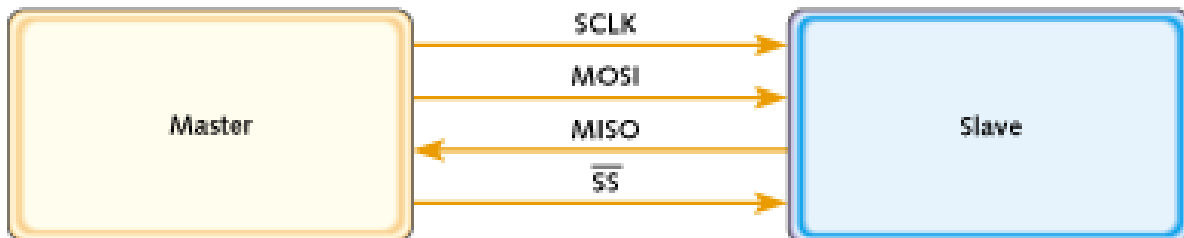


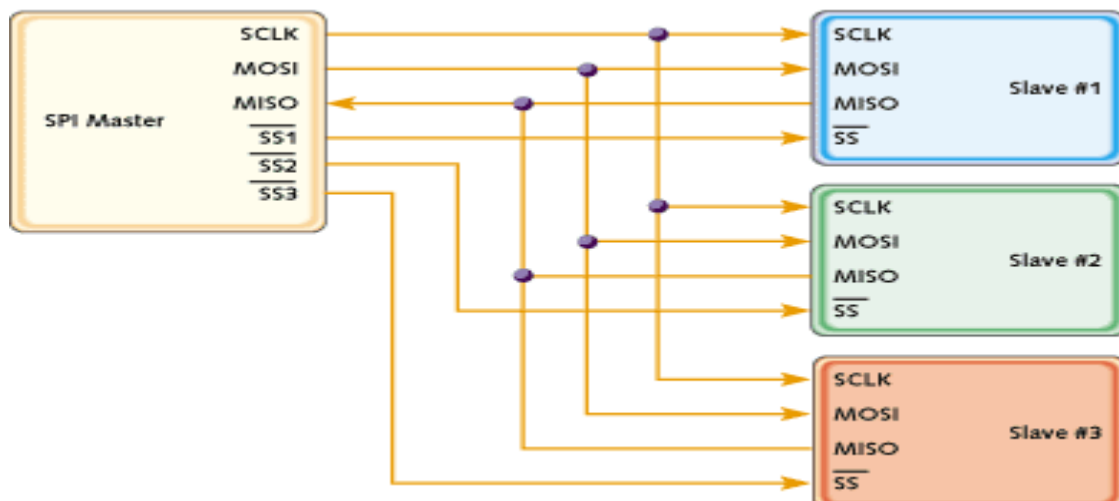
## Sihrona serijska komunikacija

SPI (*“The Serial Peripheral Interface Bus”*) predstavlja vid sinhronne serijske komunikacije koju je razvila Motorola. SPI je full duplex komunikacija koja se ostvaruje između master-slave uređaja. Za ovaj vid komunikacije je potrebno četiri fizičke linije SCK (Serial **C**Lock), MOSI (Master **O**ut Slave **I**n), MISO (Master **I**n Slave **O**ut) i CS (Chip Select).



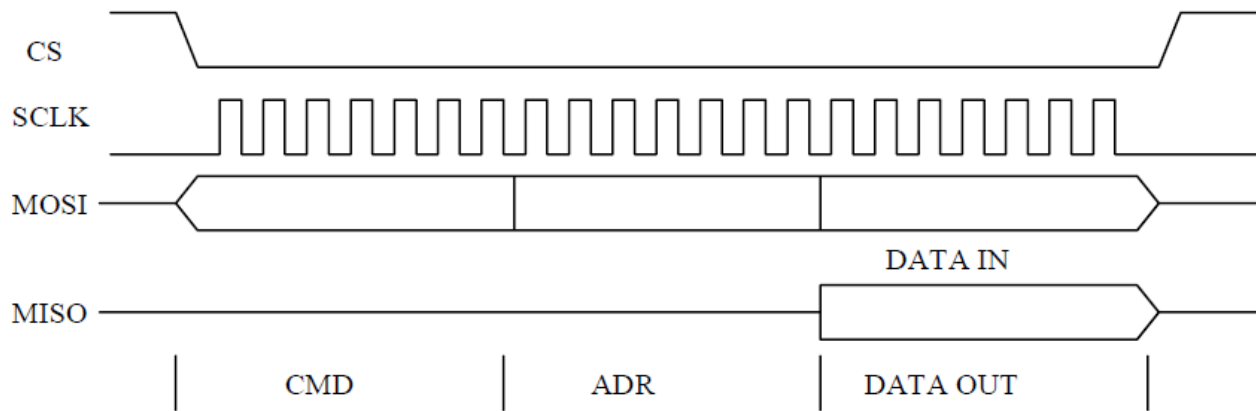
Slika 1.1. Povezivanje jednog master i jednog slave uređaja

Moguće je povezati i više slave uređaja ali u tom slučaju je potrebno da na strani mastera postoji CS (chip select) za svaki slave uređaj, što je prikazano na slici 1.2.



Slika 1.2. Povezivanje jednog master i tri slave uređaja

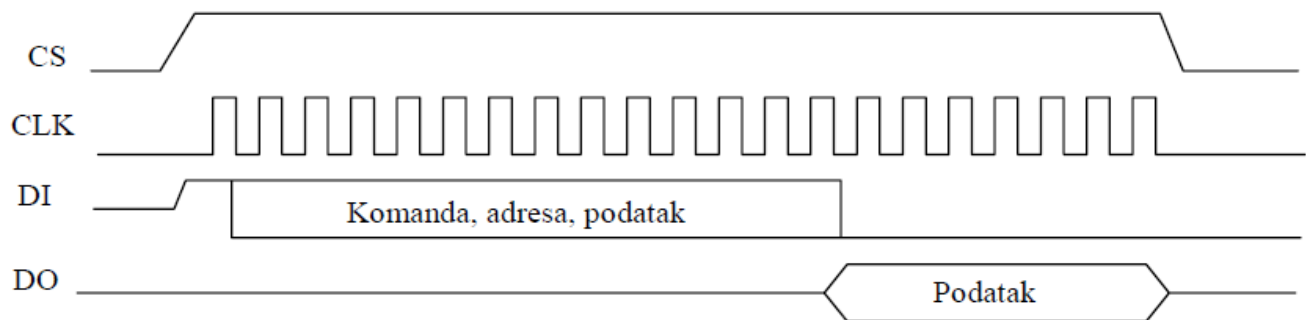
Komunikaciju inicira master uređaj tako što spusti CS signal na logičku nulu, a zatim sledi prenos podataka preko MOSI linije. Ukoliko master uređaj ne očekuje odgovor od periferije onda će odmah nakon poslatog podatka signal CS vratiti na logičku jedinicu, a ukoliko periferija treba da odgovori ona će to učiniti preko MISO linije pa će nakon toga master uređaj CS postaviti na jedinicu. Prenos podataka preko MOSI i MISO linija je sinhron sa taktom SCLK. Na slici 1.3 je prikazan primer komunikacije sa memorijom. CMD je komanda, a ADR je adresa sa koje se čita (upisuje). Ukoliko se upisuje u memoriju tada će master uređaj zanemariti ono što dolazi kao DATA OUT preko MISO linije, dok ako je upitanju čitanje iz memorije tada će memorija zanemariti DATA IN paket sa MOSI linije. Bitna napomena je da CMD, ADR, DATA IN i DATA OUT su u paketima po 8 bita.



*Slika 1.3 Primer SPI komunikacije*

### MicroWire serijska komunikacija

MicroWire komunikacija je vrlo slična SPI komunikaciji i ovde se koriste četiri linije CS (Chip Select), CLK (CLock), DI (Data In) i DO (Data Out). Linije DI i DO se posmatraju sa strane periferije tako da je ovde DI ulaz u periferiju, a DO izlaz iz periferije.



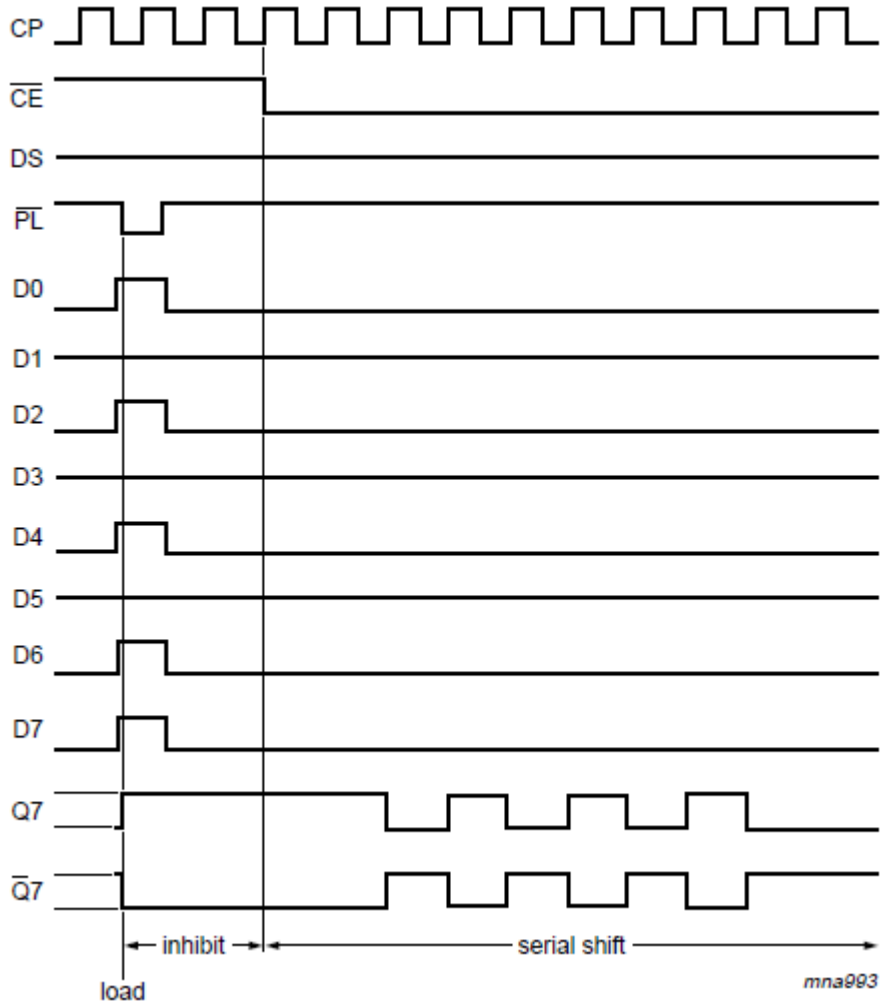
*Slika 1.4 Primer MicroWire komunikacije*

Za razliku od SPI komunikacije CS je ovde aktivan na jedinici. Komunikacija započinje na rastuću ivicu CLK pod uslovom da su CS i DI na logičkoj jedinici. Posle ovoga sinhrono sa taktom se prenose komanda, adresa i podatak preko DI linije. Ako ima potrebe da periferija odgovori onda se pošalje podatak preko DO linije. Posle završene komunikacije master spušta CS na logičku nulu. Prilikom ovog prenosa nije potrebno da paketi budu u dužini od 8 bita.

### Primer rada serijskog EEPROM-a

Možemo da iskoristimo kolo 74HC595 koje je serial in/ parallel out da bi proširili broj izlaznih linija. Ovde se serijski pošalje bajt koji je onda na izlazu kola dostupan u paralelnom obliku. Komunikacija može da bude SPI gde je SCK povezan na SH\_CP, MOSI bi bio povezan na DS, dok bi CS bio povezan na ST\_CP.

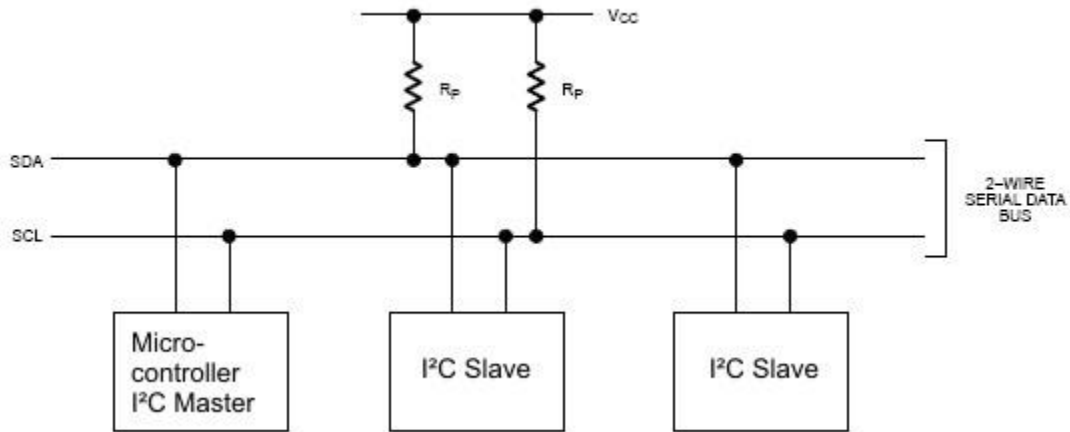
Ako iskoristimo kolo 74HC165 kao parallel in/serial out možemo proširiti broj ulaznih linija. Tako što bi u shift registar paralelno upisali bajt (D0-D7), a zatim taj bajt serijski pročitali. Serijska komunikacija može da bude SPI gde bi SCK bio CP signal, CS bi bio CE i MISO bi bio Q7. MOSI bi mogao da bude DS ali ovde serijski ništa nećemo upisati tako da on nije potreban. Kod SPI komunikacije se prvo šalje komanda i adresa, ali to ovde nije potrebno jer se uvek samo čita i to samo iz jednog shift registra.



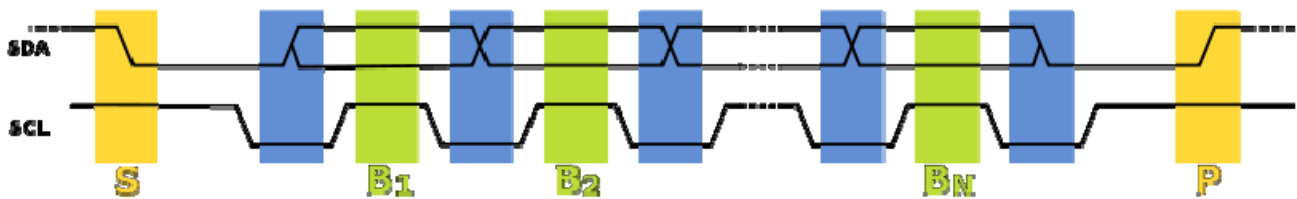
Slika . 74HC165

## I2C komunikacija

I2C („*Inter-Integrated Circuit*”) komunikaciju je razvio Philips. Ovaj vid komunikacije se koristi za mala rastojanja, najčešće za povezivanje integriranih kola na štanpanoj pločici. Za ovaj protokol su potrebne dve fizičke linije tipa open drejn. Ove linije se nazivaju SCL (Serial CLock) i SDA (Serial DAta). Na I2C magistralu se mogu vezati 112 kola, a njima se pristupa preko 7-bitne adrese.



Slika . Način povezivanja uređaja na I2C magistralu



Slika . Primer I2C komunikacije

Komunikacija započinje start bitom (S) kada se SDA linija sa jedinice obori na nulu dok je SCL još na logičkoj jedinici. Zatim dolazi do slanja podataka preko SDA linije tako što se bit na SDA liniji postavlja dok je SCL na nuli (plavo), a sepmplije se kada je SCL na jedinici (zeleno). Kada se završi prenos podataka tada se šalje stop bit (P) tako što se SDA postavi sa nule na jedinicu dok je SCL na jedinici.

Ukoliko se na istoj magistrali nalaze više master uređaja onda se mora voditi računa o koliziji, odnosno situaciji kada dva ili više mastera pokušava da istovremeno pošalje podatak. U ovom slučaju master uređaj posle postavljenog bita mora i da pročita da li se zaista na magistrali nalazi postavljeni bit.

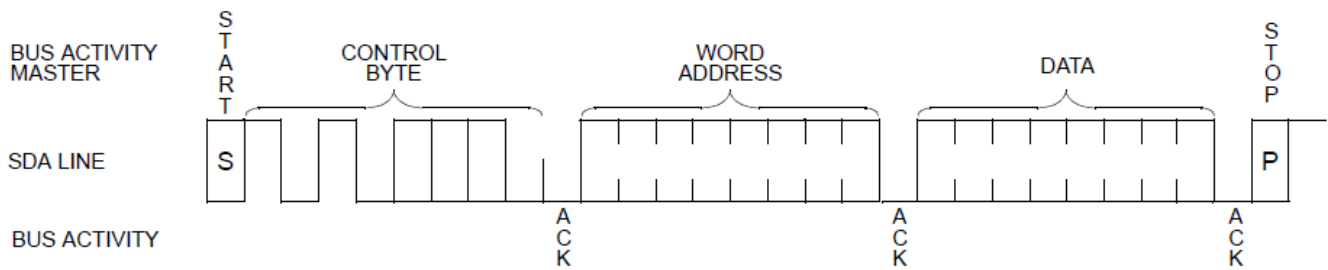
### Primer I2C komunikacije

Prvo će biti obrađen primer rada sa memorijom 24C04A. Kod ove memorije u kontrolnom bajtu gornja četiri bita su rezervisana dok donja četiri služe za adresu i komandu za čitanje ili upis.

1	0	1	0	A2	A1	A0	R/W
---	---	---	---	----	----	----	-----

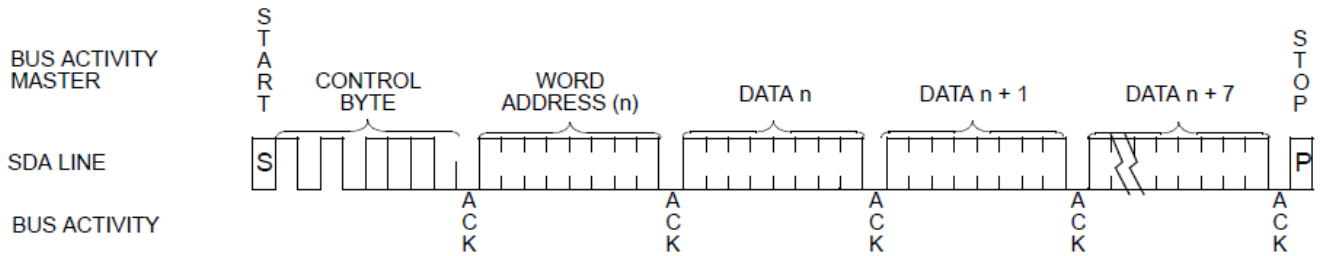
Pošto poseduje samo tri bita za adresu moguće je samo 8 ovakvih memorija povezati na istu magistralu.

Prilikom upisa prvo ide start bit zatim kontrolni bajt nakon toga sledi odgovor prijemnika (ACK – potvrda prijema), nakon toga sledi adresa na koju treba upisati podatak zatim sledi ACK posle koga se šalje podatak i posle ACK ide stop bit.



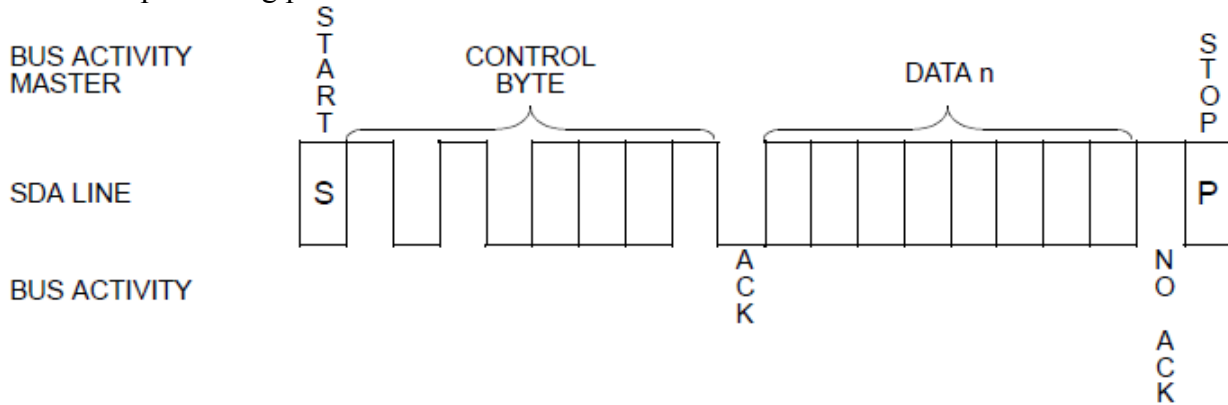
Slika . Dijagram upisa bajta

Razlika između slanja jednog bajta i niza bajtova je u tome što se posle ACK za prvi bajt nastavlja komunikacija



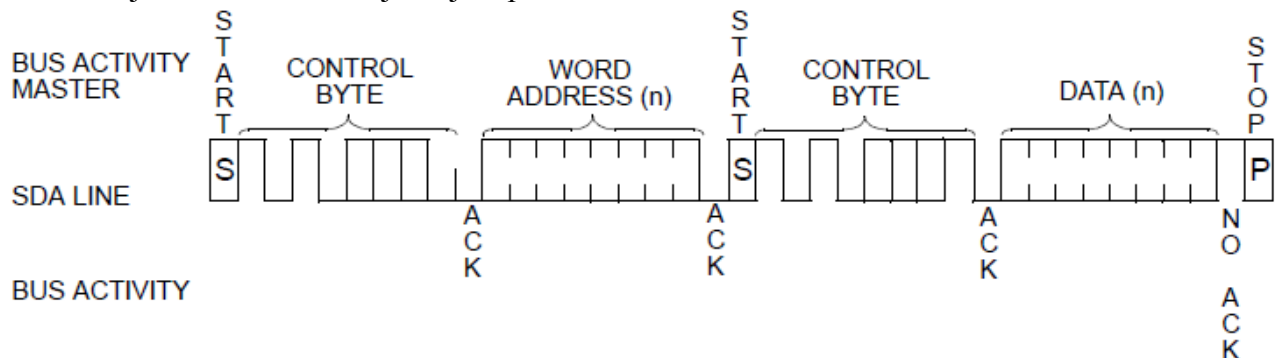
Slika . Dijagram upisa niza bajtova

Na slici je prikazana procedura čitanja ali sa početne adrese u memoriji, pa zbog toga nema adrese i ovde nakon pročitane podatka ne sledi ACK.



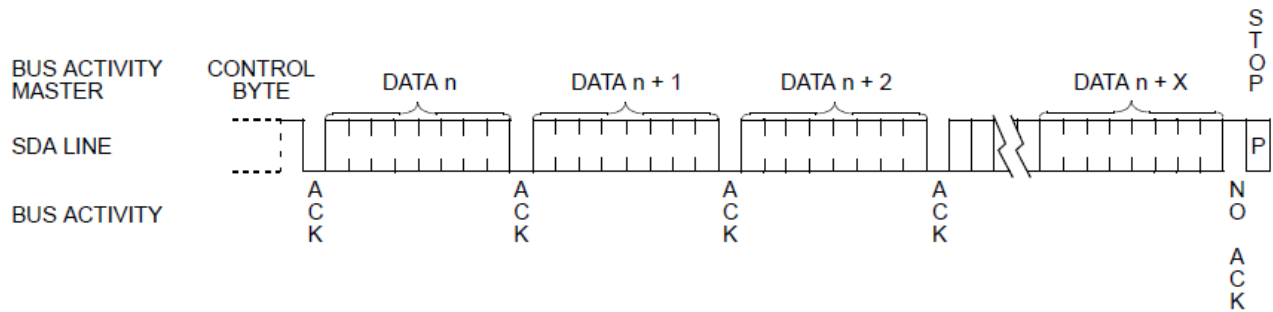
Slika . Čitanje bajta

Kod čitanja sa proizvoljne adrese se prvo upiše adresa sa koje se čita pa se ponovo pošalje start bit i kontrolni bajt i tek onda se dobija željeni podatak.



*Slika . Čitanje bajta sa adrese*

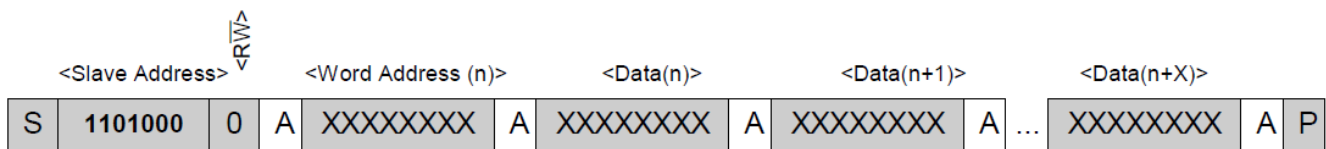
Kod čitanja niza bajtova sa proizvoljne adrese procedura je ista samo što se nastavlja čitanje postavljanjem ACK osim na poslednjem.



*Slika . Čitanje niza bajtova*

**Primer rada sa RTC (Real Time Clock) DS1307**

Kod komunikacije sa RTC procedura je ista kao i u slučaju memorije stom razlikom da je kontrolni bajt, odnosno ovde nazvan slave address nešto drugačijeg oblika.

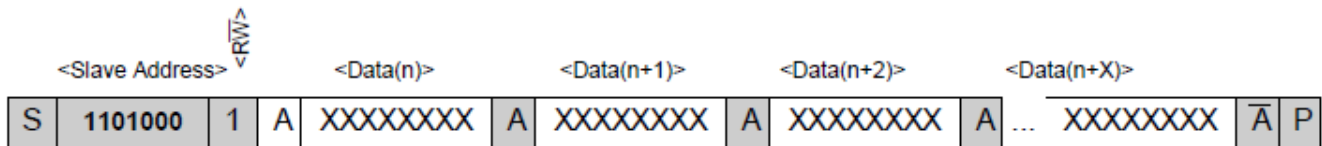


S - Start  
 A - Acknowledge (ACK)  
 P - Stop

Master to slave  
 Slave to master

DATA TRANSFERRED  
 (X+1 BYTES + ACKNOWLEDGE)

*Slika . Dijagram upisa*



S - Start  
 A - Acknowledge (ACK)  
 P - Stop  
 $\bar{A}$  - Not Acknowledge (NACK)

Master to slave  
 Slave to master

DATA TRANSFERRED  
 (X+1 BYTES + ACKNOWLEDGE); NOTE: LAST DATA BYTE IS  
 FOLLOWED BY A NOT ACKNOWLEDGE ( $\bar{A}$ ) SIGNAL)

*Slika . Dijagram čitanja*