

## UGRADBENI RAČUNALNI SUSTAVI

### TREĆA DOMAĆA ZADAĆA

#### Zadatak 1 (1 bod)

Nabrojite osnovne porodice PLD sklopova, navedite njihove nazive?

##### Rješenje:

PROM (programmable ROM), PLA (programmable logic array), PAL (programmable array logic), GAL (generic array logic), CPLD, FPGA.

#### Zadatak 2 (1 bod)

Koja je najbitnija razlika između GAL i PAL sklopova?

##### Rješenje:

GAL – višestruko programiranje, PAL – samo jednom.

#### Zadatak 3 (2 boda)

Za funkcije:

$F(A,B,C) = (A \text{ xor } B \text{ xor } C)$  i

$G(A,B,C) = ((A \text{ and } B) \text{ or } (A \text{ and } C) \text{ or } (B \text{ and } C))$ ,

odrediti veličinu memorije i njezin cjelokupni sadržaj.

##### Rješenje:

Tablica:

A	B	C	F	G
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Veličina memorije je 8x2 bita.

#### Zadatak 4 (2 boda)

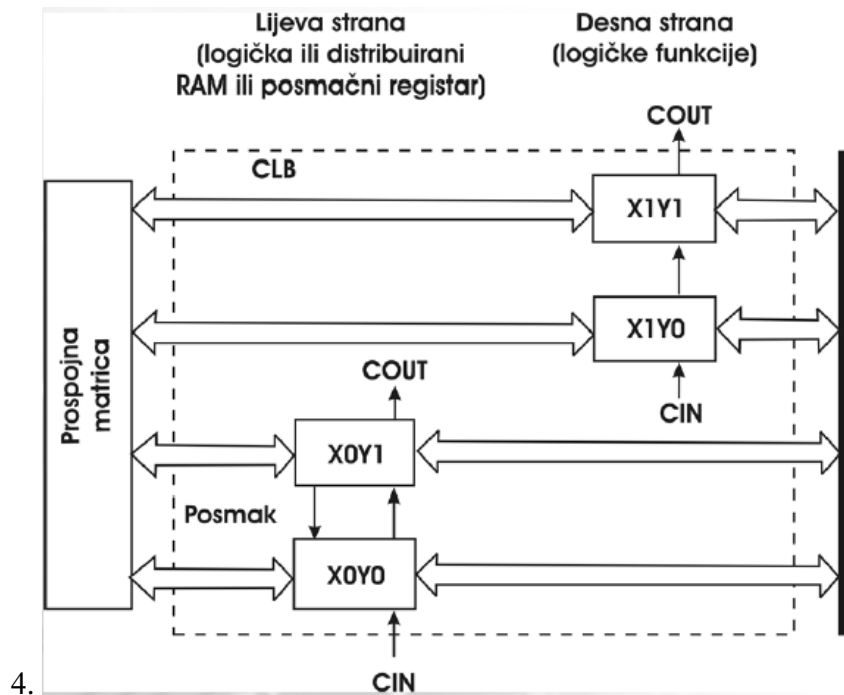
Objasnite i skicirati strukturu CLB blokova unutar Spartan 3 uređaja.

##### Rješenje:

Konfigurabilni logički blokovi (eng. Configurable Logic Blocks – CLB) predstavljaju glavni dio sklopa zadužen za izvedbu slijednih i kombinacijskih funkcija. Svaki CLB sastoji se od 4 međusobno povezana dijela kako je prikazano na slici koja slijedi. Dijelovi su grupirani u parove, a svaki par organiziran je u stupce sa nezavisnim lancem za prijenos (eng. Carry look ahead). Sva četiri dijela sadrže slijedeće elemente:

1. Dva generatora funkcijske logike
2. Dva memorijska elementa
3. Multipleksore

#### 4. Prijenosni lanca sa dodatnom logičkim sklopovima



Lijevi par podržava i distribuirani RAM i 16-bitni posmak.

#### Zadatak 5 (1 bod)

Navesti osnovne vrste poveznica u Spartan 3 arhitekturi.

#### Rješenje

Long lines, hex lines, double lines, direct connections.

#### Zadatak 6 (1 bod)

Što je entitet a što opis arhitekture u VHDL-u?

#### Rješenje

Entitet opisuje sučelje sklopa, a arhitektura njegovu funkcionalnost (ponašanje).

#### Zadatak 7 (1 bod)

U VHDLu definirati signal s tipa `std_logic` koji će nakon globalnog reseta imati vrijednost '1'.

#### Rješenje:

```
Signal s : std_logic := '1';
```

#### Zadatak 8 (1 bod)

Koja je razlika između kombinajske i procesne logike i na koje se dvije vrste logike može podijeliti procesna logika?

#### Rješenje:

Kombinacijska logika – jednoznačna pridruživanja, redosljed nije bitan. Korištenje procesne logike razlikuje se od kombinacijske logike po tome što se pridruživanje radi određenim redosljedom. Zadnje pridružena vrijednost unutar procesne logike poništava prethodnu pridruženu vrijednost. Procesna logika može biti kombinacijska i slijedna.

**Zadatak 9 (1 bod)**

Objasnite memorijski sustav Picoblaze procesora (programskog i podatkovnog)? S obzirom na memoriju u koju aritekturu pripada?

**Rješenje:**

Sastoji se od 2 odvojene memorije: Programska memorija (1024 18-bitne memorijske lokacije sa kojih se samo čita), Scratch-pad memorija od 64 8 bitne memorijskih lokacija za spremanje varijabli, i 16 8-bitnih registara. Harvard arhitektura.

**Zadatak 10 (2 boda)**

Korištenjem kombinacijske logike napišite VHDL kod (entitet i arhitekturu) za izvedbu multipleksora 8 na 1, pri čemu je defaultna vrijednost na izlazu ona sa ulaza(0)?

**Rješenje:**

```
Entity bhv is
Port(in_data : in std_logic_vector(7 downto 0);
      Selector : in std_logic_vector(2 downto 0);
      Out_data : out std_logic_vector
        );
End entity bhv;

architecture bhv of select__bhv is
begin
with selector select
T <= In_data(1) when "001",
      In_data(2) when "010",
      ...
      In_data(0) when others;

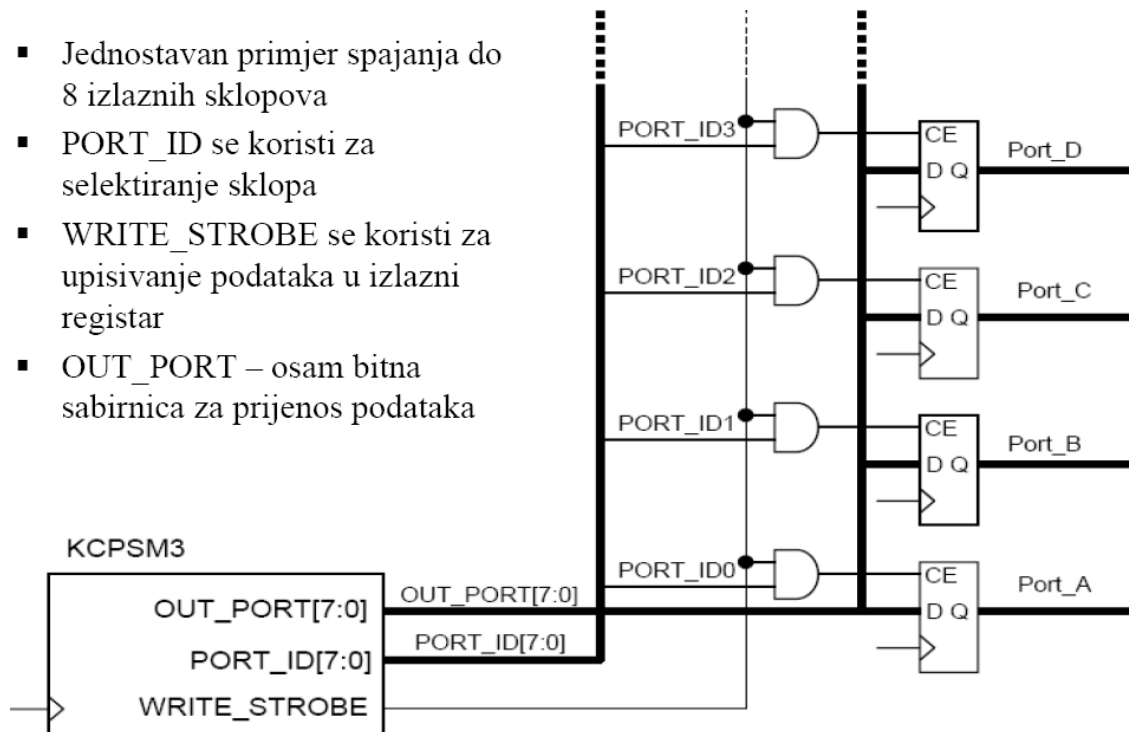
end bhv;
```

**Zadatak 11 (2 boda)**

Napišite dio VHDL koda (potrebno je napisati samo proces) koji služi za spajanje izlaznih vanjskih jedinica na PicoBlaze? Objasnite kako radi? Napisati primjer korištenja vanjske jedinice u assembleru. Nacrtati shematski spajanje na KCPSM3 procesor.

**Rješenje:**

- Jednostavan primjer spajanja do 8 izlaznih sklopova
- PORT\_ID se koristi za selektiranje sklopa
- WRITE\_STROBE se koristi za upisivanje podataka u izlazni registar
- OUT\_PORT – osam bitna sabirnica za prijenos podataka



```
output_ports: process(clk)
begin
    if clk'event and clk='1' then
        if write_strobe='1' then
            -- Write to LEDs at address 80 hex.
            if port_id(7)='1' then
                led <= out_port;
            end if;
        end if;
    end if;
end process output_ports;
```

```
CONSTANT LED_PORT, 40
...
LOAD    s0, 08
OUTPUT s0, LED_PORT
```

**Zadatak 12 (1 bod)**

Objasnite koje je razlika između Diskretnih, Hard i Soft Core procesora? Koje su prednosti i nedostaci svakih od njih?

**Rješenje:****Diskretni procesori**

- Diskretni ili “off the shelf”
- Tradicionalan pristup
- Veliki izbor različitih proizvođača
- Veliki izbor različite funkcionalnosti
- Izvedeni kako ASIC sklopovi koji osim samoga procesora uglavnom sadrže i neke periferije
- Ispravan odabir odgovarajućeg procesora može biti vrlo zahtjevan zadatak

**Hard Processor Core**

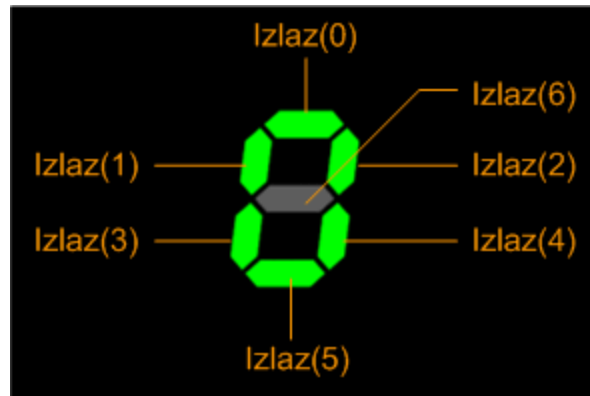
- Posebni namjenski sklopovi unutar komponente (uglavnom FPGA)
- S obzirom na izvedbu mogu raditi na velikim brzinama
- Prednost im je što postoje u okruženju gdje se njegova okolina može jednostavno prilagoditi potrebama dizajna.
- Ne omogućavaju prilagodbu procesora dizajnu
- Samo pojedini članovi porodica FPGA sklopova imaju takvu mogućnost
- Vrlo ograničen broj proizvođača i FPGA sklopova

**Soft Processor Core**

- SoftCore procesori
- U potpunosti izveden u logici
- Zbog toga ne može raditi na maksimalnim brzinama samog sklopa
- Uglavnom višestruko sporiji od diskretnih i “hard” procesora
- Uglavnom pogodni za sustave gdje brzina izvođenja algoritma nije presudna
- Kontrola GPIO
- Iako nije ograničena na samo takvu primjenu.

**Zadatak 13 (2 boda)**

Isprojektirati u VHDLu komponentu (entitet i arhitekturu) za 7-segmentni pokaznik korištenjem kombinacijske procesne logike. Ulaz je signal tipa int koji prima vrijednosti od 0 do 9 a izlaz je std\_logic\_vector veličine 7 koji predstavlja kod za 7-segmentni pokaznik kako je prikazano na slici:



### Rješenje:

```
Entity SevenSegDecoder is
Port(in_data : in integer;
      Out_data: out std_logic_vector(6 downto 0)
);
End entity SevenSegDecoder;
```

```
architecture Beh of SevenSegDecoder is
begin
```

```
    process(in_data)
    begin
        case in_data is
            when 1 =>
                out_data <= "0010100";
            when 2 =>
                out_data <= "1011011";
            when 3 =>
                out_data <= "1010111";
            when 4 =>
                out_data <= "0100101";
            when 5 =>
                out_data <= "1100111";
            when 6 =>
                out_data <= "1101111";
            when 7 =>
                out_data <= "1010100";
            when others =>
                out_data <= "0000000";
        end case;
    end process;
end Beh;
```

### Zadatak 14 (2 boda)

Za PicoBlaze procesor napisati sljedeću prekidni potprogram: Na svaki zahtjev za prekid potrebno je očitati stanje vanjske jedinice koja je spojena na IN\_PORT. Na dobivenom

podatku ispituje se parnost: Ukoliko je on paran, potrebno je povećati brojač parnih podataka u ScratchPad memoriji na adresi hex(10), a ukoliko je neparan, povećati brojač neparnih podataka koji je u ScratchPad memoriji na adresi hex(11).

**Rješenje:**

```
ADDRESS 000
CONSTANT IN_PORT, 00

; brojac parnih i neparnih
LOAD s1, 00
LOAD s2, 00

PREKID
; učitaj podatak
INPUT s0, IN_PORT
;ispitaj parnost
AND s0, 01
COMPARE s0, 00
JUMP Z, PARAN
NEPARAN
ADD s1, 01
STORE s1, (10)
RETURNI
PARAN
ADD s2, 01
STORE s2, (11)
RETURNI

; prekidna adresa
ADDRESS 3FF
JUMP PREKID
```