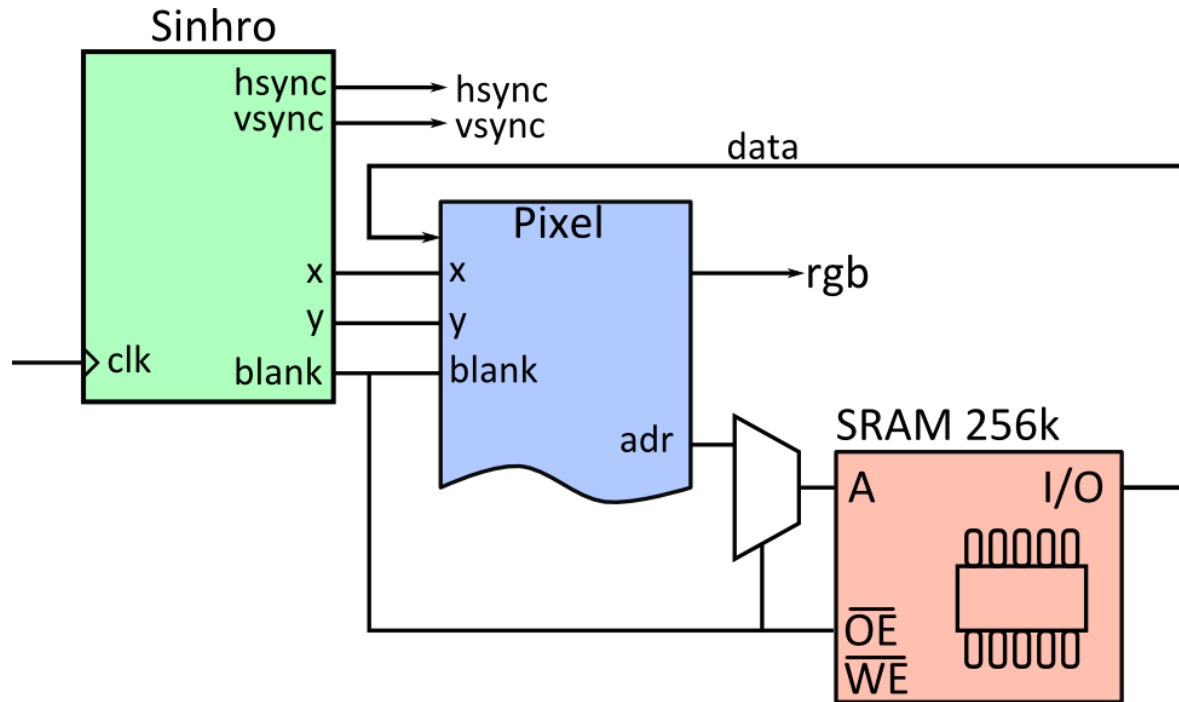
Arhitektura: spisek elementov

Qty	Value	Device	Package	Parts
1	FPGA	XC3S50A-VQ100	XILINX-VQ100	U1
1	SRAM256KX16	CY7C1041DV33	SOJ44	U2
1	74HC245DW	74HC245DW	SO20W	U3
1	W25X20BV	W25X10BV-SNIG	SOIC 8	U4
1	+3.3V Reg.	LM1117MP-3.3	SOT223	U5
1	+1.2V Reg.	LM1117MP-ADJ	SOT223	U6
1	25.000 MHz	SMDOSC	CFPS-73	OSC
2	green LED	LED0805	CHIP-LED0805	LED1, LED2
2	0.1uF	C-USC0603	C0603	C24, C26
15	0.1uF	C-USC0603	C0603	C1-3, C5, C7, C9, C10-13, C15, C18, C19, C21, C22
4	10nF	C-USC0603	C0603	C4, C16, C17, C20
1	1uF	C-EUC0805	C0805	C6, C23
2	10uF	C-USC1210	C1210	C25, C27
3	330E	R-EU_R0805	R0805	R14, R17, R19
1	390E	R-EU_R0805	R0805	R15
3	510E	R-EU_R0805	R0805	R1, R4, R7
5	1k	R-EU_R0805	R0805	R2, R5, R8, R10, R11
3	2k	R-EU_R0805	R0805	R3, R6, R9
5	4.7k	R-EU_R0805	R0805	R12, R13, R16, R18, R20

Arhitektura: poročilo

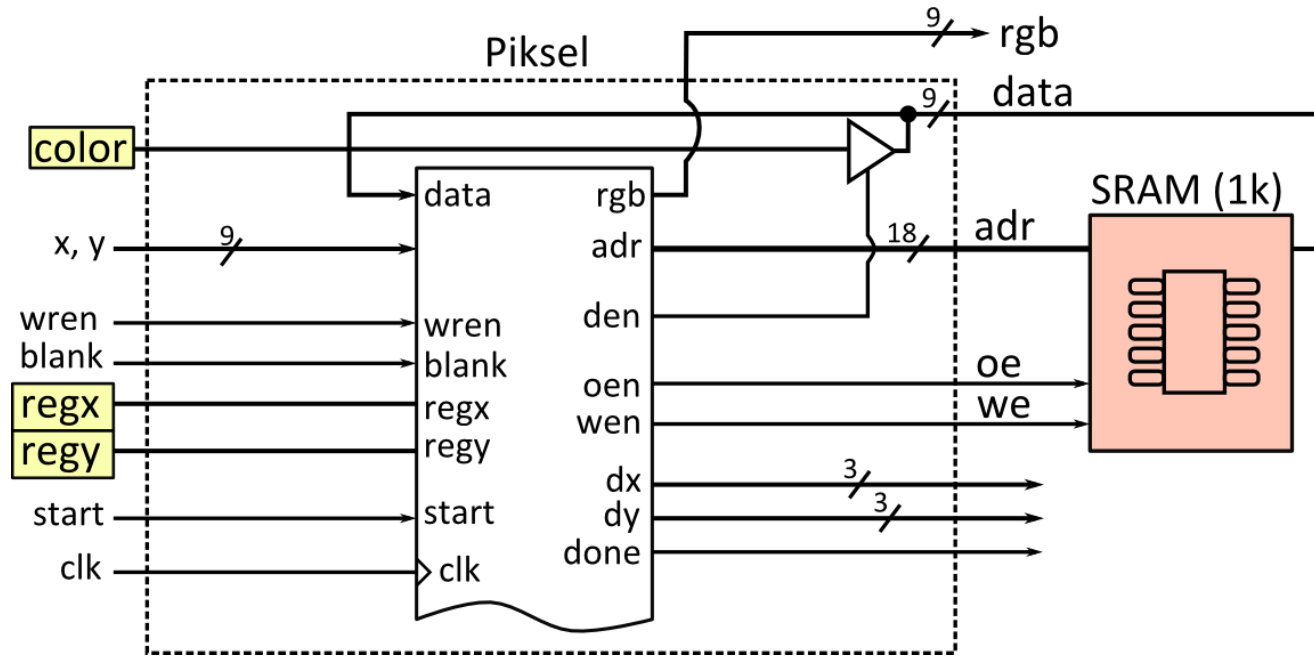
- ▶ Predstavi glavne komponente vezja
 - ▶ izseki iz sheme vezja
 - ▶ povezava na Arduino
 - ▶ programiranje vezja FPGA
- ▶ Predstavi napajalni del
 - ▶ logični nivoji, statični red
- ▶ Naredi izračun uporov za VGA in izpiši napetostne nivoje za eno barvno komponento (vse kombinacije bitov)
- ▶ Seznam uporabljenih elementov in možnosti za razširitev vezja
- ▶ Poročilo 3-5 strani oddajte do 7.6.2013

Logika: prikaz slike iz video pomnilnika



- ▶ Komponenta Pikel določa naslove in prenaša podatke na izhod VGA
- ▶ Komponenta Sinhro poskrbi za koordinate, blank in kontrolne signale za monitor

Logika: nastavitvev in prikaz slike iz pomnilnika



- ▶ Komponenti Piksel dodamo kontrolne vhode:
 - ▶ color – barva kvadrata, regx, regy – položaj prve točke
 - ▶ wren – omogoči pisanje v SRAM, blank – temni interval, start
- ▶ in izhode:
 - ▶ we – pisanje v pomnilnik, dx, dy – trenutni odmik
 - ▶ done – zaključek pisanja, pripravljen na nov start

Logika: vezje za risanje kvadrata

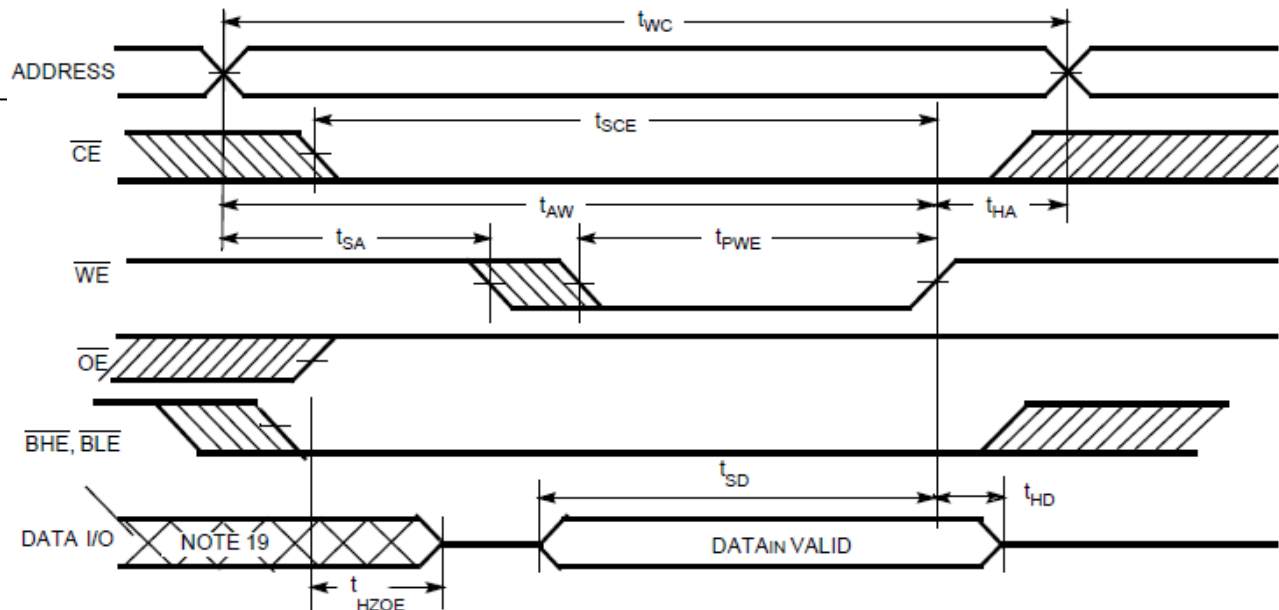
Algoritem

- ▶ Na ukaz start pobarvaj kvadratek 8x8

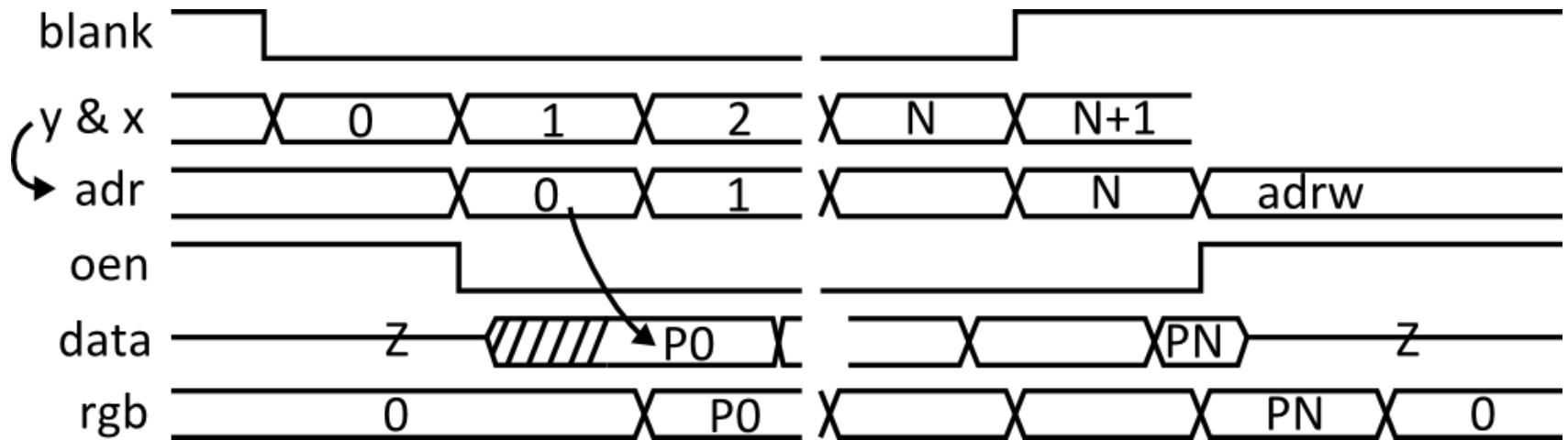
```
if (start=='1') {  
  for (dxy=0; dxy<64; dxy++) {  
    dx, dy = dxy  
    M [regy+dy] [regx+dx] = color  
  }  
  done = '1'  
}
```

Komponente & časovni diagrami

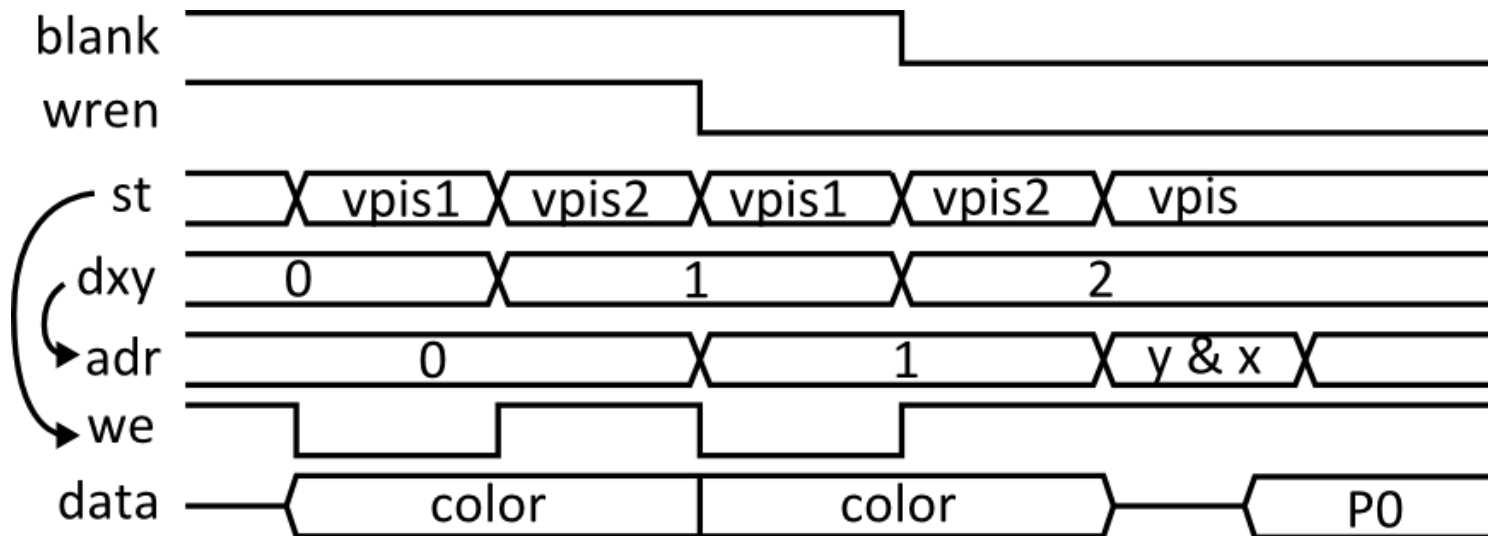
- ▶ Slika ozadja v asinhronem pomnilniku (SRAM)
- ▶ pisanje je možno, ko ne beremo (blank='1')
- ▶ upoštevamo časovne parametre pomnilnika



Logika: časovni potek branja in vpisa v SRAM



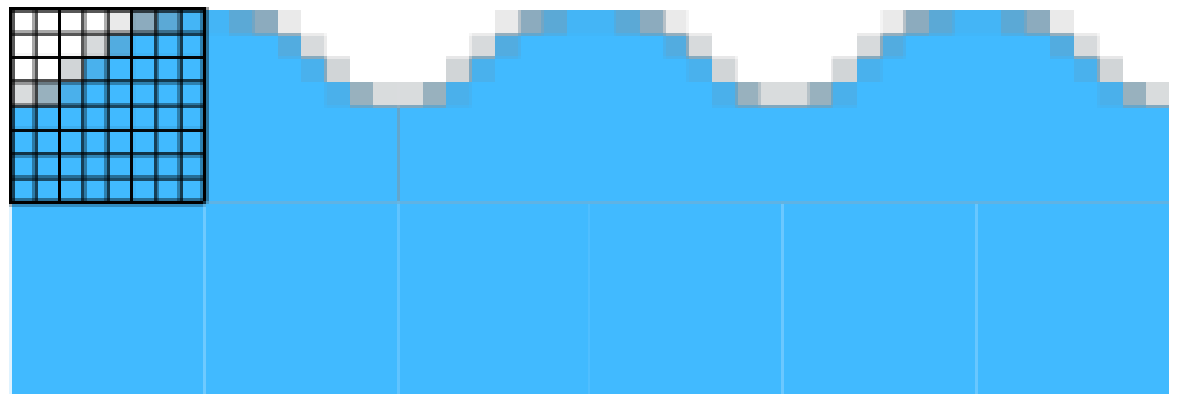
- ▶ Vpis podatka v 2 ciklih, zato potrebujemo stanja (st)



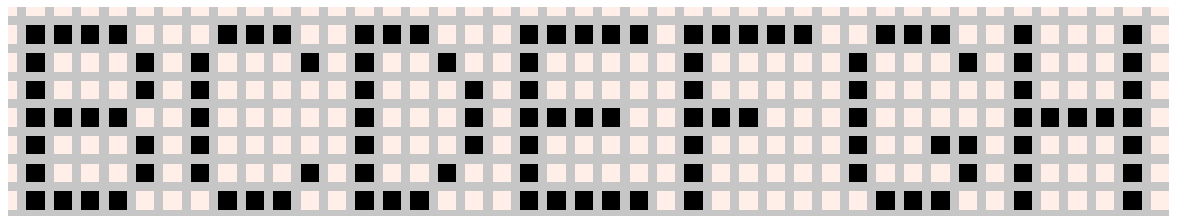
Princip risanja slikovne grafike

- ▶ Slika zasede 256k, procesor ATmega32 ima 32kB pomnilnika
- ▶ Grafika iz sličic (tile) velikosti 8x8 2-bitnih točk (4 barve)
 - ▶ strojna oprema začasno shrani 16 različnih sličic

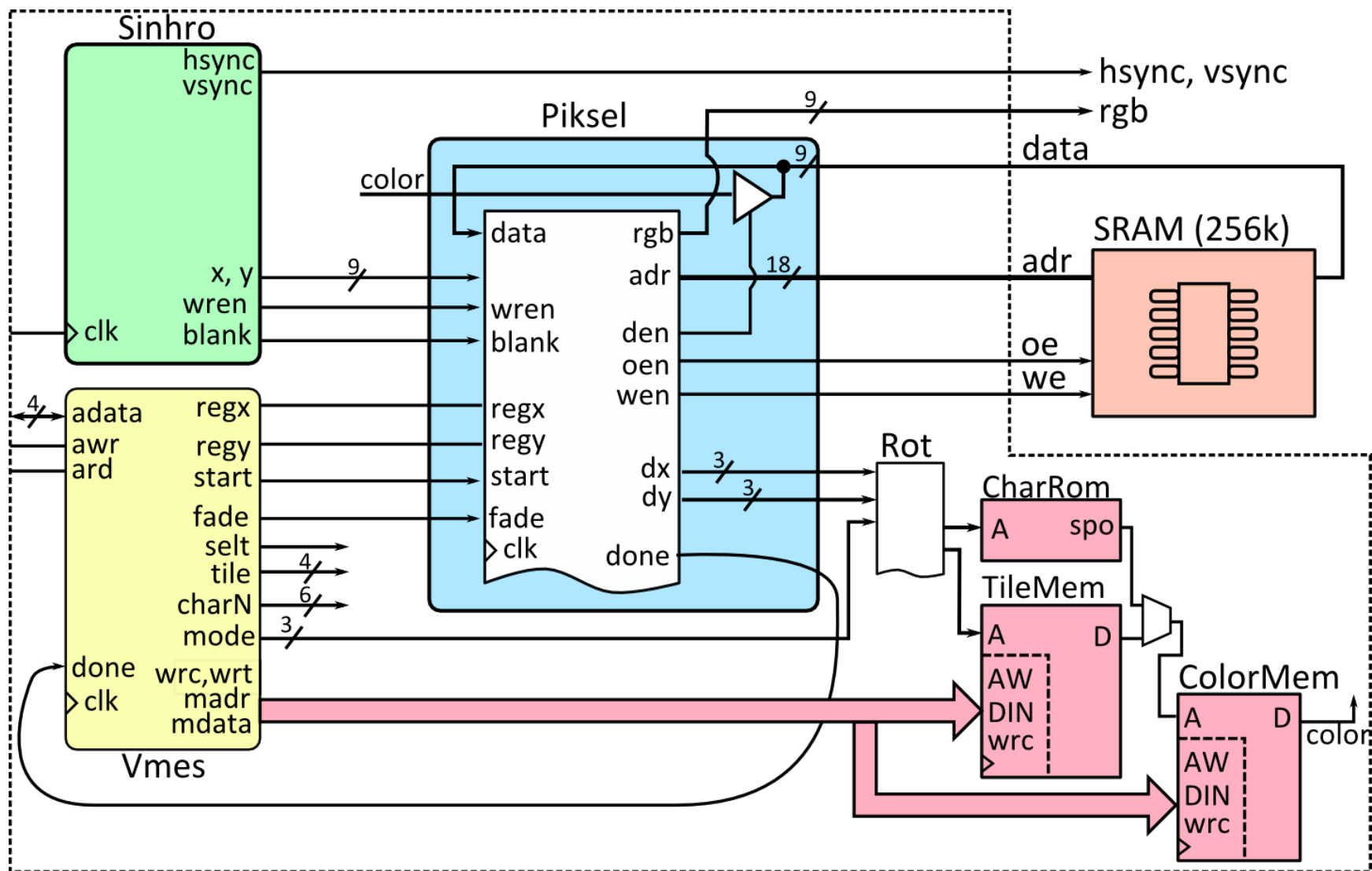
```
B0000000000011111,  
B0000000000111111,  
B0000000111111111,  
B0001111111111111,  
B1111111111111111,  
B1111111111111111,  
B1111111111111111,  
B1111111111111111
```



- ▶ ROM: 64 znakov
 - ▶ 8x8 črke, številke

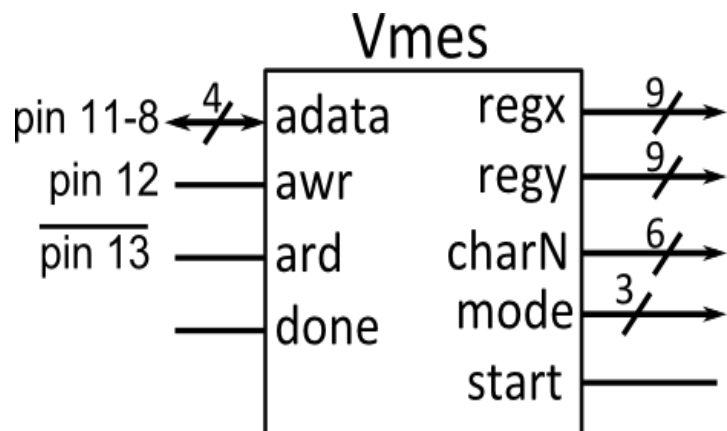


Logika: blokovna shema grafičnega vmesnika

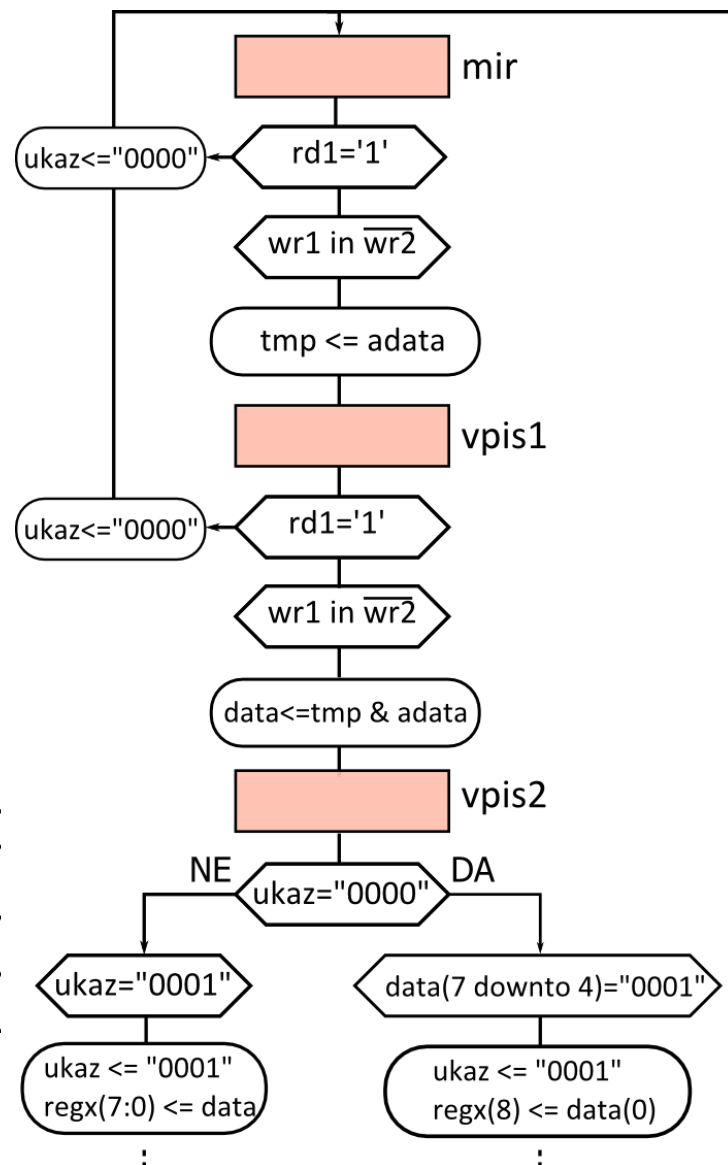
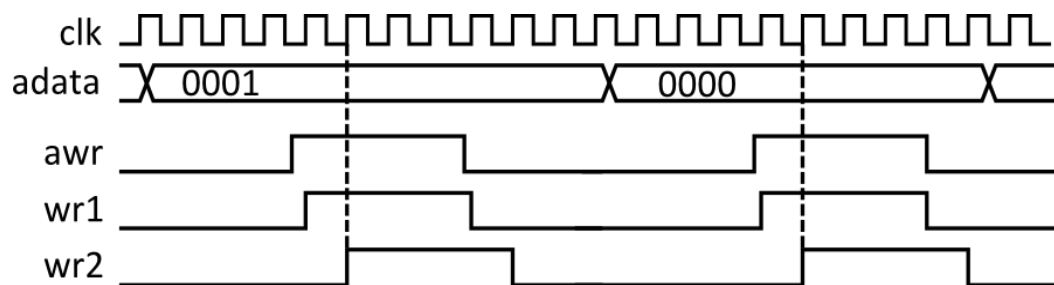


Logika: vmesnik

▶ Asinhroni 4-bitni vmesnik



▶ Sinhronizacija krmilnih signalov



Logika: poročilo

- ▶ **Razloži delovanje vmesnika VGA**
 - ▶ časovni potek signalov za prikaz slike na monitorju
 - ▶ sinhronizacijska komponenta
 - ▶ na kratko razloži vlogo pomnilnika in ostalih komponent vezja
- ▶ **Predstavi eno izmed glavnih komponent: Piksel ali Vmesnik**
 - ▶ blokovna shema in opis vhodnih in izhodnih signalov
 - ▶ časovni potek signalov – izsek iz simulacije
 - ▶ predstavi diagram stanj in primer (izsek) iz kode
 - ▶ naredi sintezo predstavljene komponente in zapiši v poročilo iz katerih logičnih gradnikov je sestavljena (registri, seštevalniki...)
- ▶ **Poročilo 3-5 strani oddajte do 12.6.2013**

C: inicializacija zunanjih vrat procesorja AVR

- ▶ Arduino: funkcija setup() za inicializacijo in loop() za kodo v zanki
- ▶ V setup() določimo smer in začetna stanja priključkov
 - ▶ naredimo reset pulz (RD=0, RD=1)

```
void setup() {  
  PORTD = PORTD & B10111111; // WR=0 (PD6)  
  DDRD = DDRD | B01000000;   // PD6: OUTPUT  
  PORTC = PORTC | B10000000; // RD=1 (PC7)  
  DDRC = DDRC | B10000000;   // PC7: OUTPUT  
  
  PORTC = PORTC & B01111111; // DIR = 0, RD pulz za resetiranje vmesnika  
  PORTC = PORTC | B10000000; // DIR = 1  
  PORTB = B00000000;         // ADATA=0  
  DDRB = DDRB | B11110000;   // PB(7:4): OUTPUT  
  
  GrafPuts("Digitalni elektronski sistemi 2013"); // primer klica funkcije za izpis  
}
```

C: osnovni ukazi grafičnega vmesnika

- ▶ 4-bitna komunikacija, impulz na WR za vpis v grafični vmesnik
- ▶ Osnovni ukazi: nastavi koordinati regx, regy in nariši znak 8x8

ime ukaza		koda ukaza		podatek
SetRegX	0 0 0 1	- - - r(8)	r(7) r(6) r(5) r(4)	r(3) r(2) r(1) r(0)
SetRegY	0 0 1 0	- - - r(8)	r(7) r(6) r(5) r(4)	r(3) r(2) r(1) r(0)
DrawChar	1 0 0 0	- m(2) m(1) m(0)	- - c(5) c(4)	c(3) c(2) c(1) c(0)

- ▶ Primer: SetRegX
 1. PORTB = B00010000; in naredi impulz na WR
 2. nastavi najvišji bit koordinate X na PORTB in impulz WR
if (X & 0x0100) PORTB = B00010000; else PORTB = 0;
 3. nastavi naslednje 4 bite koordinate in impulz WR
 4. nastavi spodnje 4 bite koordinate in impulz WR

C: funkcija za prikaz znakov

ime ukaza	koda ukaza	podatek
DrawChar	1 0 0 0 - m(2) m(1) m(0)	- - c(5) c(4) c(3) c(2) c(1) c(0)

```
#define AWR PORTD = PORTD | B01000000; PORTD = PORTD & B10111111
```

```
// Prikaži znak, m = 1 za normalno orientacijo, znak = koda ASCII – 32
```

```
void DrawChar(char c, byte m)
```

```
{
```

```
char znak = c-32;
```

```
PORTB = B10000000;
```

```
AWR;
```

```
PORTB = m << 4;
```

```
AWR;
```

```
PORTB = znak;
```

```
AWR;
```

```
PORTB = znak << 4;
```

```
AWR;
```

```
delayMicroseconds(40); // čakaj, da se operacija izvede
```

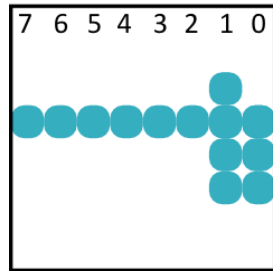
```
}
```

```
! " # $ % & ' ( ) * + , - . /  
0 1 2 3 4 5 6 7 8 9 : ; < = > ?  
@ A B C D E F G H I J K L M N O  
P Q R S T U V W X Y Z [ \ ] ^ _
```

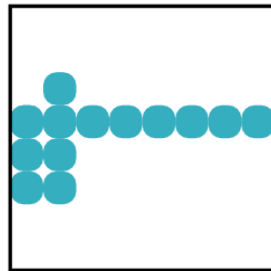
Zrcaljenje narisanih likov

- ▶ Zrcaljenje in/ali rotacija znakov oz. sličic: $m(2) m(1) m(0)$
 - ▶ naslov sestavimo iz števcov dy in dx , z negacijo spremenimo vrstni red
 - ▶ SW nastavlja parameter: mode (m)

mode=001
adr= dy & \overline{dx}

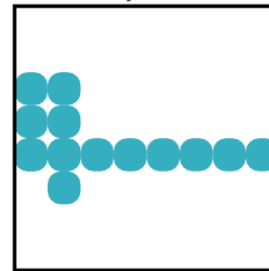


mode=000
adr= dy & dx



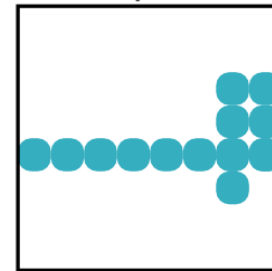
zrcaljenje X

mode=010
adr= \overline{dy} & dx



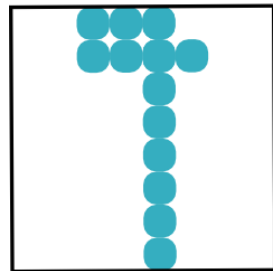
zrcaljenje X,Y
rotacija 180°

mode=011
adr= \overline{dy} & \overline{dx}

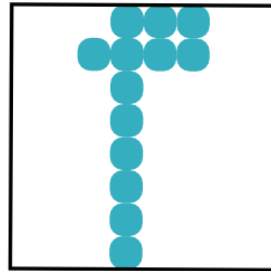


zrcaljenje Y

mode=101
adr= dx & \overline{dy}

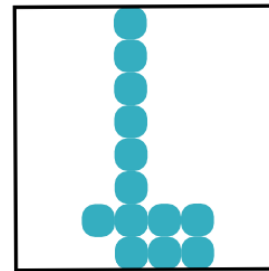


mode=100
adr= dx & dy

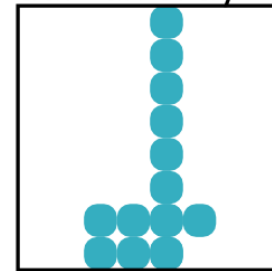


rotacija 90°

mode=110
adr= \overline{dx} & dy



mode=111
adr= \overline{dx} & \overline{dy}



rotacija 270°

C: ukazi za delo s sličicami in barvno paleto

ime ukaza	koda ukaza	Podatek
SetTile	0 1 0 0 t(3) t(2) t(1) t(0)	32 podatkovnih ciklov
SetColor	0 1 0 1 c3(8) c2(8) c1(8) c0(8)	8 podatkovnih ciklov
DrawTile	1 0 0 1 - m(2) m(1) m(0)	- - - - t(3) t(2) t(1) t(0)
Fade	1 0 1 0 - - - F	

- ▶ SetTile(t, *p) nastavi točke sličice, DrawTile(t, m) nariše sličico
 - ▶ t(3:0) določa indeks sličice, vezje FPGA lahko shrani 16 sličic
 - ▶ podatki (*p) so zaporedne vrednosti 8x8 2-bitnih točk
- ▶ SetColor: nastavi 4-bitno barvno paleto
 - ▶ za kodo ukaza pošljemo MSB vseh štirih barv, nato pa spodnjih 8 bitov barv: color0 (ozadje), color1 (barva znakov), color2 in color3
 - ▶ Kode nekaterih barv:
0x007, **0x02F**, **0x03F**, **0x038**, **0x1F8**, **0x100**, **0x1C7**, **0x000**, **0x1FF**
- ▶ Fade(f), zatemni sliko pri F=1 oz. odtemni pri F=0

C: poročilo

- ▶ Predstavi grafični vmesnik s stališča programerja
 - ▶ napiši in predstavi osnovne grafične funkcije
 - ▶ primer klica funkcije in rezultat na monitorju
- ▶ Naredi in predstavi preprosto aplikacijo, ki prikazuje animirano sliko
 - ▶ opiši podatkovne strukture in diagram poteka na nivoju glavnih funkcij
 - ▶ predstavi potek izdelave programa in
 - ▶ opiši možnosti za nadalnje delo
- ▶ Poročilo 3-5 strani oddajte do 17.6.2013, namesto poročila lahko naredite predstavitev 30.5. na predavanjih (10-12 prosojnic)