



Laboratorij za načrtovanje integriranih vezij

Univerza *v Ljubljani*
Fakulteta *za elektrotehniko*



Digitalni Elektronski Sistemi

Vmesniki in sekvenčna vezja

Zaporedna in vzporedna vodila

Vmesniki in vodila

- ▶ **Električni vmesnik (interface)**
 - ▶ sestavljen iz fizičnih povezav in priključkov ter logičnega vezja
- ▶ **Naloga digitalnih vmesnikov**
 - ▶ prenos podatkov in dogodkov med dvema napravama
- ▶ **Vodilo (bus)**
 - ▶ podatki se prenašajo po vodilu
 - ▶ vodilo omogoča prenos podatkov med dvema ali več napravami
- ▶ **Pasivno vodilo**
 - ▶ npr. direktne povezave med procesorjem in pomnilnikom
- ▶ **Vodilo s krmilnikom**
 - ▶ krmilnik poskrbi za usklajevanje prenosa podatkov
 - ▶ izvaja najnižji – fizični nivo prenosa (phy)

Kaj prenašamo po vodilu

- ▶ **Podatke**
 - ▶ prenašajo se po besedah (*byte*)
- ▶ **Ukaze oz. kontrolne signale**
 - ▶ izberi napravo / beri / piši
 - ▶ naslov izvora ali ponora podatkov
- ▶ **Signale, ki jih določa protokol prenosa**
 - ▶ zahteve (*request*), prekinitve (*interrupt*)
 - ▶ potrditve (*acknowledge*)
 - ▶ časovno usklajevanje (sinhrono / asinhrono)



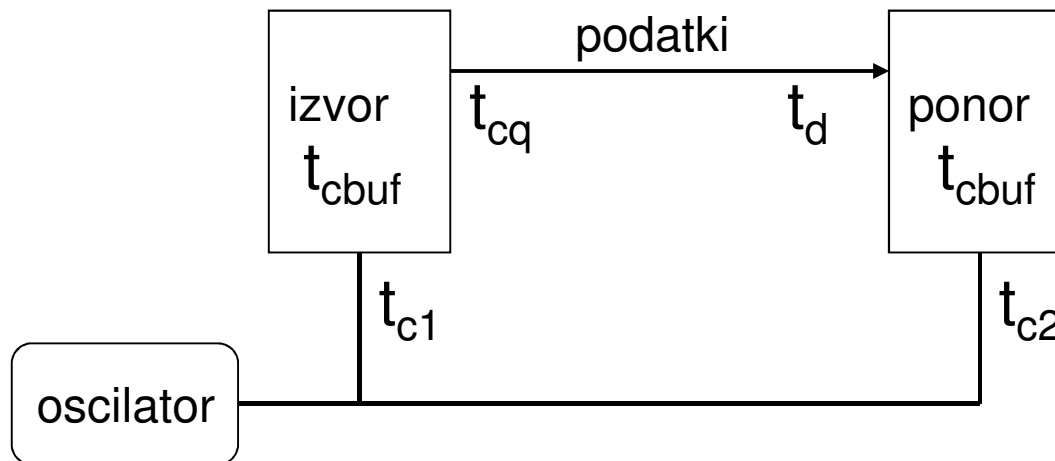
Kako prenašamo podatke

- ▶ vzporedno po besedah (paralelni) ali zaporedno po bitih (serijski prenos)
- ▶ po smeri
 - ▶ enosmerno
 - ▶ dvosmerno (dvojno ali 3-stanjsko vodilo)
 - ▶ hkrati v obe smeri
- ▶ po protokolu
 - ▶ enostaven cikel
 - ▶ blokovni prenos (burst)
 - ▶ manj usklajevanja, večji izkoristek vodila



Sinhronizacija prenosa podatkov

- ▶ Sinhronizacija je pomembna za doseganje visokih hitrosti
 - ▶ dinamični red – izogibanje tekmovanju med signali
- ▶ Sistemska sinhronizacija
 - ▶ izvor in ponor imata skupni oscilator za urni takt

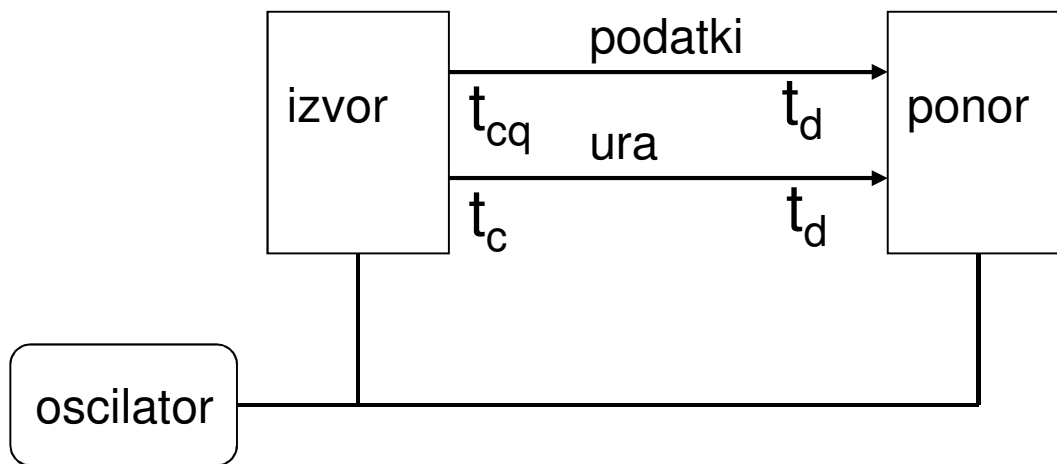


- Upoštevati moramo vse zakasnitve!
 - pri nizkih hitrostih prenosa nimamo težav



Sinhronizacija pri izvoru podatkov

- ▶ Izvor pošilja podatke in uro
 - ▶ *source synchronous / clock forwarded* (SDRAM)

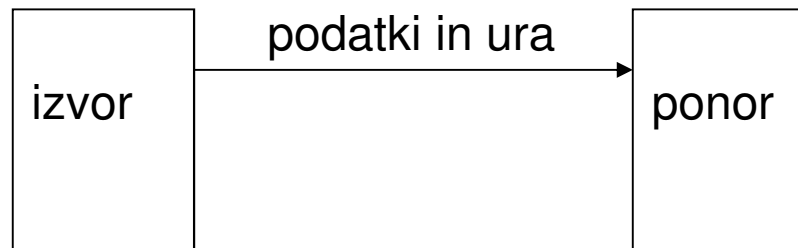


- dolžina linij za podatke in uro se mora ujemati
- v ponoru naredimo ponovno sinhronizacijo na sistemsko uro



Lastna sinhronizacija podatkov

- ▶ Podatkovni tok vsebuje podatke in uro
 - ▶ serijski prenos podatkov



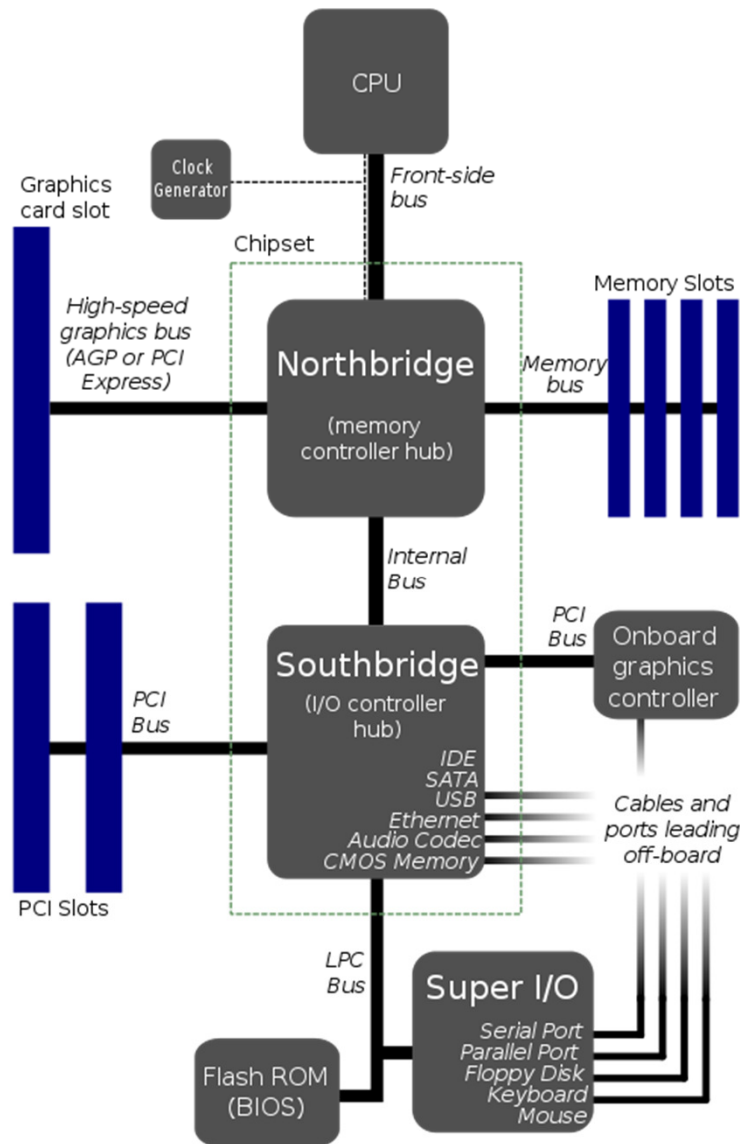
- izvor dela paralelno/serijsko pretvorbo in kodiranje podatkov in ure
- ponor dela serijsko/paralelno pretvorbo in rekonstrukcijo ure s fazno sklenjeno zanko (PLL)



Razvoj komunikacije v digitalnih sistemih

- ▶ Vzporedni prenos podatkov po tiskanem vezju s standardnimi napetostnimi nivoji (5V TTL/CMOS)
- ▶ Paralelna vodila na plošči osebnega računalnika
 - ▶ vodilo ISA (Industry Standard Architecture, 1981/84)
 - ▶ IBM AT vodilo prenaša 16 bitov pri 8 MHz, TTL
 - ▶ paralelno vodilo za disk (IDE/ATA)
- ▶ Vzporedno vodilo Centronics
 - ▶ prenos podatkov do tiskalnika, na krajše razdalje
- ▶ Zaporedna vodila
 - ▶ RS232, RS485 diferencialni nivoji omogočajo večje razdalje
- ▶ Hitra zaporedna vodila
 - ▶ USB, Ethernet, PCI Express

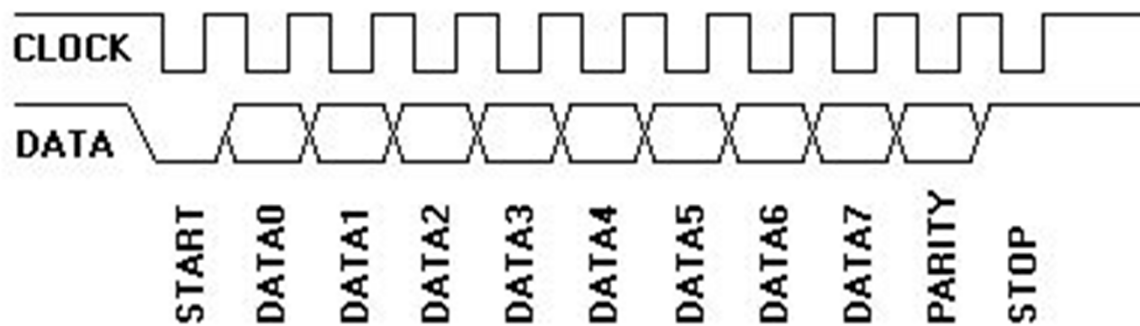
Vodila v osebnem računalniku



- ▶ **sistemsko vodilo**
 - ▶ odvisno od procesorja
- ▶ **pomnilnik SDRAM**
 - ▶ 64 bitov, 133 MHz, SSTL 2.5V
 - ▶ do 533 MHz, SSTL 1.5V
- ▶ **vzporedno vodilo PCI**
 - ▶ 32 bitov, 33 MHz (133MB/s)
 - ▶ 64 bitov, 66 MHz (533MB/s)
- ▶ **hitri zaporedni vmesniki**
 - ▶ PCIe (1-16 GB/s)
 - ▶ SATA (1.5-6 GB/s)
- ▶ **vmesniki za zunanja vodila**
 - ▶ zaporedni USB, Ethernet
 - ▶ RS232, PS/2

Zaporedni (serijski) vmesniki

- ▶ Najprimernejši za komunikacijo na daljše razdalje
- ▶ Sinhroni serijski vmesniki
 - ▶ pošiljajo tudi uro
- ▶ Asinhroni vmesniki
 - ▶ pošiljajo le podatke, uro mora sprejemnik rekonstruirati
- ▶ Protokol določa vrstni red podatkovnih in kontrolnih bitov
 - ▶ Npr. vmesnik PS/2 pošlje najprej start ('0'), nato pa podatkovne bite od najnižjega (LSB) proti najvišjemu (MSB)...

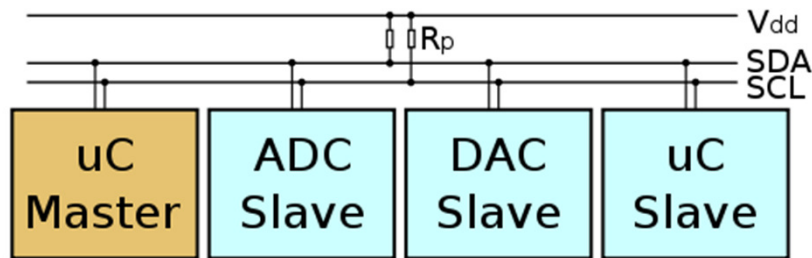


Lastnosti zaporednih vmesnikov

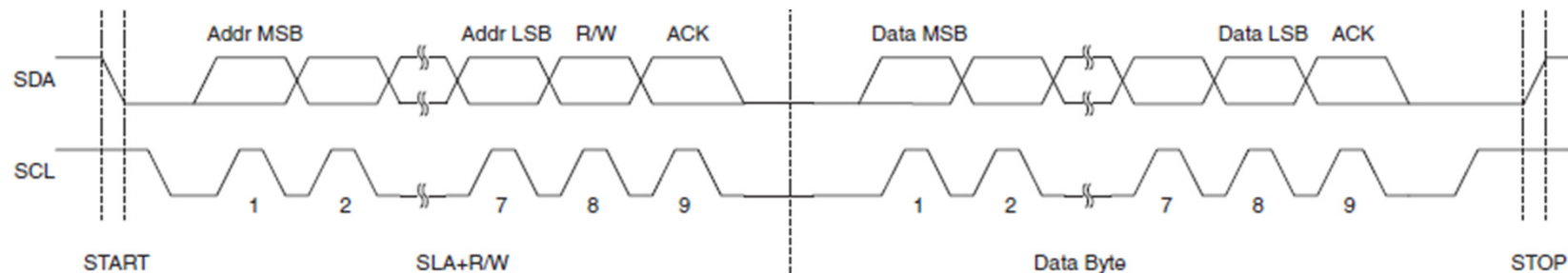
- ▶ Bitna hitrost = frekvenca s katero se prenašajo posamezni biti, določa periodo za posamezni bit ($T = f_{\text{bit}}$)
- ▶ Podatkovni okvir
 - ▶ podatkovne bite vedno spremlja nekaj kontrolnih bitov
 - ▶ podatkovni okvir – start, kontrolna vsota ali pariteta, stop
 - ▶ okvir omogoča usklajevanje 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)

Sinhroni zaporedni vmesnik – I2C

- ▶ Ločen prenos ure (SCL) in podatkov (SDA) po vodilu
- ▶ Enostavni vmesnik z dvema signalnima povezavama
 - ▶ I2C – serijski EEPROM, D/A in A/D pretvorniki, senzorji
 - ▶ Omogoča povezavo več enot na vodilo

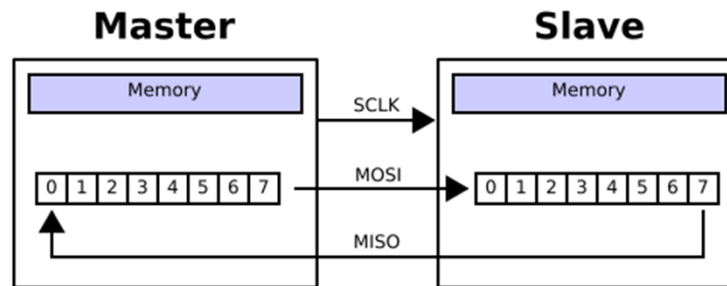


- ▶ Npr. mikroprocesor AVR ima vmesnik TWI (Two Wire Interface) za komunikacijo po I2C protokolu

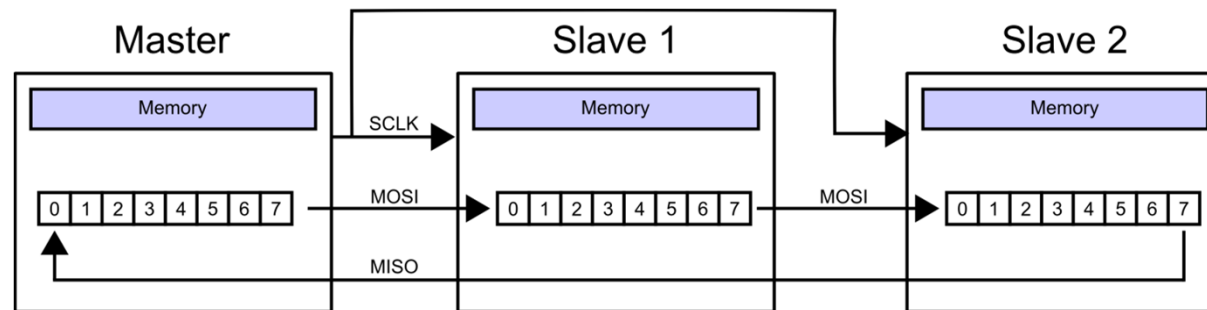


Sinhroni zaporedni vmesnik - SPI

- ▶ Prenos ure (SCLK) ter podatkov iz glavne enote (MOSI) in v glavno enoto (MISO)

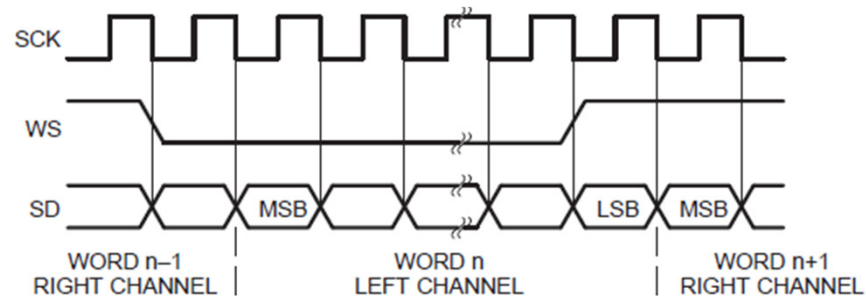


- ▶ Uporaba:
 - ▶ MMC, SD kartice, senzorji, nalaganje mikrokrmilnikov
 - ▶ prenos podatkov od MSB proti LSB
 - ▶ več podrejenih enot lahko vežemo vzporedno ali v verigo



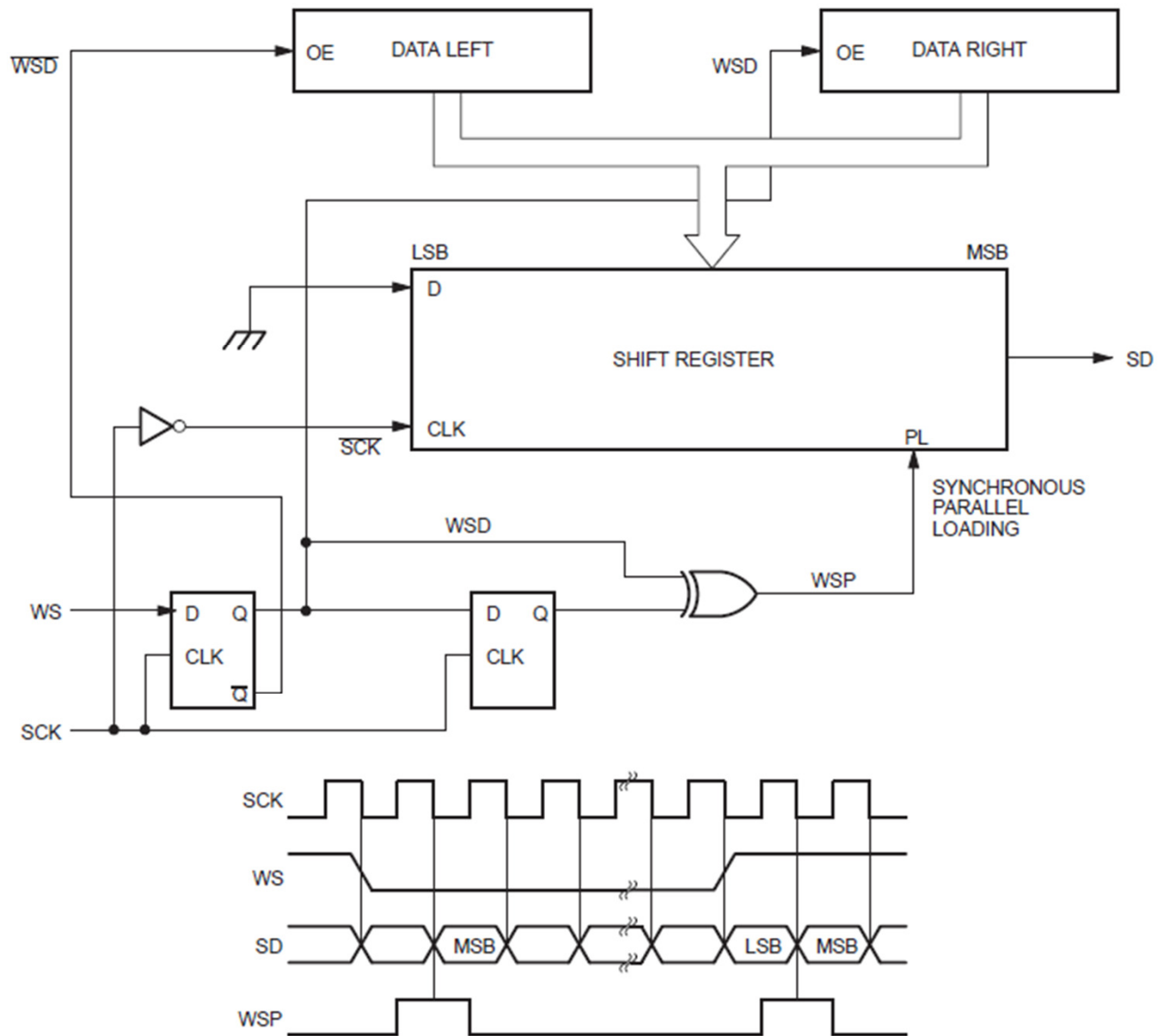
Sinhroni zaporedni vmesnik – I2S

- ▶ Serijski avdio vmesnik
- ▶ Prenos ure (SCK), podatkov (SD) in izbire kanala (WS)



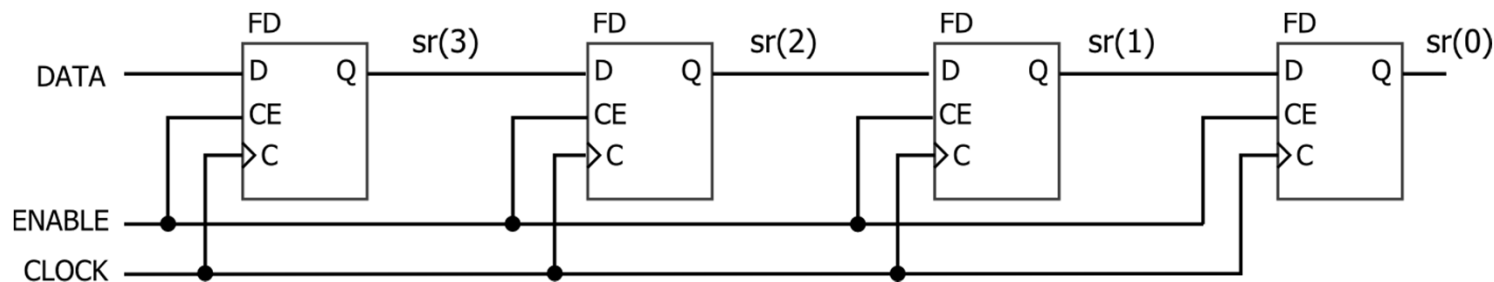
- ▶ Uporaba:
 - ▶ avdio kodeki (AD in DA), digitalni signalni procesorji
 - ▶ enostaven protokol za prenos avdio podatkov
- ▶ Ostali zaporedni protokoli za avdio
 - ▶ S/PDIF – zahteva ekstrakcijo ure iz podatkov
 - ▶ AC97 – prenos 256 bitnih paketov z avdio podatki in kontrolnimi podatki za registre kodeka

Zgradba vezja za oddajnik I2S



Zaporedna pretvorba podatkov v vzporedno

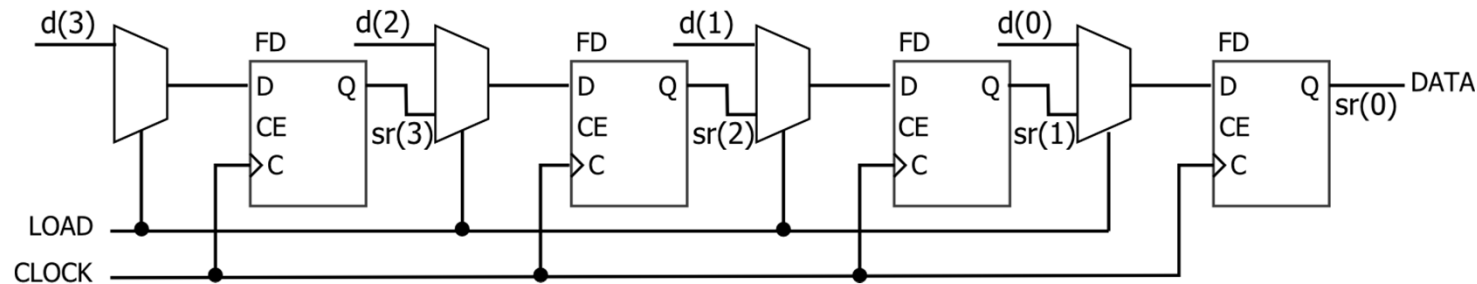
- ▶ Pretvorba s pomikalnim registrom (SIPO)
 - ▶ zaporedno vežemo toliko DFF, kolikor je dolg podatkovni paket
 - ▶ paziti moramo na zaporedje – prvi poslani bit je MSB ali LSB
- ▶ Npr. 4-bitni pomikalni register, prvi bit je LSB
 - ▶ podatki se pomikajo v desno, proti LSB



```
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)

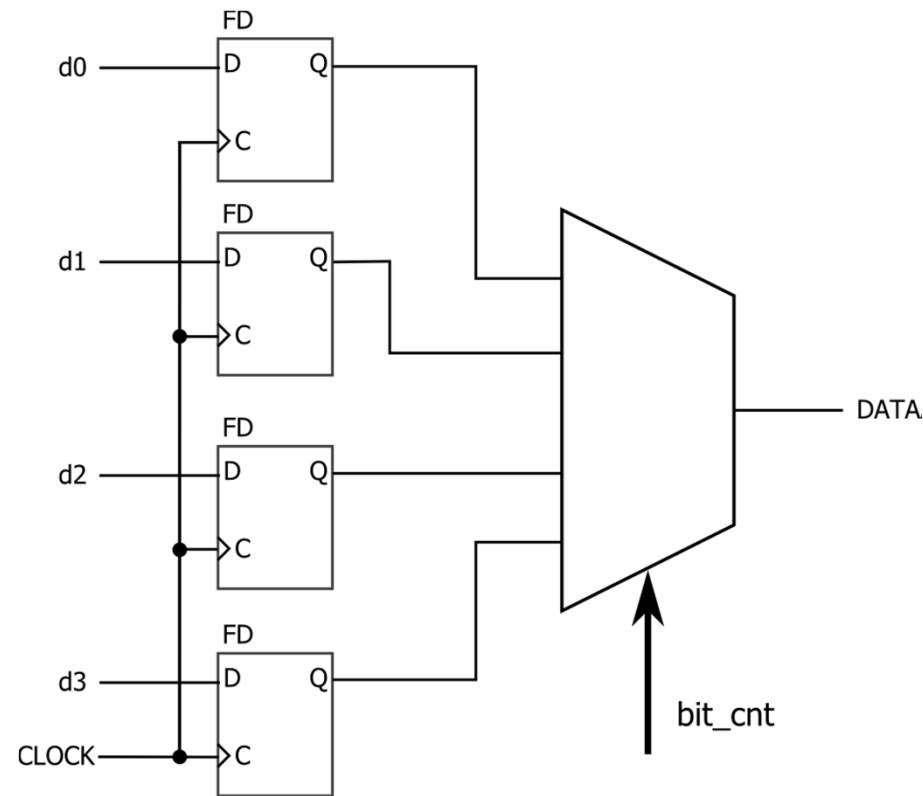
▶ Zaporedni oddajnik s pomikalnim registrom (PISO)



```
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);
```

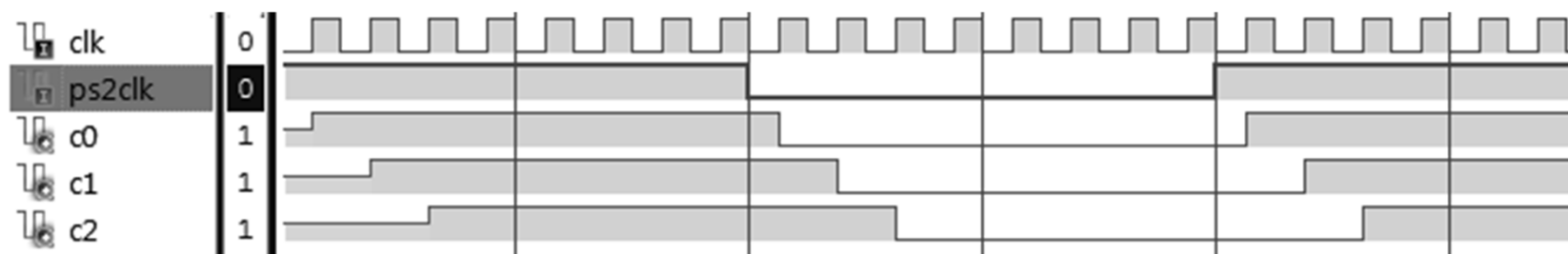
Pretvorba z izbiralnikom

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



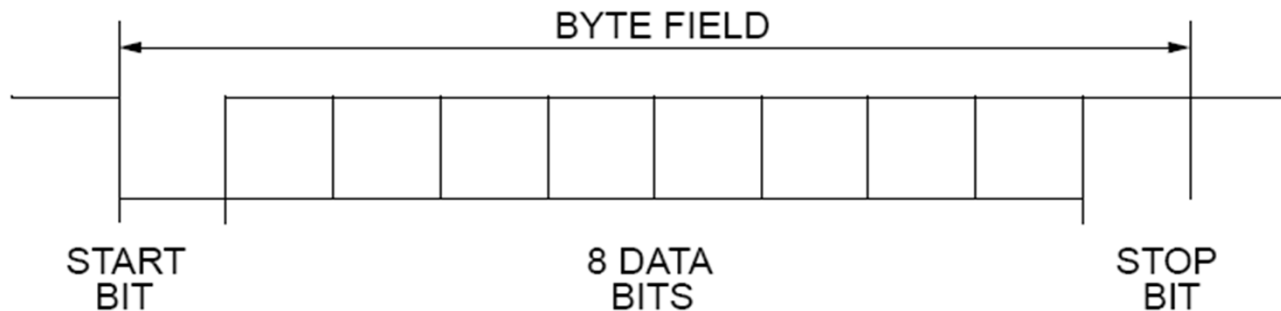
Sinhronizacija sprejetih podatkov

- ▶ 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



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
 - ▶ cenena izvedba vmesnika z mikrokrmilnikom
 - ▶ ne vsebuje robustnega odpravljanja napak
 - ▶ ni uporaben za varnostno kritične aplikacije

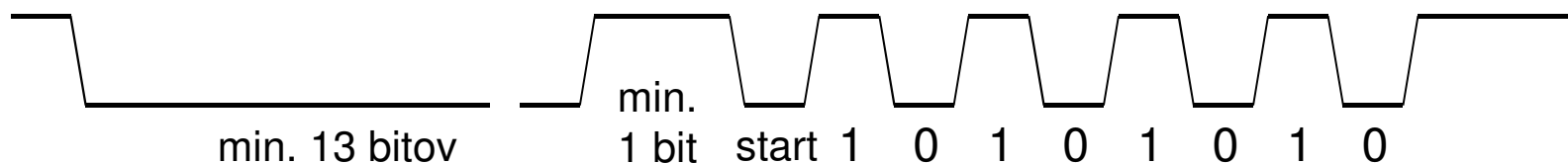


Podrobnosti prenosa podatkov



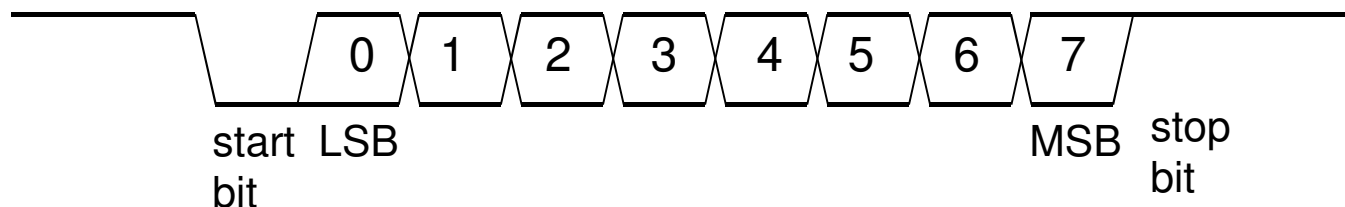
▶ Sinhronizacija sprejemnika

- ▶ 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)

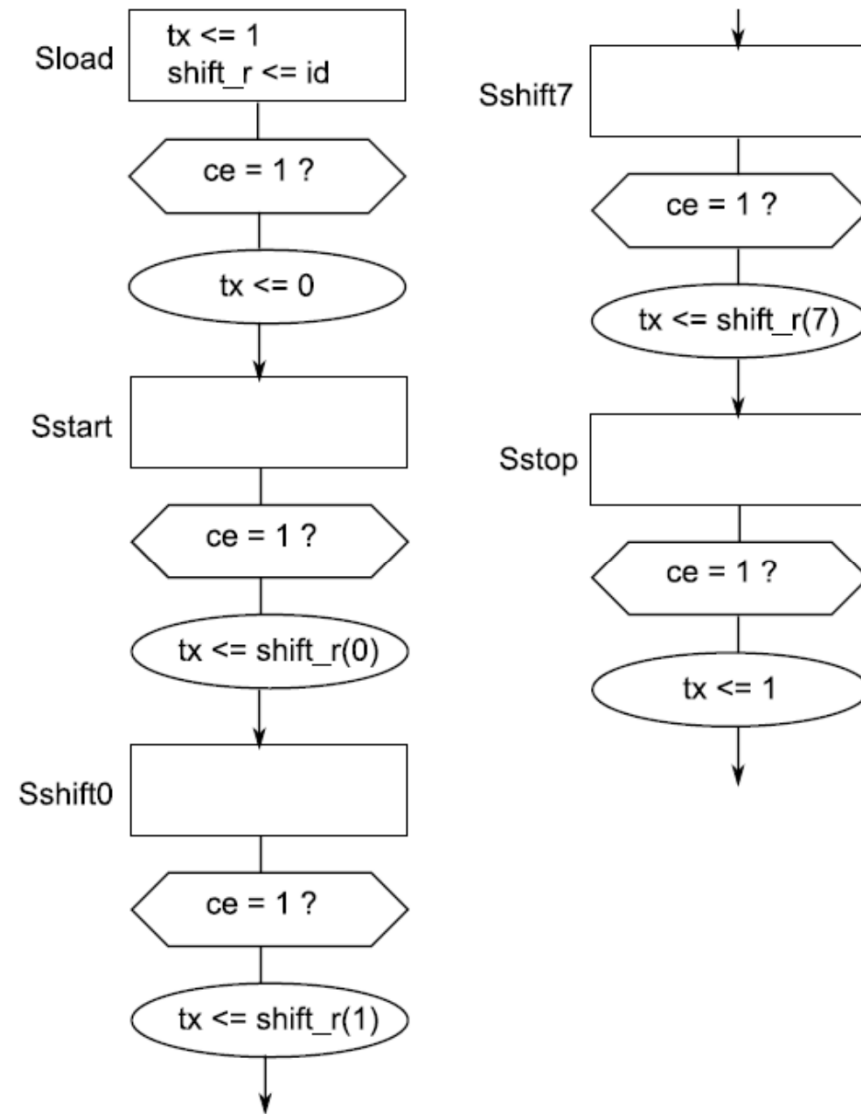


Asinhroni serijski oddajnik

▶ Algoritmični avtomat

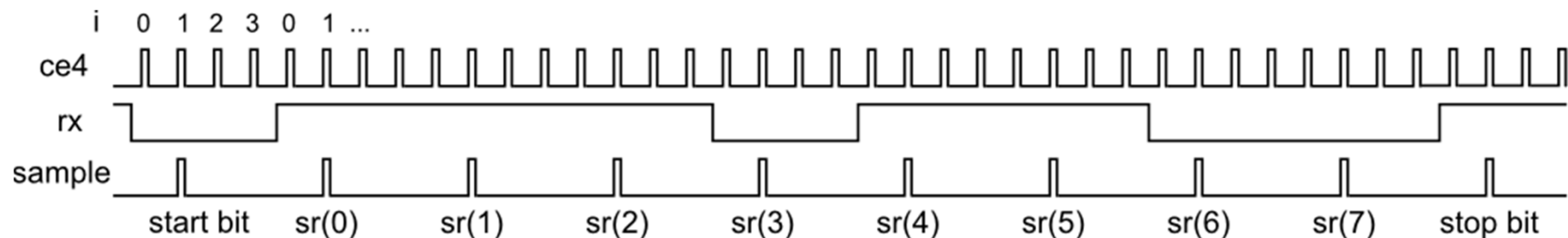
- ▶ nov podatkovni bit se pošlje ob clk in pogoju $ce='1'$
- ▶ s števcem določimo hitrost prenosa, ce je preliv števca
- ▶ npr. $f_{clk} = 1\text{MHz}$, $f_{bit} = 38.4\text{kHz}$ razmerje je 26,0416... zato uporabimo števec po modulu 26 s prelivom:

```
if rising_edge(clk) then
  if st < 25 then
    st <= st + '1'; ce='0'
  else
    st <= "00000"; ce='1'
  ...
```



Asinhroni serijski sprejemnik

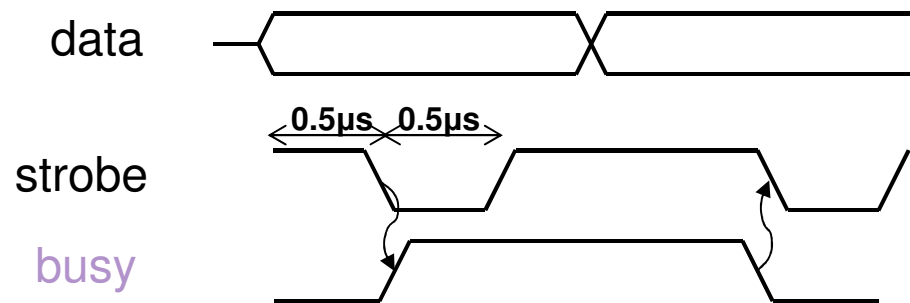
- ▶ Glavna naloga sprejemnika je sinhronizacija paketa
- ▶ Sinhronizacija s prevzorčenjem
 - ▶ vhodni signal vzorčimo z večkratnikom frekvence prenosa
 - ▶ Npr. signal ce4 je 4 x bitne frekvence



- ▶ Delovanje sinhronizacije paketa
 - ▶ ob zaznanem startnem bitu ($rx=0$) postavimo števec $i = 0$
 - ▶ števec se nato povečuje z večkratnikom bitne frekvence
 - ▶ podatke beremo približno na sredini časovnega intervala za posamezni bit

Asinhroni vzporedni prenos podatkov

- ▶ Vodilo Centronics
- ▶ 8-bitni asinhroni prenos podatkov za zunanje naprave
- ▶ Preprost usklajevalni protokol
 - ▶ PC z impulzom (strobe) označi nov podatek
 - ▶ naprava signalizira zasedenost (*busy*)

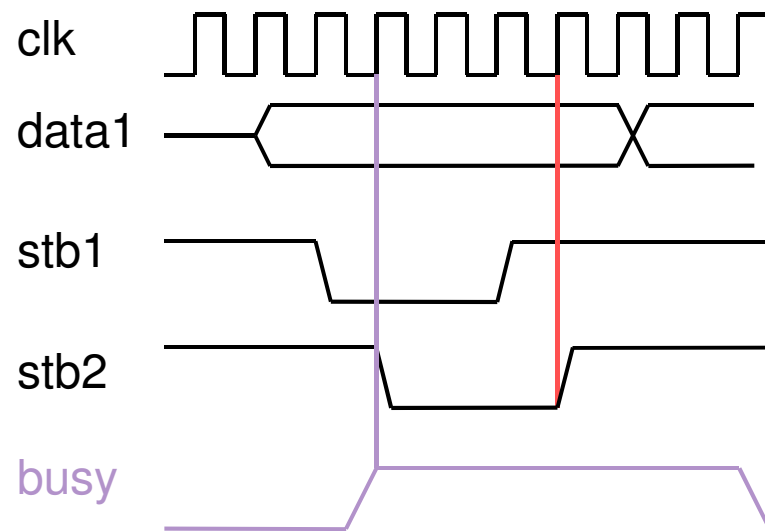


- Dodatni signali
 - prekinitev ob zaključku
 - status naprave (out of paper, error...)



Izvedba vmesnika za vodilo Centronics

- ▶ Podatke in kontrolne signale vzorčimo z lokalno uro
 - ▶ strobe vzorčimo dvakrat za detekcijo fonte!



```
p: process (clk)
begin
  if rising_edge(clk) then
    if stb1='0' and stb2='1' then
      busy <= '1';
    end if;

    if stb1='1' and stb2='0' then
      d <= data1;
    end if;
  end if;
end process;
```

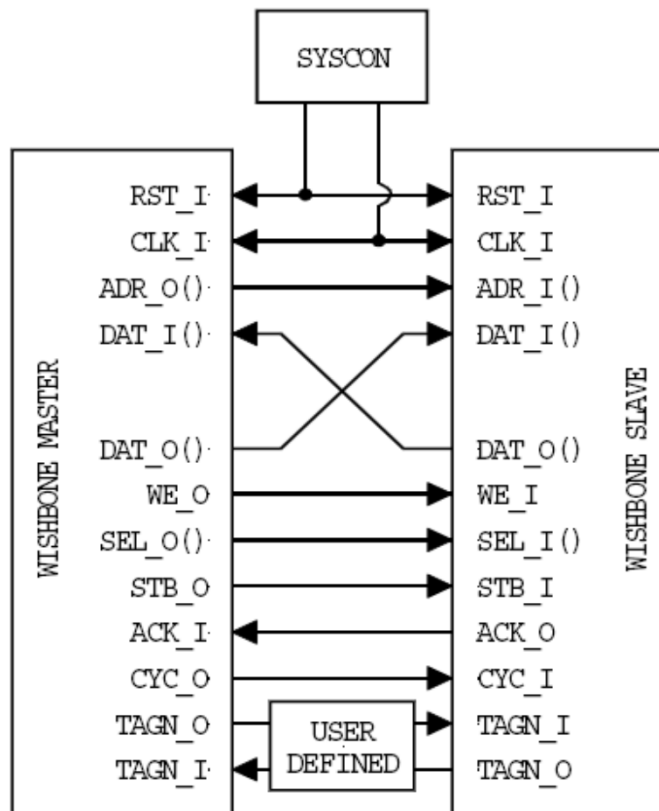
Sinhrono vzporedno vodilo Wishbone

- ▶ odprt standard za vodilo v integriranem vezju
- ▶ enostaven protokol
 - ▶ bralni, pisalni, blokovni prenos in RMW prenos
 - ▶ prenos podatka v enem urinem ciklu
- ▶ različne povezovalne arhitekture
 - ▶ točka s točko, deljeno vodilo, stikalo
 - ▶ sistem gospodar/podrejena enota (master/slave)
- ▶ različne možnosti razširitev
 - ▶ osnovni prenos, registrski prenos
 - ▶ lastni signali (address, data ali cycle tag)



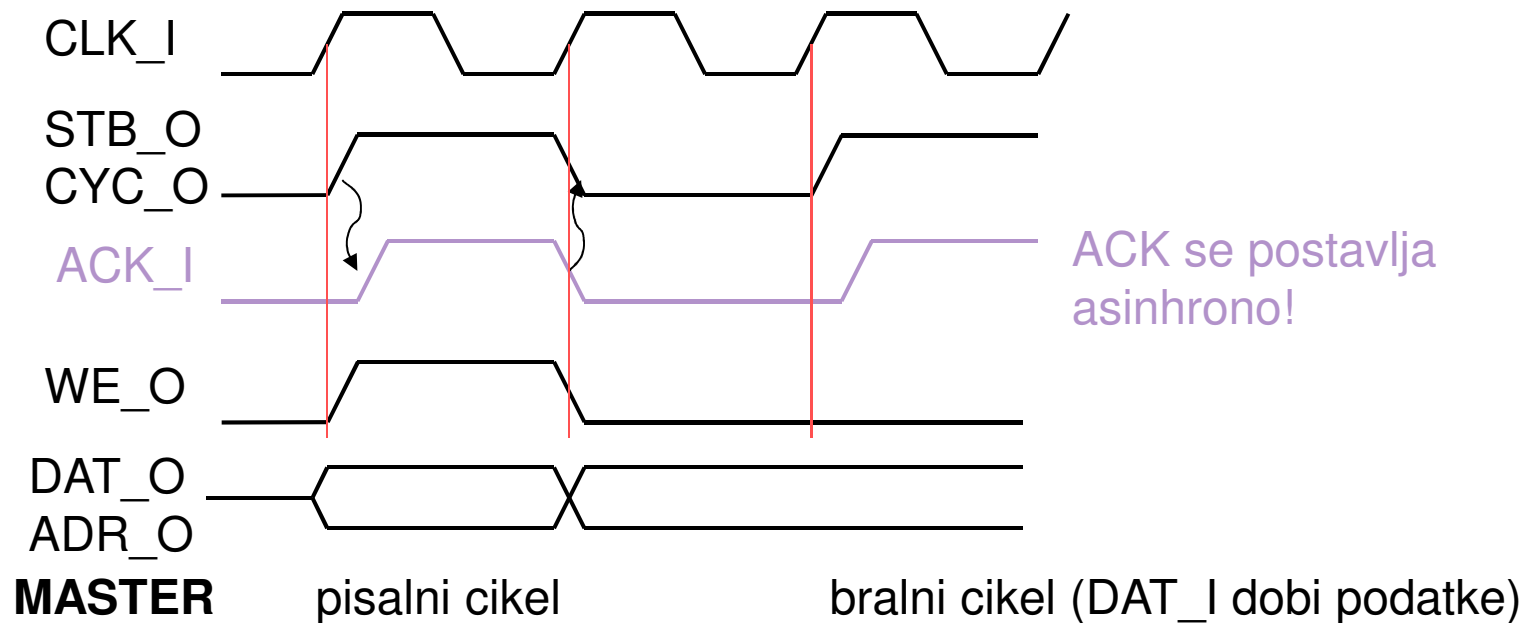
Povezovanje dveh enot

- ▶ Osnovna povezava master/slave



Protokol vodila Wishbone

- ▶ Sinhrono vodilo za povezovanje komponent znotraj integriranega vezja
 - ▶ kontrolne signale gledamo ob fronti ure



Primer podrejene enote (slave)

- ▶ Npr. register na vodilu Wishbone

```
p: process (clk_i, rst_i)
begin
  if rst_i='1' then
    q <= "00000000";
  elsif rising_edge(clk_i) then
    if stb_i='1' and we_i='1' then
      q <= dat_i;
    end if;
  end if;
end process;
ack_o <= stb_i;
```

