



Laboratorij za načrtovanje integriranih vezij

Univerza *v Ljubljani*  
Fakulteta *za elektrotehniko*



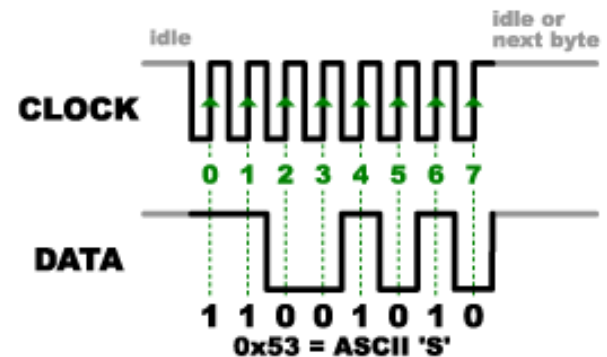
Digitalni Elektronski Sistemi

## Zaporedni vmesniki

Zaporedni (serijski) vmesniki

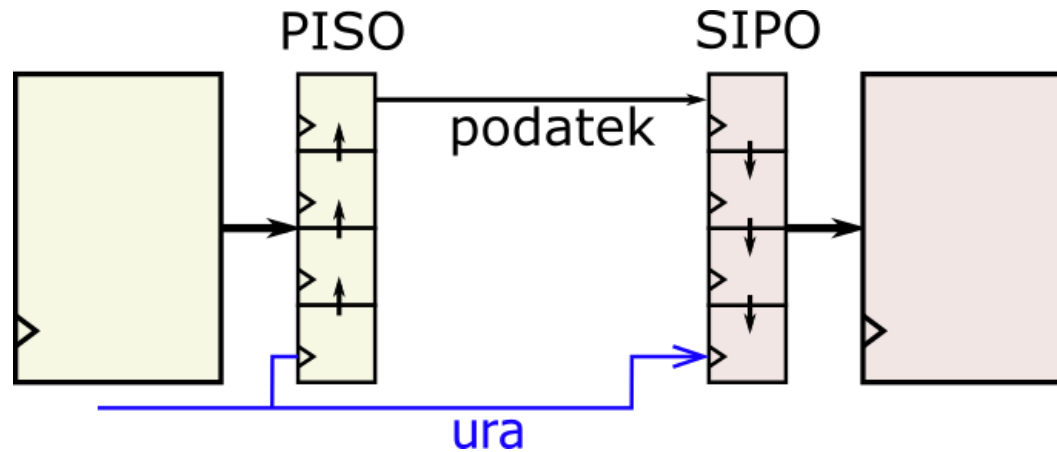
# Zaporedni (serijski) vmesniki

- ▶ Najprimernejši za komunikacijo na daljše razdalje
- ▶ Asinhroni vmesniki
  - ▶ pošiljajo le podatke, uro mora sprejemnik rekonstruirati
- ▶ Sinhroni vmesniki
  - ▶ na vodilu so podatki in ura



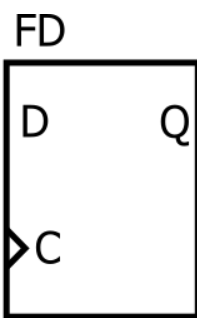
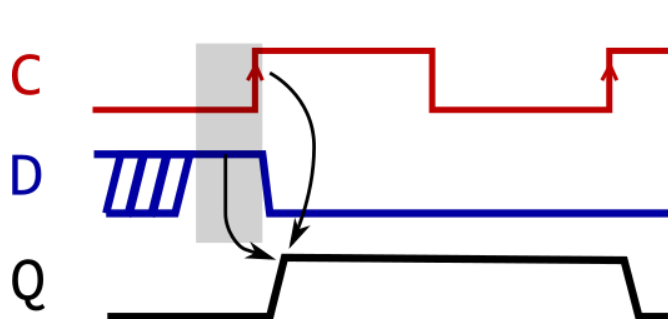
# Zaporedni prenos podatkov

---



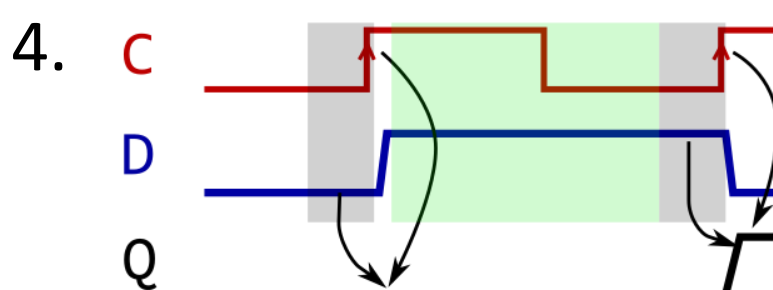
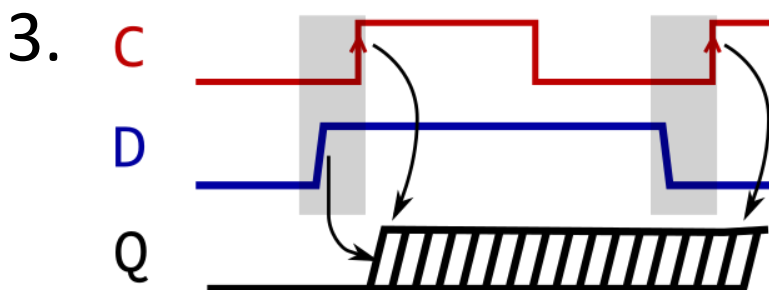
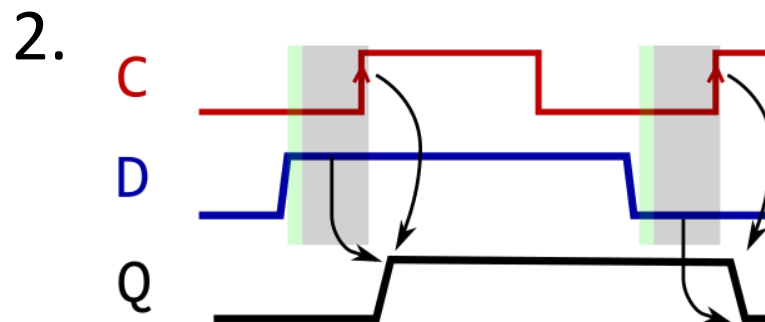
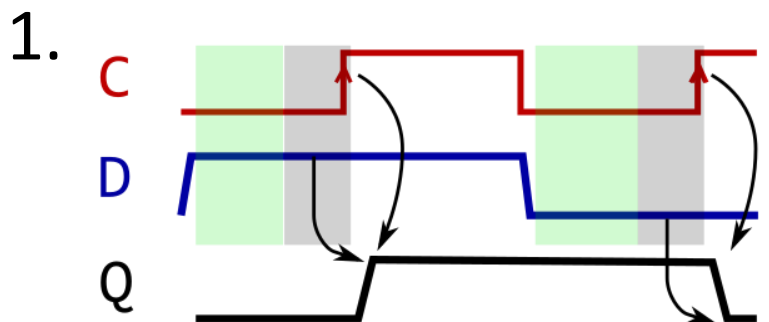
- ▶ pomikalni registri
  - ▶ PISO – parallel in serial out
  - ▶ SIPO – serial in parallel out
- ▶ Sinhroni - ura je del podatkovnega vodila
- ▶ Asinhroni - ura je kodirana med podatki
  - ▶ prenosna ura ima lahko drugačno frekvenco od systemske ure v sprejemniku in oddajniku, zato potrebujemo sinhronizacijo !

# Delovanje flip-flopa D

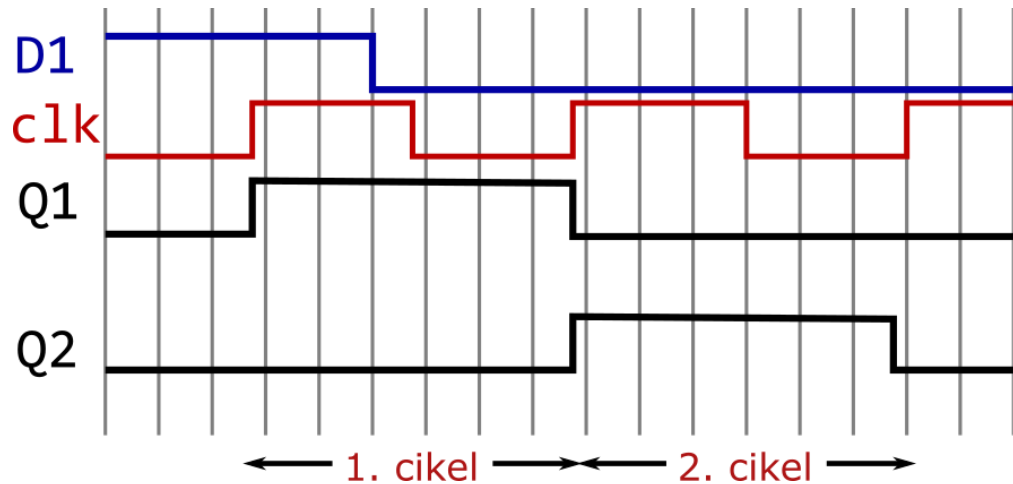
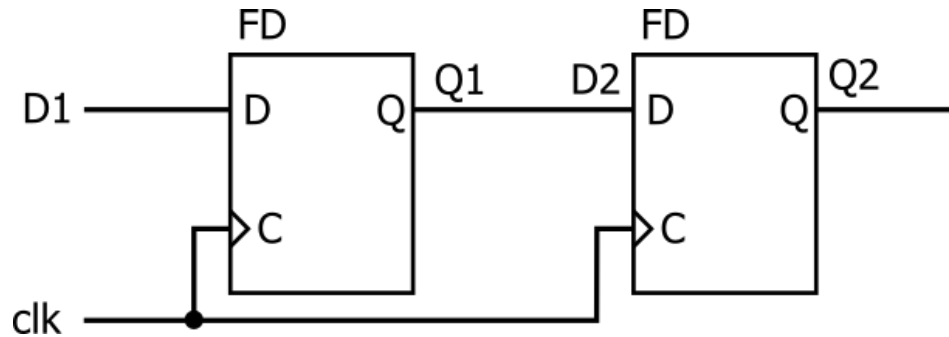


Dinamični red !

-  postavitveni čas
-  časovna rezerva

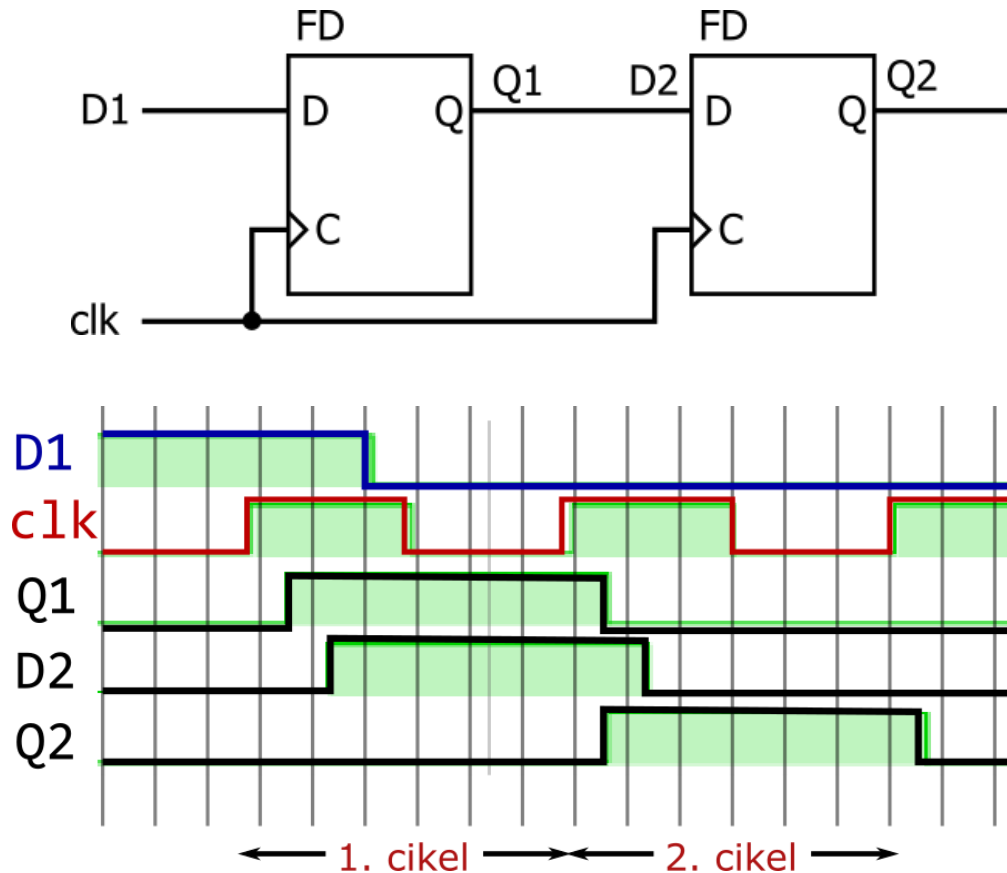


# Delovanje pomikalnega registra



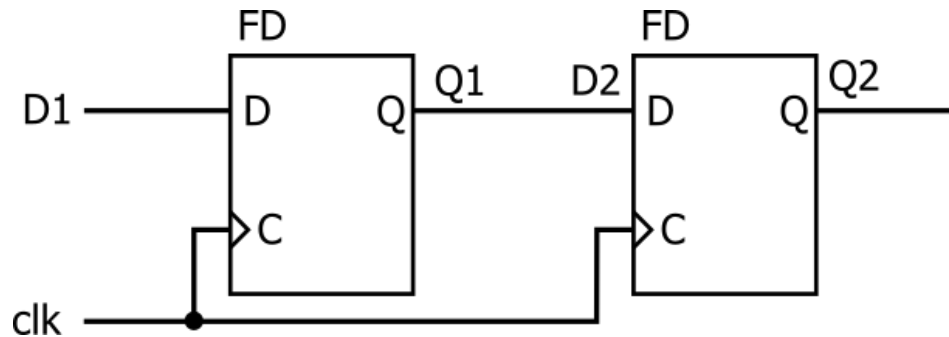
- ▶ funkcionalna simulacija (brez upoštevanja zakasnitev)

# Delovanje pomikalnega registra

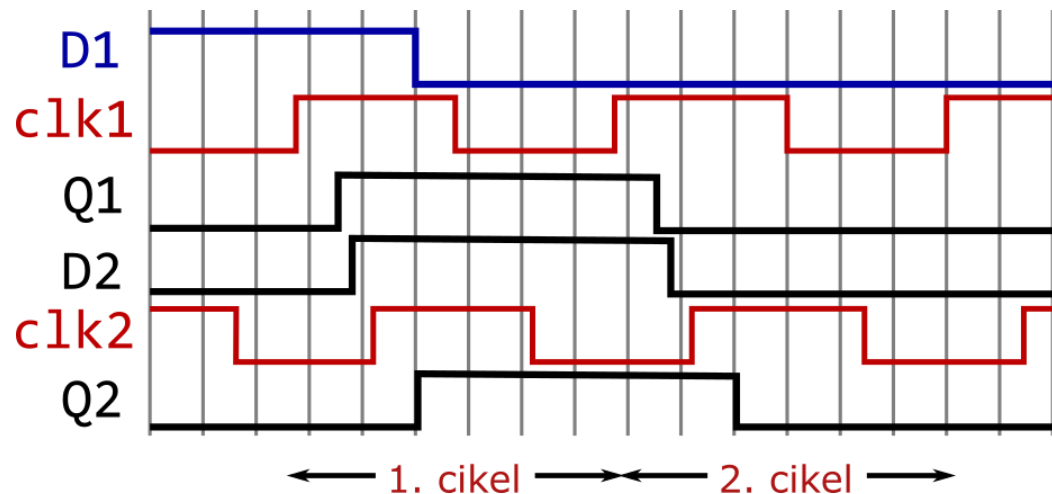


- ▶ simulacija z zakasnitvami signalov
  - ▶ izhod FD je zakasnen za fronto ure
  - ▶ propagacijska zakasnitev Q1-D2

# Delovanje pomikalnega registra



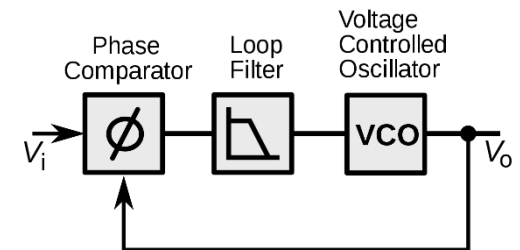
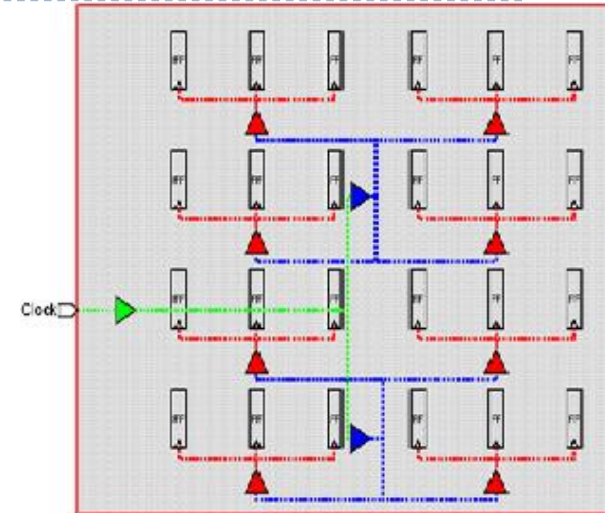
- ▶ Kaj če FD ne dobita enake ure (propagacija do FD1,2) ?



- ▶ drugi FD s proži prezgodaj
- ▶ **Vezje ne deluje kot 2-bitni pomikalni register !**

# Rešitev za distribucijo ure

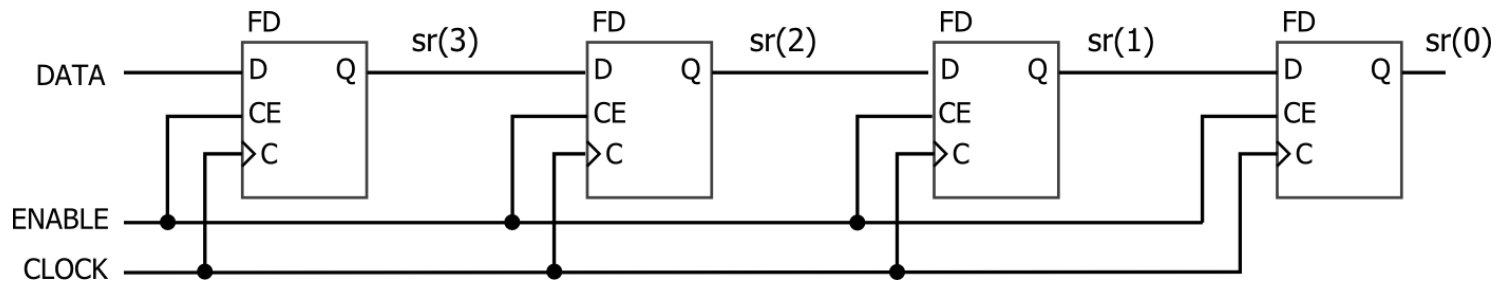
1. Izenačitev zakasnitve ure znotraj čipa
  - ▶ drevesna struktura ojačevalnikov (H-tree)
  - ▶ uro povezujemo prek globalnih ojač. (global buffer)
2. Zamik ure: PLL, DLL
  - ▶ fazno sklenjene zanke (Phase Locked Loop) in zakasnilne zanke (Delay Locked Loop)
  - ▶ uporabno za visoke frekvence
3. Menjava aktivne fronte ure
  - ▶ npr. izhodni FD dela ob padajoči fronti ure, vhodni FD vzorči ob naraščajoči fronti ure
  - ▶ primerno za počasne serijske vmesnike





# Zaporedna pretvorba podatkov v vzporedno

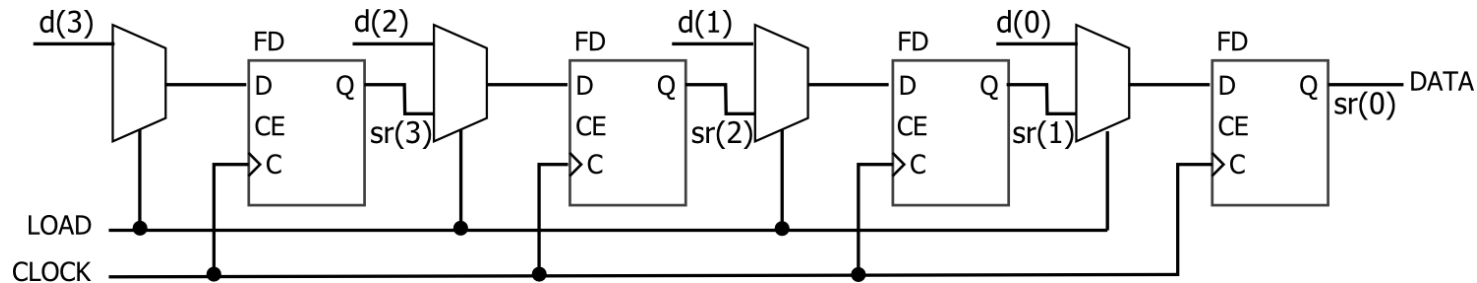
- ▶ Pretvorba s pomikalnim registrom (SIPO)
  - ▶ zaporedno vežemo toliko DFF, kolikor je dolg podatkovni paket
  - ▶ paziti moramo na zaporedje – prvi poslani bit je MSB ali LSB
- ▶ Npr. 4-bitni pomikalni register, prvi bit je LSB
  - ▶ podatki se pomikajo v desno, proti LSB



```
p: process (CLOCK)
begin
  if rising_edge(CLOCK) and ENABLE='1' then
    sr <= DATA & sr(3 downto 1);
  end if;
end process;
```

# Vzporedna zaporedna pretvorba (oddajnik)

## ▶ Zaporedni oddajnik s pomikalnim registrom (PISO)

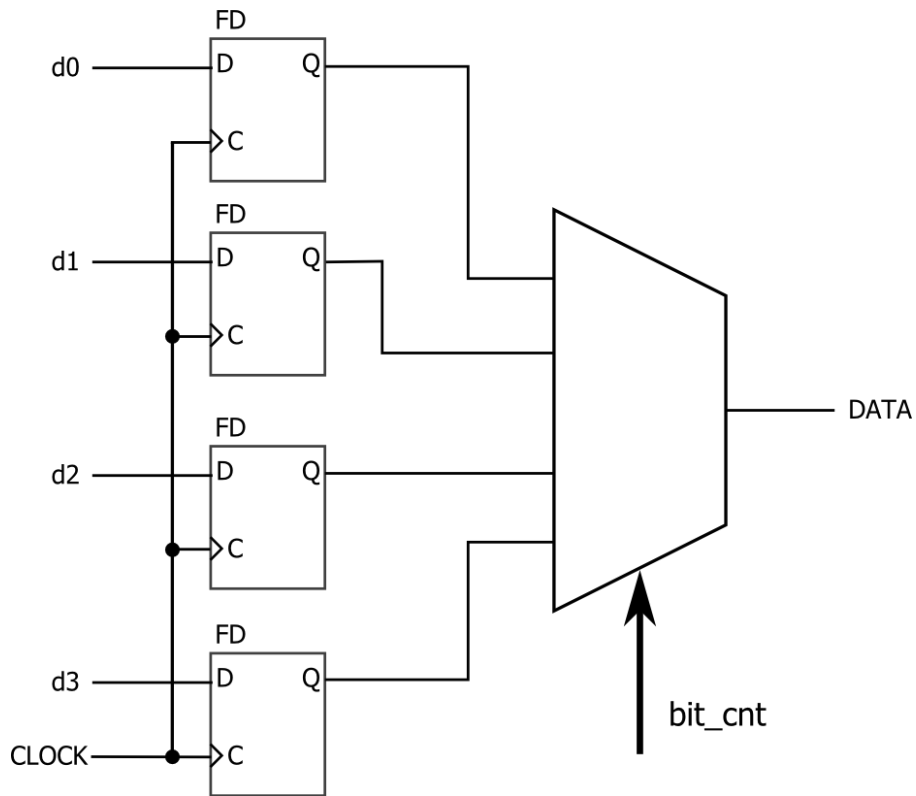


```
p: process (CLOCK)
begin
  if rising_edge(CLOCK) then
    if LOAD='1' then
      sr <= d;
    else
      sr <= '0' & sr(3 downto 1);
    end if;
  end if;
end process;

DATA <= sr(0);
```

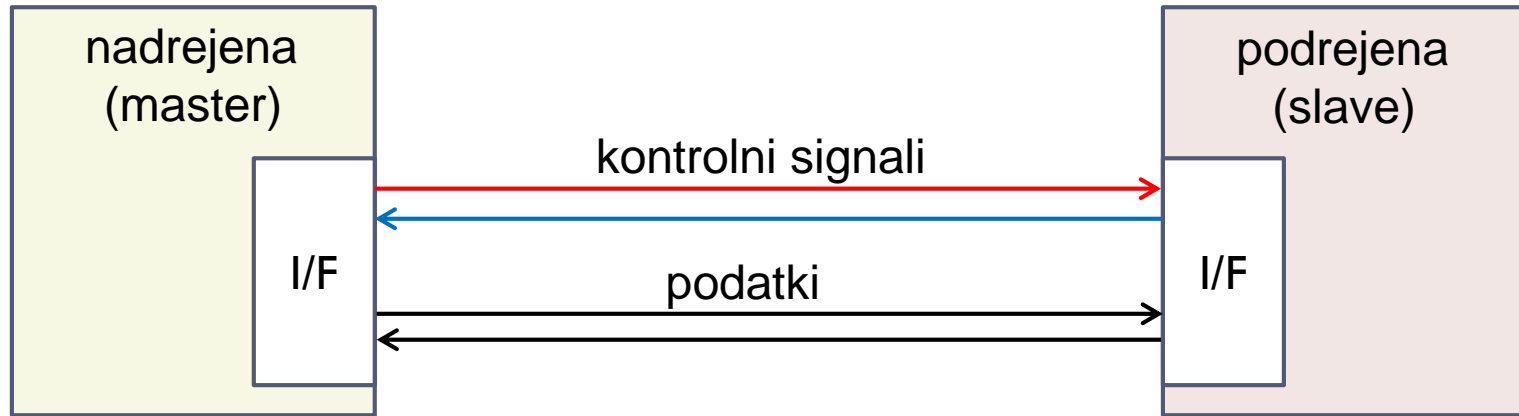
# Pretvorba z izbiralnikom

- ▶ Zaporedno vzporedno pretvorbo lahko naredimo z registrom in izbiralnikom
  - ▶ števec bitov (`bit_cnt`) določa kateri podatek gre na izhod



# Usklajevanje zaporedne komunikacije

---



- ▶ Usklajevanje določa začetek in konec prenosa, status...
- ▶ Načini usklajevanja:
  - ▶ z dodatnimi signali na vodilu, npr. load (start), select...
  - ▶ s protokolom, ki določa kaj prenašamo poleg podatkov npr. okvir: start, naslov, kontrolno vsoto, stop...

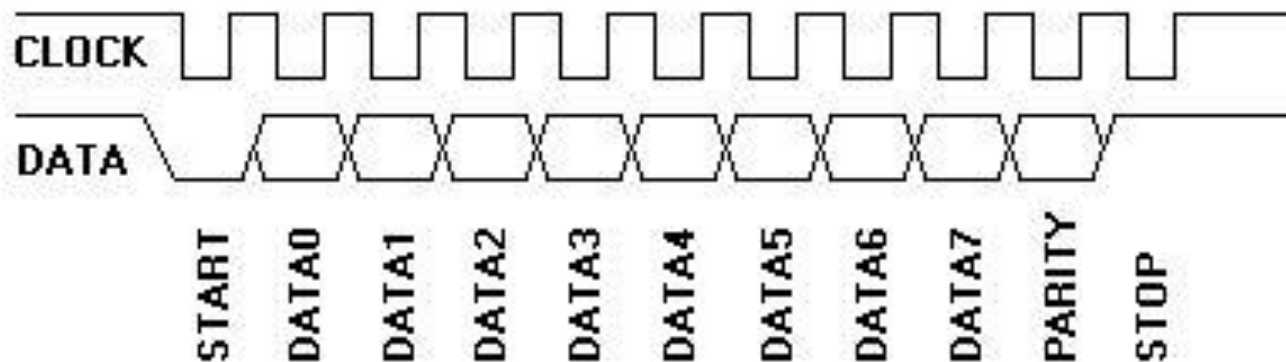
# Lastnosti zaporednih vmesnikov

---

- ▶ Bitna hitrost = frekvenca s katero se prenašajo posamezni biti, določa periodo za posamezni bit ( $T = f_{\text{bit}}$ )
- ▶ Serijski okvir (**frame**)
  - ▶ podatkovnim bitom dodamo pred prenosom kontrolne bite
  - ▶ vsebuje – start, kontrolno vsoto ali pariteto, stop
  - ▶ okvir omogoča usklajevanje sprejemnika in oddajnika in detekcijo ali odpravljanje napak pri prenosu
  - ▶ hitrost prenosa podatkov (podatkovni pretok) je zaradi okvirja manjša od bitne hitrosti
- ▶ Prenos poteka po povezavah z običajnimi logičnimi nivoji (npr. na tiskanem vezju) ali diferencialnih povezavah (za večje razdalje ali hitrosti prenosa)

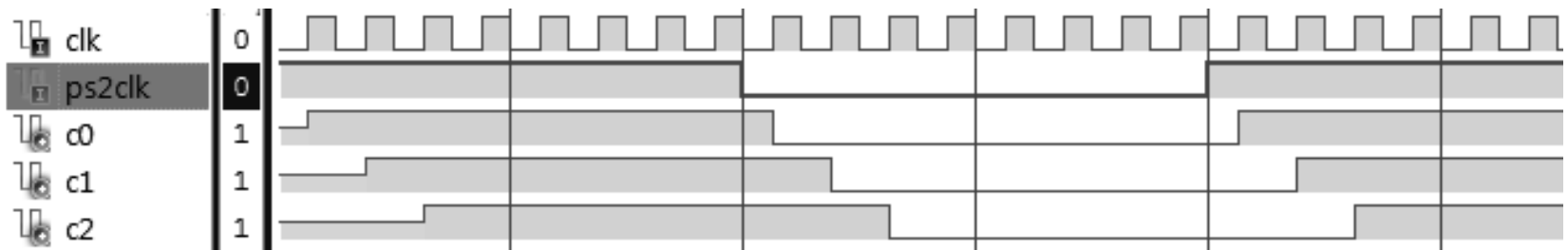
# Primer serijskega protokola

- ▶ Protokol določa vrstni red podatkovnih in kontrolnih bitov
- ▶ Asinhroni serijski vmesnik (RS232) ali sinhroni vmesnik (PS/2)
  - ▶ podatkovna povezava je na '1', kadar se nič ne prenaša
  - ▶ začetek prenosa (start) je '0', nato sledijo podatki
  - ▶ na koncu podatkov je kontrolna vsota (parity), ki ji sledi '1' (stop)
- ▶ Npr. vmesnik PS/2 pošlje najprej start ('0'), nato pa podatkovne bite od najnižjega (LSB) proti najvišjemu (MSB)...



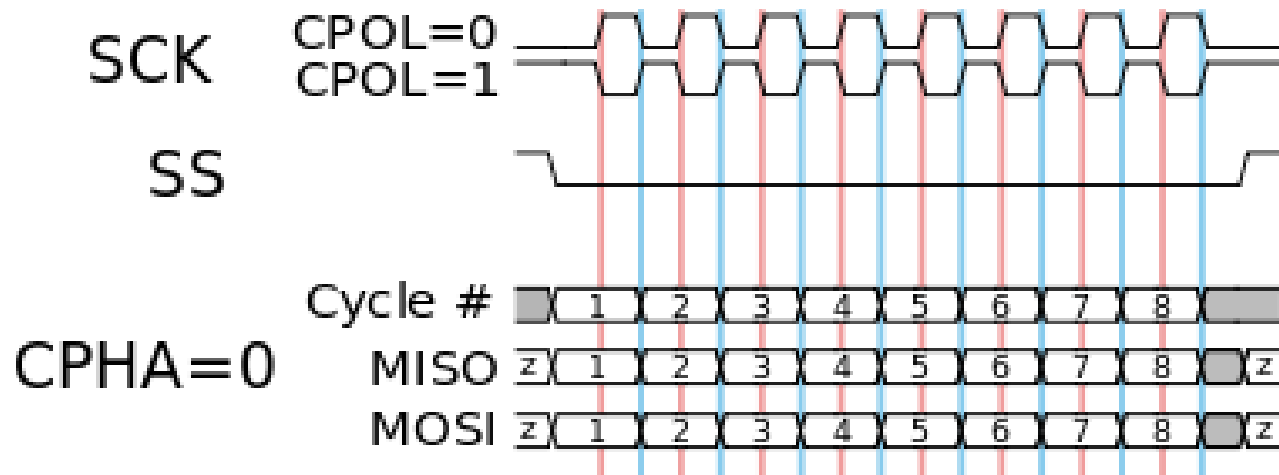
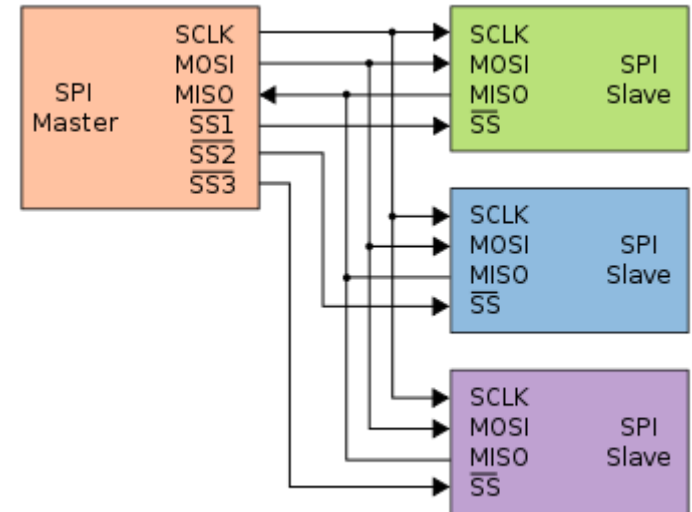
# Sinhronizacija sprejetih podatkov

- ▶ Digitalni sistem običajno dela s svojo uro
  - ▶ potrebujemo sinhronizacijo podatkov, ki se pomikajo z uro, ki jo določa hitrost prenosa
- ▶ Sinhronizacija z medpomnilniki
  - ▶ uporabimo pomnilnik FIFO (First-In First-Out) z dvema vrati, na eni strani vpisuje podatke sprejemnik, na drugi jih bere sistem
- ▶ Sinhronizacija ob vzorčenju
  - ▶ potrebujemo vsaj 2x višjo frekvenco kot je hitrost prenosa
  - ▶ vzorčimo uro in podatke ter detektiramo prehode ure za določanje trenutka zajemanja podatkov



# Sinhroni zaporedni vmesnik - SPI

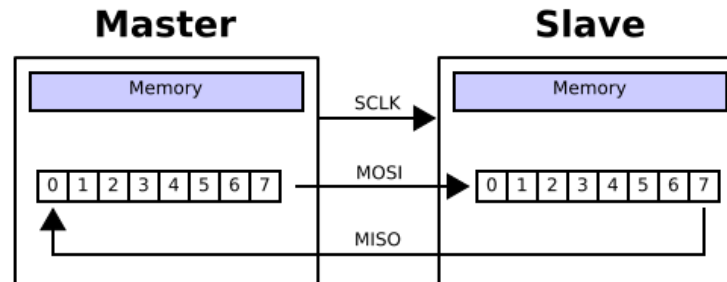
- ▶ Serial Peripheral Interface
- ▶ master – slave
  - ▶ ura (SCLK)
  - ▶ podatkovni izhod (MOSI)
  - ▶ podatkovni vhod (MISO)
  - ▶ izbira (SS0, SS1...)
- ▶ časovni potek signalov:





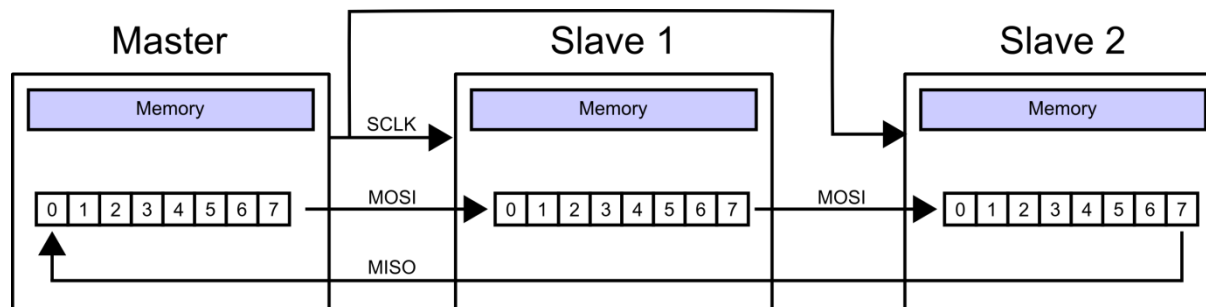
# Sinhroni zaporedni vmesnik - SPI

- ▶ oddajnik pomakne podatke na vodilo in hkrati sprejme podatkovne bite iz podrejene enote



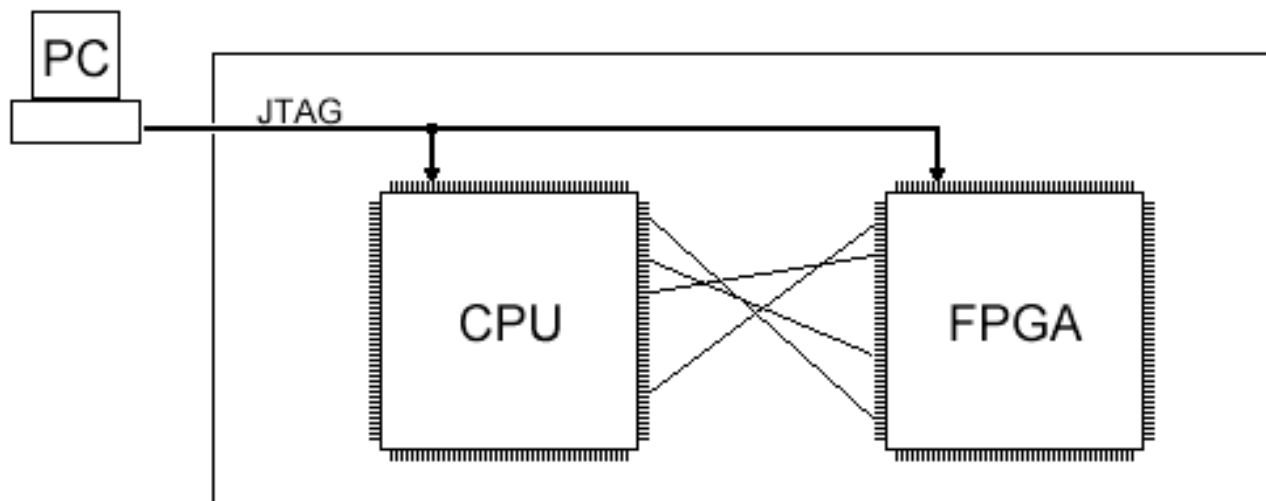
- ▶ Uporaba:

- ▶ MMC, SD kartice, senzorji, nalaganje mikrokrmilnikov
- ▶ prenos podatkov od MSB proti LSB
- ▶ več podrejenih enot lahko vežemo vzporedno ali v verigo

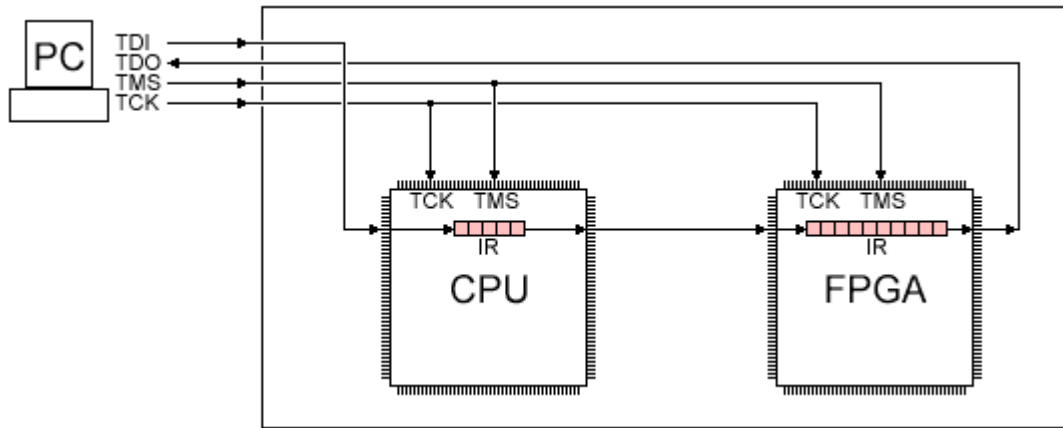


# Sinhroni zaporedni vmesnik - JTAG

- ▶ vodilo sestavlja ura (TCK), kontrolni izhod (TMS), podatkovni izhod (TDO) in vhod (TDI) v nadrejeno enoto
- ▶ JTAG je zaporedni vmesnik po IEEE standardu (1149.1) za periferno testiranje vezij
  - ▶ proizvodno testiranje povezav med dvema integriranima vezjema
  - ▶ preko JTAG odklopimo priključke od notranjosti vezja, nastavljamo stanja na priključkih in beremo stanja na drugem vezju



# Testiranje in programiranje vezij z JTAG

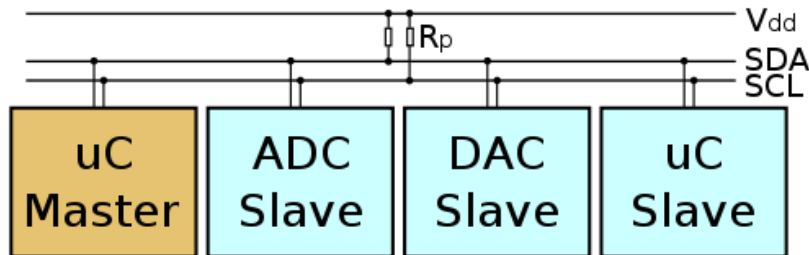


- ▶ zaporedni prenos ukazov in podatkov
- ▶ več vezij povežemo v JTAG verigo

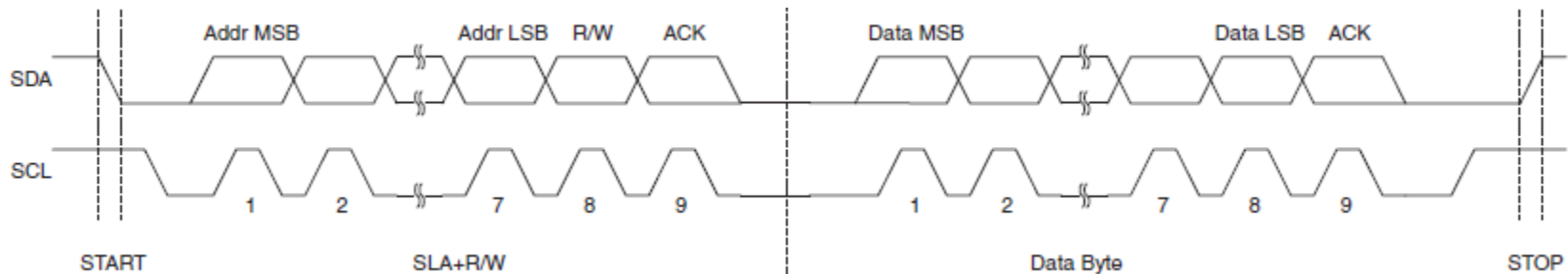
- ▶ Poleg ukazov po standardu še dodatni ukazi za programiranje vezij CPLD ali FPGA
- ▶ Programiranje in debugiranje mikrokontrolerov preko JTAG
  - ▶ ustavitev izvajanja ukazov in dostop do vseh registrov preko vmesnika JTAG

# Sinhroni zaporedni vmesnik – I2C

- ▶ Vodilo sestavlja ura (SCL) in podatkovna linija (SDA)
  - ▶ I2C – serijski EEPROM, D/A in A/D pretvorniki, senzorji
  - ▶ Omogoča povezavo več enot na vodilo, pasivna logična '1'

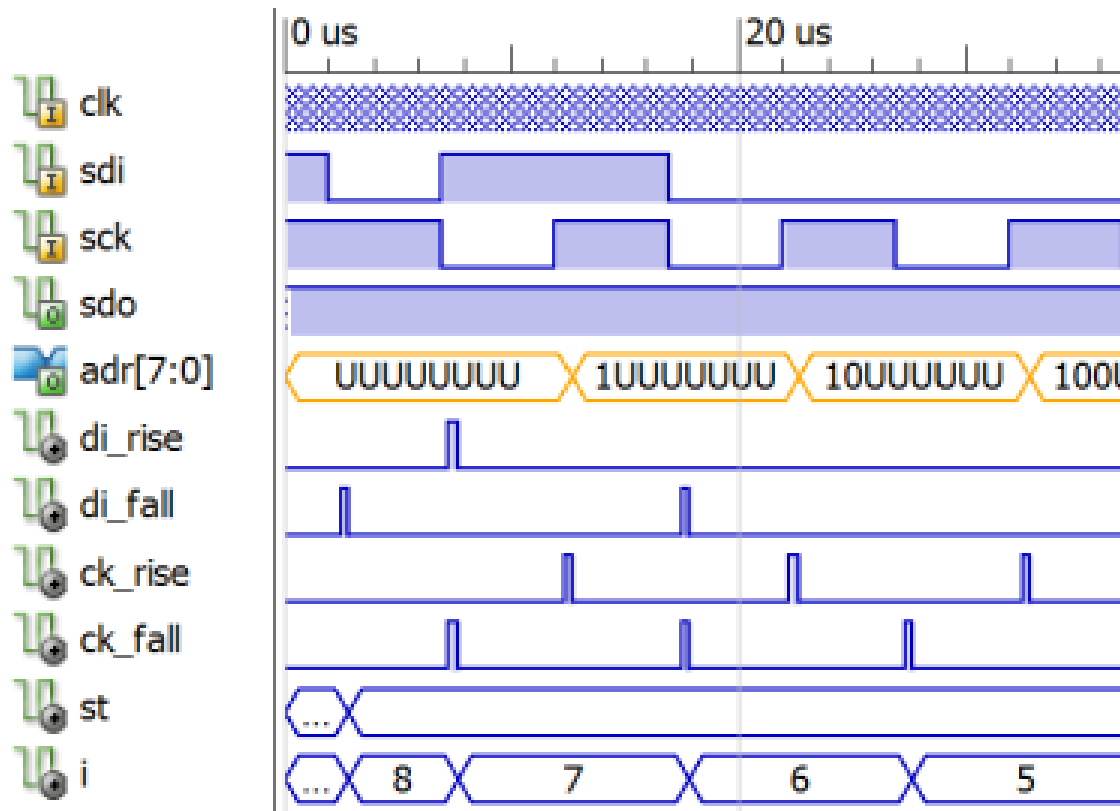


- ▶ Okvir:
  - ▶ start, naslov, r/w, potrdi, start, podatek, potrdi...



# Načrtovanje vezja vmesnika I2C

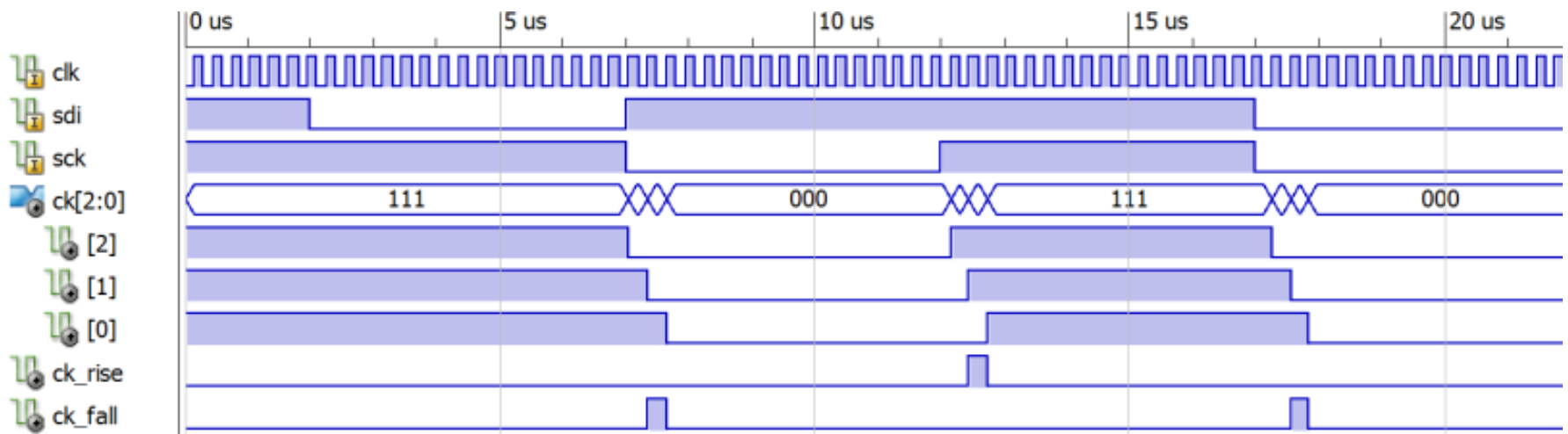
- ▶ z uporabo hitre ure (clk) zaznamo fronte signalov SDI, SCK
  - ▶ ob SCK=0 se SDI nastavi (ck\_fall)
  - ▶ ob SCK=1 se SDI ne spreminja – branje SDI (ck\_rise)
- ▶ **START:** SCK=1 in SDI gre na 0 (di\_fall)



# Sinhrono zaznavanje fronte signala

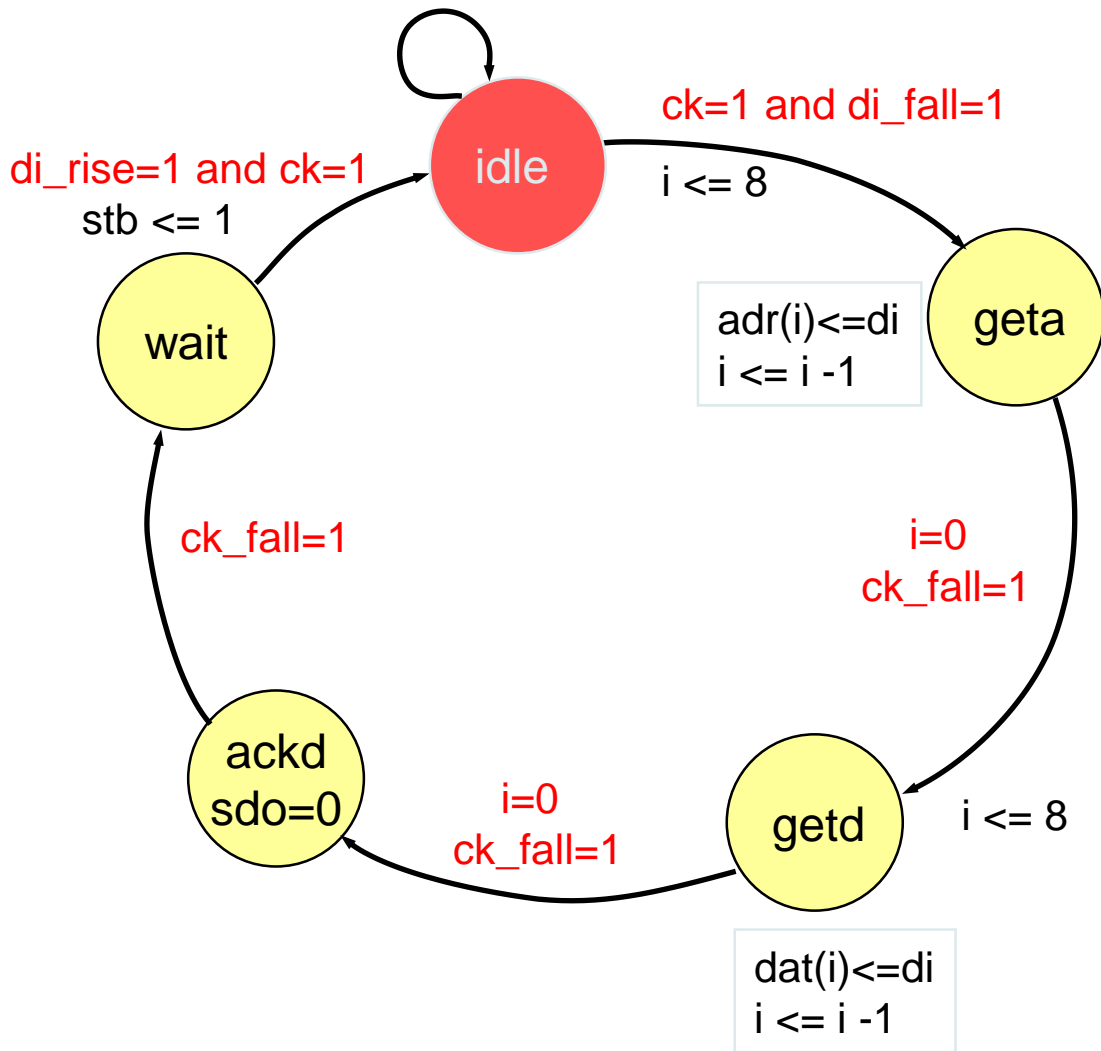
- ▶ 3x vzorčenje s pomikalnimi registri
  - ▶ eno vzorčenje za sinhronizacijo (odpravo metastabilnosti) in dve za zaznavo fronte

```
if rising_edge(clk) then
  ck <= sck & ck(2 downto 1);
end if;
```

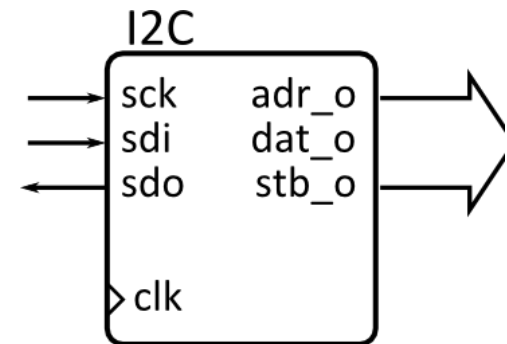


```
ck_rise <= '1' when ck(1)='1' and ck(0)='0' else '0';
ck_fall <= '1' when ck(1)='0' and ck(0)='1' else '0';
```

# Zajem podatkov s sekvenčnim strojem



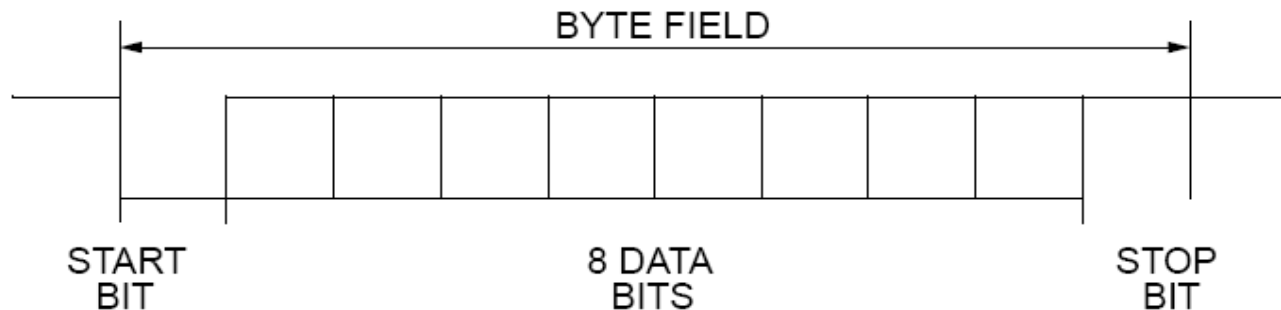
- Prenos 8-bitnega naslova in podatka



# Asinhroni zaporedni vmesnik – RS232

---

- ▶ Asinhroni sprejemnik in oddajnik (UART)
  - ▶ asinhroni podatkovni paketi, sprejemna ura se sinhronizira za vsak paket – sinhronizacija paketov
  - ▶ vnaprej dogovorjene bitne frekvence
    - ▶ 9600, 38400, 115200 bit/s
  - ▶ npr. podatkovni paket 8N1:





# Asinhroni zaporedni vmesnik – LIN

---

- ▶ **LIN (Local Interconnect Network)**
  - ▶ enostaven protokol, ki je podoben RS232
  - ▶ razvit za avtomobilsko industrijo
- ▶ V primerjavi z zmogljivejšim avtomobilskim vmesnikom **CAN** je počasnejši in enostavnejši
  - ▶ uporaben za monitoring senzorjev, prenos ukazov za odpiranje oken...
  - ▶ cenena izvedba vmesnika z mikrokrmilnikom
  - ▶ ne vsebuje robustnega odpravljanja napak
    - ▶ ni uporaben za varnostno kritične aplikacije

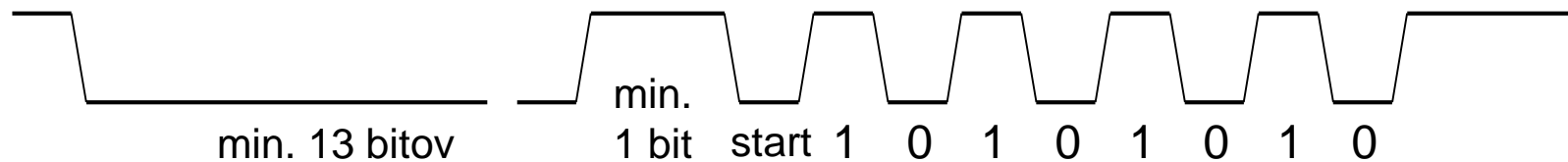


# Prenos podatkov po protokolu LIN

---

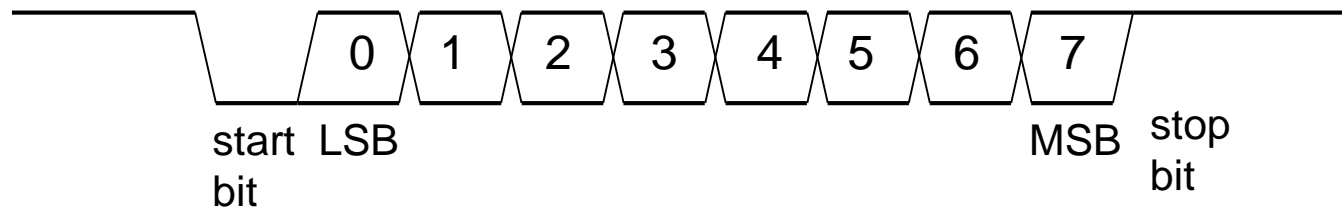
## ▶ Sinhronizacija sprejemnika

- ▶ min. 13 bitov logična 0, ki mu sledi 1 bit premora
- ▶ sledi start bit (logična 0) in kombinacija  $55_{16}$



## ▶ Prenos v serijski obliki 8N1

- ▶ start bit (logična 0), 8 podatkov in stop bit (log. 1)



# Povzetek

---

- ▶ Kakšne so prednosti zaporednega (serijskega) prenosa podatkov pred vzporednim?
- ▶ Opiši osnovni princip zaporednega prenosa podatkov s pomikalnimi registri.
- ▶ Opiši kako naredimo z logičnim vezjem vzporedno-zaporedno pretvorbo podatkov (2 načina).
- ▶ Navedi nekaj primerov zaporednih (serijskih) vmesnikov.
  - ▶ V čem se razlikujejo zaporedni vmesniki ?
  - ▶ Kje se uporabljajo ?