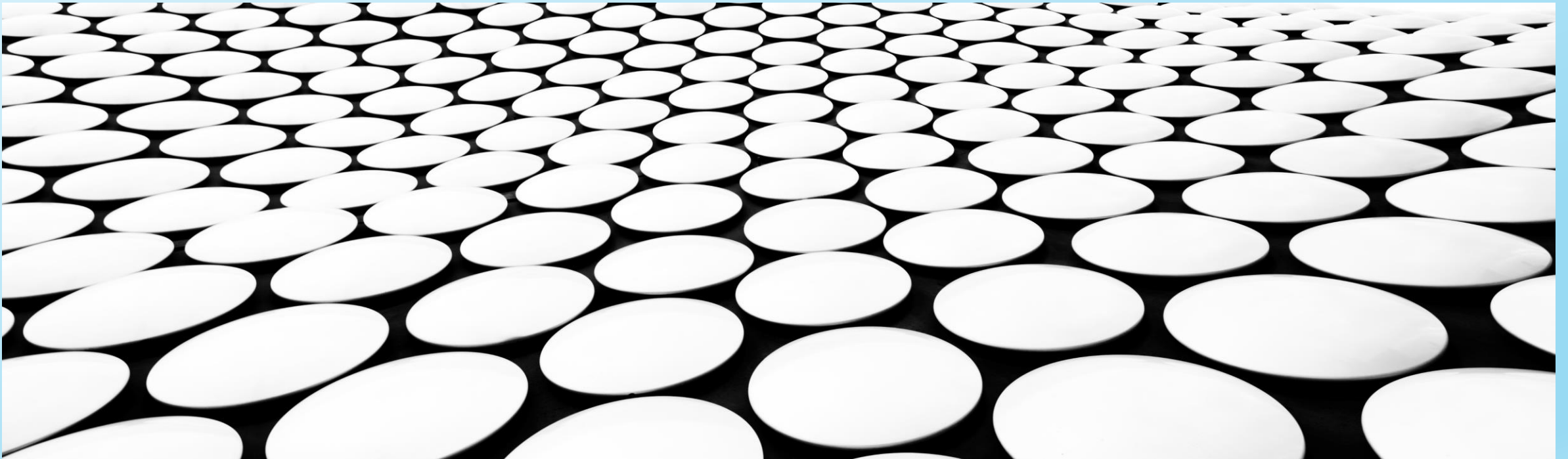


---

# ARHITECTURA SISTEMELOR DE CALCUL

UB, FMI, CTI, ANUL III, 2022-2023



**MAGISTRALE I2C**

I2C este o magistrală serială, sincronă, multi-*master*, multi-*slave*, cu un singur capăt

Este o magistrală de bandă îngustă și de distanță mică.

A fost inventată la Philips Semiconductor (acum NXP Semiconductors).

Este de obicei folosit pentru atașarea circuitelor integrate periferice cu viteză redusă la procesoare și microcontrolere

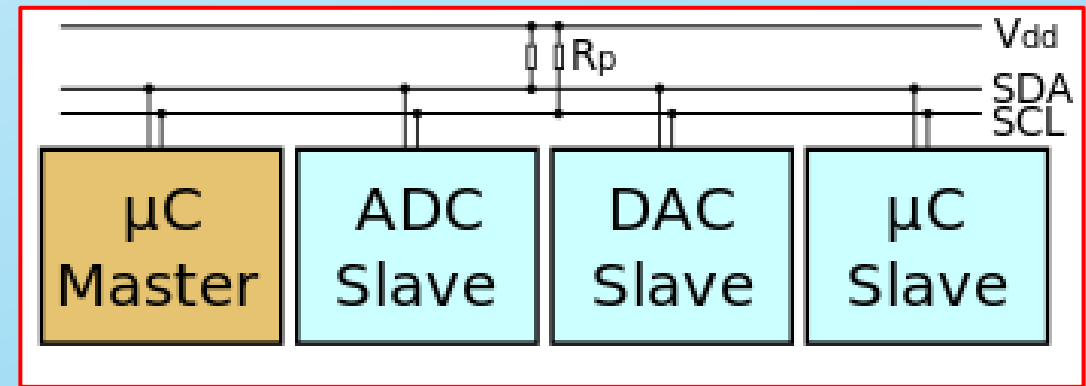
I2C folosește doar două linii bidirecționale,  
linia de date în serie (**SDA**=Serial Data Line) și  
linia de ceas serial (**SCL**=Serial Clock Line),

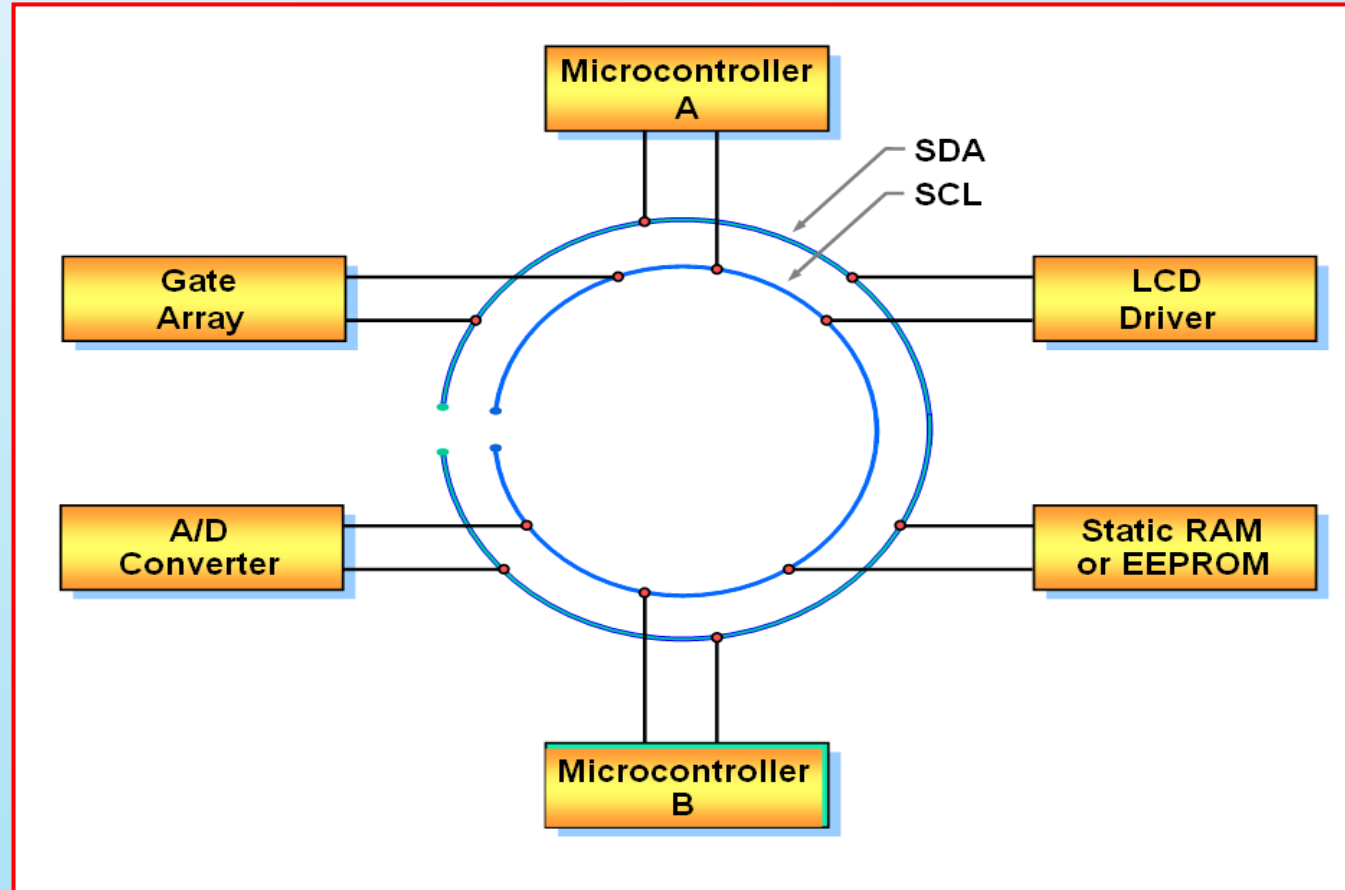
Tensiunile tipice utilizate ( $V_{dd}$ ) sunt +5 V sau +3,3 V,  
deși sunt permise și sistemele cu alte tensiuni.

Vitezele obișnuite ale magistralei I2C sunt:

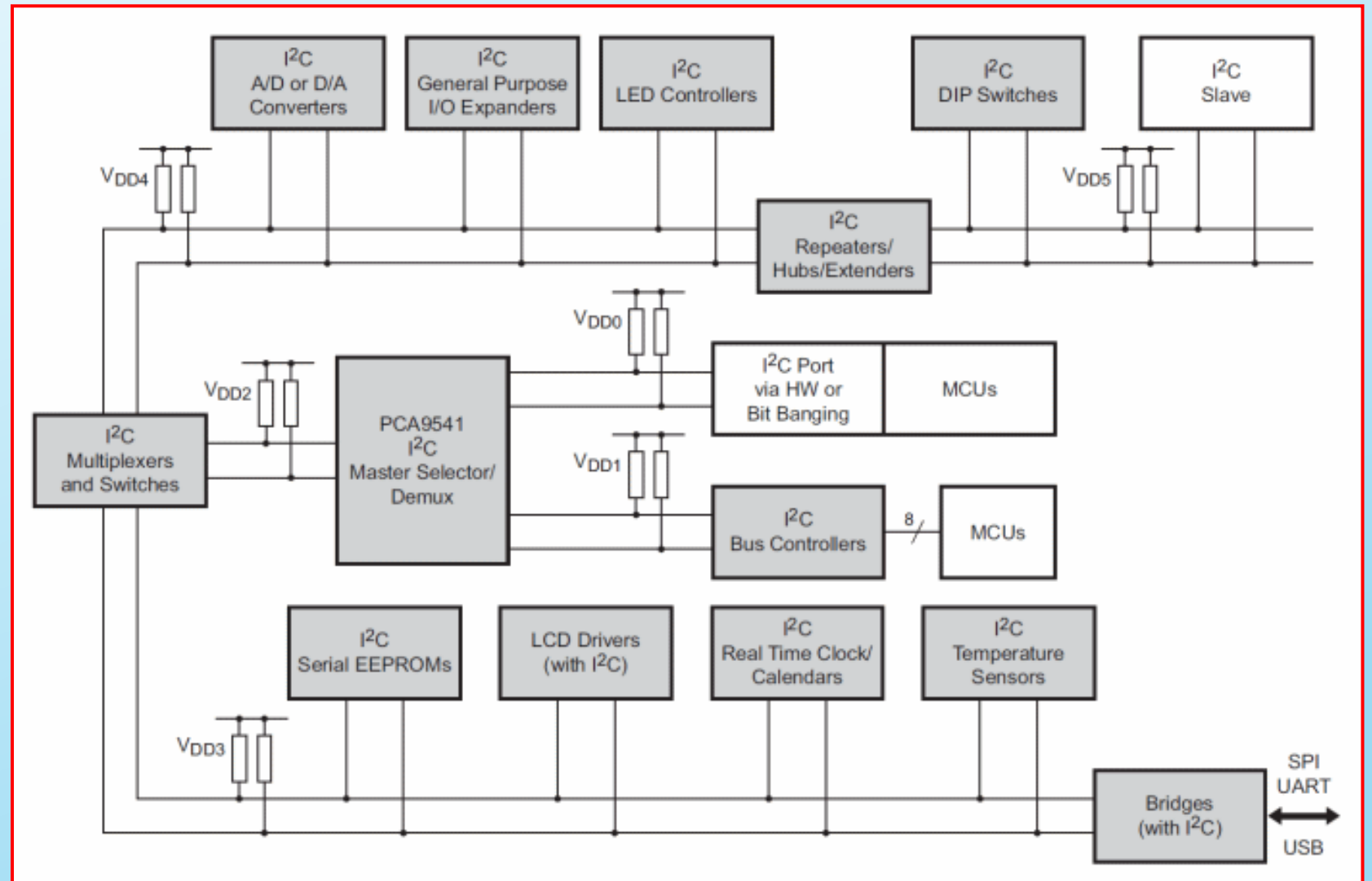
- modul de viteză redusă de 10 kbit/s,
- modul standard de 100 kbit/s
- Modul de viteză mare 3.4 Mbit/s
- sunt permise și frecvențe de ceas arbitrar scăzute.

Topologie simplă





Exemplu de  
topologie complexa:



Magistrala I2C folosește două fire: date seriale (SDA) și ceasul serial (SCL).

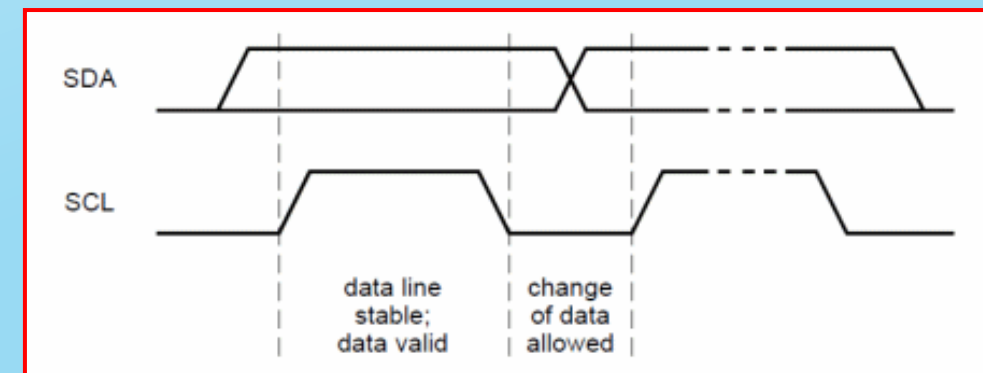
Toate dispozitivele *master* și *slave* I2C sunt conectate doar cu acele două fire.

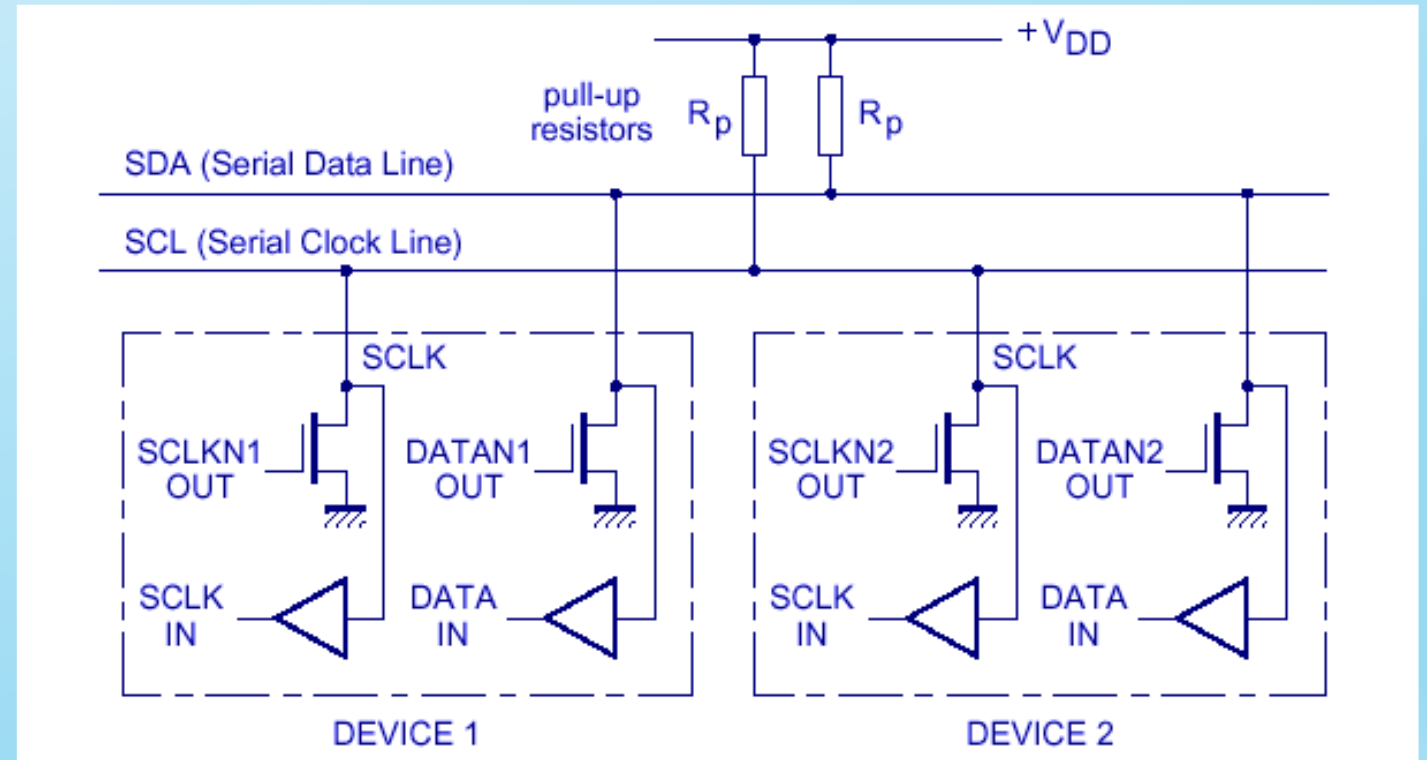
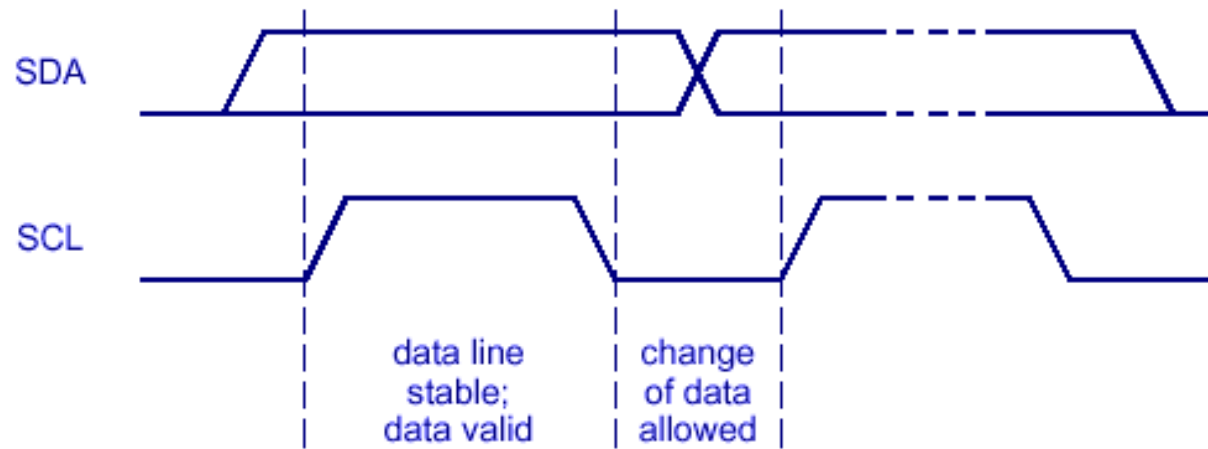
Fiecare dispozitiv poate fi un transmițător, un receptor sau ambele.

- Dispozitivele *master* generează semnalul de ceas și inițiază comunicarea pe magistrală,
- Dispozitivele *slave* răspund la comenzile de pe magistrală.

Pentru a se comunica cu un anumit dispozitiv, fiecare dispozitiv *slave* trebuie să aibă o adresă unică pe magistrală.

Dispozitivele *master* I2C (de obicei microcontrolere) nu au nevoie de o adresă, deoarece niciun alt dispozitiv (*slave*) nu trimite comenzi către master.





Dispozitivele I2C sunt conectate **AND** împreună, prin cablu.

Dacă un singur nod scrie zero, întreaga linie este zero



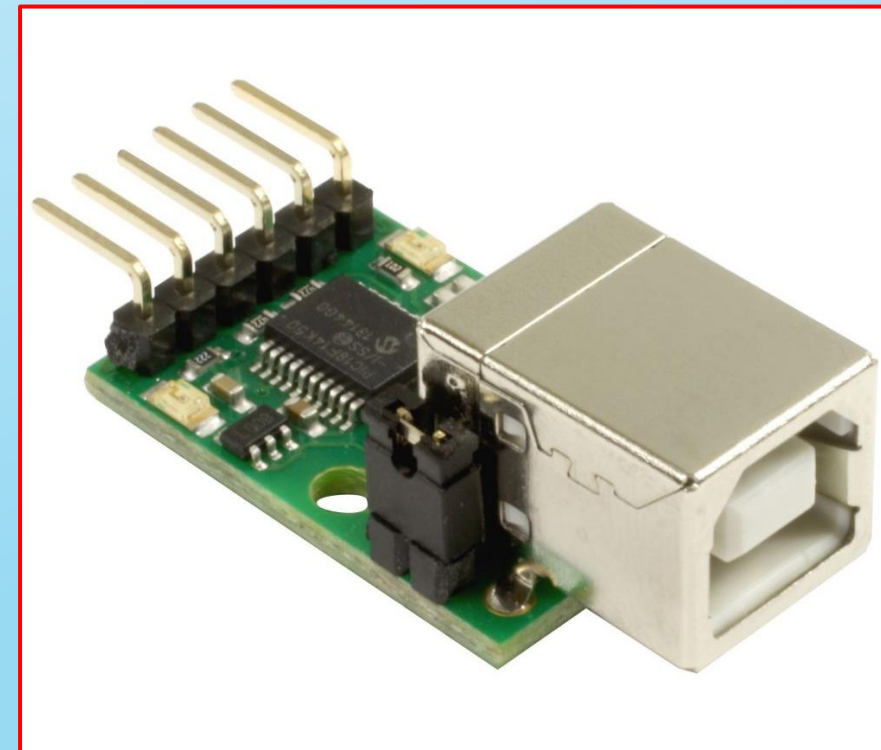
Sisteme de calcul dotate cu interfata I2C (dispozitiv *master*)

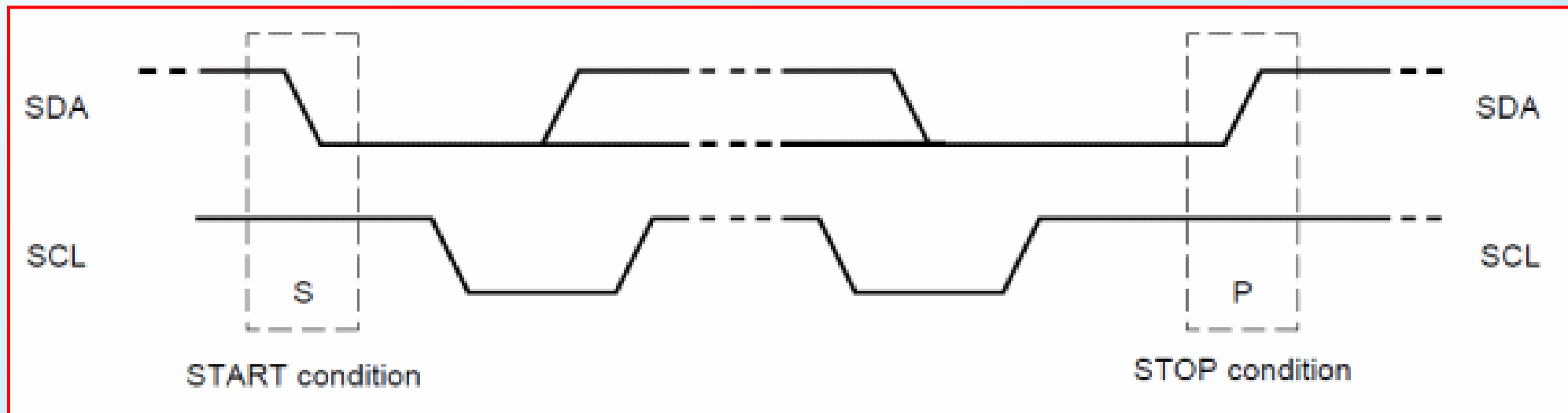
Raspberry PI

Arduino

Sisteme de calcul fara interfata I2C

Adaptor USB-I2C





## Condițiile de pornire și oprire

Fiecare comandă I2C inițiată de dispozitivul *master*

începe cu o condiție START și se termină cu o condiție STOP.

Pentru ambele condiții, semnalul SCL trebuie să fie ridicat.

o tranziție **înalt spre scăzut** a semnalului SDA este considerată **START** și

o tranziție **scăzut spre înalt** ca STOP.

În ambele cazuri semnalul SCL este generat de *master*.

După condiția de pornire, magistrala este considerată ocupată și poate fi utilizată de un alt *master* numai după ce este detectată o condiție de oprire.

## Transferul de date

Este generat un puls de ceas pentru fiecare bit transferat.

Două reguli simple (sincronizare) dictează modul de operare:

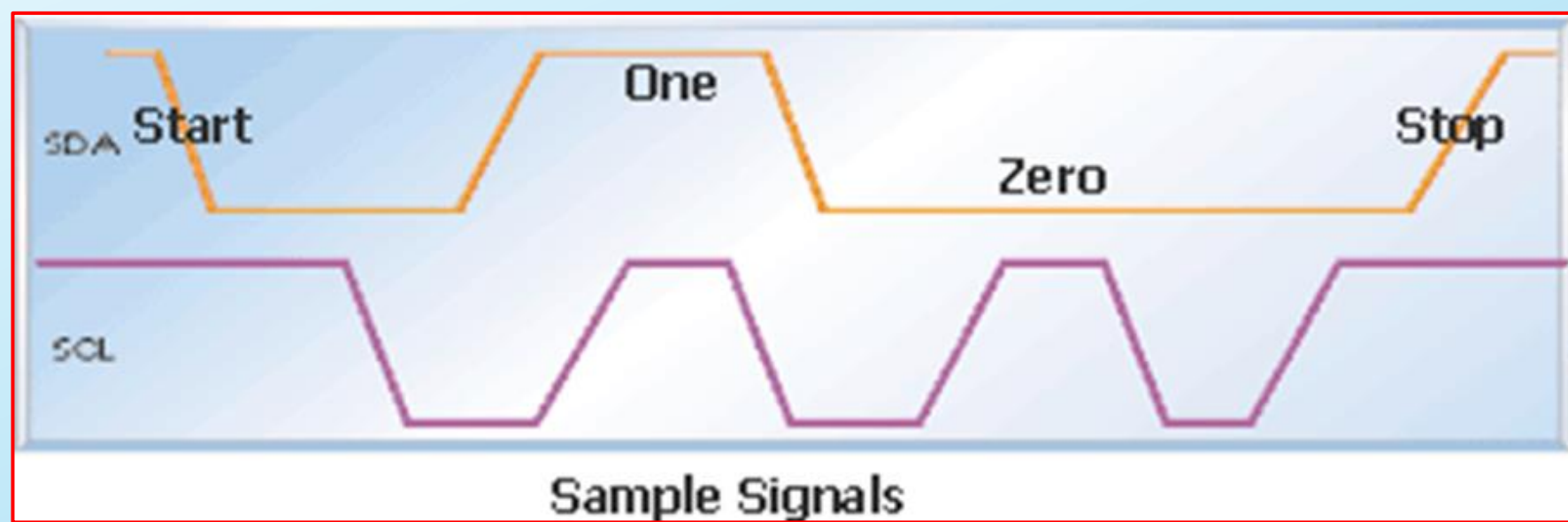
1. Când linia SCL are valoare scăzută starea liniei SDA se poate schimba.
2. Când linia SCL are valoare ridicată, starea liniei SDA indică valoarea unui bit.

Două excepții de la regula 2 creează condiții speciale care sunt folosite pentru a delimita începutul și sfârșitul fiecărei tranzacții între două dispozitive de pe magistrală.

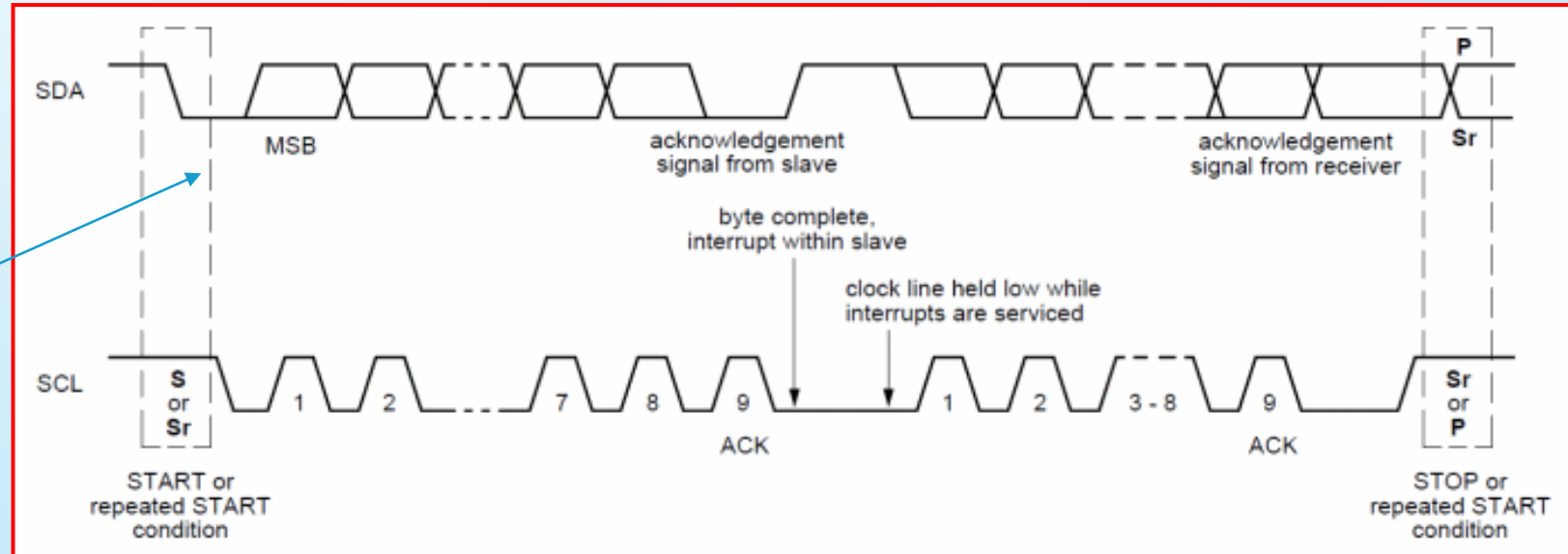
Când SCL este ridicat:

Condiția START este indicată de schimbarea liniei SDA de la ridicat la scăzut

Condiția STOP este indicată de schimbarea liniei SDA de la scăzut la ridicat



*Most Significant Bit*



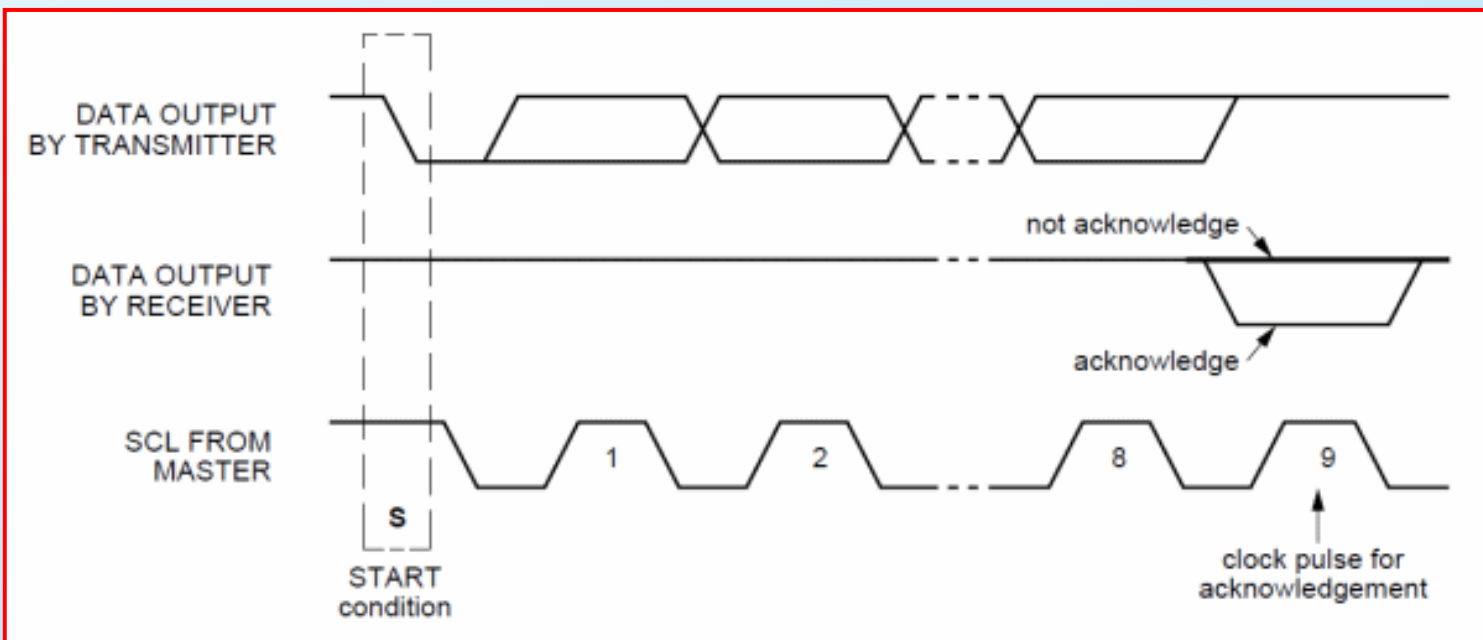
Datele de pe magistrala I2C sunt transferate în pachete de 8 biți (octeți).

Nu există nicio limitare a numărului de octeți, totuși, fiecare octet trebuie să fie urmat de un bit de confirmare. Acest bit semnalează dacă dispozitivul este pregătit să continue cu următorul octet.

Pentru toți biții de date, inclusiv bitul de confirmare, *masterul* trebuie să genereze impulsuri de ceas.

Dacă dispozitivul *slave* nu confirmă transferul, înseamnă că nu mai există date sau că dispozitivul nu este încă pregătit pentru transfer.

Dispozitivul *master* trebuie să genereze starea de oprire sau de pornire repetată.



## Confirmarea

La al 9-lea bit recepționat dispozitivul *slave* pune SDA la nivel scăzut.

Dacă acest lucru nu se întâmplă *slave* nu a detectat date.

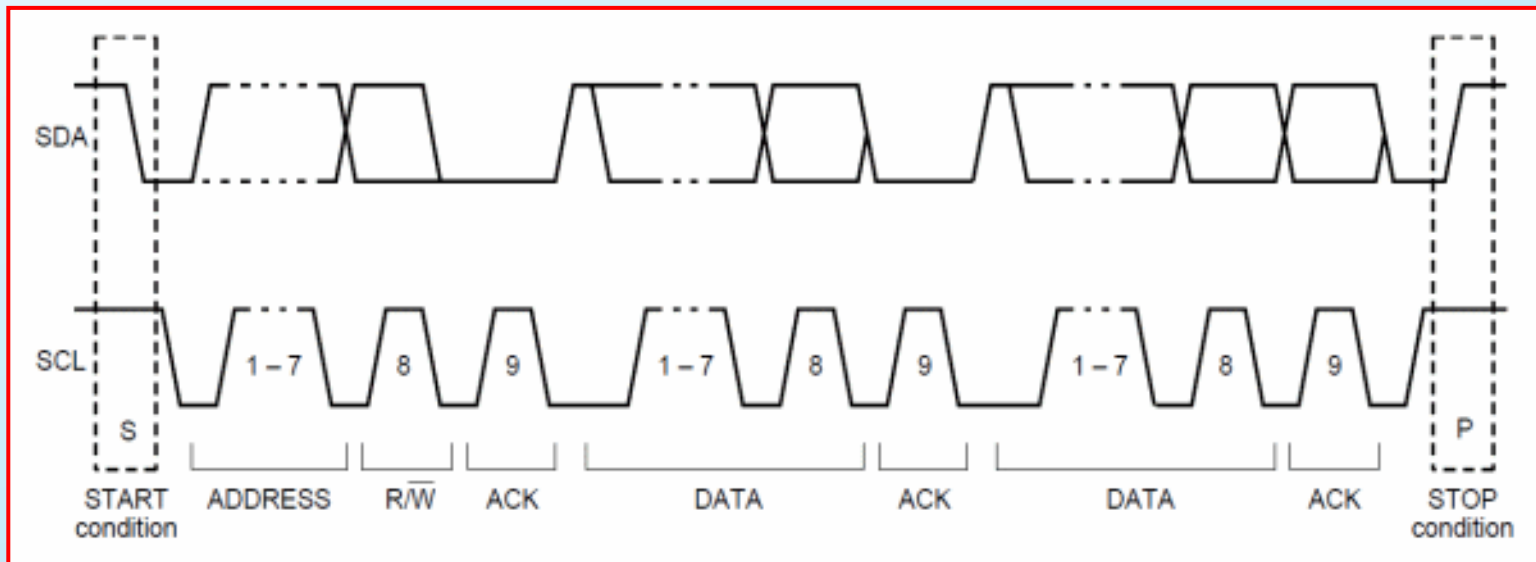
În acest caz trebuie reîncercata transmiterea datelor sau trebuie semnalată o eroare.

## Sincronizarea ceasului și handshaking

Dispozitivele *slave* care au nevoie de ceva timp pentru a procesa octetul primit sau care nu sunt încă gata să trimită următorul octet, pot cobori semnalul de ceas pentru a semnala *masterului* că ar trebui să aștepte.

Odată ce semnalul ceasul este eliberat, *masterul* poate continua transmiterea următorului octet.





## Comunicarea cu adrese I2C pe 7 biți

Fiecare dispozitiv *slave* de pe magistrală trebuie să aibă o adresă unică de 7 biți. (sau 10 biti)  
Comunicarea începe cu condiția Start, urmată de adresa dispozitivului *slave* pe 7 biți și bitul de direcție a datelor (citire/scriere).

Dacă acest bit este **0**, atunci *masterul* va **scrie** pe dispozitivul *slave*.

Dacă bitul de direcție a datelor este **1**, *masterul* va **citi** de pe dispozitivul *slave*.

După ce adresa *slave* și direcția datelor sunt trimise, masterul poate continua cu citirea sau scrierea.  
Comunicația se încheie cu condiția Stop care semnalează și că magistrala I2C este liberă.

Dacă masterul trebuie să comunice cu alți *slave*, acesta poate genera o pornire repetată cu o altă adresă *slave* fără generarea stării de oprire.

Toți octeții sunt transferați cu bitul MSB transportat mai întâi.

# Adresarea I2C pe 7 biți



Alocarea adreselor I2C este administrată de **I2C Bus Committee** care se ocupă de alocări.

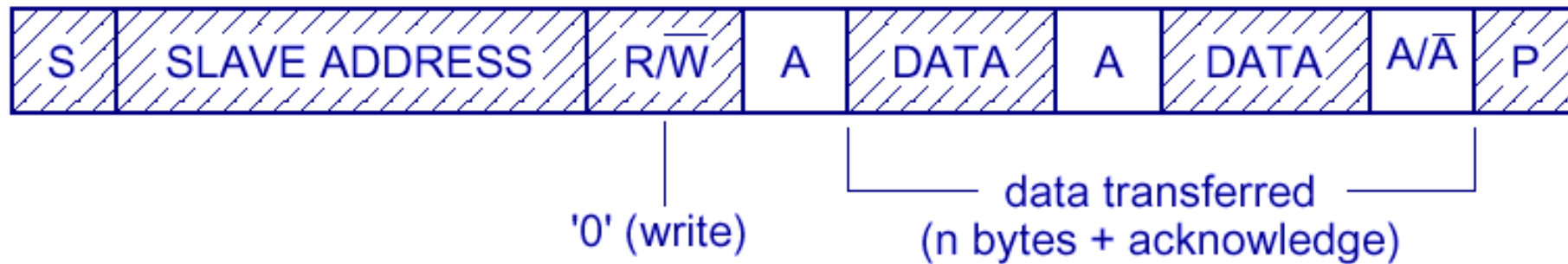
Două grupuri de 8 adrese I2C sunt rezervate pentru utilizări viitoare și o adresă este utilizată pentru adresarea I2C pe 10 biți.

SLAVE ADDRESS	R/ $\overline{W}$ BIT	DESCRIPTION
0000 000	0	General call address
0000 000	1	START byte
0000 001	X	CBUS address
0000 010	X	Reserved for different bus format
0000 011	X	Reserved for future purposes
0000 1XX	X	Hs-mode master code
1111 1XX	X	Reserved for future purposes
1111 0XX	X	10-bit slave addressing



## Transfer de date de la *Master*-Transmitter la *Slave*-Receiver

Directia de transmisie nu se schimba niciodata.  
Configurarea si transferul sunt simple



 from master to slave

 from slave to master

A = acknowledge (SDA LOW)

$\bar{A}$  = not acknowledge (SDA HIGH)

S = START condition

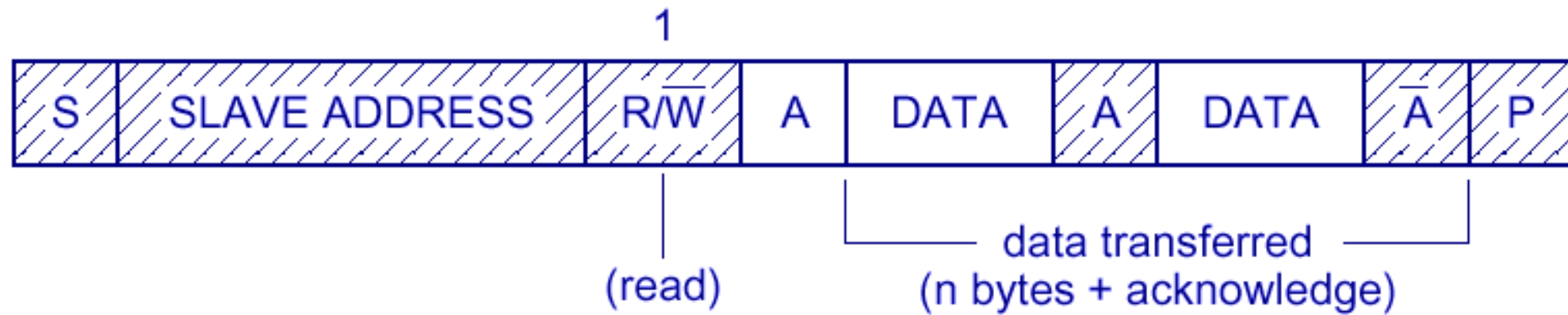
P = STOP condition

## Transfer de date *Master-Receiver* și *Slave-Transmitter*

*Masterul* inițiază transferul de date prin generarea condiției START, urmată de octetul de pornire (cu bitul de citire/scriere setat la 1, adică modul de citire)

După prima confirmare de la *slave*, direcția datelor se schimbă și *masterul* devine receptor iar transmițător *slave*.

Condiția STOP este încă generată de *master* (*masterul* trimite not-ACK înainte de a genera STOP)



# Citire și scriere în același transfer de date

Schimbarea direcției transferului de date poate avea loc prin generarea de către *master* a unei alte condiții de START (numită condiție de START repetată) cu adresa *slave* repetată

Dacă *masterul* a fost un receptor înainte de schimbare, atunci *masterul* trimite un not-ack (A') înainte de condiția START repetată

