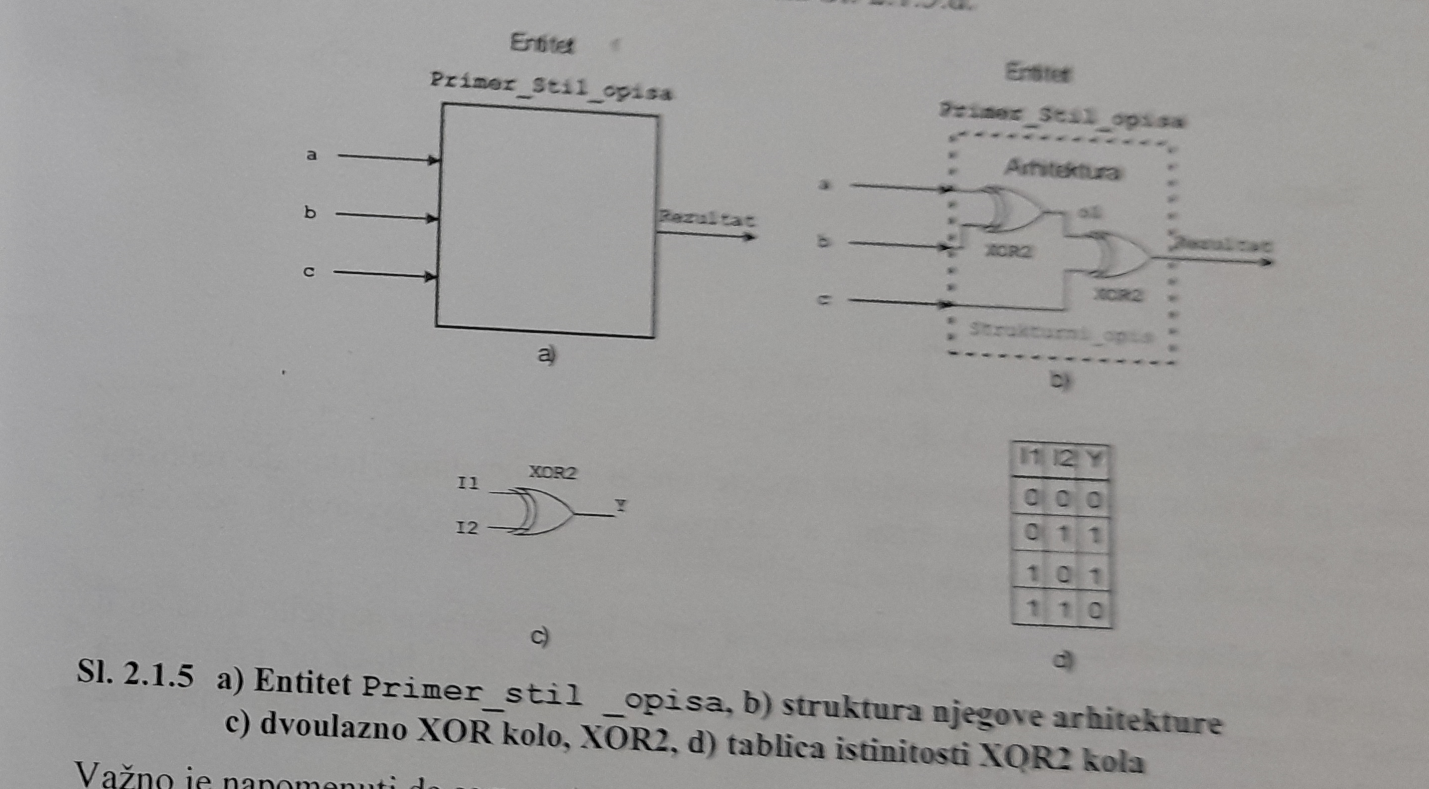
**Računske vežbe PPR – Termin 4**

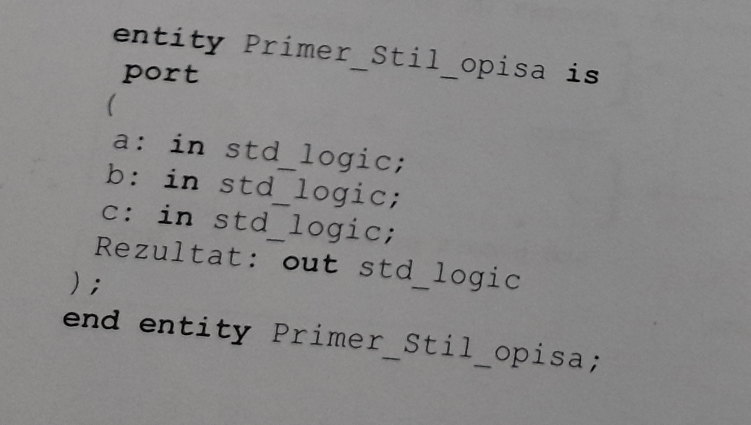
Stilovi opisa u VHDL-u:

1. Strukturni opis (structural)
2. Opis toka podataka (dataflow)
3. Opis ponašanja (behavioral)

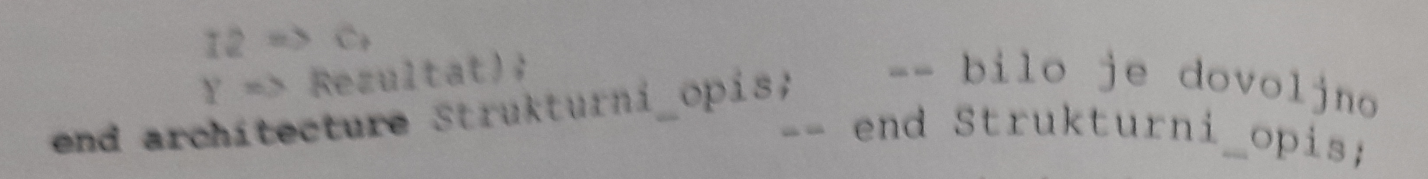
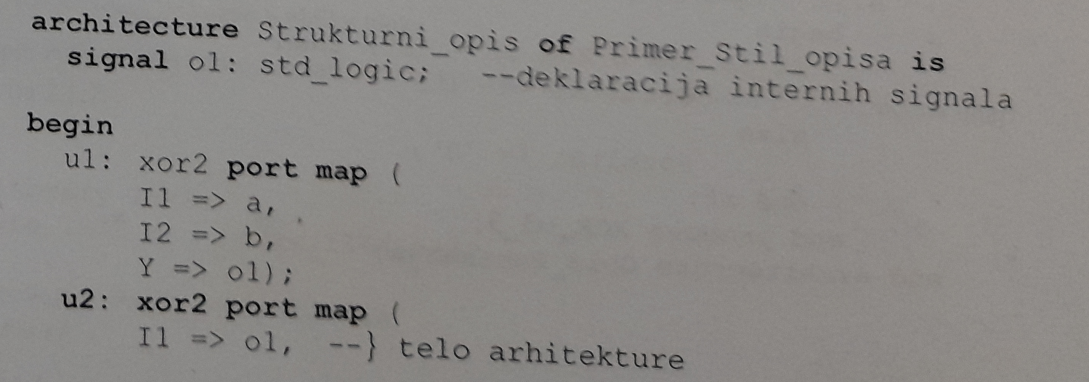
STRUKTURNI OPIS

Ako je poznata struktura nekog kola koje projektujemo koje je dato na slici 2. 1.5, a strukturni opis arhitekture je prikazan u nastavku.

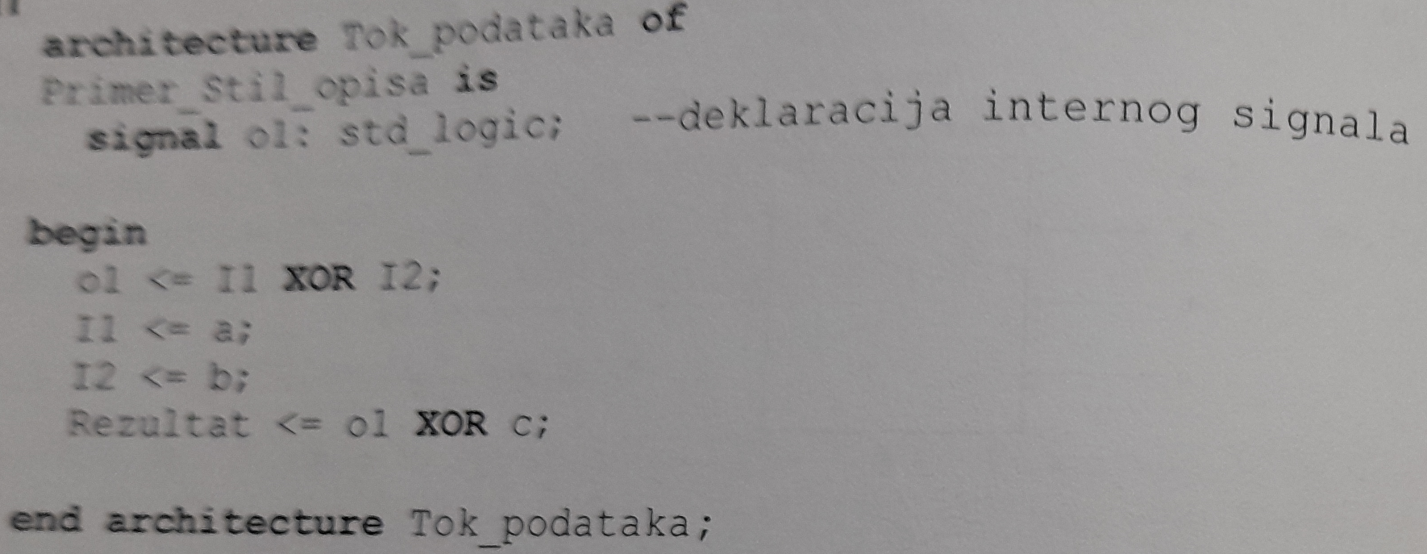




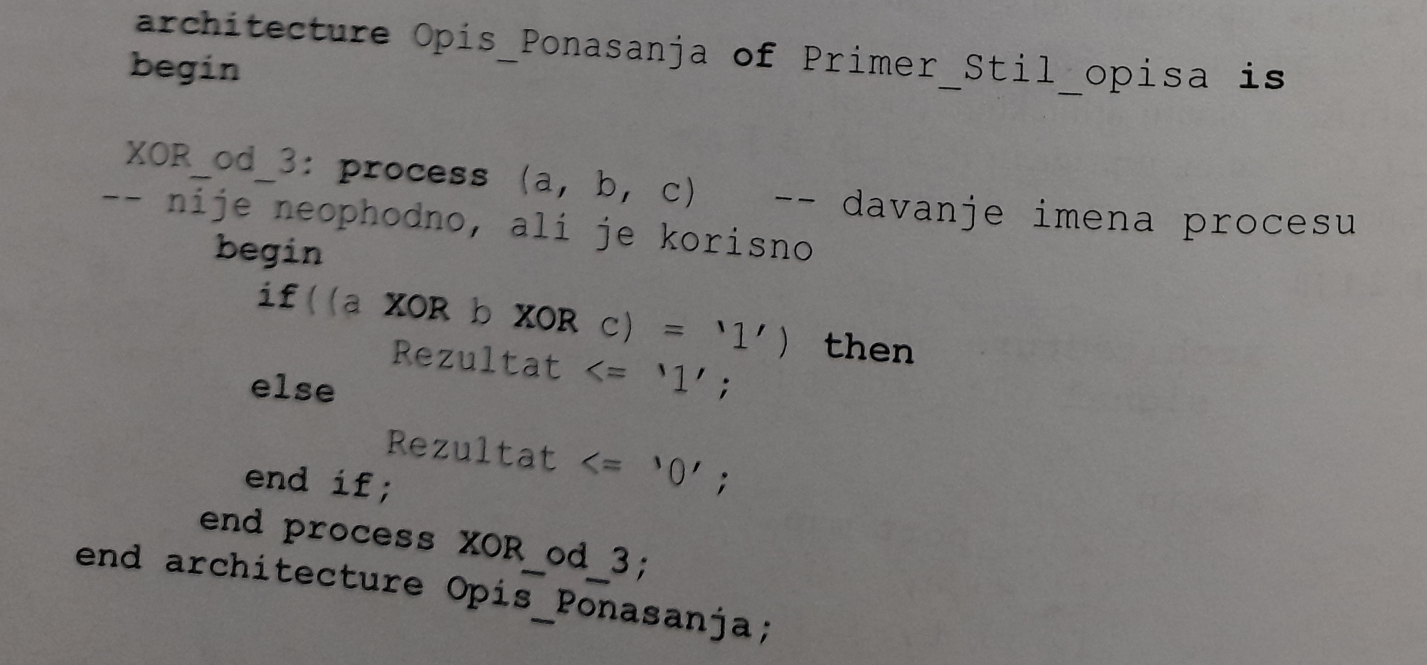
Strukturni opis arhitekture:



Opis toka podataka:

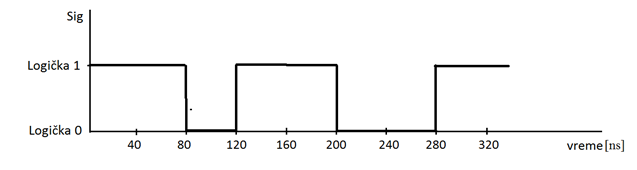


Opis ponašanja:



Zadavanje pobude u testbench- u

1. Za signal čiji je identifikator ***Sig*** koji je dat na slici, napisati konkurentnu specifikaciju, a zatim takođe i sekvencijalnu specifikaciju.



REŠENJE:

Konkurentna specifikacija:

Sig <=’1’,’0’ after 80ns,’1’ after 120ns, ‘0’ after 200ns, ‘1’ after 280ns;

Sekvencijalna specifikacija:

architecture ime\_arh of ime\_ent is

constant Kašnjenje: time := 80ns;

begin

Seq: process

begin

Sig<=’1’; wait for Kašnjenje;

Sig<=’0’; wait for Kašnjenje/2;

Sig<=’1’; wait for Kašnjenje;

Sig<=’0’; wait for Kašnjenje;

Sig<=’1’; wait for Kašnjenje;

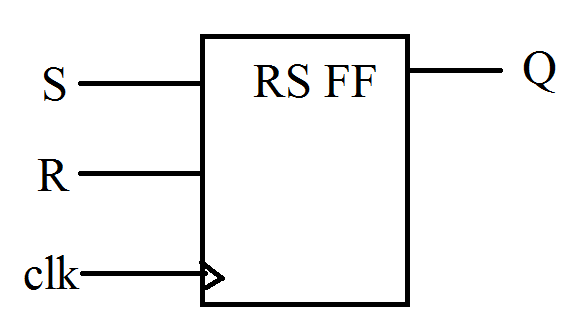
end process seq;

end erchitecture;

UKOLIKO UNUTAR PROCESA POSTOJI WAIT NAREDBA, NE TREBA DA NAPIŠEMO LISTU SIGNAL NA KOJE JE PROCES OSETLJIV.

Može da bude: wait for (vreme) ...., wait on (signal)..., wait until (Bulov iskaz true/false)...

**Zadatak 1**. Napisati entitet i arhitekturu za RS Flip-flop koji je dat i opisan tablicom istinitosti na slici 1.



|  |  |  |  |
| --- | --- | --- | --- |
| S | R | clk | Q |
| 0 | 0 | X | Q'(PRETHODNO STANJE) |
| 0 | 1 | \_↑¯ | 0 |
| 1 | 0 | \_↑¯ | 1 |
| 1 | 1 | X | U |

Slika 1. RSFF

*Pojašnjenje:* U datom slučaju S predstavlja SET, R je RESET, clk je TAKT signal i Q je izlaz. U slučaju da na ulaz S i R dovedemo nizak naponski nivo (logička 0), bez obzira na takt signal, na izlazu RF flip-flopa ostaje ono što je bilo, tj. prethodno stanje koje označavamo sa Q'. Ukoliko nastupi rastuća ivica takt signala, S=0,R=1, vrši se resetovanje flip flopa, odnostno izlaz se postavlja u Q=0, a ko je S=1, R=0, vrši se setovanje flip flopa (izlazu Q se dodeljuje logička jedinica). Ukoliko su i S=1 i R=1, bez obzira na takt signal, izlaz će biti nedefinisano stanje.

library IEEE;

use IEEE.STD\_LOGIC\_1164.all;

**entity** RSFF **is**

**port**(

R: **in** STD\_LOGIC;

S: **in** STD\_LOGIC;

clk: **in** STD\_LOGIC;

Q: **out** STD\_LOGIC);

**end entity** RSFF;

**architecture** RSFF\_ponasanje **of** RSFF **is**

**begin**

rsff: **process**

**begin**

**wait until** clk=’1’; --čekaj da clk promeni stanje na 1(HIGH) i kreni da izvršavaš proces

--tako znamo da je nastupila rastuća ivica clk signala

**if** S=’0’ **and** R=’1’ **then** Q<=’0’; -- ukoliko je set=0, reset=1, Q postavi na 0 (resetuj FF)

**elsif** S=’1’ **and** R=’0’ **then** Q<=’1’; -- ako je set=1, reset=0, setuj izlaz Q=1

**end if**;

**end process** rsff;

**end architecture** RSFF\_ponasanje;