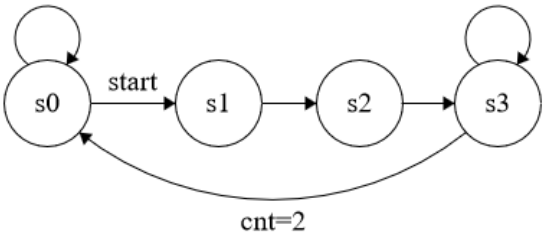
5. vaja: Sekvenčni stroj

Končni stroj stanj (avtomat)

Naredi model vezja, ki ob pritisku tipke pošlje na izhod Morsejev znak A: en kratek in en dolg impulz.



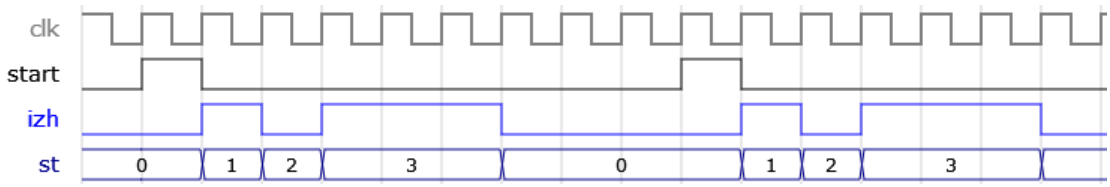
Deklaracija stanj: **SHDL**

```
st: (s0,s1,s2,s3);
```

VHDL

```
type stanja is
 (s0,s1,s2,s3);
signal st:stanja;
```

1. Vezje ima enobitni vhod (**start**) in enobitni izhod, med notranjimi signali pa 2-bitni števec in signal naštevnega podatkovnega tipa, ki določa stanje vezja (**st**).
2. Preglej diagram stanj in zapiši pogoje za prehajanje stanja. Stanje vezja naj se spremeni ob uri in dodatnemu pogoju (npr. start=1, cnt=2). Števec cnt je v vezju zato, da počakamo v stanju s3 tri cikle ure. V tem stanju naj se števec povečuje, ko pride do 2 pa se postavi nazaj na 0.
3. Dodaj še stavek za določanje izhoda: izhod naj bo 1 v stanju s1 in s3, sicer pa 0. Preizkusi delovanje s simulacijo.



Mikrosekvenčnik

Mikrosekvenčnik je sekvenčno vezje pri katerem je zaporedje izhodov zapisano v pomnilniku ROM.

1. Dodaj v vezje iz prejšnje vaje pomnilnik ROM v katerem naj bo zaporedje kode B (— . . .)

   SHDL:
   ```
   rom: 16u2=1,1,1,0,1,0,1,0,1,2;
   ```

   VHDL:
   ```
   type memory is array (0 to 15) of unsigned(1 downto 0);
   signal ROM: memory := ("01","01","01","00","01","00","01","00","01",
   "00","10", "00","00","00","00","00");
   ```

   Pomnilnik smo deklarirali kot 16-bitno zbirko 2-bitnih vrednosti: najnižji bit predstavlja izhod (zaporedje dolgega in treh kratkih impulzov), najvišji pa je 1 ob koncu zaporedja.

2. Deklarirajmo še 2-bitni nepredznačen signal (**data**) in 4-bitni števec (**n**) s katerim beremo pomnilnik. Branje pomnilnika opisuje stavek:

   SHDL:
   ```
   data = rom(n)
   ```

   VHDL:
   ```
   data <= ROM(to_integer(n));
   ```

3. Števec naj se povečuje ob start=1 in resetira kadar ob data(1)=1. Izhod je določen z data(0).