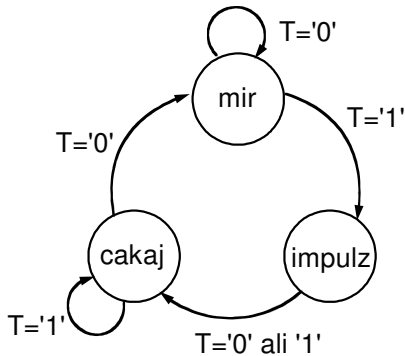


## 6. Vaja: sekvenčno vezje z registrom stanj (avtomat)

## 1. Vezje za detekcijo pritiska tipke



Vezje ima en vhodni signal (tipka), prehodi med stanji pa naj se vršijo ob prvi fronti zunanje ure.

Delovanje vezja predstavimo z diagramom stanj. Ob pritisku na tipko, gremo iz mirovnega stanja v stanje impulz, ob naslednjem urinem ciklu pa v stanje čakaj. V stanju čakaj bo vezje tako dolgo, dokler ni signal tipke na '0'. Vezje v stanju impulz generira izhodni signal, ki je dolg natanko eno urino periodo.

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

```

architecture opis of avtomat is
  type tip_stanja is (mir, impulz, cakaj);
  signal stanje: tip_stanja;
  
```

**begin**

```

  Avt: process(clk)           -- določitev prehajanja stanj
  begin
  
```

```

    if rising_edge(clk) then
      case stanje is
        when mir =>
          if t='1' then
            stanje <= impulz;
          end if;
        
```

```

        when impulz =>           -- prehod se izvrši ob naslednji fronti
          stanje <= cakaj;
        
```

```

        when others =>         -- namesto when cakaj =>
          if t='0' then
            stanje <= mir;
          end if;
        end case;
      end if;
    
```

```

  end process;
  
```

```

  izhod <= '1' when stanje=impulz else '0'; -- izhodna logika
  
```

**end opis**;

- a) Naredi opis in simulacijo vezja za detekcijo pritiska tipke.

### NAŠTEVNI TIP PODATKOV

Deklaracija:

```

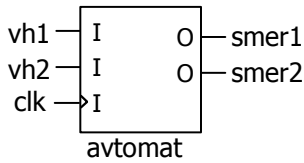
type ime_tipa is (ime1, ime2, ime3...);
signal ime_signala: ime_tipa;
  
```

Prireditveni stavek:

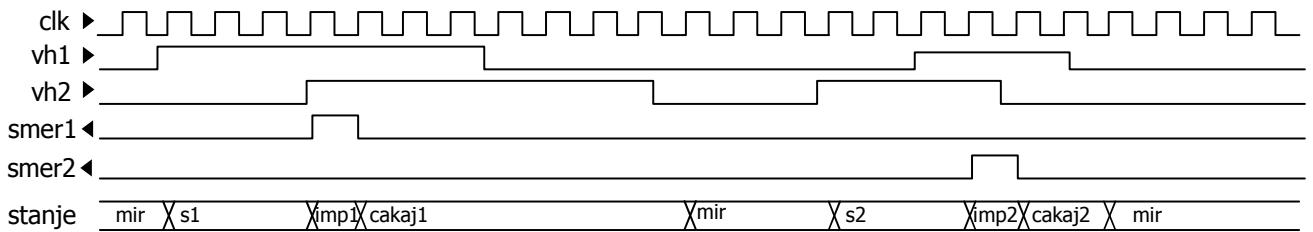
```

  ime_signala <= ime1;
  
```

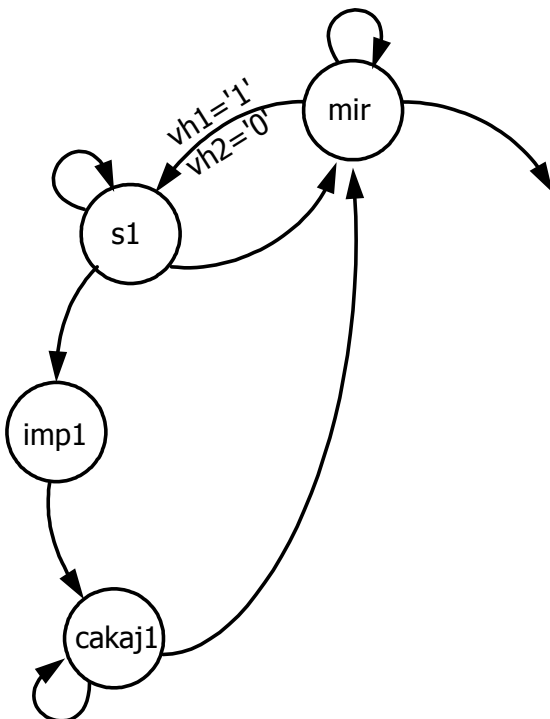
## 2. Avtomat za detekcijo smeri



Načrtaj sekvenčno vezje, ki spremlja dva vhodna signala (**vh1** in **vh2**). Mirovno stanje obeh vhodov je '0'. Če se najprej spremeni **vh1** na '1' in nato **vh2** na '1', naj se postavi izhodni signal **smer1** na '1', če se vhoda spreminjata v obratnem vrstnem redu pa naj aktivira izhod **smer2**:



- a) Dopolnite diagram prehajanja stanj z vsemi potrebnimi stanji za detekcijo smeri in pogoji za prehode med stanji. Pri pogojih upoštevate tudi možnosti nepravilnih vhodnih kombinacij, pri katerih se izhodi ne smejo postaviti (npr. oba vhoda se naenkrat postavita na '1' ali pa en vhod se postavi na '1' in takoj nato na '0'...).



Deklariramo nov podatkovni tip:

```
type tip_stanja is (mir, s1, imp1...);
signal stanje: tip_stanja;
```

in opišemo prehode stanj:

```
if rising_edge(clk) then
  case stanje is
    when mir =>
      if vh1='1' and vh2='0' then
        stanje <= s1;
      elsif ... end if;
    when s1 =>
      if vh1='1' and vh2='1' then
        ...
```

- b) Naredi opis vezja v jeziku VHDL in simulacijo!