



Laboratorij za načrtovanje integriranih vezij

Univerza *v Ljubljani*
Fakulteta *za elektrotehniko*



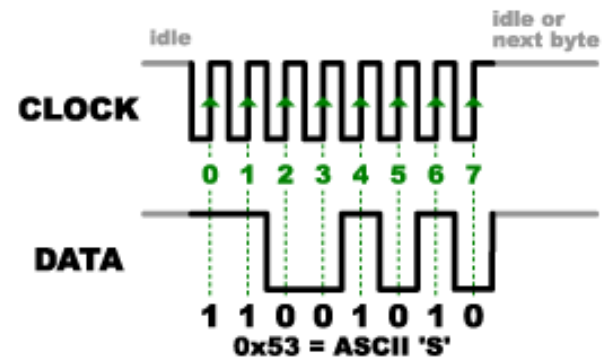
Digitalni Elektronski Sistemi

Zaporedni (serijski) vmesniki

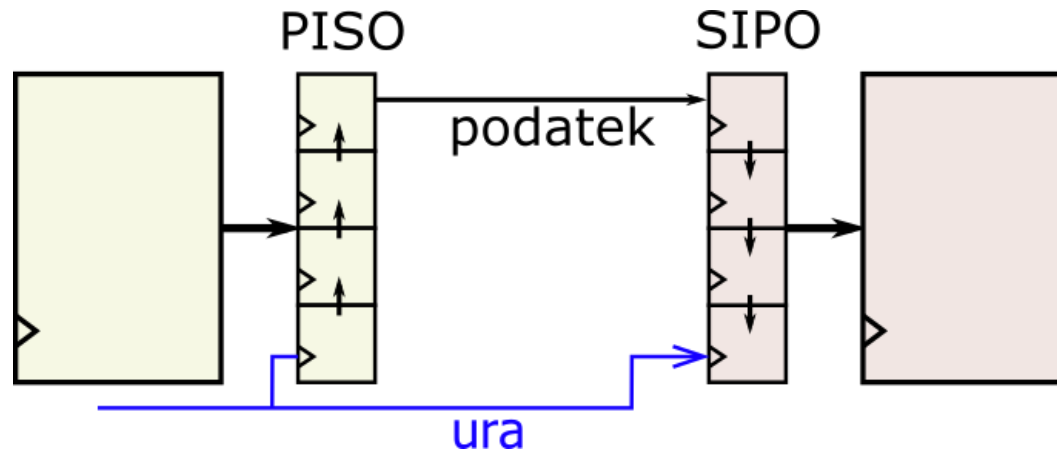
Osnove zaporednega prenosa in primeri vmesnikov

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

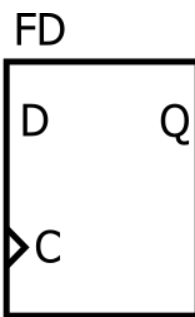
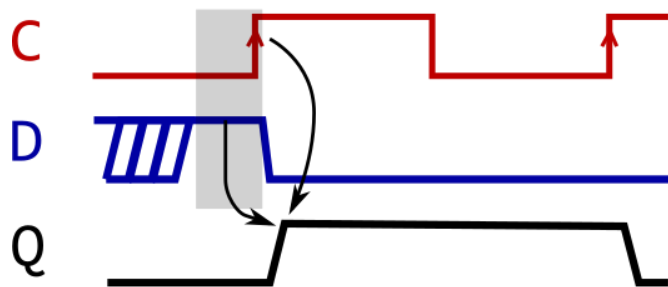


Zaporedni prenos podatkov



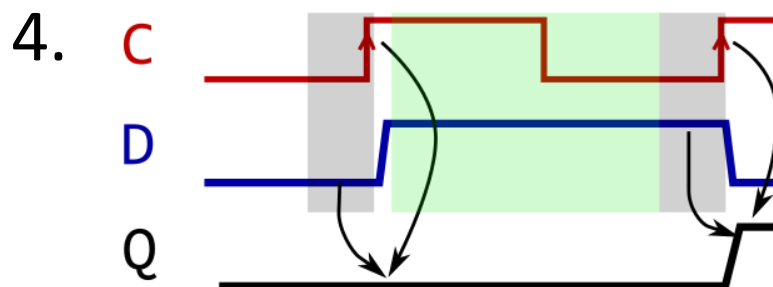
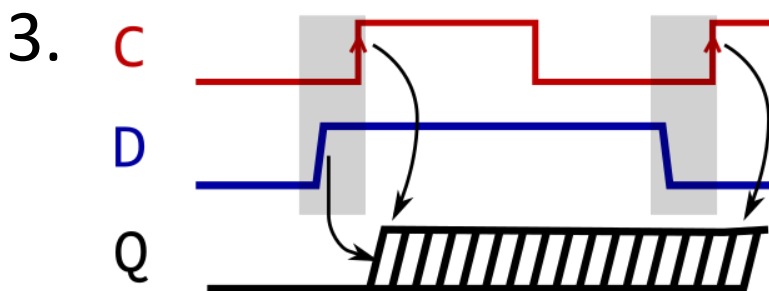
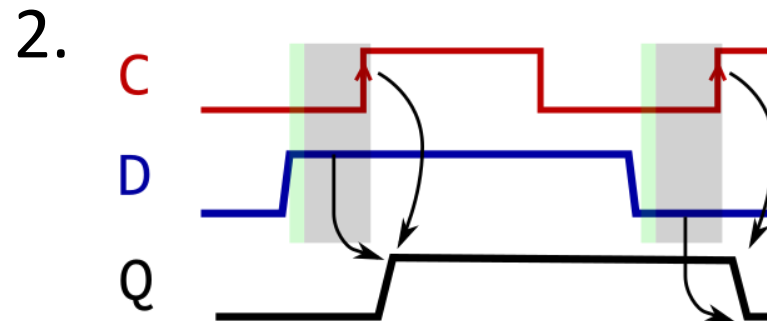
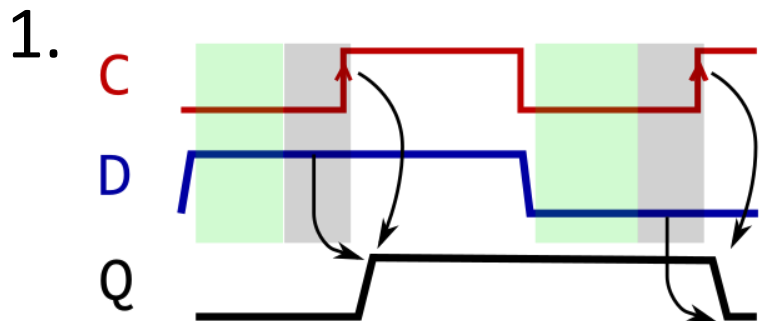
- ▶ pomikalni registri
 - ▶ PISO – parallel in serial out
 - ▶ SIPO – serial in parallel out
- ▶ Sinhroni - ura je del podatkovnega vodila
- ▶ Asinhroni - ura je kodirana med podatki
 - ▶ prenosna ura ima lahko drugačno frekvenco od systemske ure v sprejemniku in oddajniku, zato potrebujemo sinhronizacijo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

Osnove: delovanje flip-flopa D

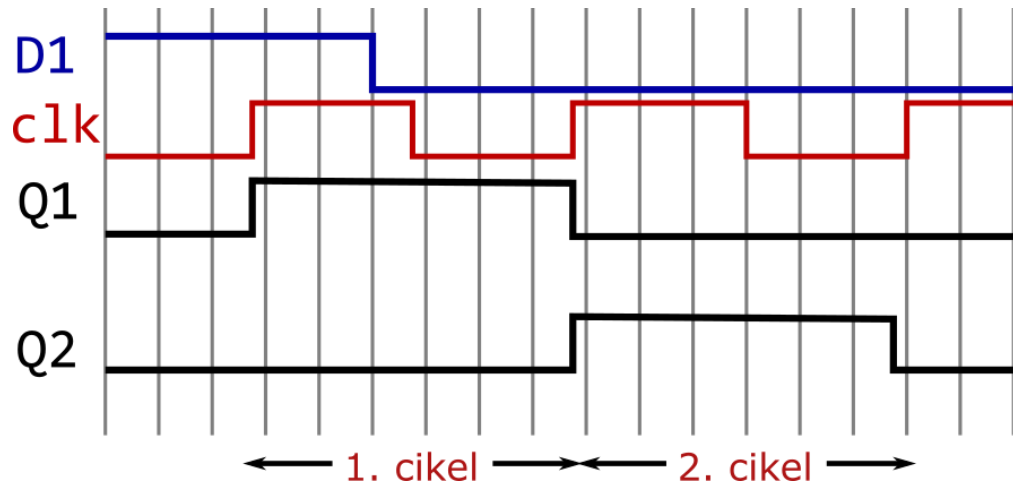
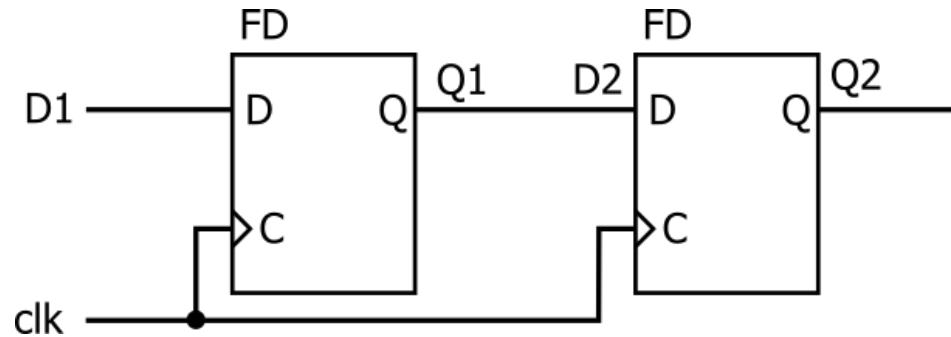


Dinamični red !

-  postavitveni čas
-  časovna rezerva

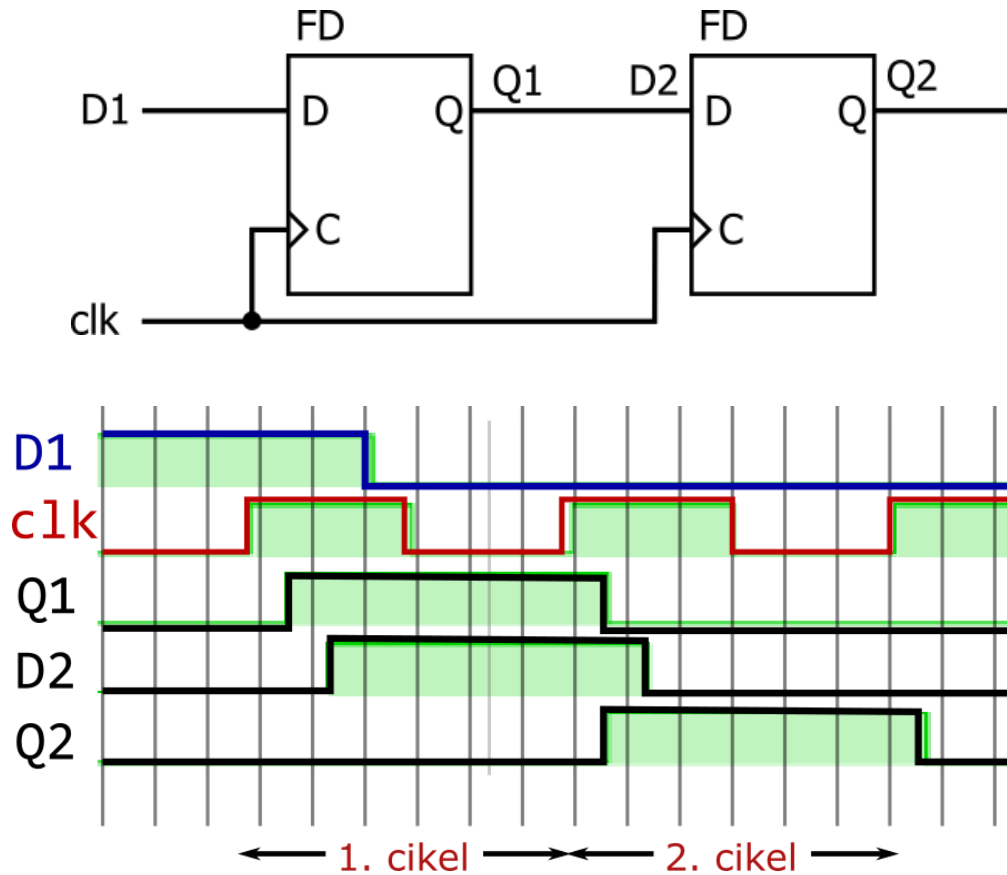


Osnove: delovanje pomikalnega registra (1)



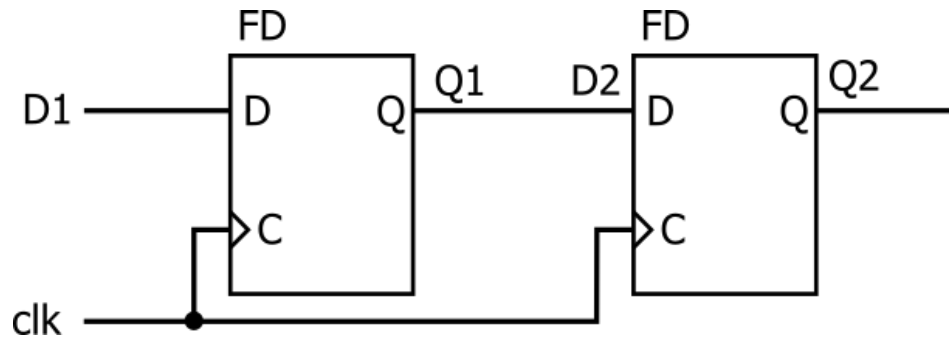
- ▶ funkcionalna simulacija (brez upoštevanja zakasnitev)

Osnove: delovanje pomikalnega registra (2)

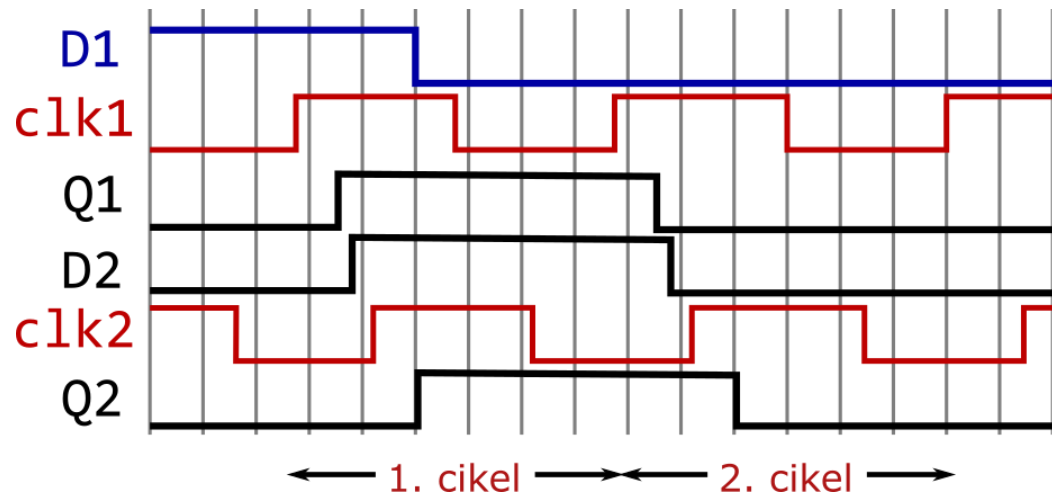


- ▶ simulacija z zakasnitvami signalov
 - ▶ izhod FD je zakasnen za fronto ure
 - ▶ propagacijska zakasnitev Q1-D2

Osnove: delovanje pomikalnega registra (3)



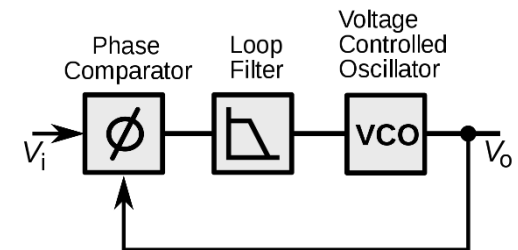
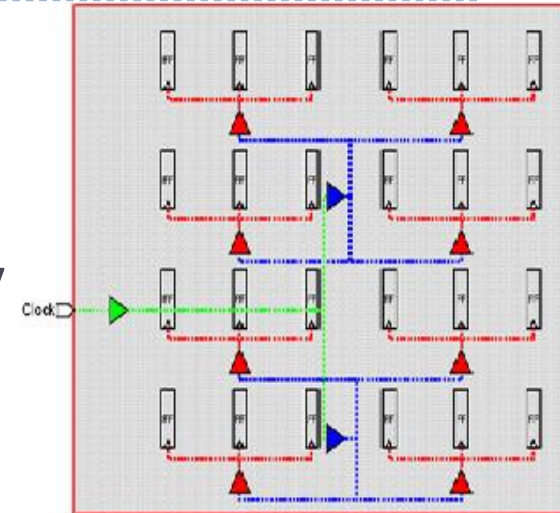
- ▶ Kaj če FD ne dobita enake ure (propagacija do FD1,2) ?



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

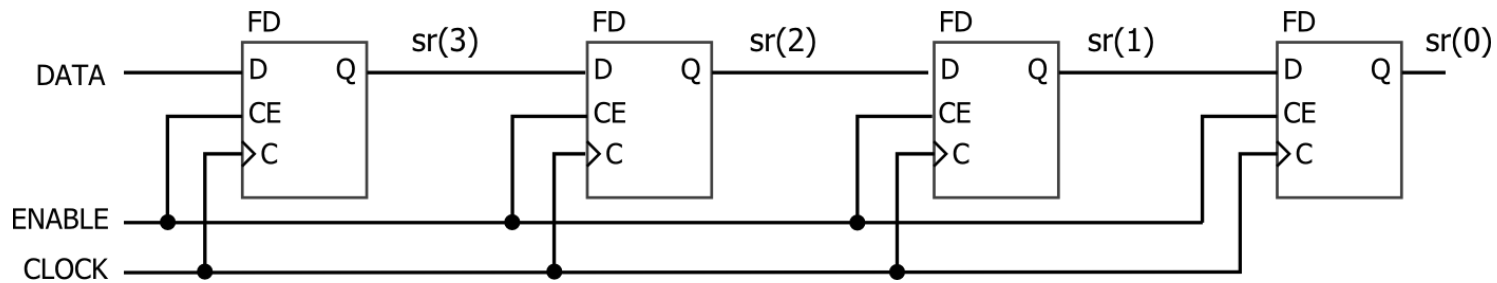
Rešitev za distribucijo ure

1. Izenačitev zakasnitve ure znotraj čipa
 - ▶ drevesna struktura ojačevalnikov (H-tree)
 - ▶ uro povezujemo prek globalnih ojačevalnikov (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

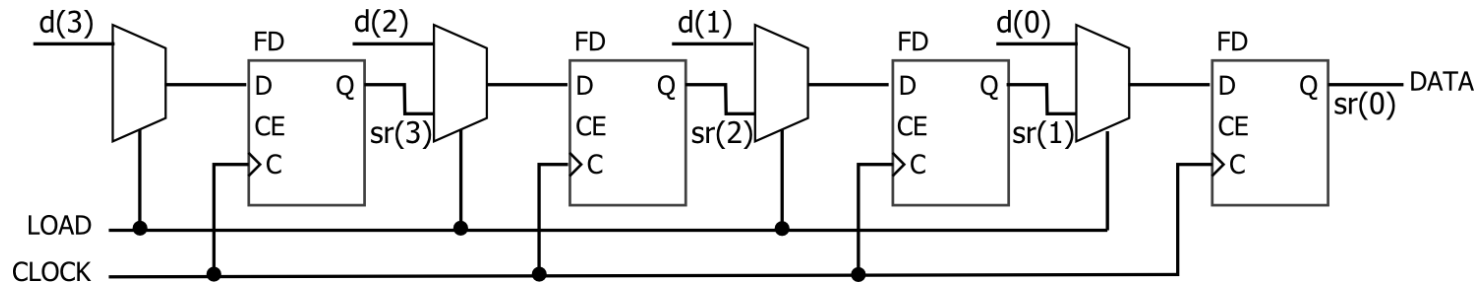
- ▶ Pretvorba s pomikalnim registrom (SIPO)
 - ▶ zaporedno vežemo toliko DFF, kolikor je dolg podatkovni paket
 - ▶ paziti moramo na zaporedje – prvi poslani bit je MSB ali LSB
- ▶ Npr. 4-bitni pomikalni register, prvi bit je LSB
 - ▶ podatki se pomikajo v desno, proti LSB



```
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)

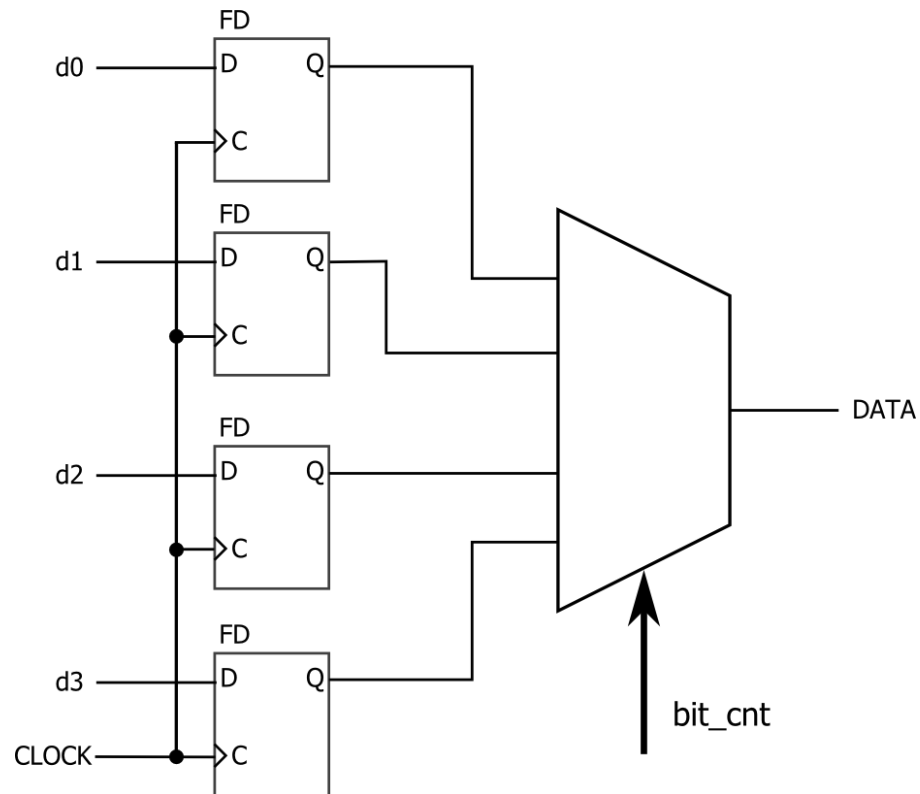


```
p: process (CLOCK)
begin
  if rising_edge(CLOCK) then
    if LOAD='1' then
      sr <= d;
    else
      sr <= '0' & sr(3 downto 1);
    end if;
  end if;
end process;

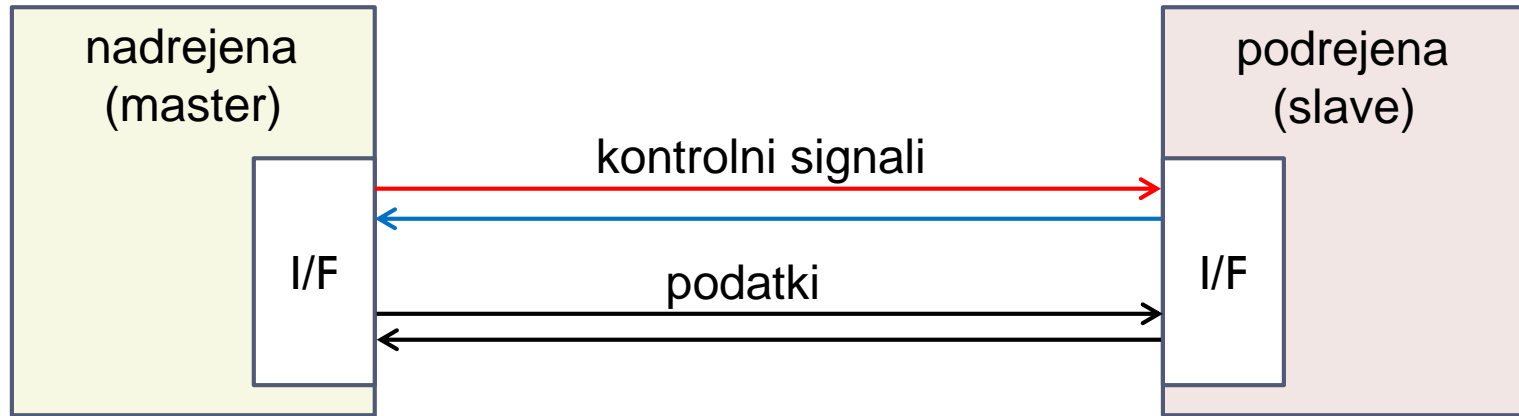
DATA <= sr(0);
```

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



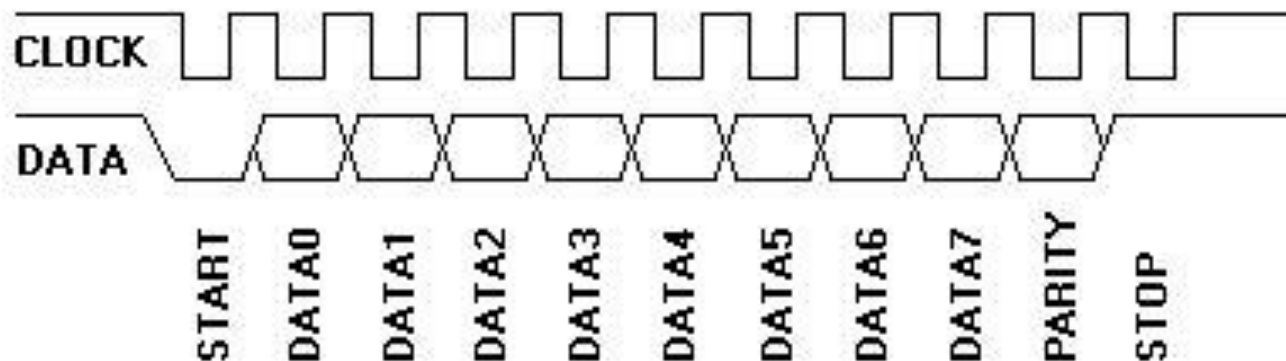
- ▶ Usklajevanje določa začetek in konec prenosa, status...
- ▶ Načini usklajevanja:
 - ▶ z dodatnimi signali na vodilu, npr. load (start), select...
 - ▶ s protokolom, ki določa kaj prenašamo poleg podatkov npr. okvir: start, naslov, kontrolno vsoto, stop...

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)...

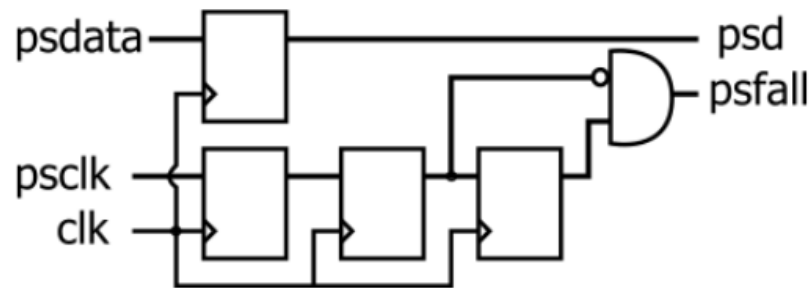


Sinhronizacija sprejetih podatkov

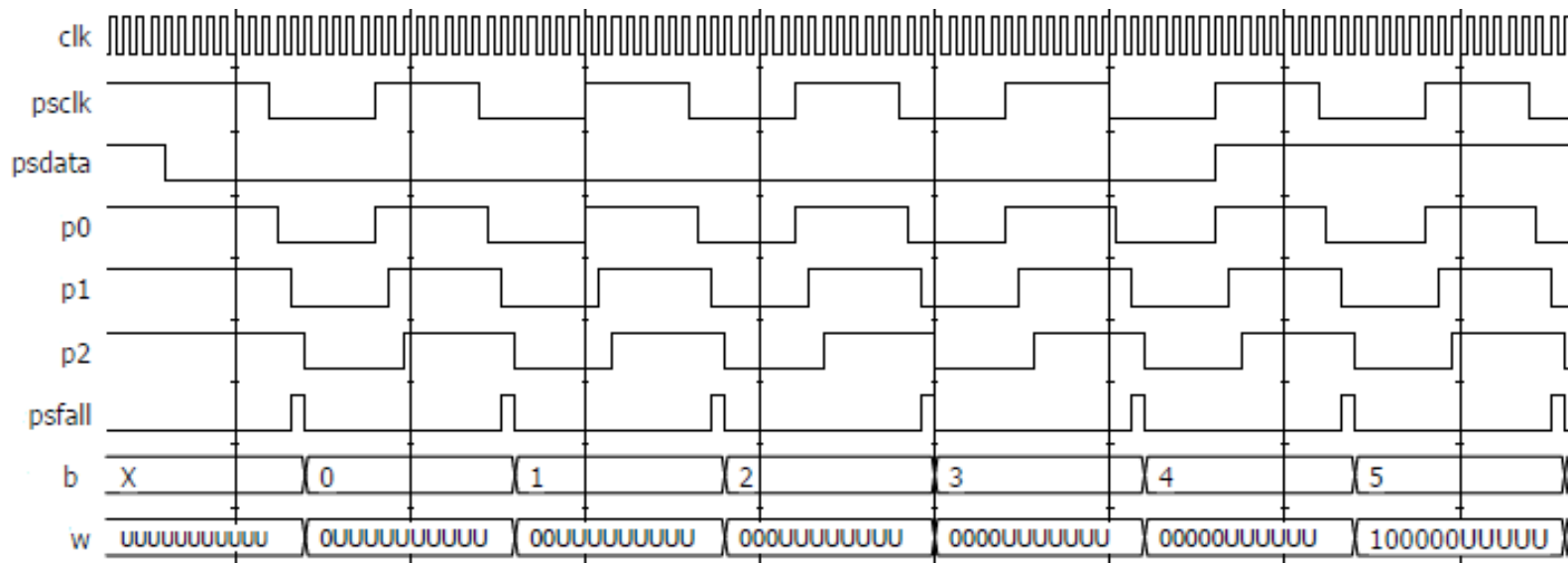
- ▶ Digitalni sistem običajno dela s svojo uro
 - ▶ potrebujemo sinhronizacijo podatkov, ki se pomikajo z uro, ki jo določa hitrost prenosa
- ▶ Sinhronizacija z medpomnilniki
 - ▶ uporabimo pomnilnik FIFO (First-In First-Out) z dvema vrati, na eni strani vpisuje podatke sprejemnik, na drugi jih bere 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

Vzorčenje signalov vmesnika PS/2

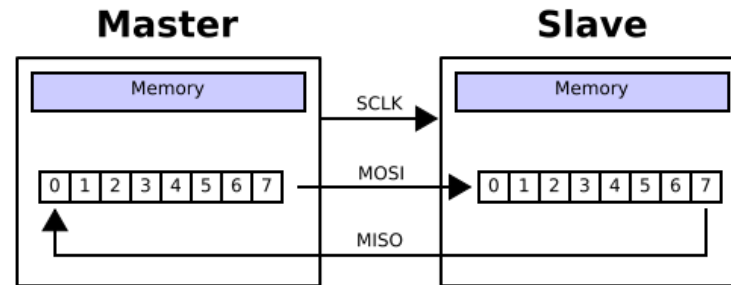
- ▶ s hitro sistemsko uro vzorčimo uro (psclk) in podatke (psdata)
- ▶ z dvema zaporednima flip-flopoma in vrati zaznamo fronto ure psclk (psfall)



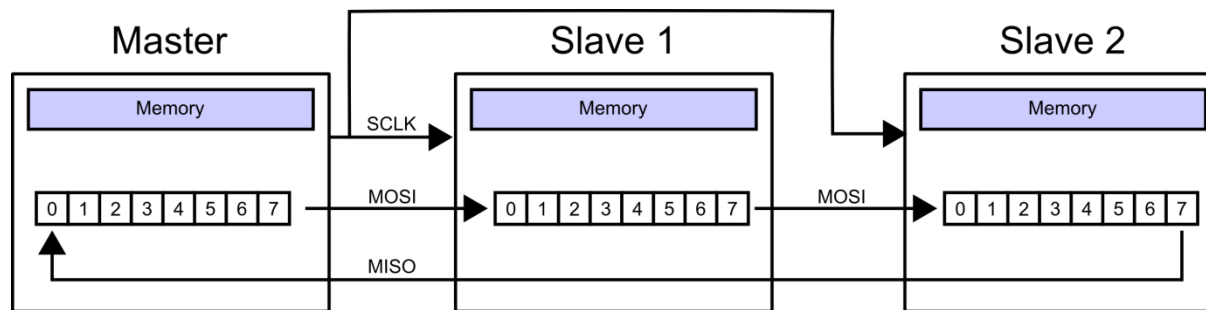
- ▶ signal psfall uporabimo za sinhrono pomikanje in štetje bitov



Delovanje in uporaba vmesnika SPI



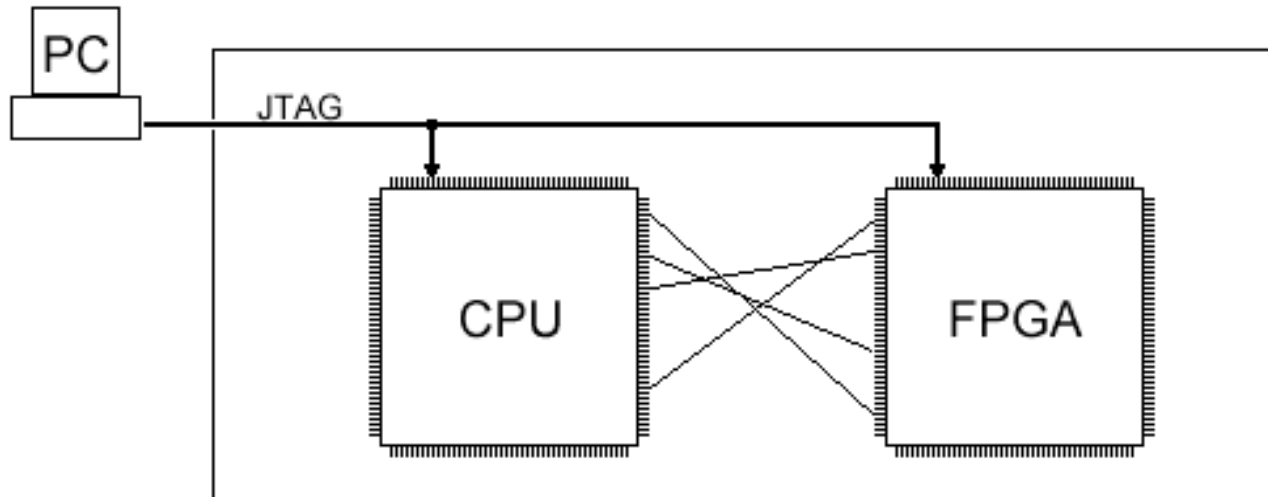
- ▶ oddajnik pomika podatke od MSB proti LSB po vodilu in hkrati sprejme podatkovne bite iz podrejene enote
 - ▶ več podrejenih enot lahko vežemo vzporedno ali v verigo



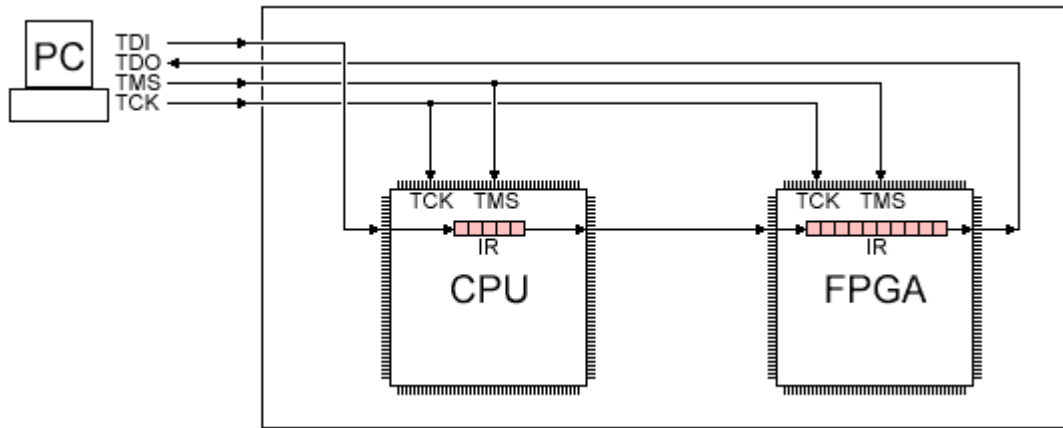
- ▶ Uporaba: MMC, SD kartice, senzorji, nalaganje mikrokrmilnikov

Primer: JTAG

- ▶ Sinhroni zaporedni vmesnik s signali:
 - ▶ ura (TCK), kontrolni izhod (TMS), podatkovni izhod (TDO) in vhod (TDI) v nadrejeno enoto
- ▶ JTAG je zaporedni vmesnik po standardu IEEE 1149.1 za robno testiranje vezij
 - ▶ proizvodno testiranje povezav med čipi na tiskanem vezju
 - ▶ med testom odklopimo priključke od notranjosti vezja, nastavljamo stanja na priključkih in beremo stanja na drugem vezju



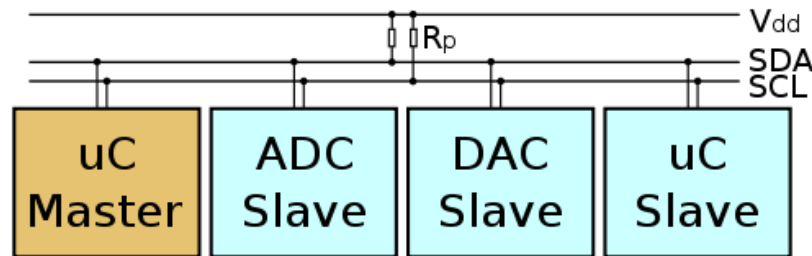
Delovanje in uporaba JTAG



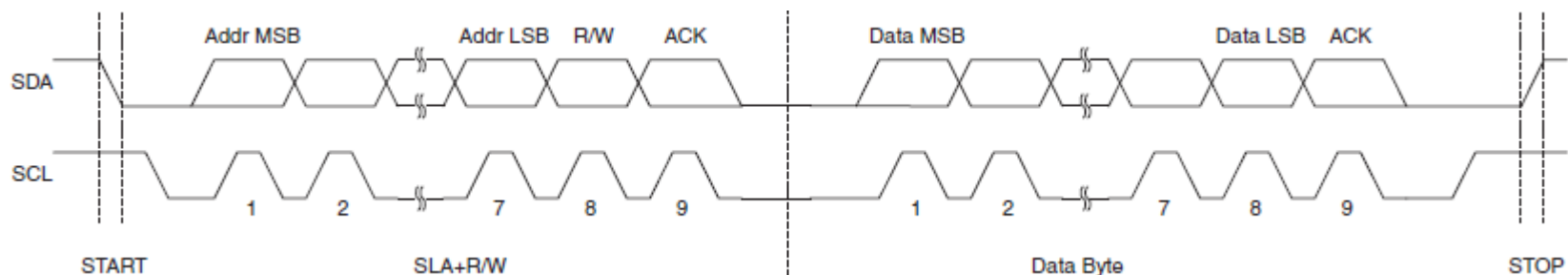
- ▶ zaporedni prenos ukazov in podatkov
 - ▶ več vezij povežemo v zaporedno verigo
-
- ▶ Uporaba: testiranje tiskanin, programiranje, debugiranje
 - ▶ programirljiva vezja (CPLD, FPGA) imajo poleg standardnih ukazov še dodatne za programiranje vezij
 - ▶ nalaganje in debugiranje mikrokontrolerov
 - ▶ periferna enota z vmesnikom JTAG omogoča ustavitve izvajanja ukazov, dostop do registrov in vsebine pomnilnika

Primer: Inter Integrated Circuit (I2C)

- ▶ sinhroni zaporedni vmesnik (Philips) z dvema povezavama:
 - ▶ ura (SCL) in podatkovna linija (SDA)
 - ▶ vodilo omogoča povezavo več enot, ker ima pasivno logično '1'



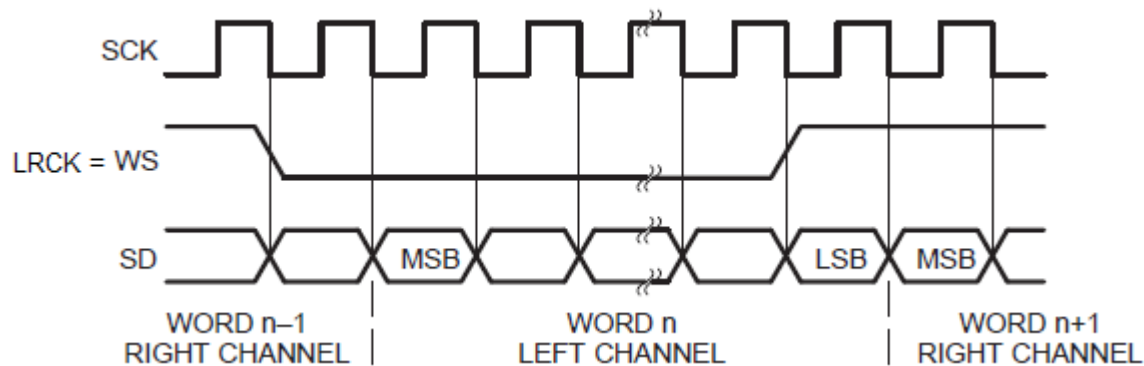
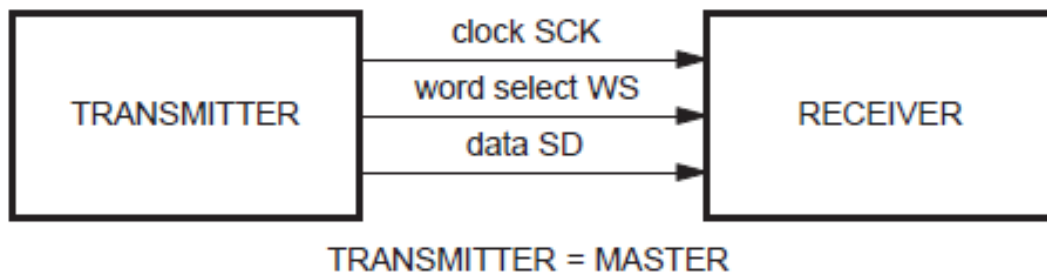
- ▶ uporaba I2C – serijski EEPROM, D/A in A/D pretvorniki, senzorji



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

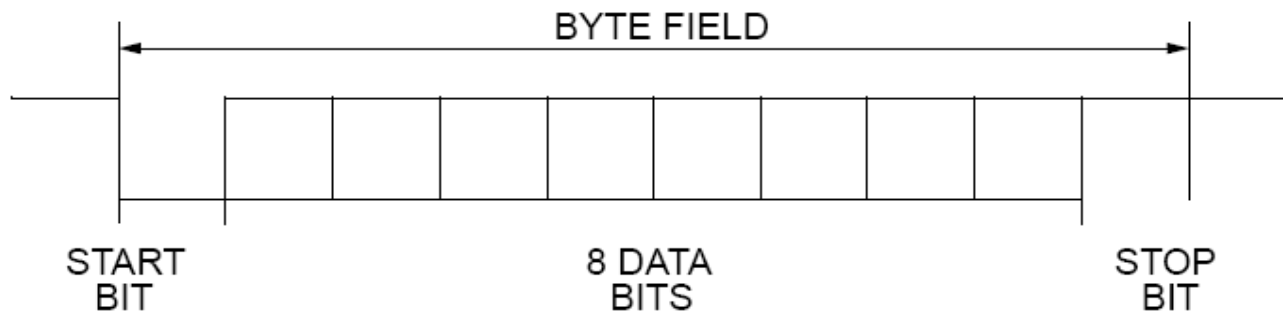
Primer: Inter-IC Sound (I2S)

- ▶ sinhroni zaporedni vmesnik za prenos zvoka (Philips)
 - ▶ naprave: A/D, D/A, signalni procesor, zvočni kodek
 - ▶ signali: ura za prenos bitov (SCK), signal za izbiro besed (WS ali LRCLK, npr. 0-levi, 1-desni kanal), serijski podatki (SD)
- ▶ nadrejena naprava (master) oddaja SCK in WE



Primer: RS232

- ▶ asinhroni zaporedni komunikacijski standard (l. 1960)
 - ▶ oddajna povezava (tx), sprejemna povezava (rx)
 - ▶ opcijski dodatni signali za usklajevanje prenosa
- ▶ vmesnik UART - asinhroni sprejemnik in oddajnik
 - ▶ asinhroni podatkovni paketi, sprejemna ura se sinhronizira za vsak paket – sinhronizacija paketov
 - ▶ vnaprej dogovorjene bitne hitrosti: 9600, 38400, 115200 bit/s...
 - ▶ npr. podatkovni paket v obliki 8N1 (8-bitov, brez paritete, 1 stop)



Uporaba asinhronega vmesnika UART

- ▶ zelo razširjen vmesnik za prenos podatkov
 - ▶ RS232: prenos po kablu na razdalji do 15m, v nekaterih izvedbah do 300 m (naprave v industriji)
 - ▶ enostaven in zanesljiv protokol za prenos manjše količine podatkov
- ▶ znakovni terminal za mikrokontrolnike in vgrajene računalnike
 - ▶ RS232 omogoča servisni dostop do programske opreme vgrajenega računalnika, usmerjevalnika, ipd.
- ▶ pretvorba iz kompleksnejših vmesnikov
 - ▶ npr. iz USB na RS232, ki ga lahko povežemo z večino mikrokontrolnikov

Primer: Local Interconnect Network (LIN)

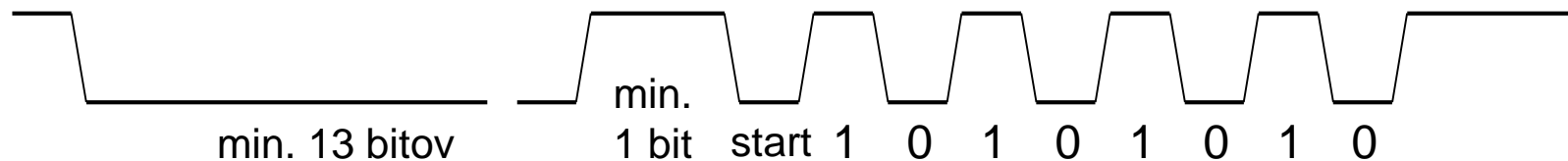
- ▶ asinhroni zaporedni vmesnik
 - ▶ enostaven protokol, ki je podoben RS232
 - ▶ razvit za avtomobilsko industrijo
- ▶ V primerjavi z zmogljivejšim avtomobilskim vmesnikom **CAN** je počasnejši in enostavnejši
 - ▶ uporaben za monitoring senzorjev, prenos ukazov za odpiranje oken...
 - ▶ cenena izvedba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



Prenos podatkov po protokolu LIN

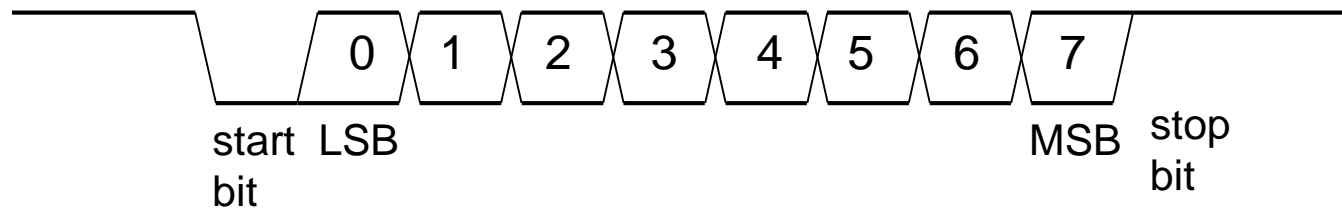
▶ Sinhronizacija sprejemnika

- ▶ 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 ?