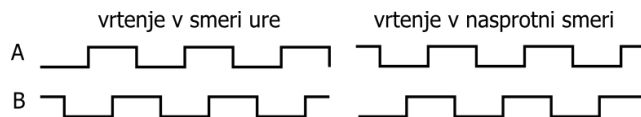


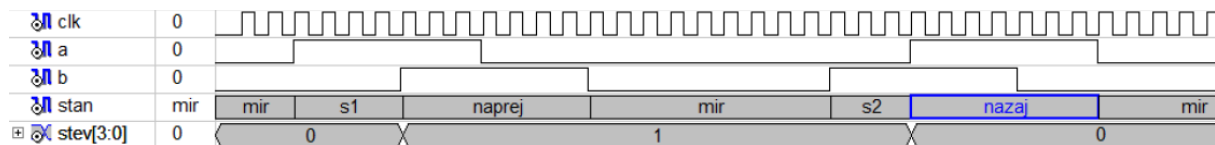
### 3. vaja: Rotacijski kodirnik in diagram stanj

Pri tej vaji bomo uporabili vezje z rotacijskim inkrementalnim kodirnikom, ki ob vrtenju proizvaja med seboj zamaknjene impulze na izhodih A in B. Digitalno vezje bomo sestavili iz procesa, ki dekodira smer vrtenja kodirnika in prišteva oz. odšteva stanje števca ter dekodirnika za 7-segmentni prikazovalnik.



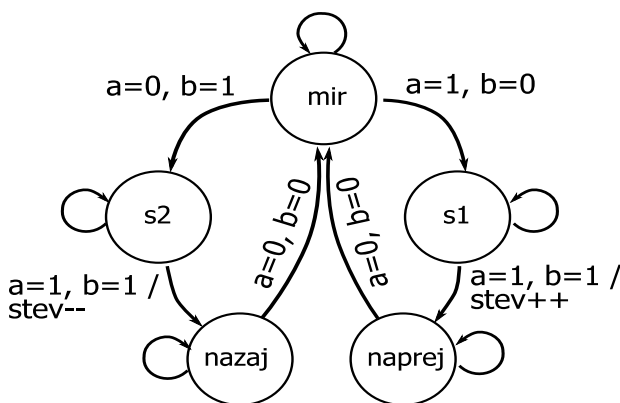
#### 3.1 Princip delovanja

Dekodiranje smeri vrtenja bomo naredili s sinhronim avtomatom. Vezje naj bo na začetku v stanju mir, dokler se ne postavi eden izmed vhodov a ali b na '1'. Če se najprej postavi a na '1', potem pa še b na '1' naj se vrednost števca poveča, pri obratnem vrstnem redu pa naj se vrednost števca zmanjša.



#### 3.2 Diagram stanj in VHDL opis

Sinhroni avtomat spreminja stanje ob spremembi vhodov in ob naraščajoči fronti ure. Vhodni signali se spreminjajo počasneje kot ura, zato bo avtomat več ciklov v posameznem stanju.



**NAŠTEVNI TIP PODATKOV**

Deklaracija:  
**type** stanja **is** (mir, s1, s2, naprej, nazaj);  
**signal** stan: stanja;

Prireditveni stavek:  
 stan <= mir;

Za zapis stanj avtomata v jeziku VHDL definiramo nov podatkovni tip, v katerem naštejemo vsa možna stanja. Avtomat opišemo s sinhronim sekvenčnim procesom, kjer definiramo v katero stanje naj gre glede na trenutno stanje in vhodne signale.

V procesu določimo za vsako stanje pogoje za prehod v novo stanje; ker moramo pogoje zapisati za vsa stanja uporabimo kar izbirni stavek (case):

```
if rising_edge(clk) then  
  case stan is  
    when mir =>  
      if a='1' and b='0' then  
        stan <= s1;  
      elsif ... end if;  
  
    when s1 =>  
      if a='1' and b='1' then  
        stan <= naprej;  
        stev <= stev + 1;  
      end if;  
    ...
```

Spreminjanje števca zapišemo kar v istem procesu, saj je vezano na pogoje za prehod med stanji. V vezju dodaj še dekodirnik za 7-segmentni prikazovalnik:

```
with stev select  
  led <= "1111001" when "0001",  
        "0100100" when "0010",  
        "0110000" when "0011",  
        "0011001" when "0100",  
        "0010010" when "0101",  
        "0000010" when "0110",  
        "1111000" when "0111",  
        "0000000" when "1000",  
        "0010000" when "1001",  
        "1000000" when others;
```

Preizkusi delovanje vezja s simulacijo in na razvojnem sistemu. Pred implementacijo vezja določi lokacije priključkov:

```
NET "a1" LOC = "C12";  
NET "b1" LOC = "A13";  
NET "clk" LOC = "M6";  
NET "led<0>" LOC = "L14";  
NET "led<1>" LOC = "H12";  
NET "led<2>" LOC = "N14";  
NET "led<3>" LOC = "N11";  
NET "led<4>" LOC = "P12";  
NET "led<5>" LOC = "L13";  
NET "led<6>" LOC = "M12";  
NET "an" LOC = "F12";
```