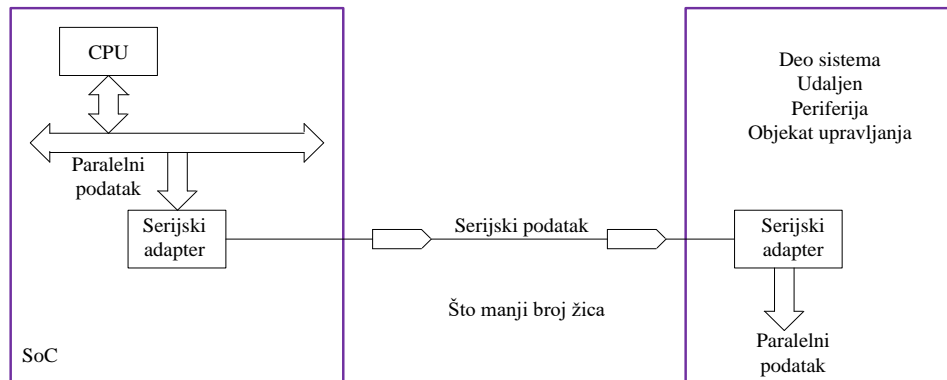


Serijski portovi



1

Osnovna ideja



Simpleks izlazna - Obrnut smer, simpleks ulazna

Uočiti

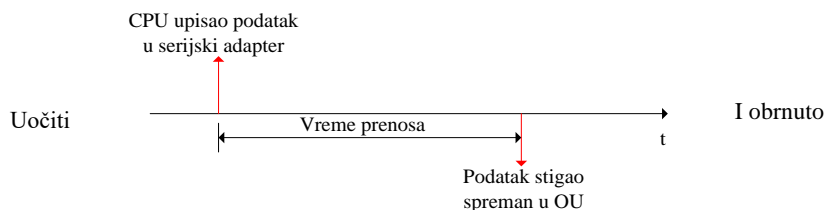
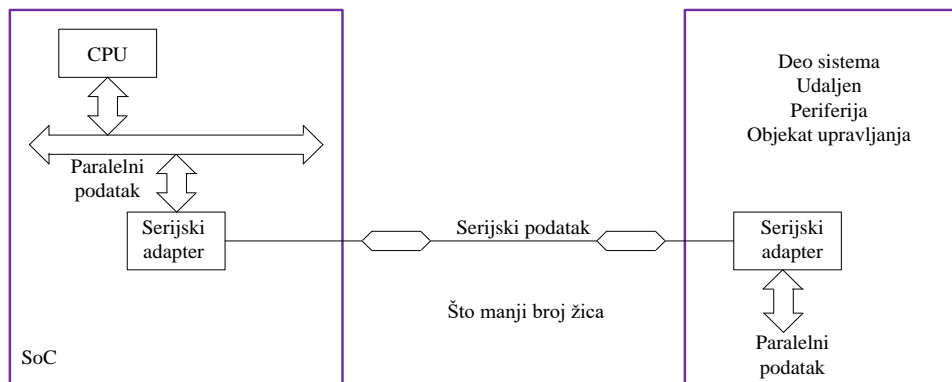
1. OU mora znati kada mu je podatak raspoloživ (treba DS data strobe kao na magistrali)
2. OU mora znati šta je taj podatak, koja informacija je sadržana u njemu (podatak koji se negde upisuje, lokalna adresa, sve zajedno – magistrala)

Imitiraćemo formatom podatka paralelne magistrale



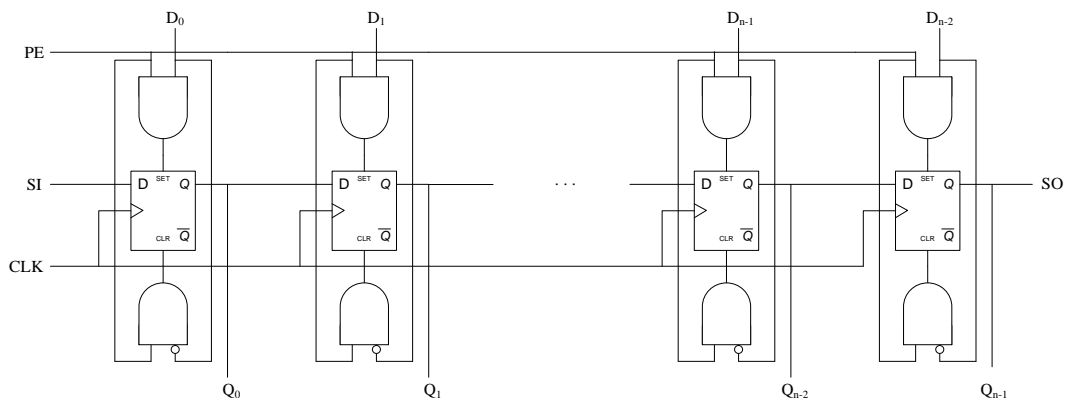
2

Odnosno poludupleks, pundupleks ...



3

Osnova serijskog adaptera – pomerački registar sa mogućnošću paralelnog upisa-čitanja



Svaki taktni signal pomera podatak

SI -> P₀ P₀ -> P₁ P₁ -> P₂ ... P_{n-2} -> P_{n-1} P_{n-1} = SO

PE – Parallel Entry
SI – Serial Input
SO – Serial Output

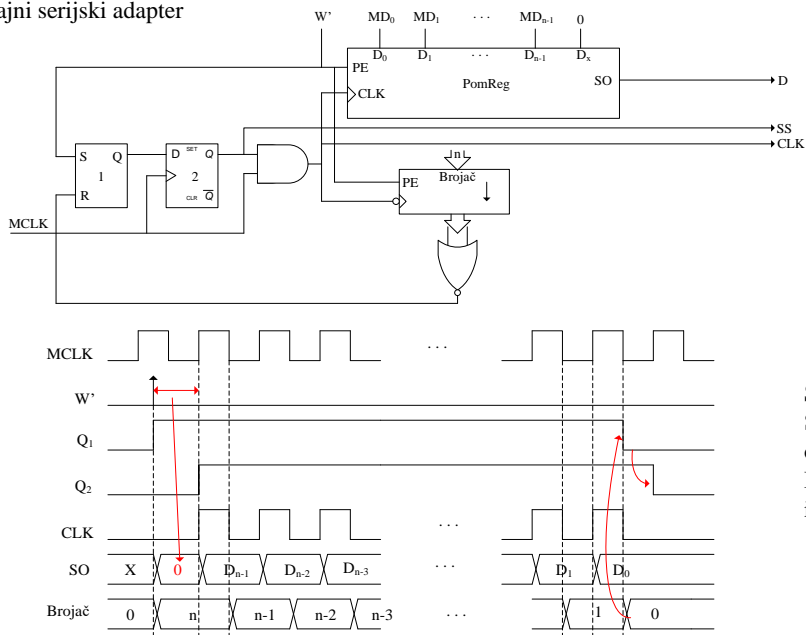
Aktivnim signalom PE moguće upisati paralelno podatak u registar.

Podatak iz registra raspoloživ u paralelnom obliku na izlazima Q₀Q₁ ... Q_{n-2}Q_{n-1}

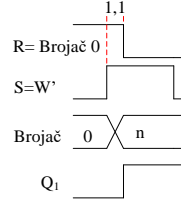


4

Predajni serijski adapter



Na prvi pogled zbunjujuće ali korektno

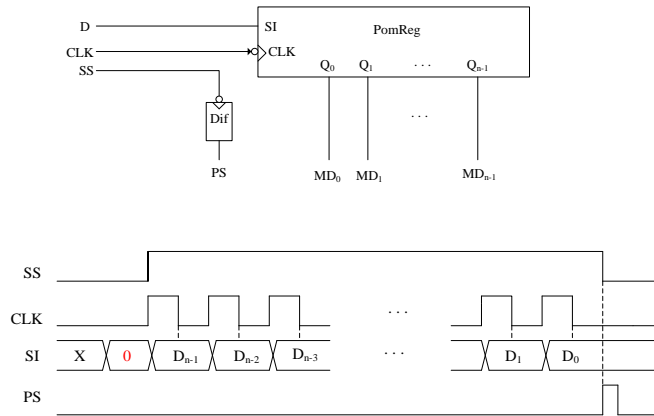


Sme $R=S=1$
Šta će biti na izlazu za vreme ovog stanja zavisi od realizacije. Bitno je šta u vremenu ostaje S ili R posle ovog stanja



5

Prijemni serijski adapter

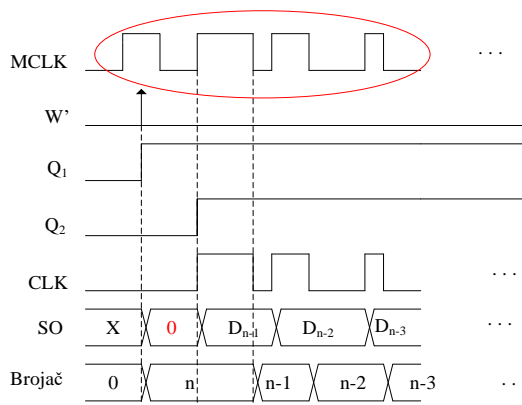


Provera da li je sve stiglo kako treba



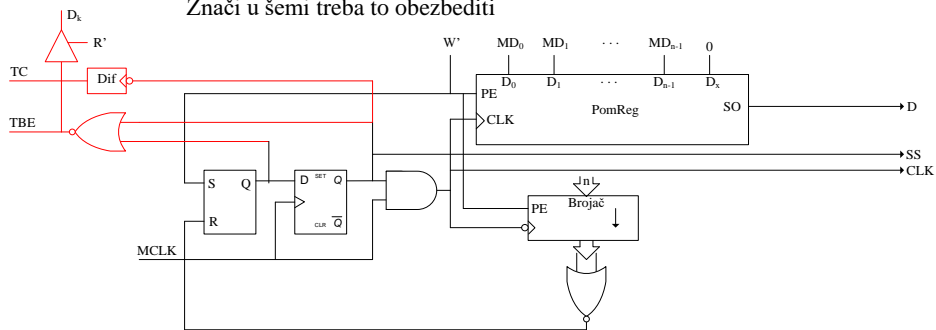
6

Ovakav način prenosa pripada klasi sinhronih serijskih prenosa.
 Prenosi se i takti signal sa kojim je sve sinhronizovano.
 Uočiti da takti signal ne mora a bude simetričan niti da su svi takti ciklusi istog trajanja.
 Bitne su usponske i silazne taktne ivice.



7

Ako je predajnik na strani CPU kako zna da sme da šalje narednu poruku?
 Ili slejv ako je predajnik na njegovoj strani?
 Odgovor: Poliranje i/ili Prekidi
 Znači u šemi treba to obezbediti



TC – Transmit Complete – ivični prekid usponskom ivicom
 TBE – Transmitter Buffer Empty – prekid sa nivoom
 D_k – bit po formatu statusnog registra, sa značenjem da li je predaja u toku $D_k = 0$, odnosno slobodan predajnik $D_k = 1$, na adresi R' - poliranje



8

Protokol

Poliranje: CPU povremeno čita statusni registar i donosi odluku

Prekid TC: Može biti stalno dozvoljen. Prva poruka iz eventualnog niza se šalje van prekidne rutine. Iz glavnog ili nekog drugog dela programa. U okviru prekidne rutine se šalje naredna poruka iz niza dok se ne dodje do kraja. Tada se ne šalje ništa i izlazi se iz prekidne rutine. Pre slanja prve poruke iz niza mora da se proveriti da li je transmitter slobodan.

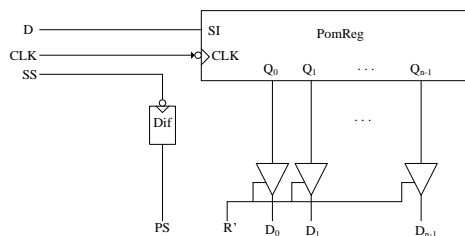
Prekid TBE: Sa nivoom. Ne sme biti stalno dozvoljen.

1. Način - Prva poruka iz eventualnog niza se šalje van prekidne rutine. Iz glavnog ili nekog drugog dela programa. Dozvoli se prekid. U okviru prekidne rutine se šalje naredna poruka iz niza dok se ne dodje do poslednje poruke. Pošalje se poslednja i maskira prekid i izlazi se iz prekidne rutine. Pre slanja prve poruke iz niza mora da se proveriti da li je transmitter slobodan.
2. Način – Iz glavnog ili nekog drugog dela programa se dozvoli prekid. U okviru prekidne rutine se šalje i prva poruka. U okviru prekidne rutine se šalje naredna poruka iz niza dok se ne dodje do poslednje poruke. Pošalje se poslednja i maskira prekid i izlazi se iz prekidne rutine. Ne mora da se proverava stanje da li je transmitter slobodan. Nije slobodan nema prvog prekida. Tek kada se oslobodi i ući će se u prekidnu rutinu.

Fleksibilnost za programera – ali treba biti oprezan



Ako je prijemnik na strani CPU kako zna da je poruka stigla?
Ili slejv ako je prijemnik na njegovoj strani?
Isti odgovori kao i u prethodnom slučaju.



PS – poruka stigla - ivični prekid
U okviru prekidne rutine se na adresi R' čita podatak koji je stigao

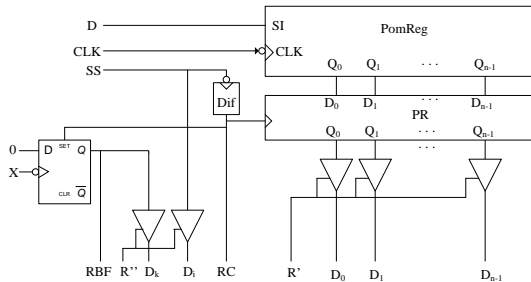
Pitanja

1. Šta ako je kašnjenje prekida „veliko“ pa je počela da stiže naredna poruka pre nego što je CPU ušao u prekidnu rutinu
2. Ako ne želimo da raimo sa prekidima ili želimo prekide sa nivoom



Dodatak

1. Baferisanje podataka
2. Pretvaranje ivičnog u prekid sa nivoom
3. Dodatak za poliranje



Signal X - mogućnosti

1. R' – pročitani podatak resetuje se informacija
2. R'' – pročitani status pa se resetuje (sticky bit)
3. R''' – čitanje u svrhu resetovanja (dummy read)

RBF – Receive Buffer Full - prekid sa nivoom – resetuje se signalom X

RC – Receive Complete – prekid sa ivicom

D_k - statusni bit na adresi R'' sa značenjem ima nova poruka

D_i - statusni bit na adresi R'' sa značenjem stiže nova poruka

Na adresi R' se čita podatak koji je stigao



Protokol

Poliranje: CPU povremeno čita statusni registar i donosi odluku. Čita podatak, resetuje statusni registar.

Prekid RC: Može biti stalno dozvoljen. U okviru prekidne rutine se čita podatak.

Prekid RBF: Sa nivoom. Kako se čitanjem resetuje može biti stalno dozvoljen. U okviru prekidne rutine se čita podatak. Resetuje statusni registar, odnosno zahtev za prekidom.

Fleksibilnost za programera – ali treba biti oprezan
Na primer, ne resetuje se statusni registar a pređe se
na prihvatanje prekida sa nivoom.



Prethodno opisani predajnik i prijemnik pripadaju klasi sinhronih serijskih adaptera.
 Prenos je sinhronizovano sa signalom takta. Sinhroni serijski prenos.



Liče na standardni serijski prenos - SPI

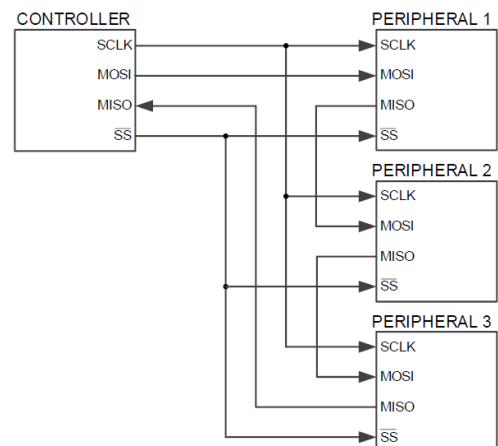
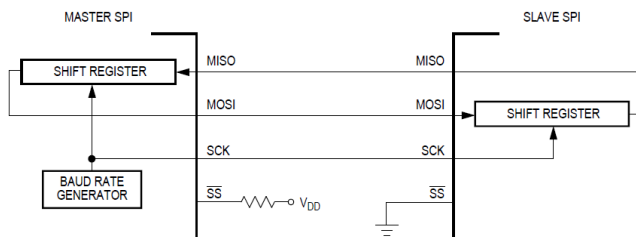
SPI - Serial Peripheral Interface

SS	Slave Select
SCK	Serial Clock
MOSI	Master Output, Slave Input
MISO	Master Input, Slave Output

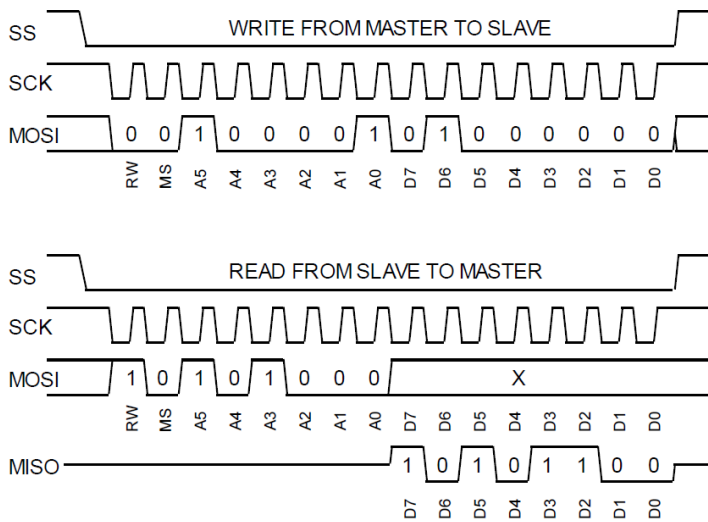


Dosta često se primenjuje u sledećoj konfiguraciji

Ulančavanje



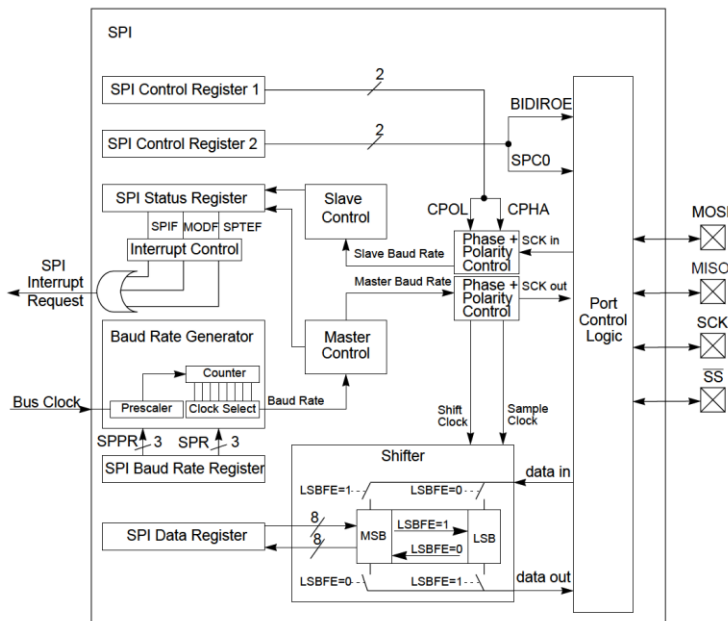
Format podatka – nije standardizovan ali može i ovako



Različite periferije – različiti formati



Jedan SPI primopredajnik



Register Address: \$__3

	Bit 7	6	5	4	3	2	1	Bit 0
R	SPIF	0	SPTEF	MODF	0	0	0	0
W								
Reset:	0	0	1	0	0	0	0	0

Read: anytime

Write: has no effect

 = Reserved

Figure 3-4 SPI Status Register (SPISR)

SPTEF — SPI Transmit Empty Interrupt Flag

If set, this bit indicates that the transmit data register is empty. To clear this bit and place data into the transmit data register, SPISR has to be read with SPTEF=1, followed by a write to SPIDR. Any write to the SPI Data Register without reading SPTEF=1, is effectively ignored.

1 = SPI Data register empty

0 = SPI Data register not empty

SPIF — SPIF Interrupt Flag

This bit is set after a received data byte has been transferred into the SPI Data Register. This bit is cleared by reading the SPISR register (with SPIF set) followed by a read access to the SPI Data Register.

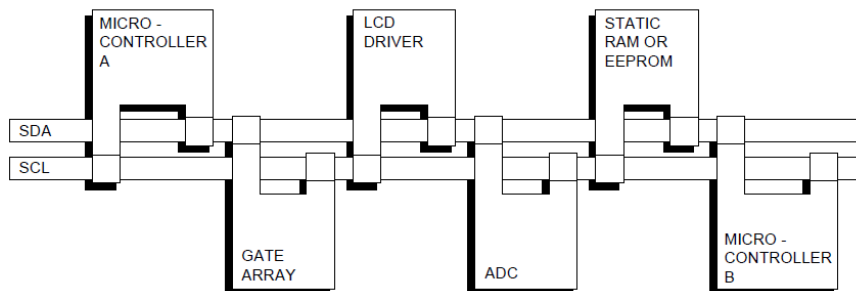
1 = New data copied to SPIDR

0 = Transfer not yet complete



Takođe dosta često I²C sinhrona serijska komunikacija.
Samo dve žice.

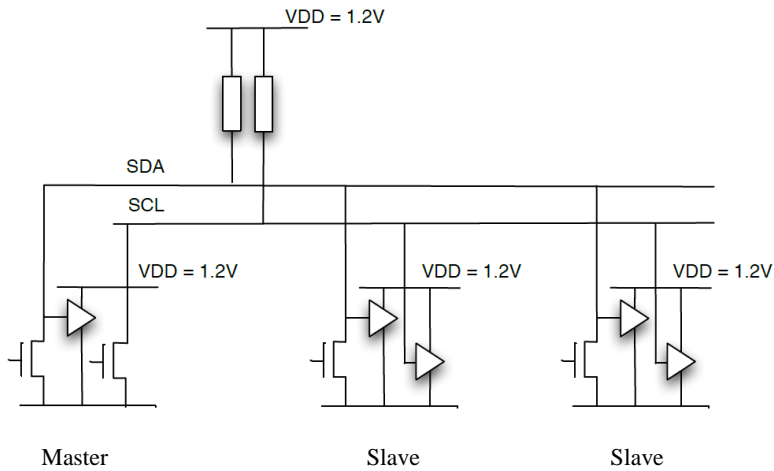
I²C - Inter-Integrated Circuit



SDA – Serial data

SCL – Serial clock

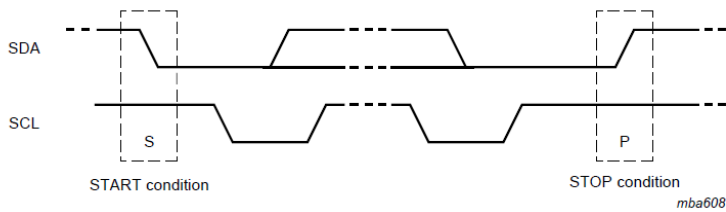




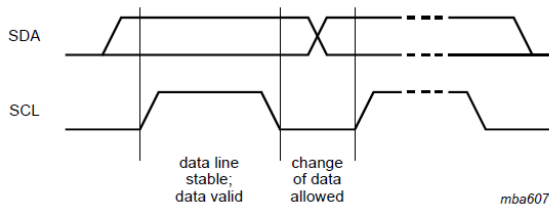
Otvoreni drejn – terminacija
Master generiše SCL



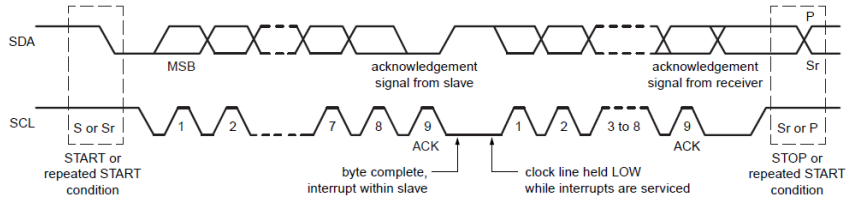
Neaktivno stanje linija – obe logičke jedinice.
Kada počinje i kada se završava poruka.



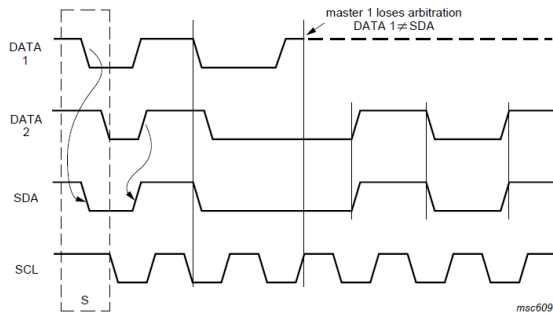
Da bi se ovo detektovalo, i u toku slanja ne bi pogrešno interpretiralo,
promena SDA linije dozvoljena samo kada je SCL na niskom nivou



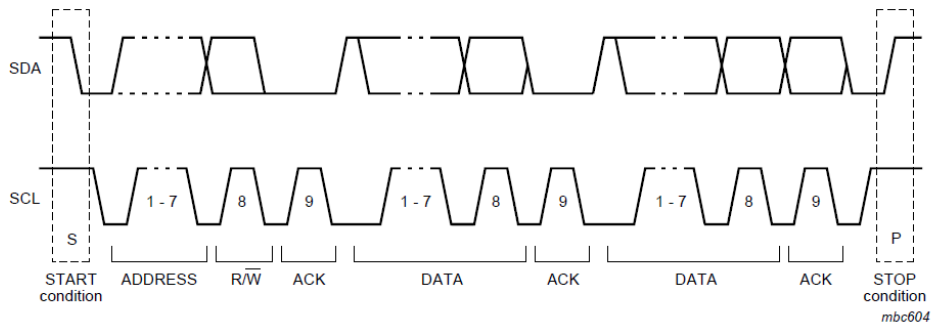
Prenos



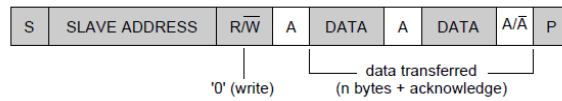
Šta ako dva mastera krenu u slanje? Arbitraža.



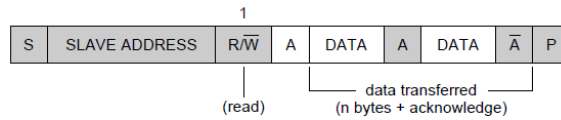
Format podatka



Ciklus upisa



Ciklus čitanja



- from master to slave
- from slave to master
- A = acknowledge (SDA LOW)
- Ā = not acknowledge (SDA HIGH)
- S = START condition
- P = STOP condition



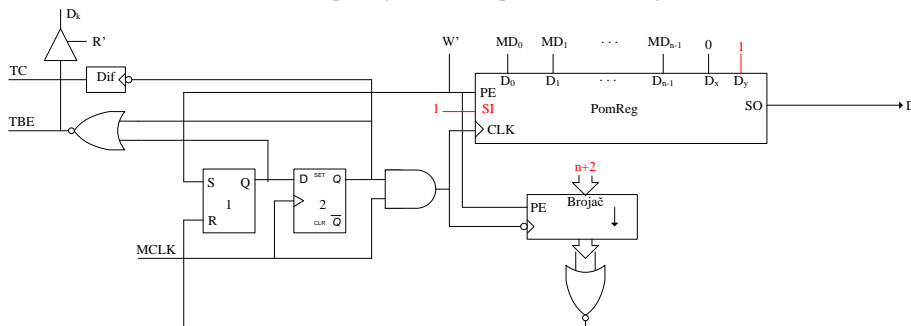
Asinhroni prenos podataka

Ideja: ne prenosi se takt. Što manji broj žica. Jedna.

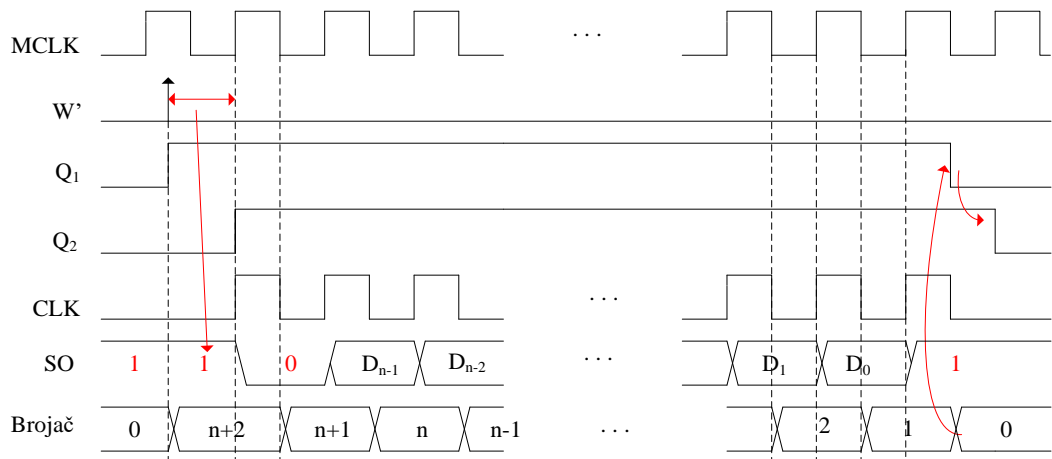
Najbitnije:

1. Uvodi se neaktivno stanje linije. Logička jedinica.
2. Da bi se identifikovao start poruke prvi bit koji se šalje mora biti 0. Start bit.
3. Da bi se linija vratila u neaktivno stanje poslednji bit poruke mora biti 1. Stop bit.
4. Kada se govori o broju bita u prenosu, na primer 8bitni transfer, ne broje se Stop i Start biti.

U samom predajniku nema puno modifikacija.

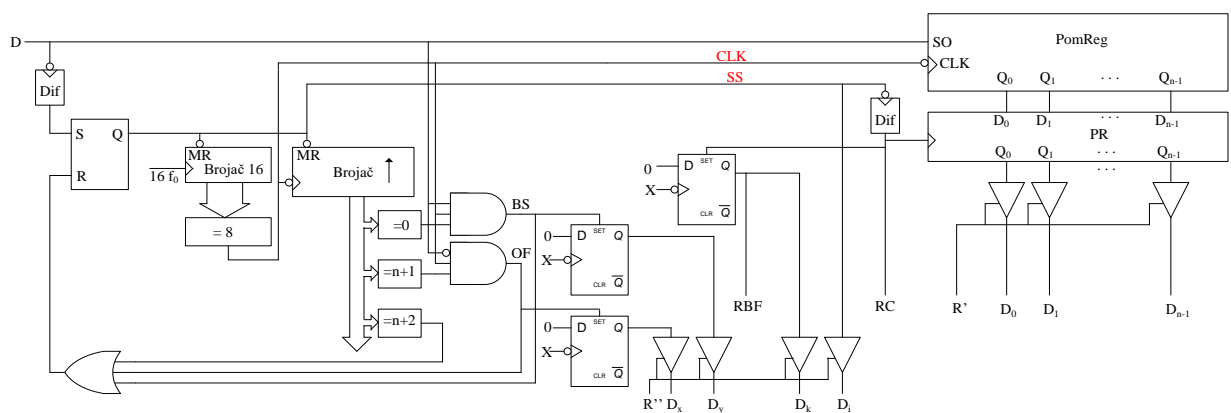


Vremenski dijagram „identičan“



Prijem

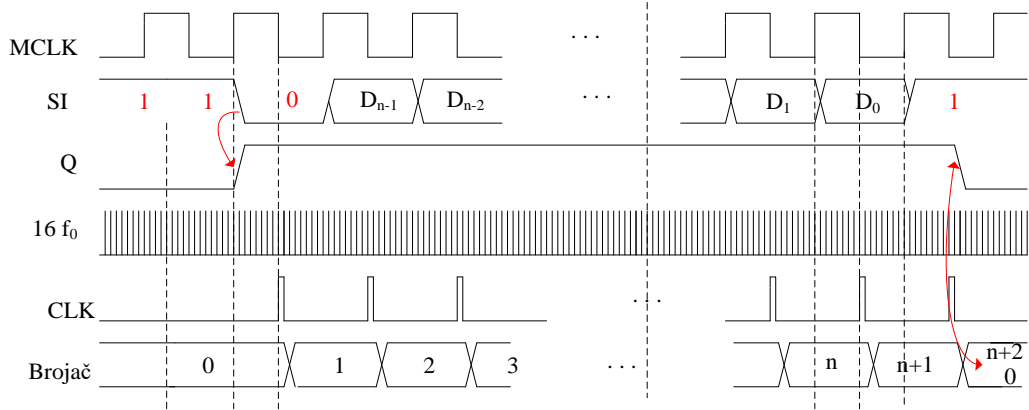
Takt se ne prenosi. Ovaj put mora da bude unapred poznat i fiksni. Ne sme da se menja. Da bi prijemnik znao kada su važeći podaci. Takt za pomerački registar na prijemu se pravi na prijemu. Sinhronizacija prijema. CLK i SS treba lokalno napraviti.



BS – Bad Start
OF – Over Frame

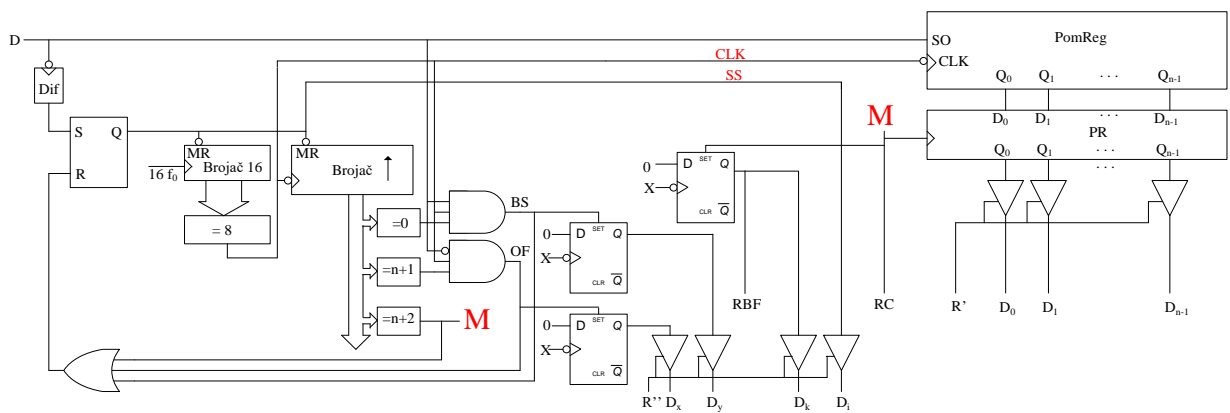


Vremenski dijagram



Modifikacije

Signali BS ili OF mogu i kao prekidi
Signali BS ili OF ne moraju da generišu signal da je poruka stigla.



Kao i mnoge druge

UART – Universal Asynchronous Receiver Transmitter



Zaključak

Bilo da je sinhrona ili asinhrona komunikacija realizacije adaptera su različite i pažljivo ih treba proučiti. Pažnju posvetiti podacima koji se prenose. Format podatka. Imitacija paralelne magistrale.

Na primer: kod UARTa česta adaptacija je mogućnost programiranja brojača koji broji bite poruke. Na taj način je moguće preneti 7, 8 ili 9 bita. Najčešće. Takođe je moguća adaptacija da se automatski u poruku dodaje bit parnosti i na prijemu proverava i generiše greška ako nije ispravan.

USB – Universal Serial BUS

asinhrona master slejv komunikacija – standardizovane poruke

USB4 standard – više hiljada strana teksta 😊

CAN - Controller Area Network

asinhrona multimaster multislejv serijska komunikacija – standardizovane poruke

Najviše u automotivu

- Pre CAN:** Car ECUs relied on complex point-to-point wiring
 - 1986:** [Bosch](#) developed the CAN protocol as a solution
 - 1991:** Bosch published CAN 2.0 (CAN 2.0A: 11 bit, 2.0B: 29 bit)
 - 1993:** CAN is adopted as international standard (ISO 11898)
 - 2003:** ISO 11898 becomes a standard series
 - 2012:** Bosch released the [CAN FD](#) 1.0 (flexible data rate)
 - 2015:** The CAN FD protocol is standardized (ISO 11898-1)
 - 2016:** The physical CAN layer for data-rates up to 5 Mbit/s standardized in ISO 11898-2
- Today, CAN is standard in automotives ([cars](#), [trucks](#), buses, tractors, ...), [ships](#), planes, [EV batteries](#), [machinery](#) and more.

