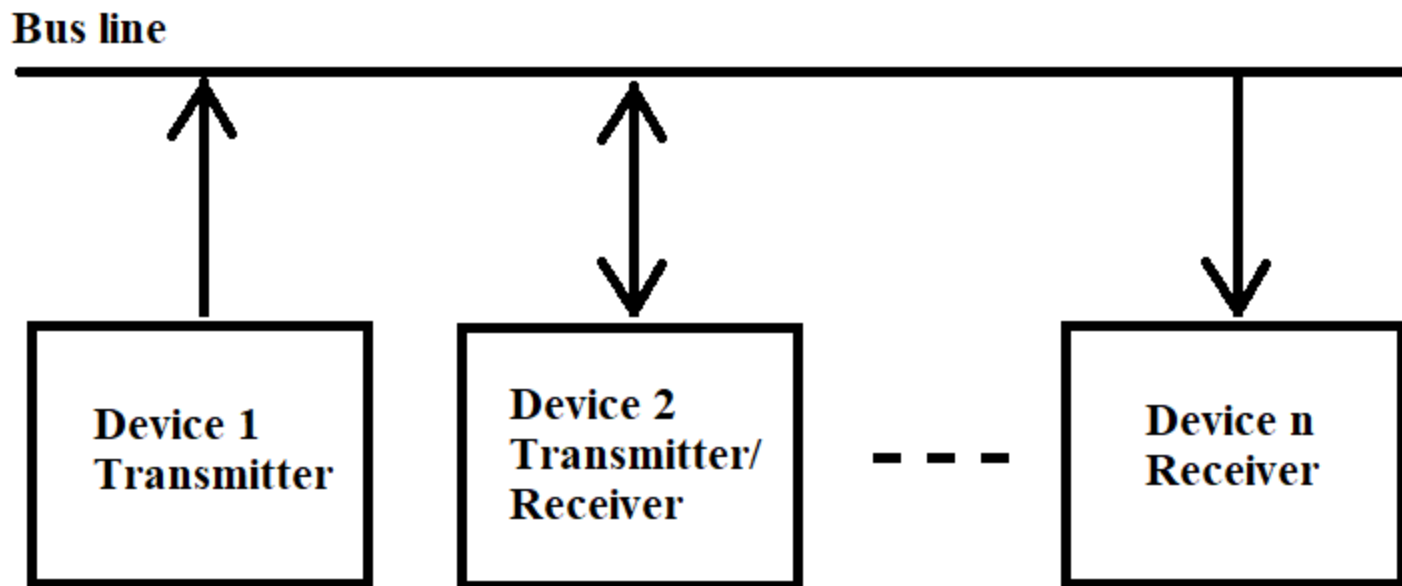


# REALIZAREA MAGISTRALELOR CU CIRCUITE LOGICE

- Introducere
- Circuite logice cu colector deschis
- Magistrala I2C
- Circuite cu trei stări

# Introducere



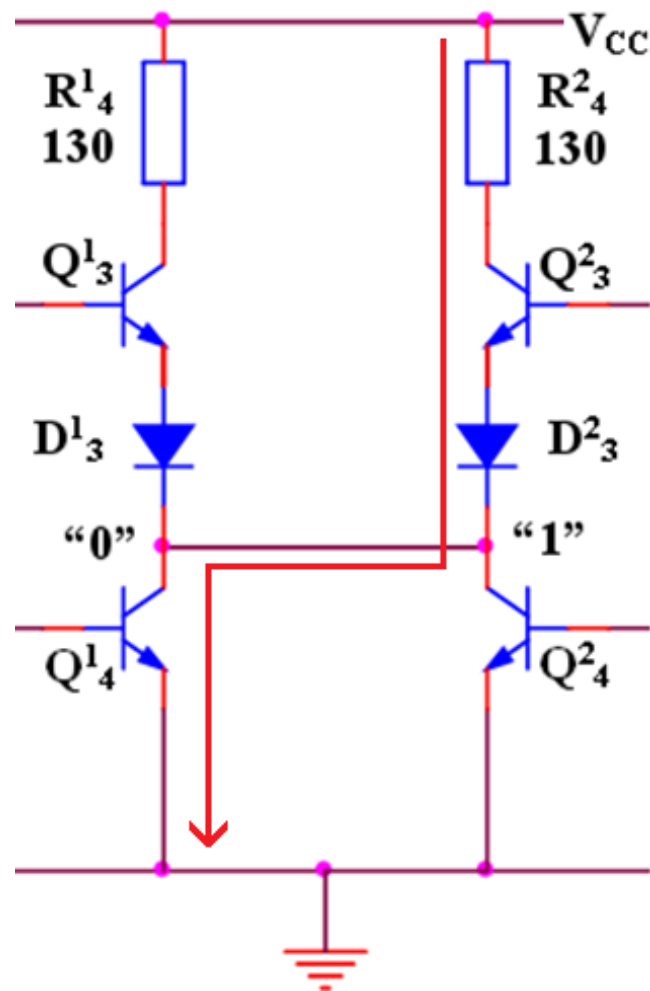
Necesită legarea împreună, prin intermediul magistralei, a ieșirilor circuitelor care transmit informații pe respectiva magistrală

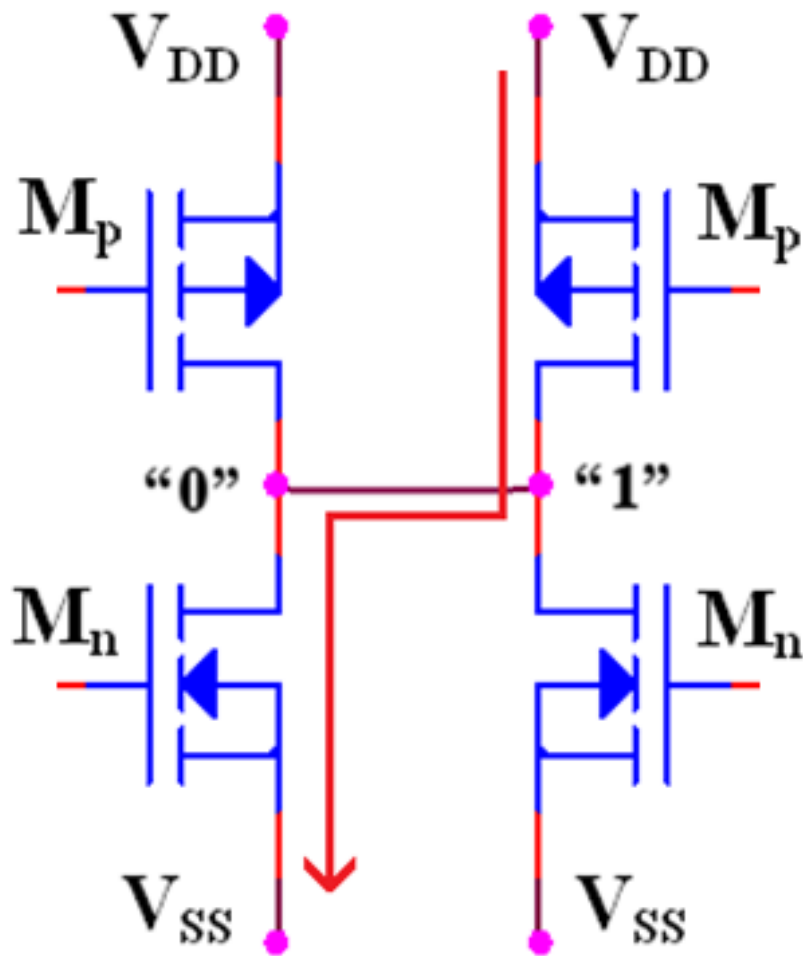
Un dezavantaj important al circuitelor integrate TTL sau MOS cu structură obișnuită este că nu permit legarea ieșirilor împreună.

Acest mod de lucru este solicitat în elaborarea magistralelor.

Dacă ambele ieșiri sunt simultan în “0” sau în “1” circuitul funcționează corect.

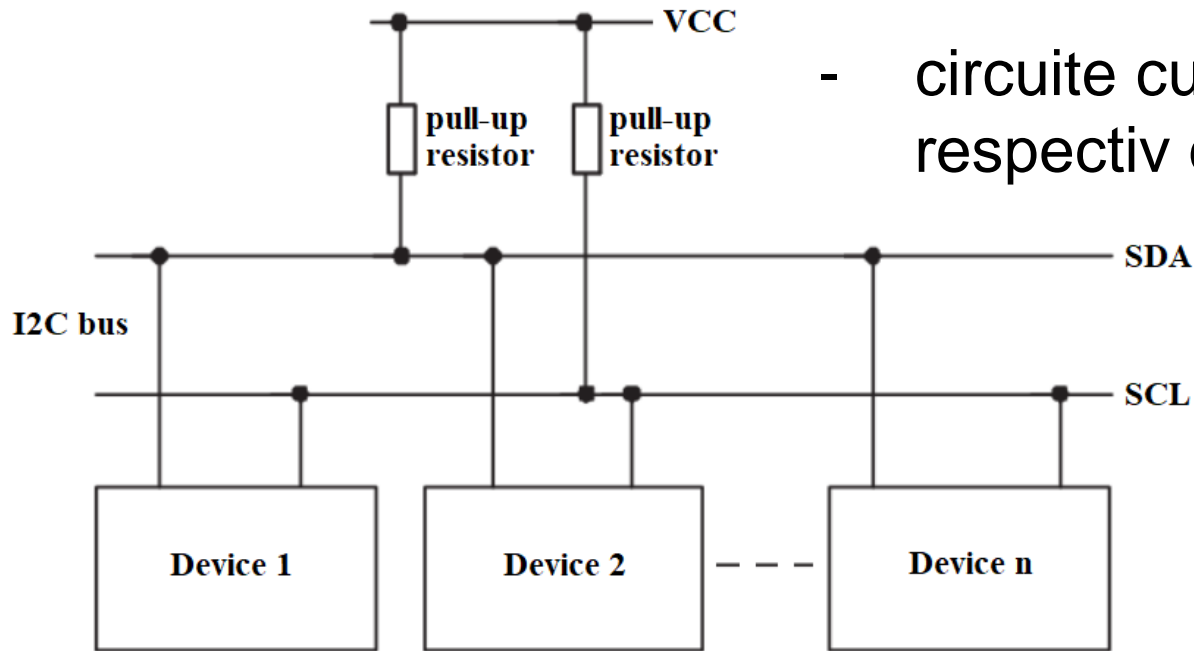
În cazul în care o ieșire se află în “0” și alta în “1” se realizează practic punerea la masă a tranzistorului  $Q_3^2$ , curentul prin acesta fiind limitat numai de dioda  $D_3^2$  și rezistența  $R_4^2$  de  $130\Omega$ . Aceasta determină disiparea unei puteri excesiv de mari în tranzistoarele  $Q_4^1$  și  $Q_3^2$  și în rezistența  $R_4^2$ . Parametri porții se modifică, sau se distrug tranzistoarele prin ambalare termică.



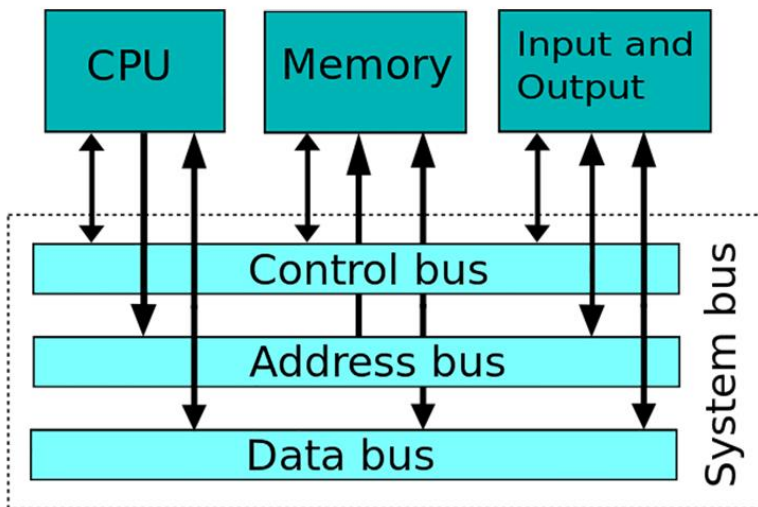


Dacă ieșirile a două porți MOS se conectează în paralel, în cazul în care o ieșire este în 1 logic și alta în 0 logic, nivelul de tensiune de la ieșire este determinat de divizorul de tensiune format de tranzistoarele cu canal n și p aflate simultan în conducție.

Pentru a cupla în paralel mai multe porți, se utilizează:

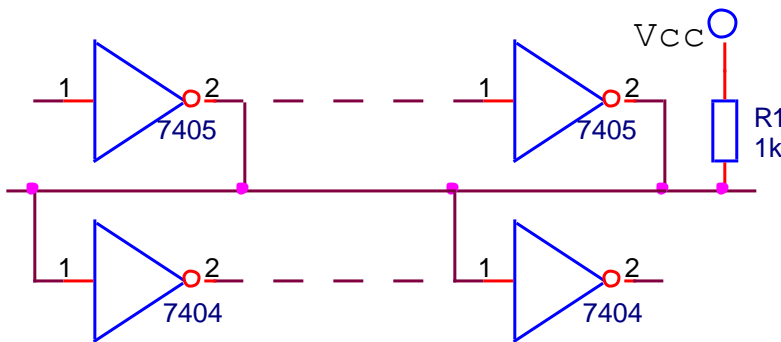
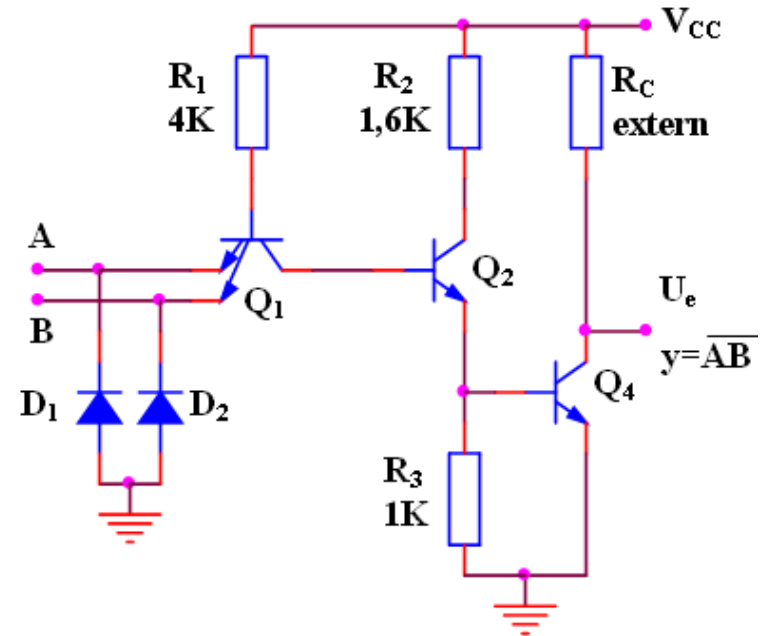
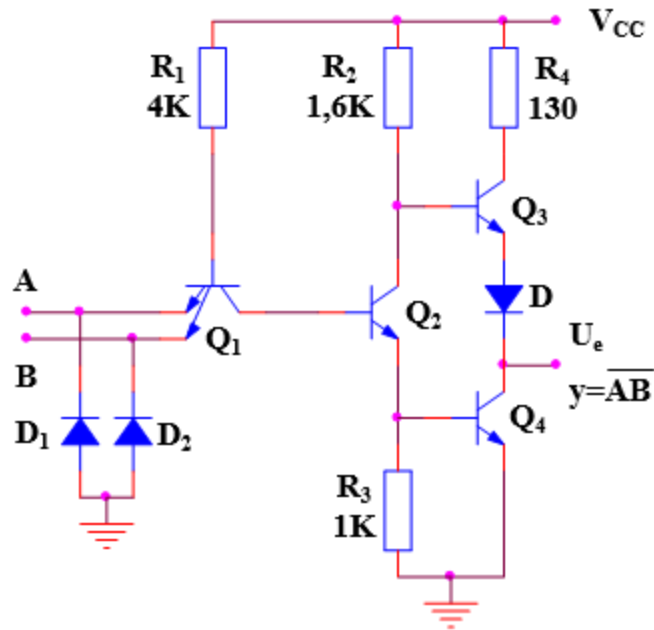


- circuite cu colector deschis respectiv drenă deschisă;



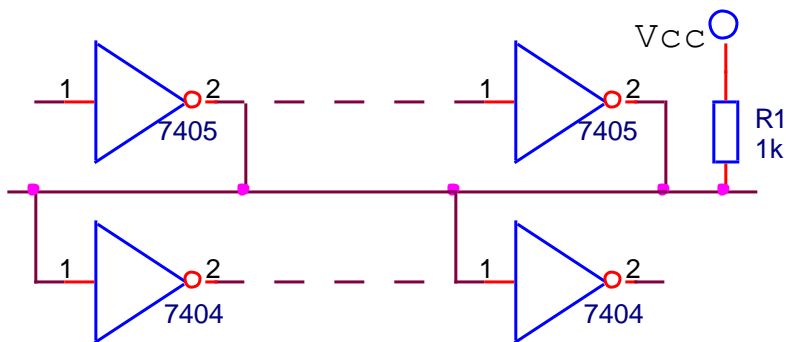
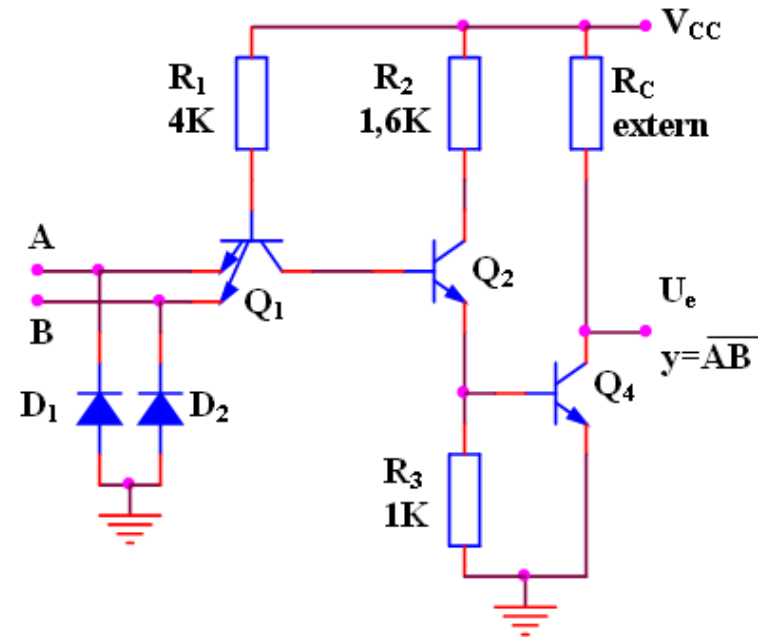
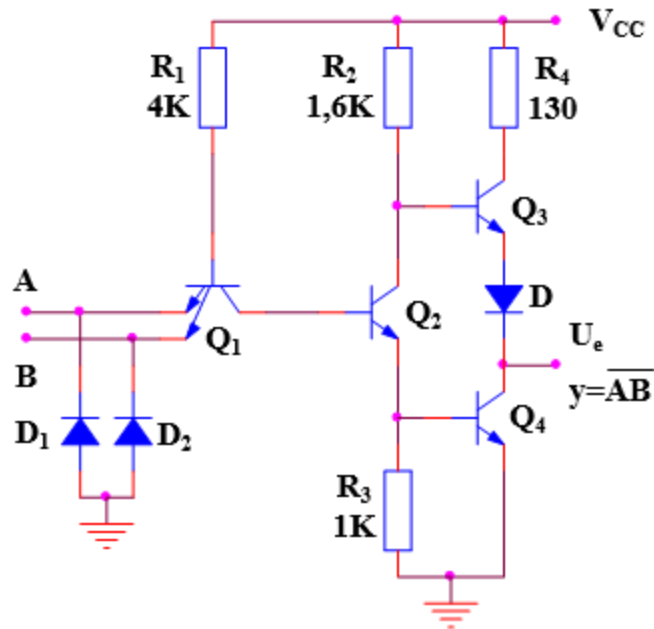
- circuite cu trei stări.

# Poarta TTL cu colector în gol



Colectoarele tranzistoarelor  $Q_4$  aparținând diferitelor circuite pot fi legate împreună, punctul comun fiind conectat printr-o rezistență la sursă. La ieșirea comună avem funcția SI.

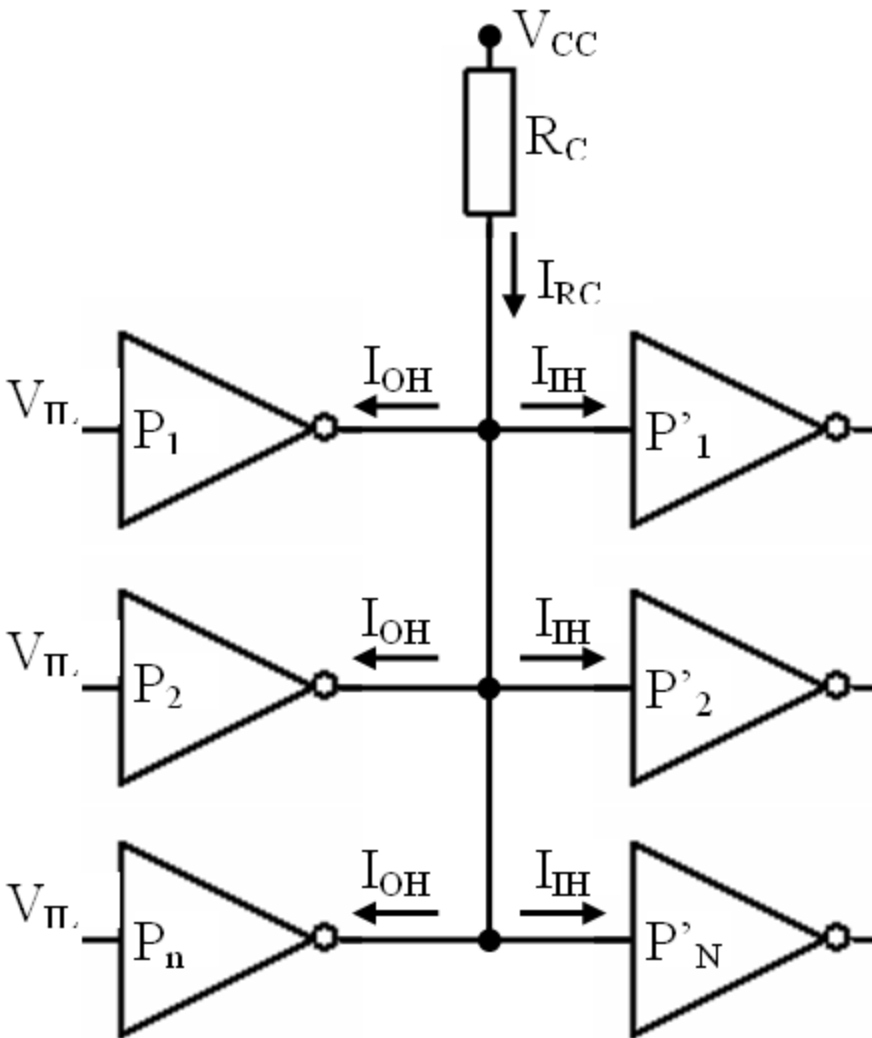
Rezistența comună ( $R_C$ ) nu este inclusă în structura integrată, ea fiind calculată de proiectantul schemei în funcție de numărul porților conectate împreună și de numărul porților TTL care trebuie comandate de către această ieșire comună.



O valoare mare a  $R_C$  reduce consumul de putere dar mărește impedența de ieșire măbind întârzierea în propagarea semnalelor și sensibilitatea la zgomot

Este necesar un compromis între timpul de comutare și consumul de putere

# Calculul rezistenței $R_C$

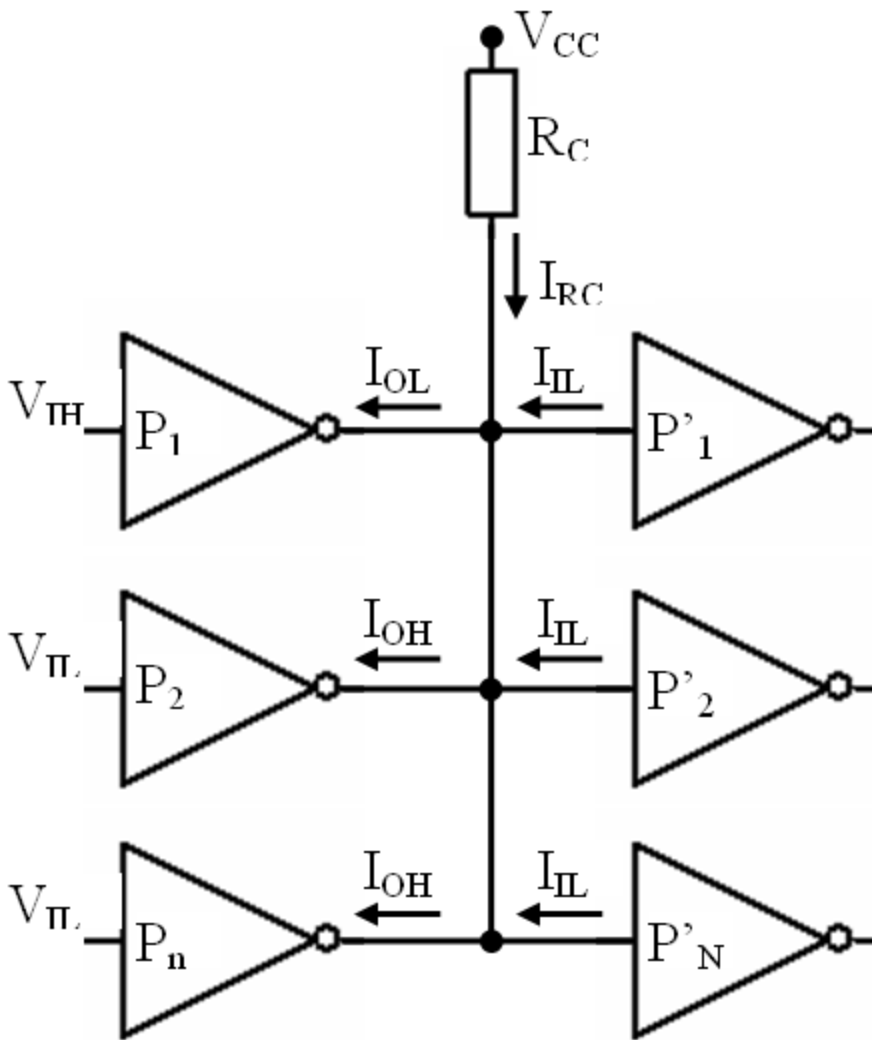


- Funcție de nivelul logic de la ieșirea comună, de curentul debitat de porțile conectate în paralel și de curenții absorbiți de porțile comandate
- În cazul nivelului 1 logic la ieșire rezulta:

$$R_{c \max} = \frac{V_{cc \min} - V_{OH \min}}{n \bullet I_{OH} + N \bullet I_{IH}}$$



# Calculul rezistenței $R_C$



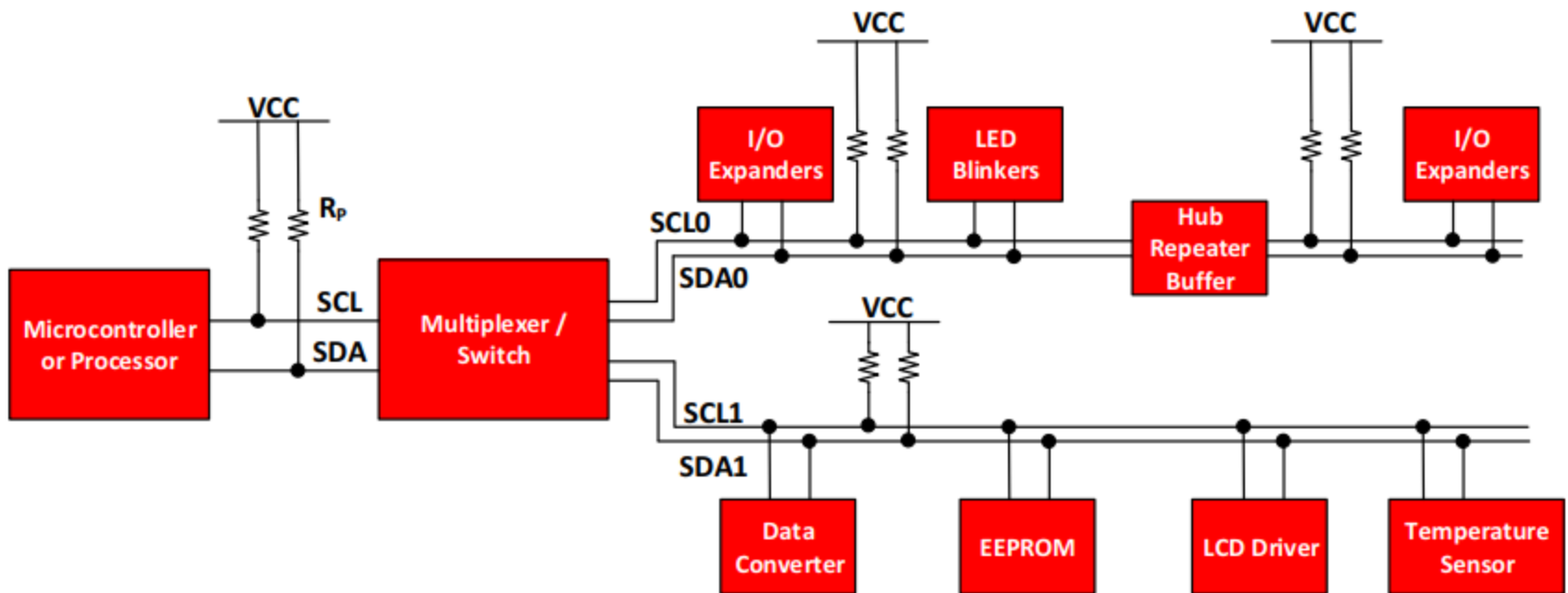
- pentru nivelul 0 logic la ieșire:

$$R_{C \min} = \frac{V_{cc \max} - V_{OL \max}}{I_{OL} + (n-1)I_{OH} - N \bullet I_{IL}}$$

- $V_{CC}=5V \pm 5\%$
- $I_{OH}=250\mu A$
- $I_{OL}=16mA$
- $I_{IL}=1.6mA$
- $I_{IH}=40\mu A$
- $V_{OH}=2.4V$
- $V_{OL}=0.4V$
- $P_D=20mW$
- $t_{pd}=13ns$

# Magistrala I<sup>2</sup>C

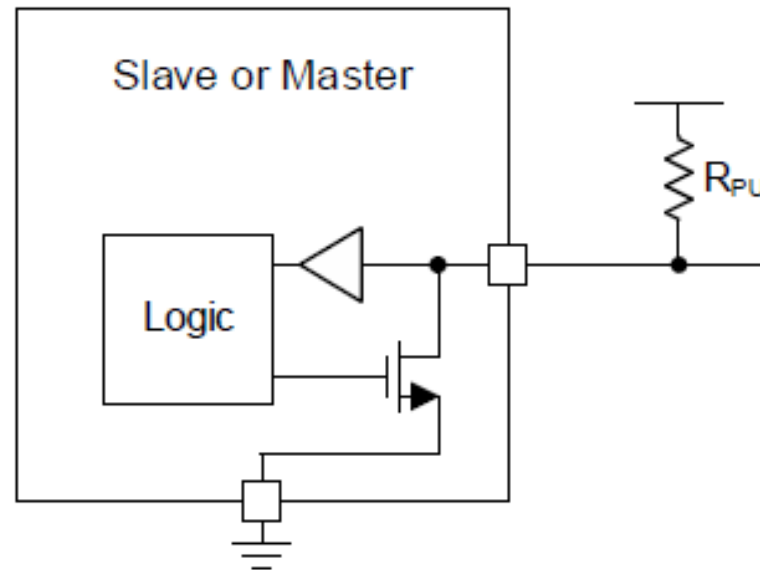
- Inter-Integrated Circuit
- Comunicatii bidirectionale intre multiple dispozitive master si slave
- Foloseste doar 2 linii



- Sursa bibliografica:
- [https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1668353137830&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1668353137830&ref_url=https%253A%252F%252Fwww.google.com%252F)

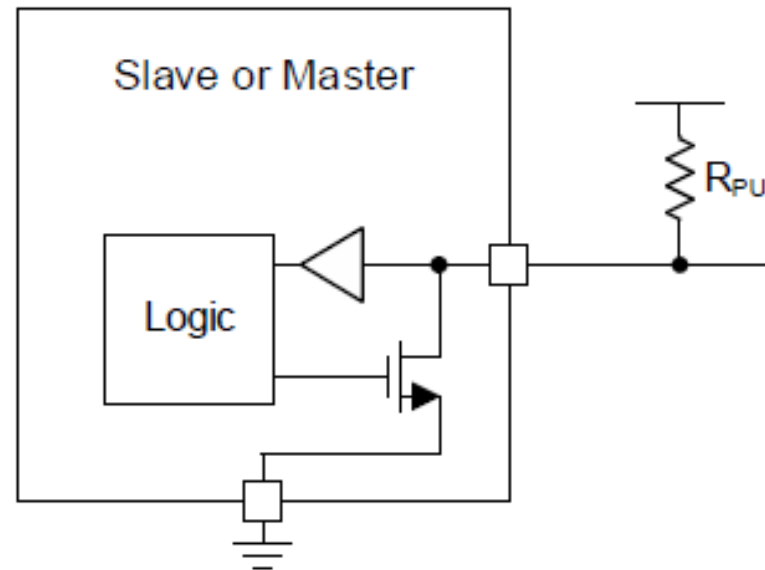
# Magistrala I<sup>2</sup>C – Caracteristicile electrice

- Open-drain/open-collector cu un buffer de intrare pe aceeași linie
- Pull-down FET pentru transmiterea datelor
- Buffer pentru citirea datelor de intrare
- Un dispozitiv poate doar să traga linia de magistrală la un nivel coborât (scurtcircuit la masă prin deschiderea FET) sau să elibereze linia de magistrală (înaltă impedanță la masă prin blocarea FET) și să permită rezistenței de ridicare să crească tensiunea
- Niciun dispozitiv nu poate ține linia de magistrală la un nivel ridicat
- O singură linie de date folosită pentru flux de date bidirecțional



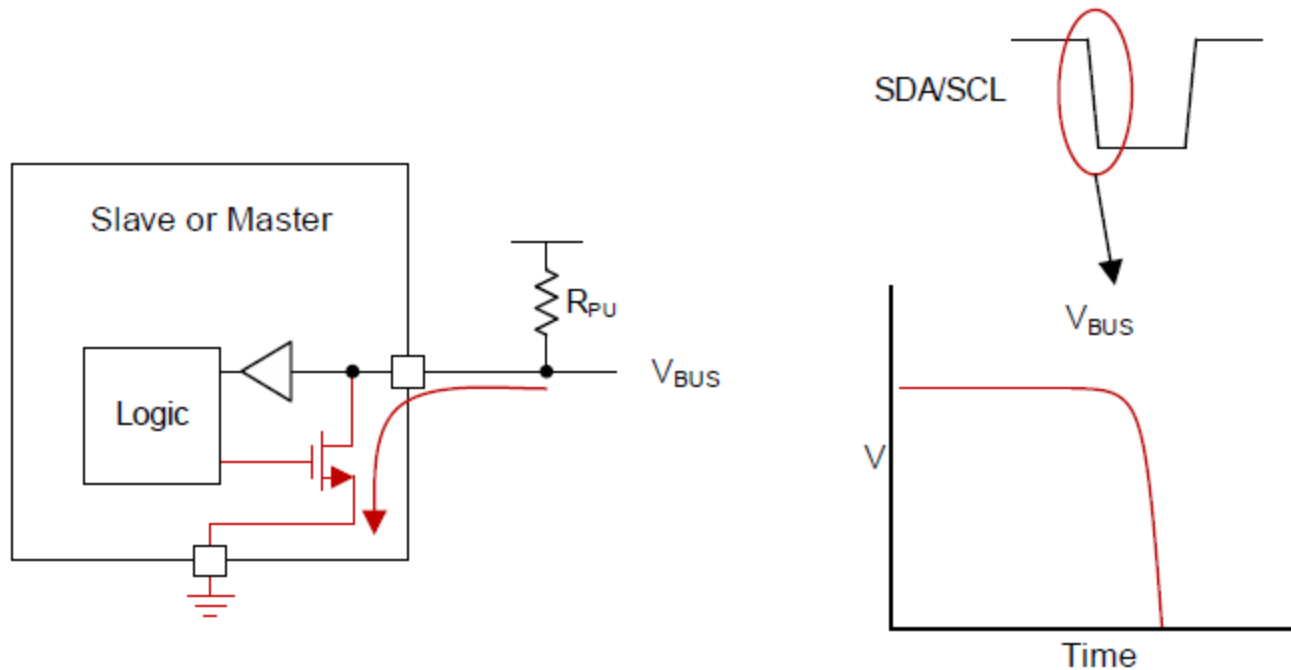
Structura internă a liniei SDA/SCL

- Niciun dispozitiv nu poate forta un nivel ridicat pe o linie
- Linia nu se va confrunta niciodată cu o problemă de comunicare unde un dispozitiv poate încerca să transmită un nivel ridicat și altul să transmită un nivel coborât, provocând un scurtcircuit (alimentare la masă)
- Dacă un master transmite un nivel ridicat, dar vede că linia este la un nivel coborât, oprește comunicațiile deoarece un alt dispozitiv folosește magistrala



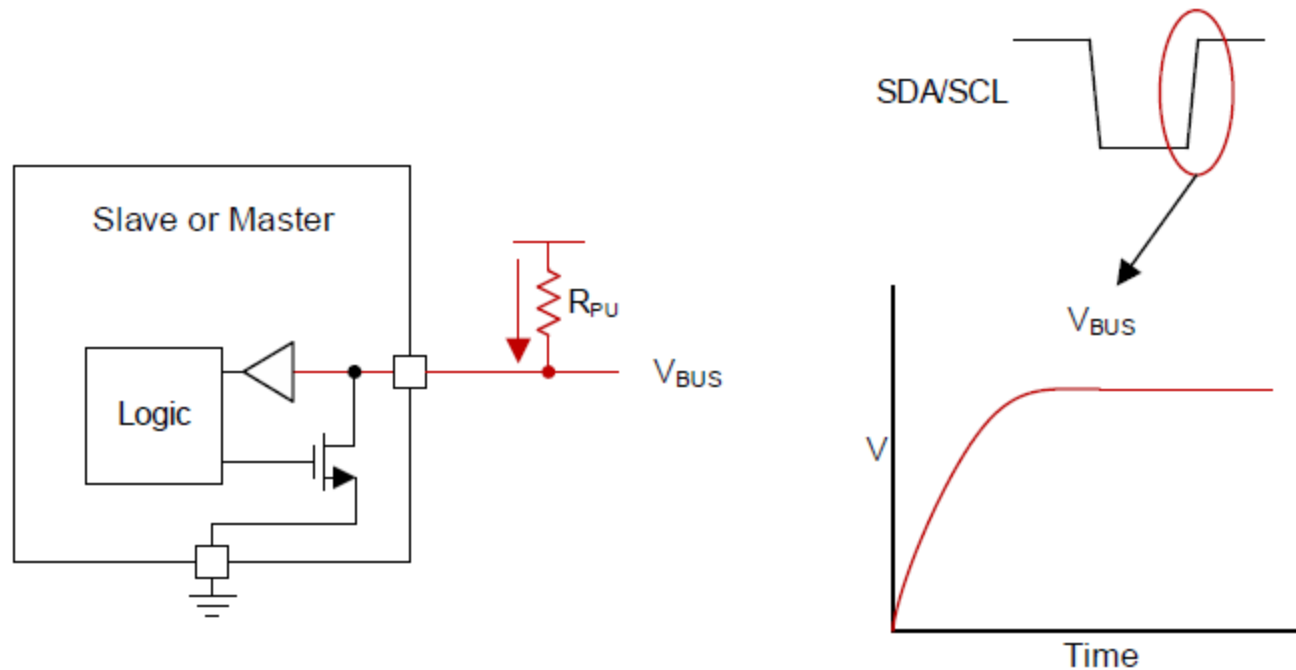
Structura internă a liniei SDA/SCL

- Daca dispozitivul doreste sa transmita un nivel coborat, va activa tranzistorul FET de coborare, care va furniza un scurtcircuit la masa, tragand linia la nivel coborat.



Coborarea nivelului magistralei cu o interfata Open-Drain

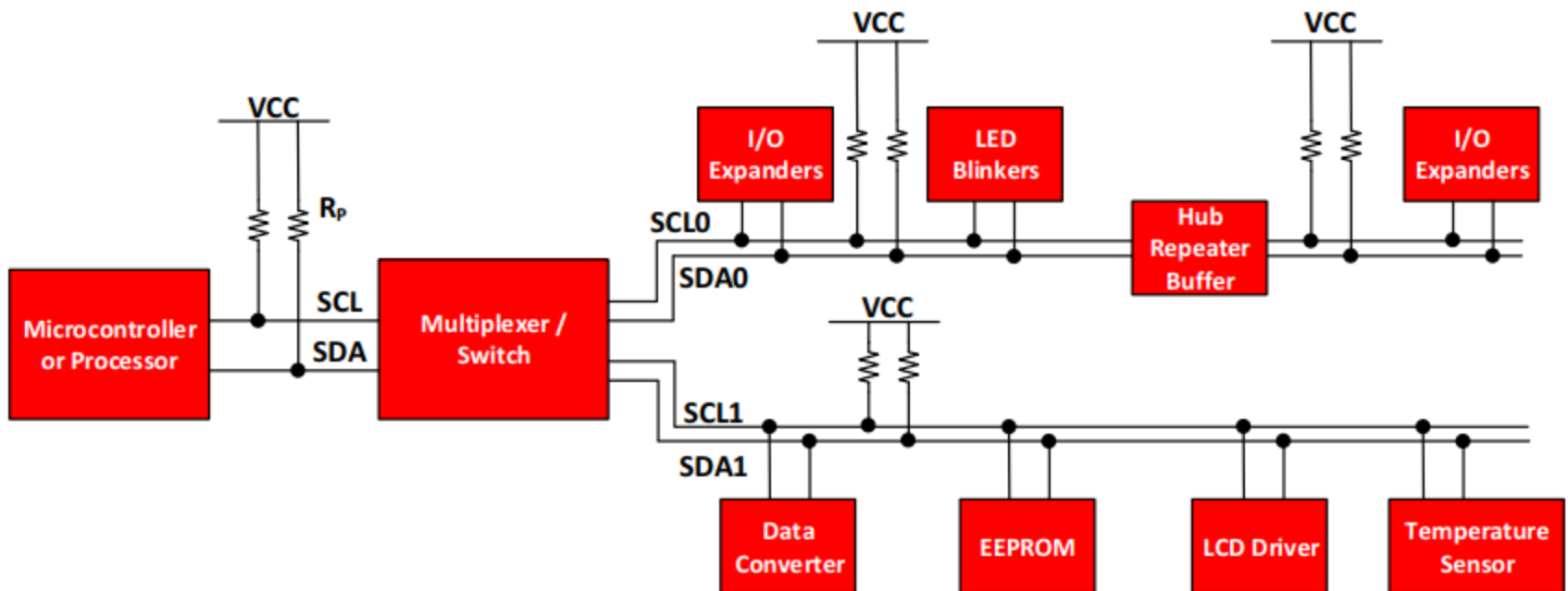
- Dacă dispozitivul dorește să transmită un nivel ridicat, poate doar elibera linia de magistrală blocând tranzistorul FET de coborare. Acesta lasă linia de magistrală flotantă, iar rezistența de ridicare va trage tensiunea în sus spre tensiunea de alimentare, care va fi interpretată ca un nivel ridicat.



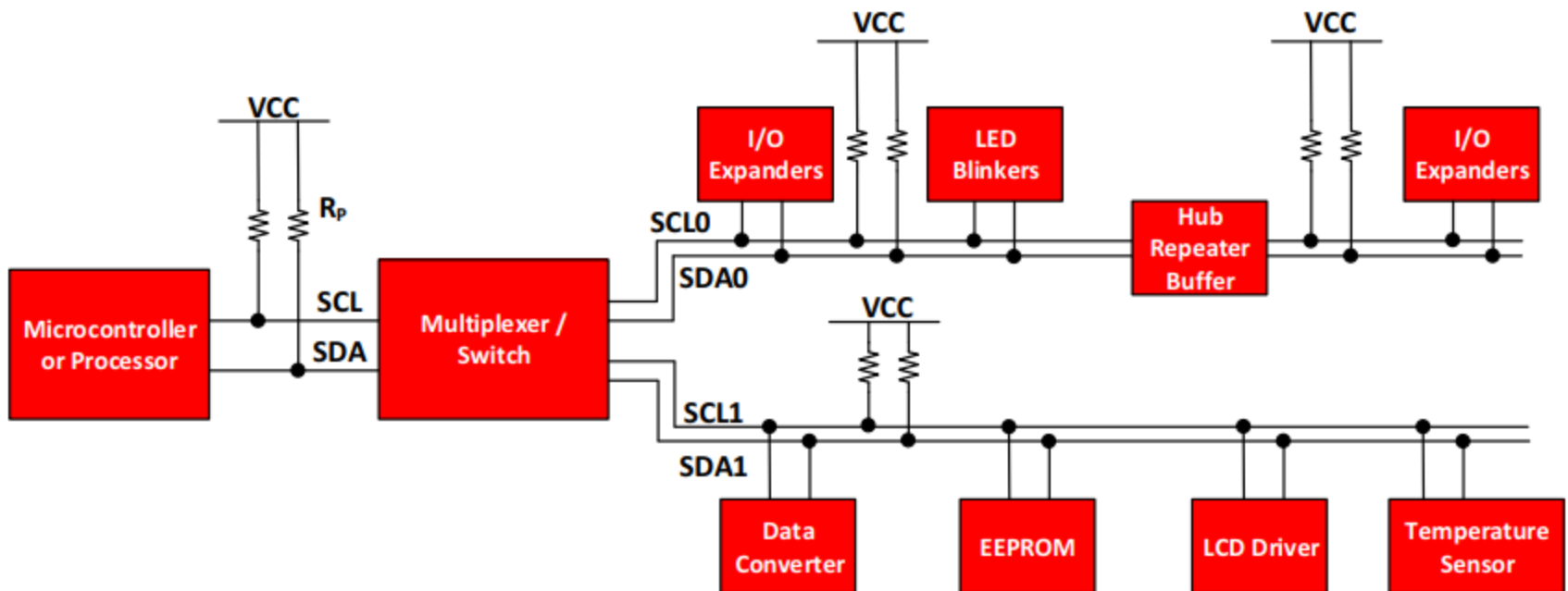
Eliberarea magistralei cu o interfata Open-Drain

# Magistrala I<sup>2</sup>C – Interfata I<sup>2</sup>C

- Magistrala I<sup>2</sup>C este o interfață bidirecțională standard
- Dispozitivele controler (master) comunica cu dispozitivele slave
- Un slave poate transmite date doar dacă acestea au fost adresate de către master
- Fiecare dispozitiv de pe magistrala I<sup>2</sup>C are o anumită adresă de dispozitiv

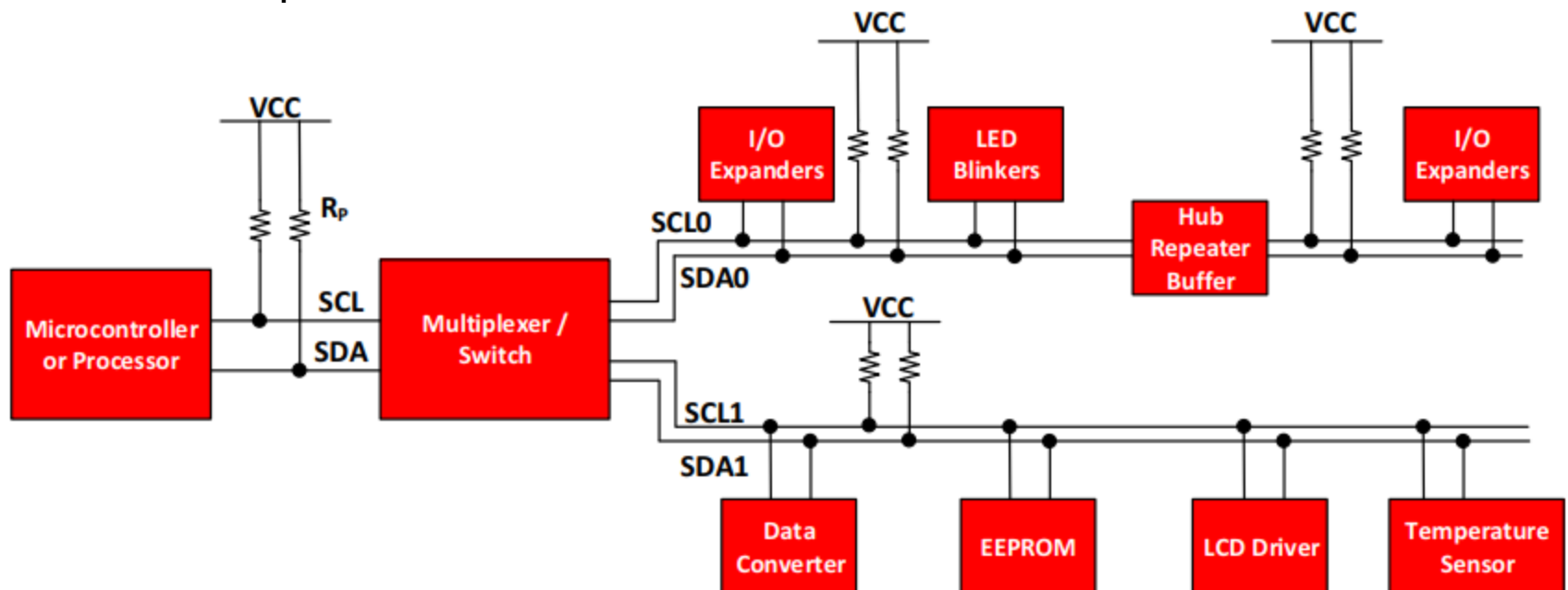


- Interfata I2C consta din liniile serial clock (SCL) si serial data (SDA).
- Ambele linii (SDA si SCL) trebuie conectate la VCC printr-o rezistenta de ridicare
- Valoarea rezistentei de ridicare este calculata de proiectant astfel incat pe linii sa se obtina functia logica Si cablat, tinand cont de capacitatea de pe liniile I2C, viteza de lucru urmarita si consumul de putere
- Transferul datelor poate fi initiat doar cand magistrala este inactiva (in stare de repaus)
- O magistrala este considerata inctiva daca ambele linii (SDA si SCL) sunt la nivel ridicat dupa o conditie STOP

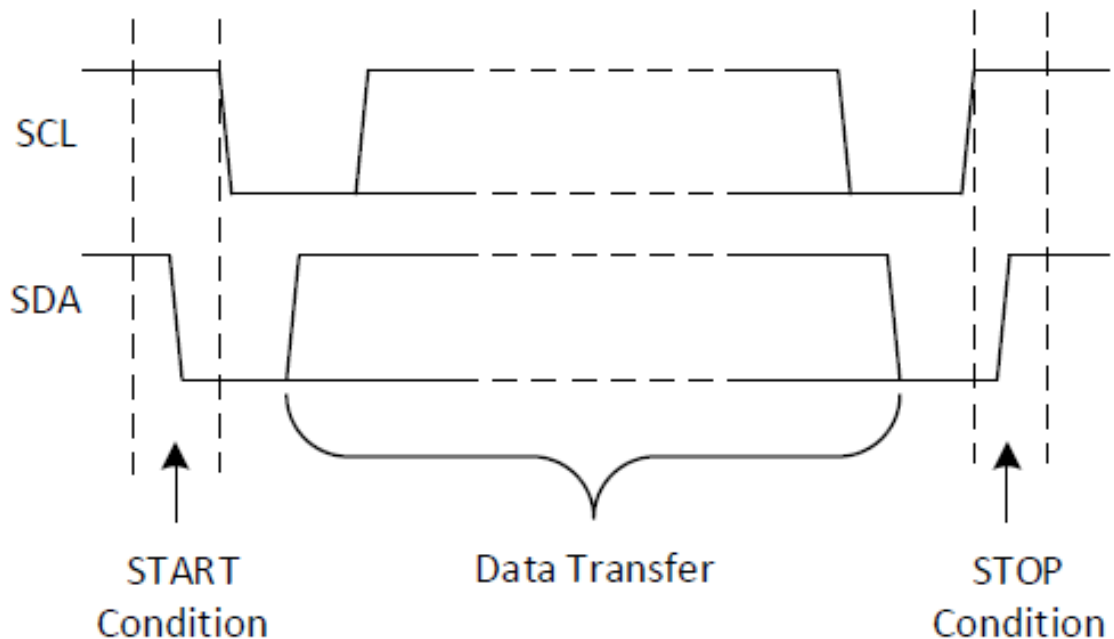




- Procedura generala pentru un dispozitiv master sa acceseze un dispozitiv slave:
- 1. Dispozitivul master doreste sa transmita date unui dispozitiv slave:
  - Master-transmitator trimite o conditie START si adreseaza slave-receptor
  - Master-transmitator trimite date spre slave-receptor
  - Master-transmitator termina transferul cu o conditie STOP
- 2. Dispozitivul master doreste sa receptioneze/citeasca date de la un dispozitiv slave:
  - Master-receptor trimite o conditie START si adreseaza slave-transmitator
  - Master-receptor trimite registrul cerut pentru citire la slave-transmitator
  - Master-receptor primeste datele de la slave-transmitator
  - Master-receptor termina transferul cu o conditie STOP

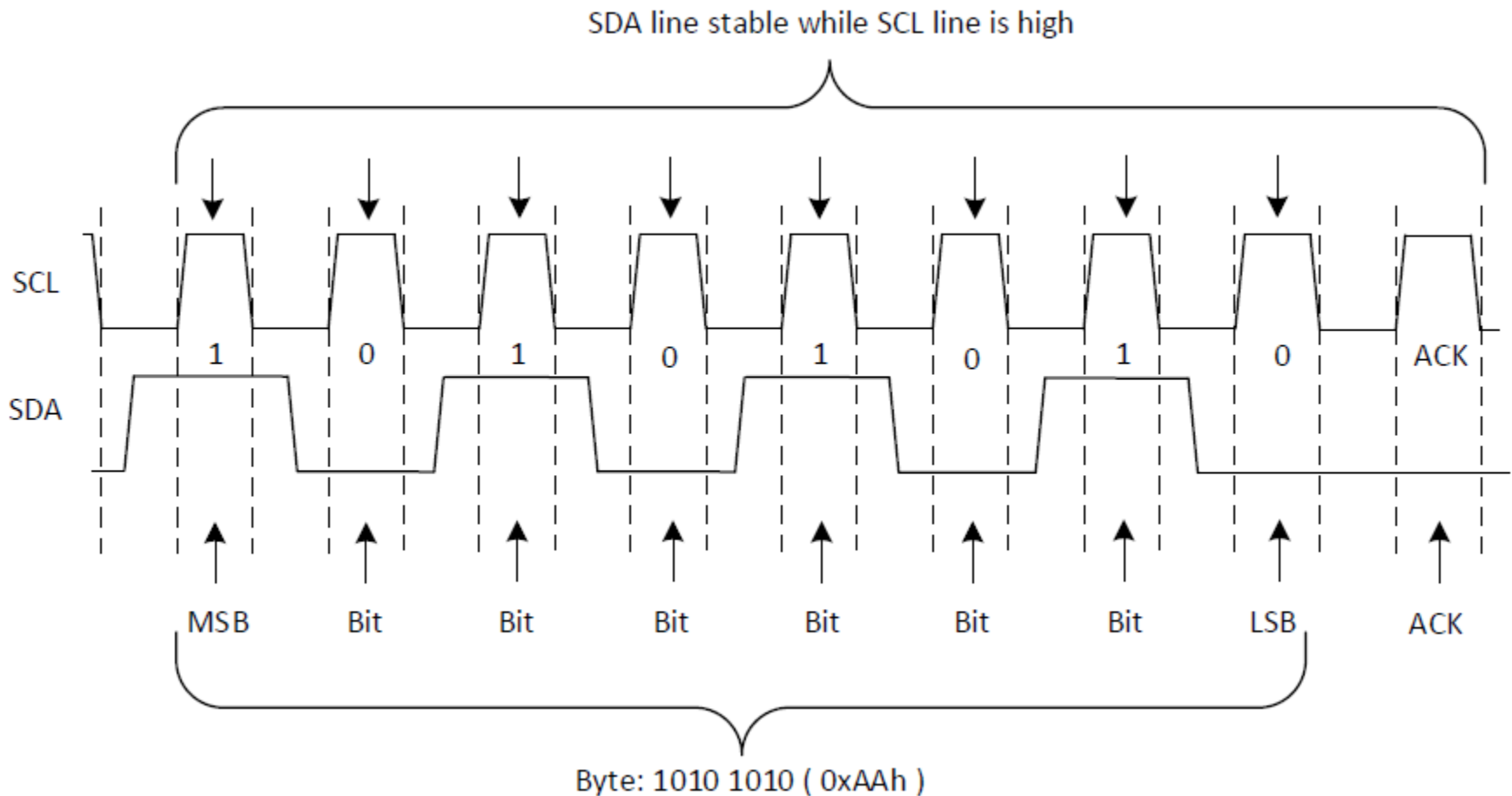


- Comunicatia I2C este initiata de dispozitivul master care trimite o conditie START si terminata de dispozitivul master care trimite o conditie STOP
- O tranzitie high-to-low pe linia SDA in timp ce linia SCL este la nivel ridicat defineste o conditie START
- O tranzitie low-to-high pe linia SDA in timp ce linia SCL este la nivel ridicat defineste o conditie STOP
- O conditie START repetata este similara unei conditii START si este folosita in locul unei conditii STOP urmata de o conditie START. Arata la fel ca o conditie START, dar difera de aceasta deoarece are loc inaintea unei conditii STOP, cand magistrala nu este inactiva.
- Este utila cand dispozitivul master doreste sa porneasca o noua comunicare, dar nu doreste sa lase magistrala sa treaca in inactivitate cu conditia STOP, care introduce posibilitatea ca dispozitivul master sa piarda controlul asupra magistralei, control care ar putea fi preluat de catre un alt dispozitiv master



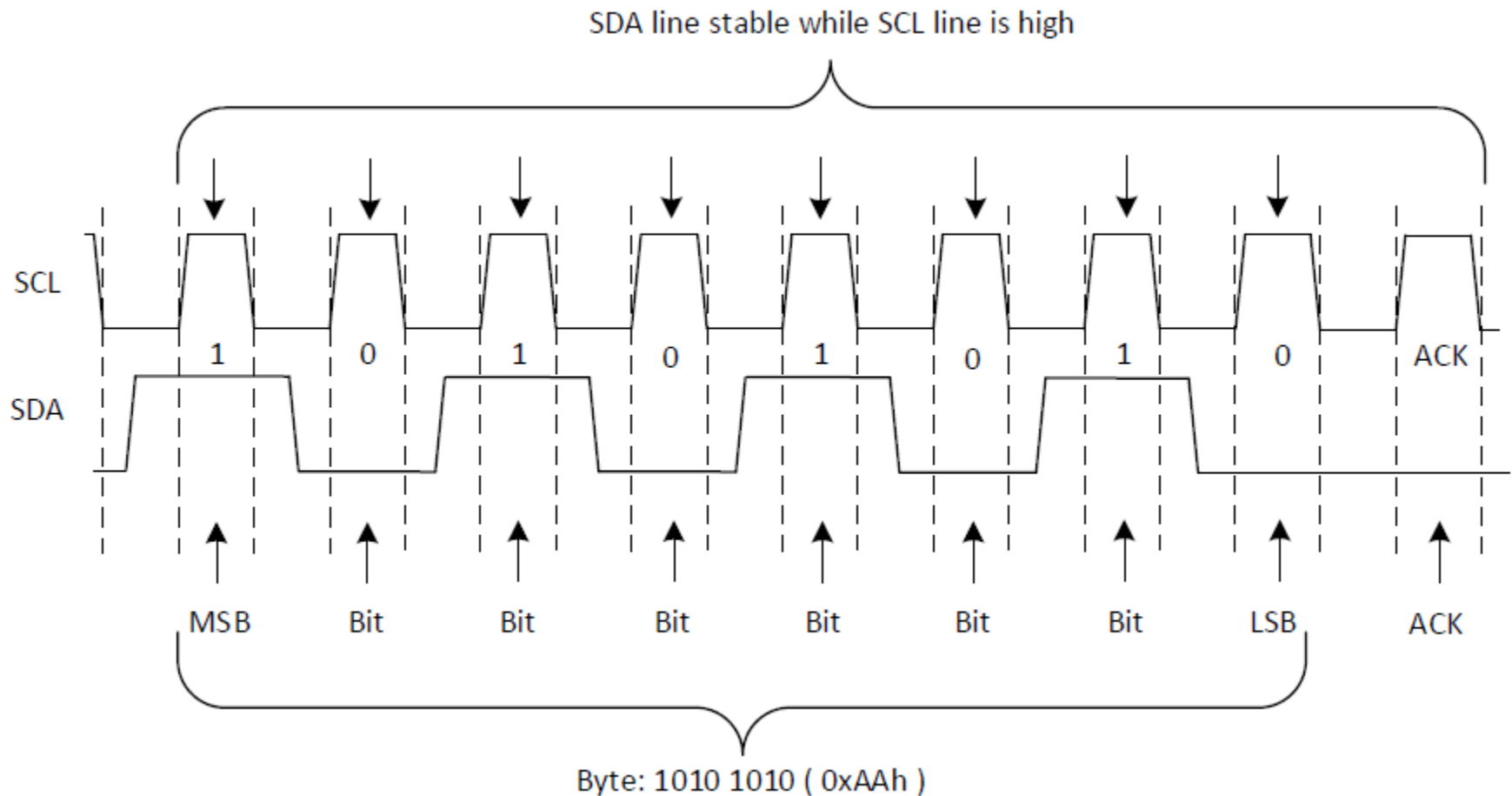
Exemplu de conditie START si conditie STOP

- Un bit de date este transferat pe linia SDA pe durata fiecarui puls de ceas pe linia SCL.
- Un octet poate fi o adresa de dispozitiv, o adresa de registru, date scrise la sau citite de la un dispozitiv slave
- Datele sunt transferate incepand cu Most Significant Bit (MSB)



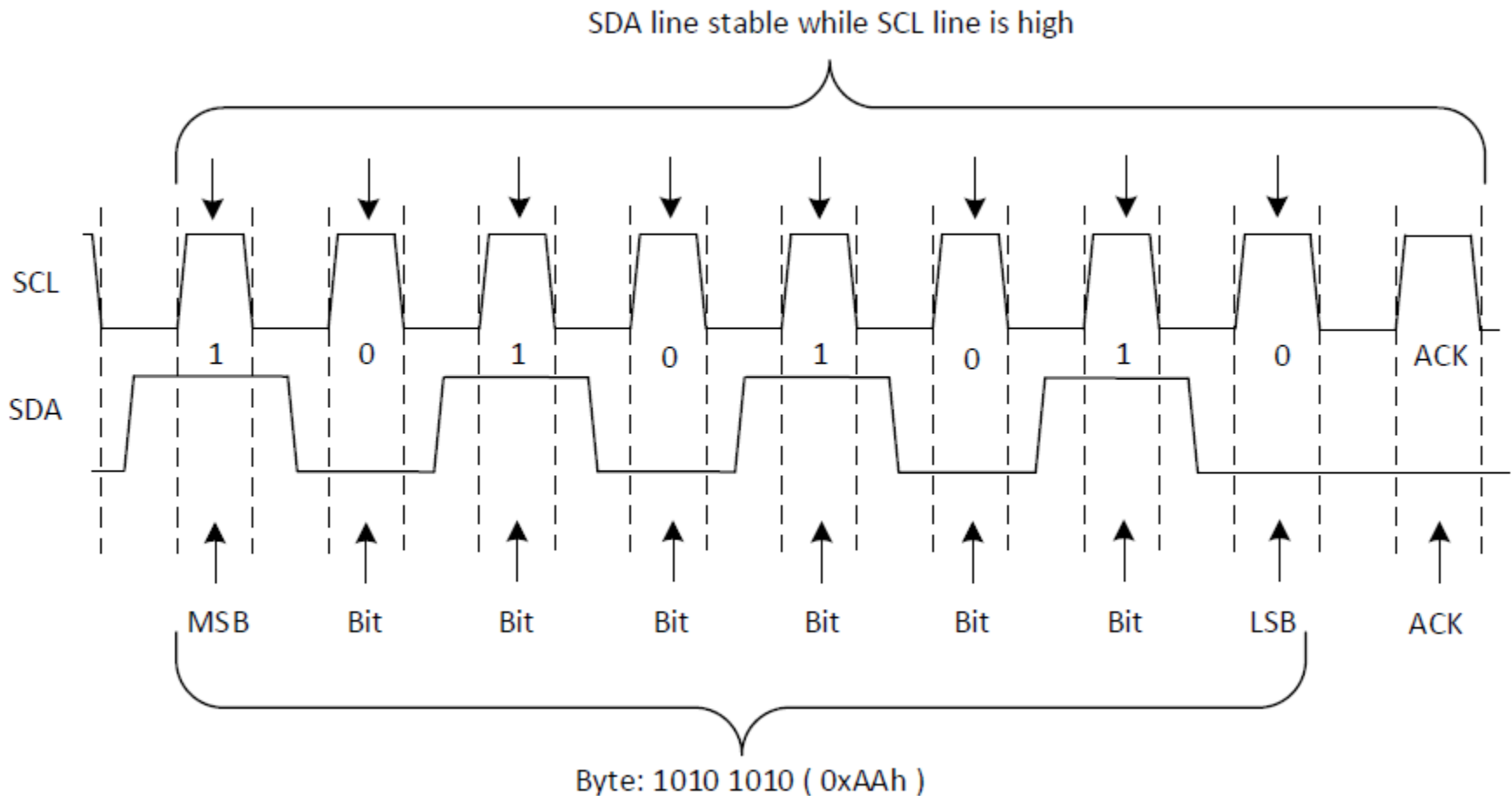
Exemplu de transfer de date

- Orice numar de octeti de date poate fi transferat de la master la slave intre conditiile START si STOP.
- Datele pe linia SDA trebuie sa ramana stabile pe durata fazei de nivel ridicat a perioadei de ceas deoarece modificari in linia de date cand SCL este la nivel ridicat sunt interpretate ca fiind comenzi de control (START si STOP)



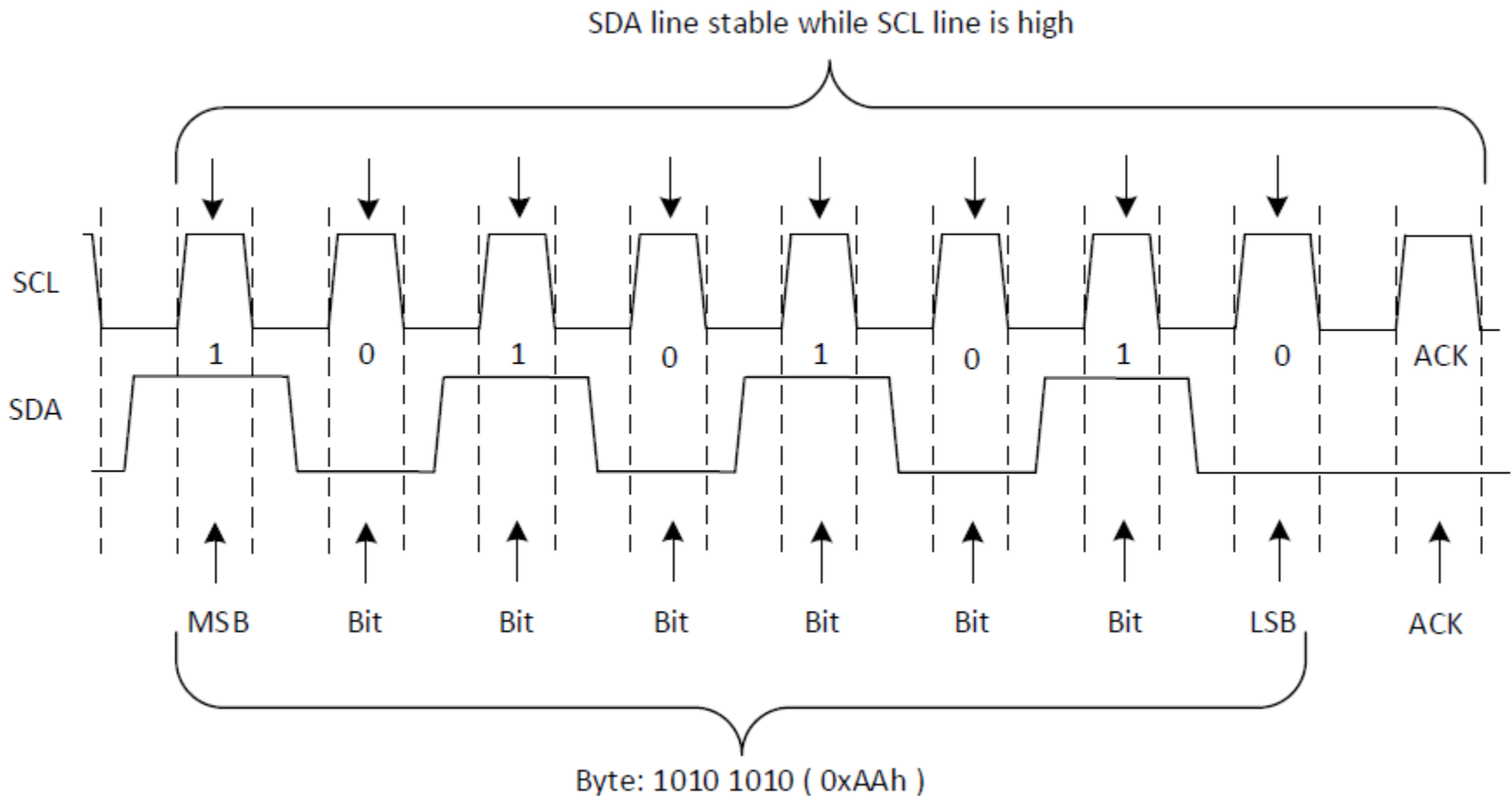
Exemplu de transfer de date

- Orice octet de date (incluzand octetul de adresa) este urmat de un bit de ACK de la receptor. Bitul ACK permite receptorului sa comunice transmitatorului ca octetul a fost receptionat cu succes si un alt octet poate fi trimis.



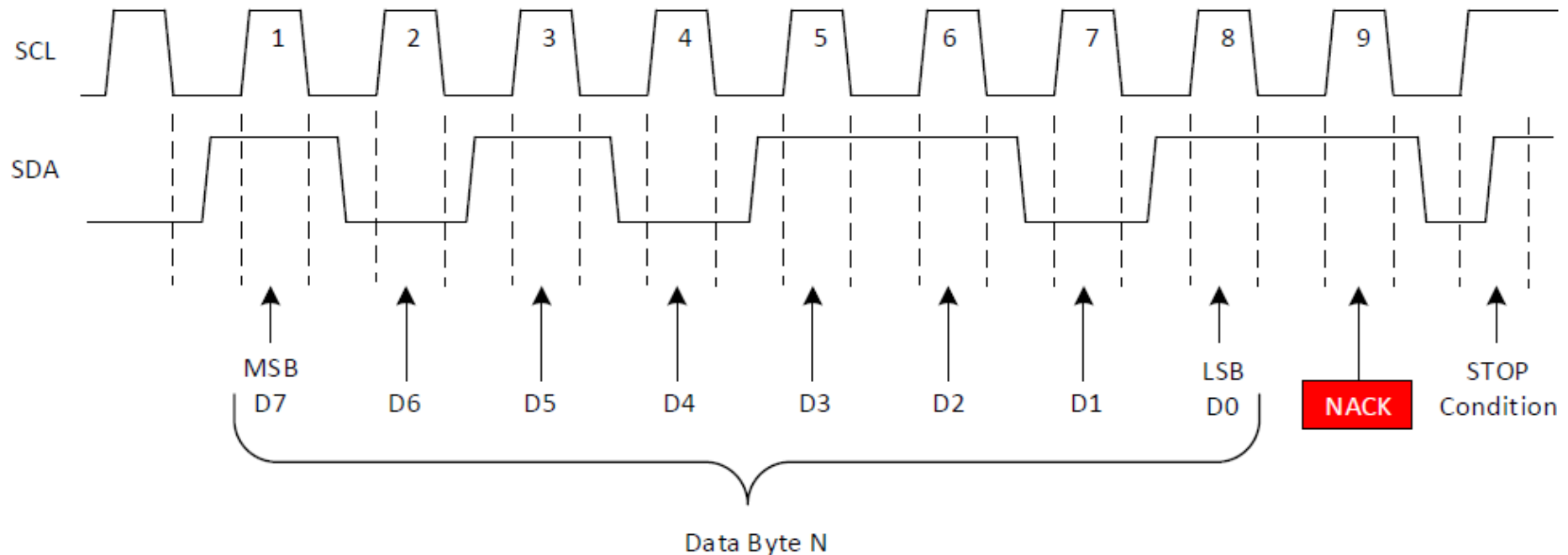
Exemplu de transfer de date si ACK

- Inainte ca receptorul sa poata sa trimita un ACK, transmitatorul trebuie sa elibereze linia SDA. Pentru a trimite un bit ACK, receptorul va trage in jos linia SDA pe durata fazei de nivel coborat a perioadei de ceas ACK/NACK-related (perioada 9), astfel incat linia SDA este stabila la nivel coborat pe durata fazei de nivel ridicat a perioadei de ceas ACK/NACK-related.



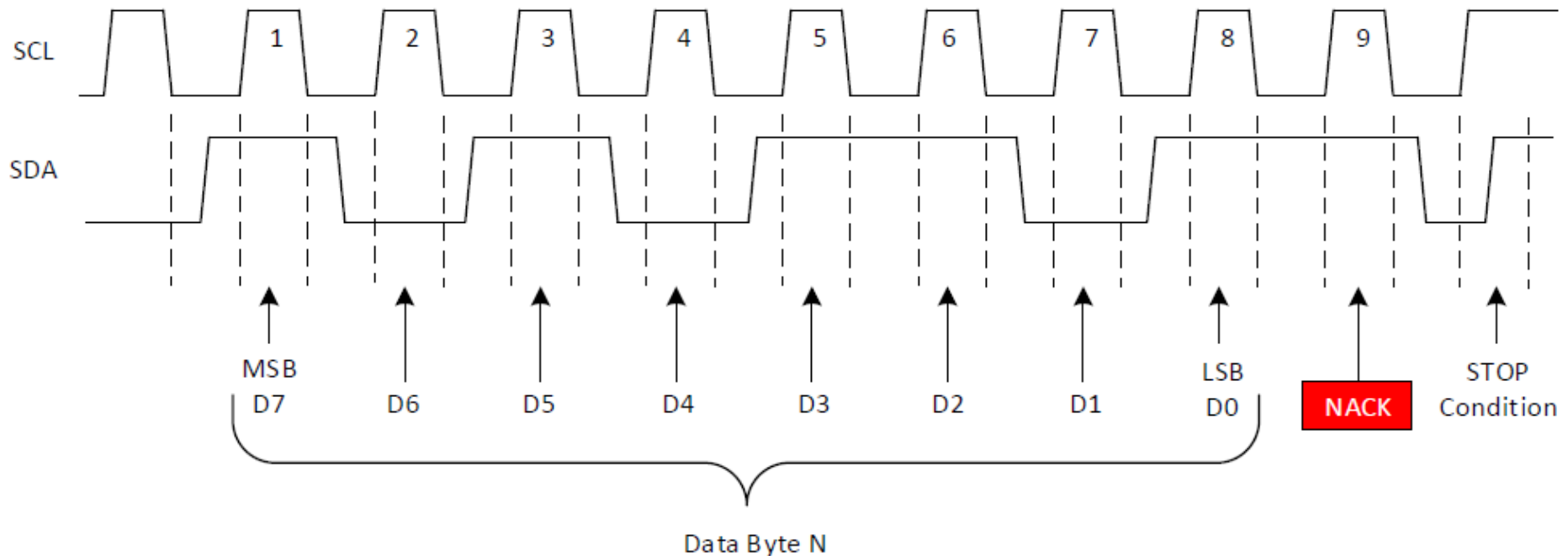
Exemplu de transfer de date si ACK

- Bitul este interpretat ca un NACK cand linia SDA ramane la nivel ridicat pe durata perioadei de ceas ACK/NACK-related



Exemplu de NACK

- Conditii care duc la generarea unui NACK:
- 1. Receptorul nu poate sa receptioneze sau sa transmita deoarece executa o functie de timp real si nu este pregatit sa inceapa comunicatia cu master-ul
- 2. Pe durata transferului, receptorul primeste date sau comenzi pe care nu le intelege
- 3. Pe durata transferului, receptorul nu mai poate receptiona octeti de date
- 4. Un master-receptor a terminat de citit datele si indica acest fapt dispozitivului slave printr-un NACK

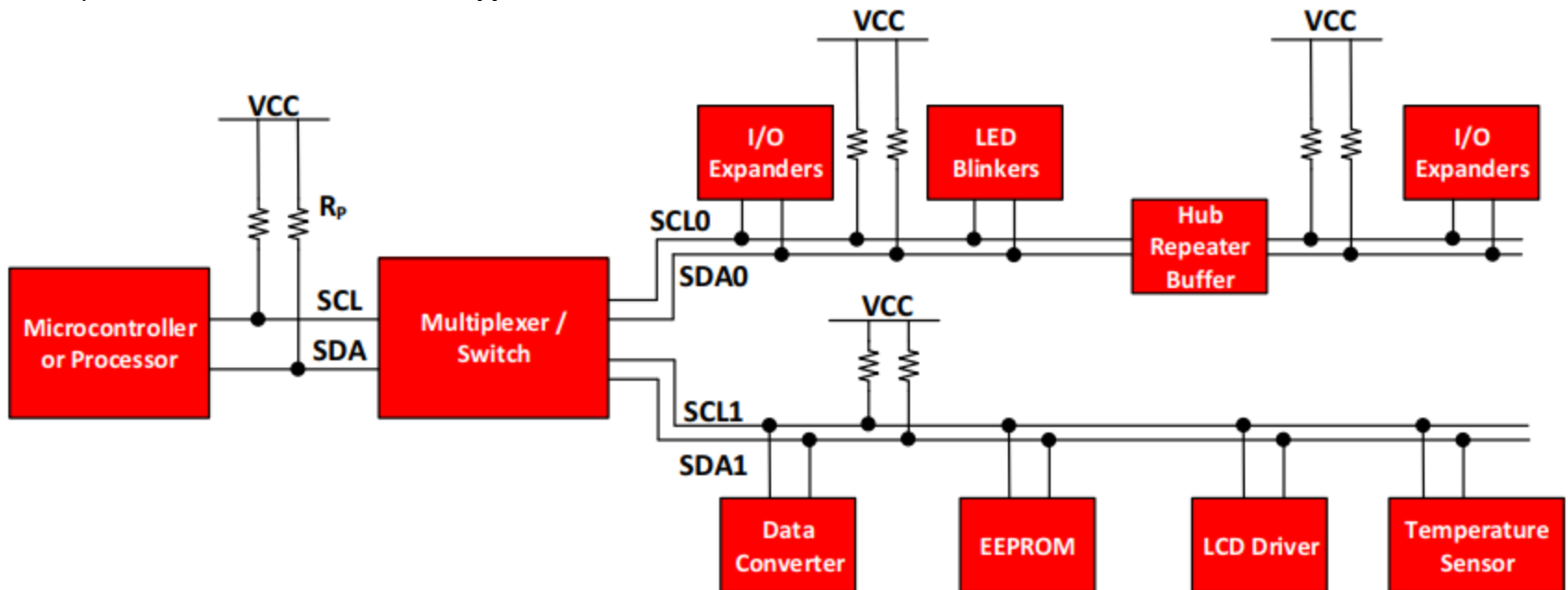


Exemplu de NACK

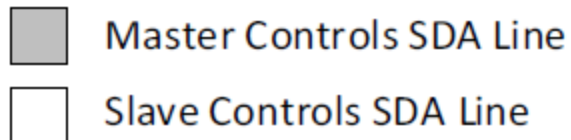


# Magistrala I<sup>2</sup>C – Datele I<sup>2</sup>C

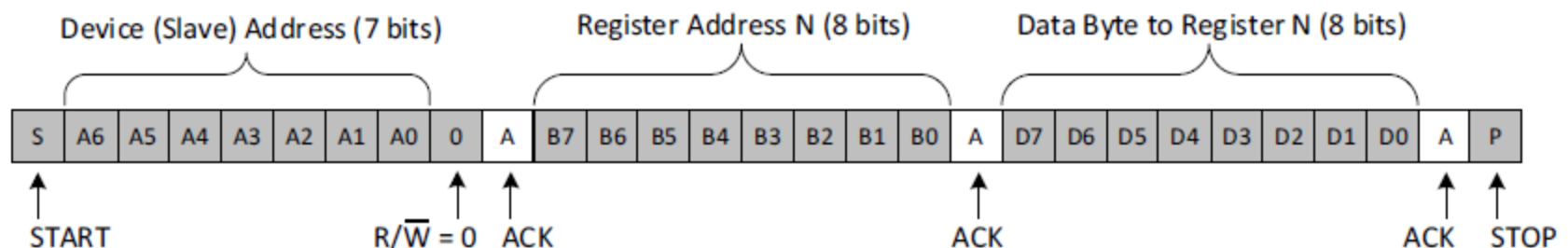
- Datele trebuie trimise si receptionate spre sau de la dispozitivele slave, dar modul în care se realizează acest lucru este prin citirea sau scrierea în sau din registrii din dispozitivul slave.
- Registrii sunt locații din memoria slave care conțin informații, fie că este vorba despre informații de configurare sau date de trimis înapoi către master. Master-ul trebuie să scrie informații în acești registrii pentru a instrui dispozitivul slave să execute o sarcină.
- Nu toate dispozitivele slave au mai multi registri. Unele au doar 1 registru, care poate fi scris direct prin trimiterea datelor registrului imediat după adresa slave-ului, în locul adresării registrului.



- Pentru a scrie pe magistrala I2C, master-ul va trimite o condiție START pe magistrală cu adresa slave-ului, precum și ultimul bit (bitul R/W) setat la 0, ceea ce înseamnă o scriere. După ce slave trimite bitul de confirmare, master-ul va trimite apoi adresa de registru a registrului în care dorește să scrie. Slave-ul va confirma din nou, anunțând master-ul că este pregătit. După aceasta, master-ul va începe să trimită datele registrului către slave, până când master-ul va trimite toate datele pe care trebuie să le trimită (uneori acestea sunt doar un singur octet), iar master-ul va termina transmisia cu o condiție STOP.

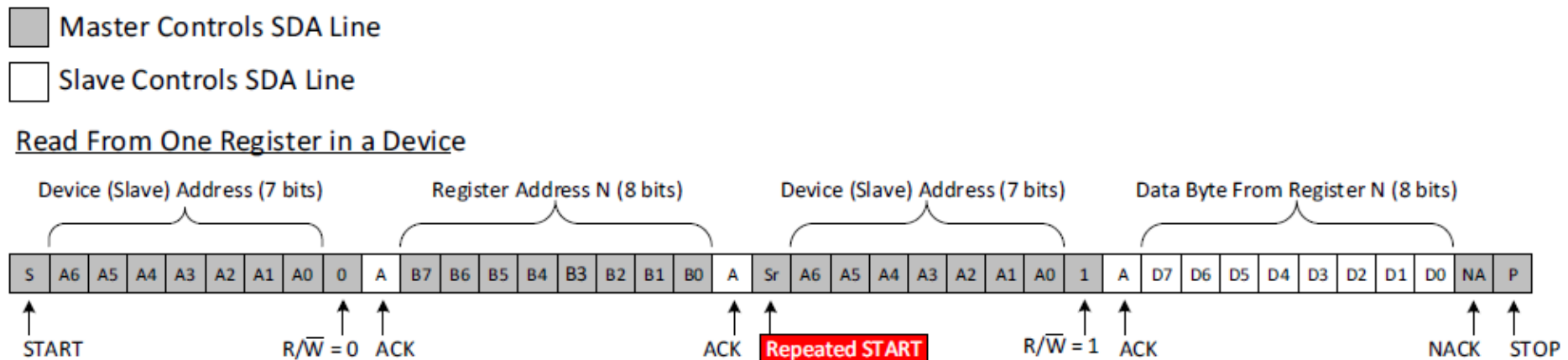


### Write to One Register in a Device



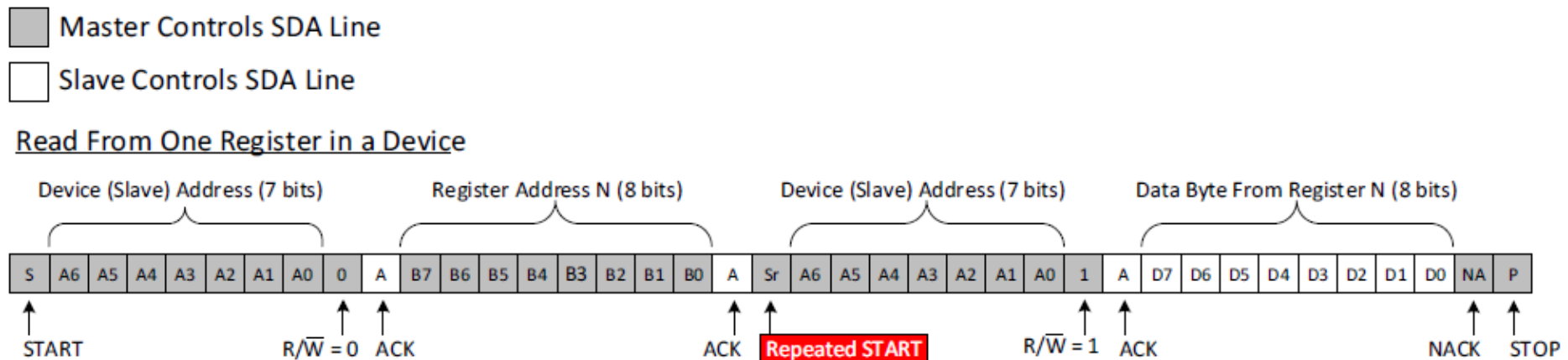
Exemplu de scriere in registrul dispozitivului slave

- Pentru a citi de la un slave, master-ul trebuie mai întâi să-l instruiască pe slave din ce registru dorește să citească. Acest lucru este realizat de către master care începe transmisia într-un mod similar cu scrierea, prin trimiterea adresei cu bitul R/W egal cu 0 (semnificând o scriere), urmată de adresa registrului din care dorește să citească. Odată ce slave confirmă această adresă de registru, master-ul va trimite din nou o condiție START, urmată de adresa slave-ului cu bitul R/W setat la 1 (semnificând o citire). De data aceasta, slave-ul va confirma cererea de citire, iar master-ul eliberează magistrala SDA, dar va continua să furnizeze ceasul slave-ului. În această parte a tranzacției, master-ul va deveni master-receptor, iar slave-ul va deveni slave-transmițător.



Exemplu de citire din registrul dispozitivului slave

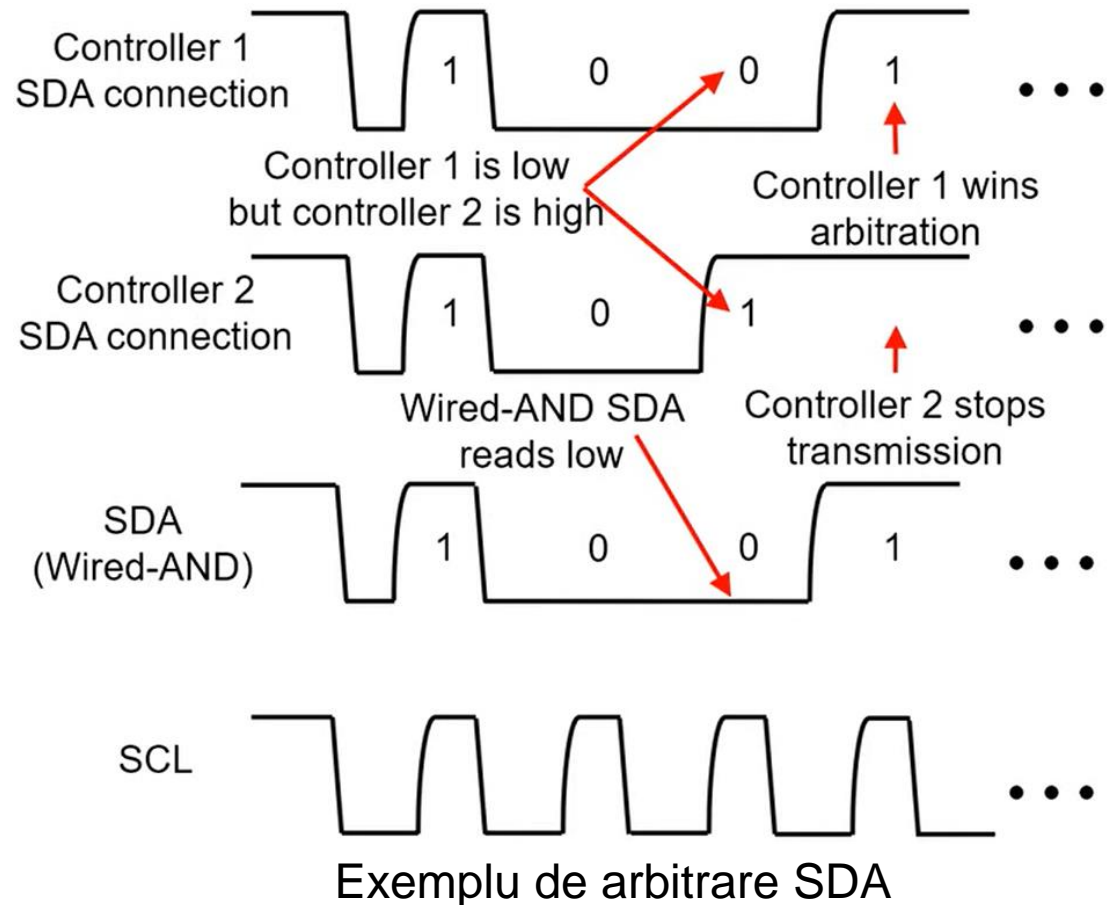
- Master-ul va continua să trimită impulsurile de ceas, dar va elibera linia SDA, astfel încât slave-ul să poată transmite date. La sfârșitul fiecărui octet de date, master-ul va trimite un ACK către slave, anunțându-l pe slave că este pregătit pentru mai multe date. Odată ce master-ul a primit numărul de octeți pe care îi aștepta, va trimite un NACK, semnalând slave-ului să oprească comunicațiile și să elibereze magistrala. Master-ul va continua cu o condiție STOP



Exemplu de citire din registrul dispozitivului slave

# Magistrala I<sup>2</sup>C – Arbitrarea SDA

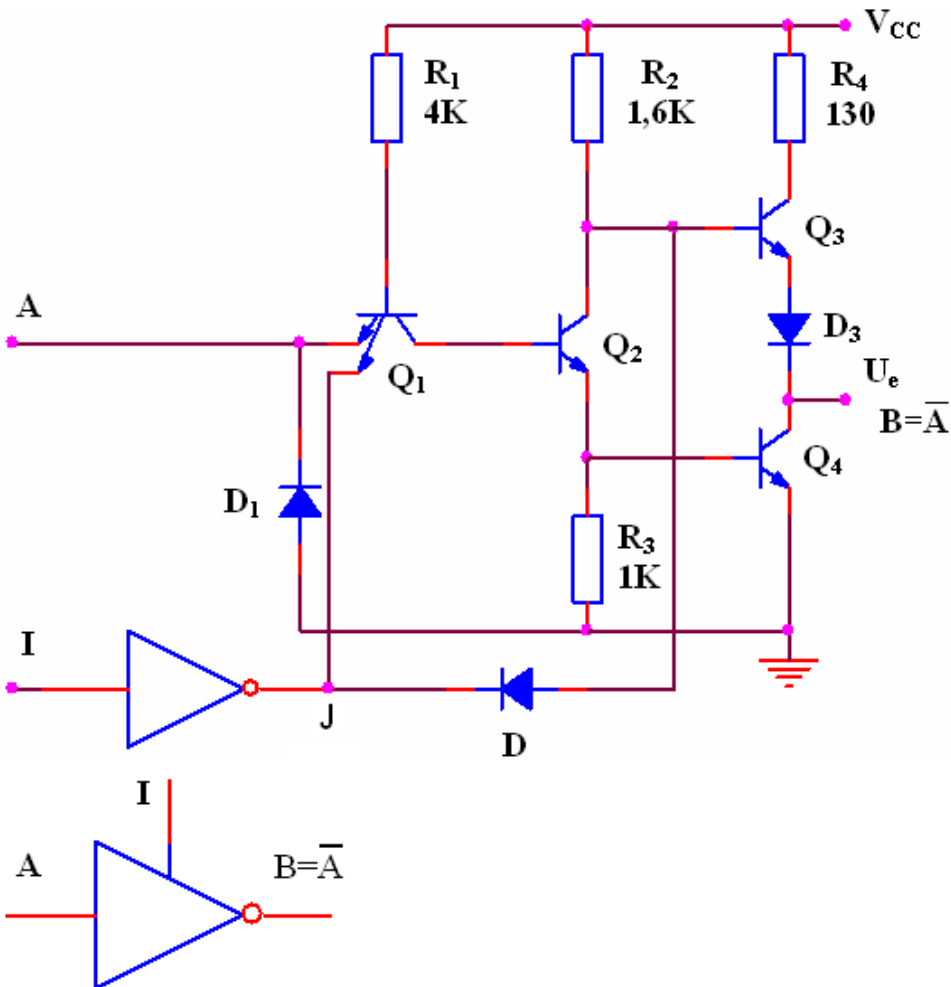
- Master-ele sincronizeaza SCL
- Arbitrarea este realizata pe SDA folosind un SCL sincronizat
- Master-ele transmit date pe SDA dar monitorizeaza fiecare bit
- Primul master care transmite un nivel coborat in timp ce celalalt master transmite un nivel ridicat, castiga arbitrarea
- Master-ul care nu castiga arbitrarea opreste transmisia si asteapta conditia STOP



# Circuite TTL cu trei stări

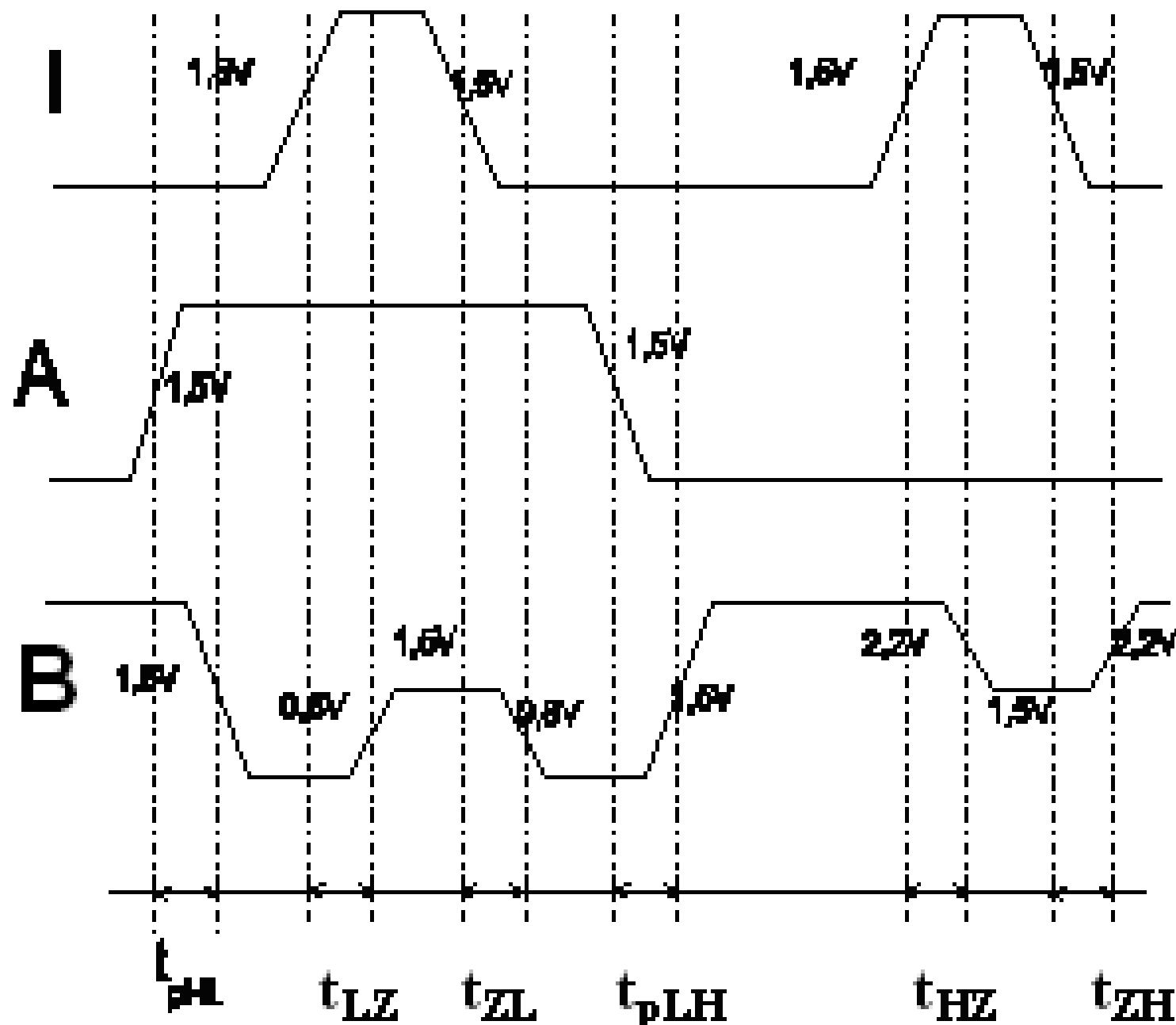
- Blocarea ambelor tranzistoare din circuitul de ieșire
- Circuitul de ieșire este izolat
- Văzută dinspre ieșire, poarta TTL se prezintă ca o impedanță mare
- Circuitul dispune de trei stări: “0”, “1” și starea de impedanță mare, care lasă ieșirea flotantă atunci când cele două tranzistoare sunt blocate

# Inversorul TTL cu trei stări



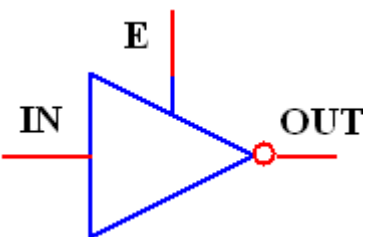
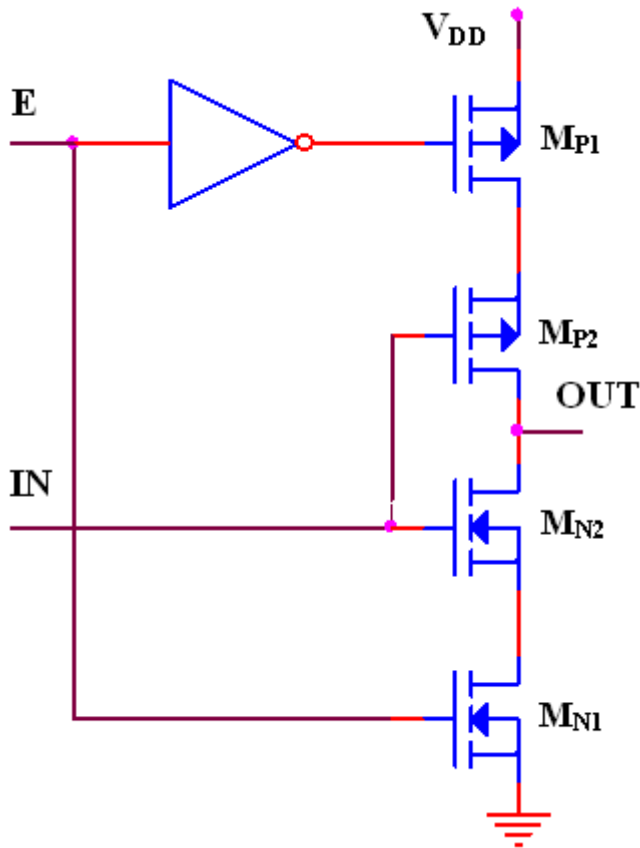
- I="0" - inversor obișnuit
- I="1", J="0", D deschis, Q<sub>1</sub> saturat, Q<sub>2</sub> și Q<sub>4</sub> blocate, Q<sub>3</sub> blocat deoarece prin dioda D deschisă potențialul bazei sale coboară la 0.7V
- Circuitul va prezenta la ieșire o impedanță mare (HZ - înaltă impedanță)
- În regim dinamic, pe lângă  $t_{pLH}$  și  $t_{pHL}$  apar următorii parametri:
- timpii de stabilire a impedanței ridicate plecând din "0",  $t_{LZ}$ , respectiv din "1",  $t_{HZ}$
- timpii de ieșire din starea de înaltă impedanță și trecerea în "0",  $t_{ZL}$ , respectiv în "1",  $t_{ZH}$

- timpul total de propagare este aproximativ 25 ns. Această valoare este superioară porților TTL obișnuite dar este inferioară circuitelor cu colector în gol
- $P_D=16mW$





# Inversorul CMOS cu trei stări

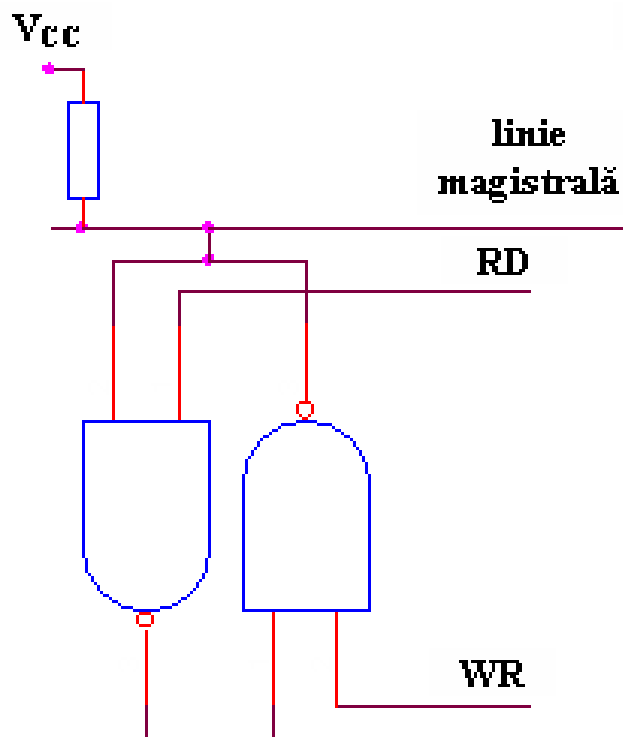


- Două tranzistoare cu canal n și două tranzistoare cu canal p
- O pereche de tranzistoare p-n operează cu funcție de inversor standard
- A doua pereche funcționează ca un comutator închis-deschis comandat de intrarea de validare  $E$  (enable)
- Dacă intrarea  $E$  este în "1",  $M_{N1}$  și  $M_{P1}$  sunt deschise iar ieșirea poate prezenta nivelele "1" și "0"
- Când intrarea  $E$  este în "0",  $M_{N1}$  și  $M_{P1}$  sunt blocate și indiferent de nivelele logice prezentate la intrare impedanța de ieșire este ridicată (mai mare de  $10^{10}\Omega$  la  $25^\circ\text{C}$ ).

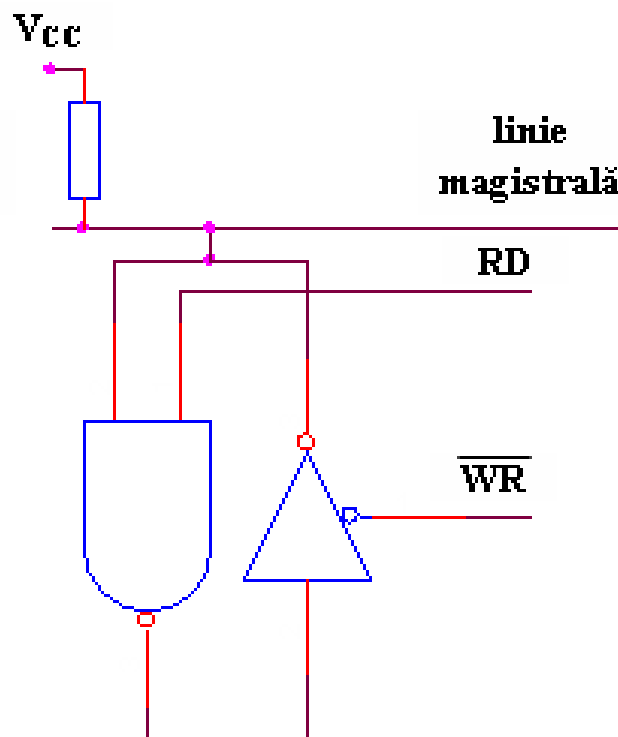
# Conectarea circuitelor la magistrala

- Unui circuit legat la o magistrală i se atribuie, de regulă, în sistem o funcționare atât de emițător, cât și de receptor. În acest caz, intrările de comandă trebuie să permită atât citirea unui cuvânt de pe magistrală, cât și scrierea unui cuvânt pe magistrală
- Printr-un semnal de comandă RD\ cuvântul de pe magistrală este introdus în circuit
- printr-un semnal de comandă WR\ cuvântul din circuit este înscris pe magistrală

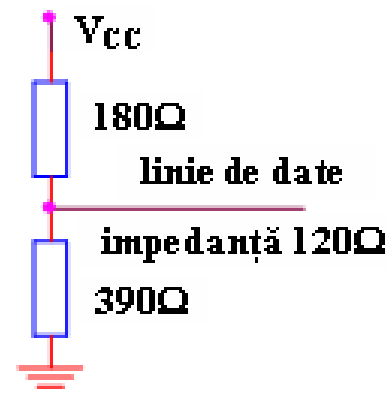
Pentru ca atunci când nu există activitate pe magistrală toate liniile să aibă același potențial, acestea se conectează prin rezistențe de ridicare la "1". Dacă la magistrală sunt cuplate numai circuite TTL, în locul rezistențelor de ridicare, se pot utiliza terminatori de magistrală (grup de rezistențe montate la extremitățile traseelor magistralei pentru adaptarea împotriva reflexiilor).



a)



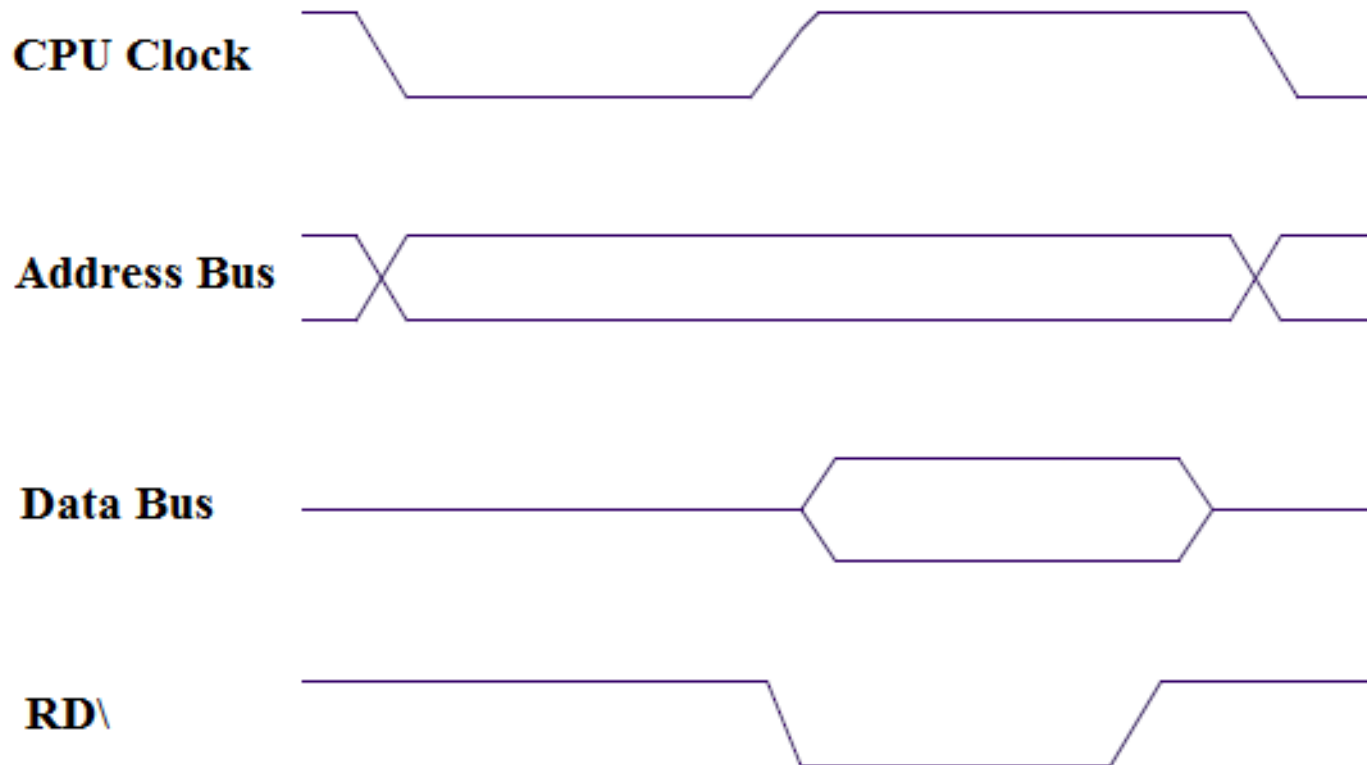
b)



c)

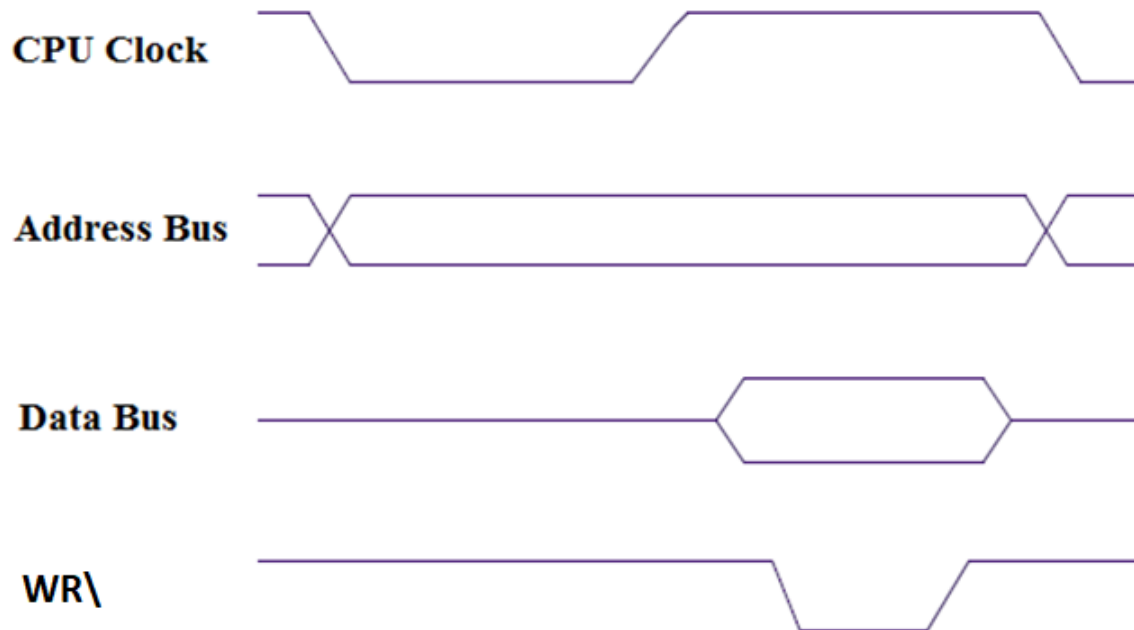
## Exemplu de ciclu CPU de citire

- CPU transmite adresa pe magistrala de adrese
- Semnalul de control RD\ este comutat in 0 logic de catre CPU pentru a semnaliza dispozitivului extern ca este pregatit sa preia datele de pe magistrala de date
- CPU citește magistrala de date



## Exemplu de ciclu CPU de scriere

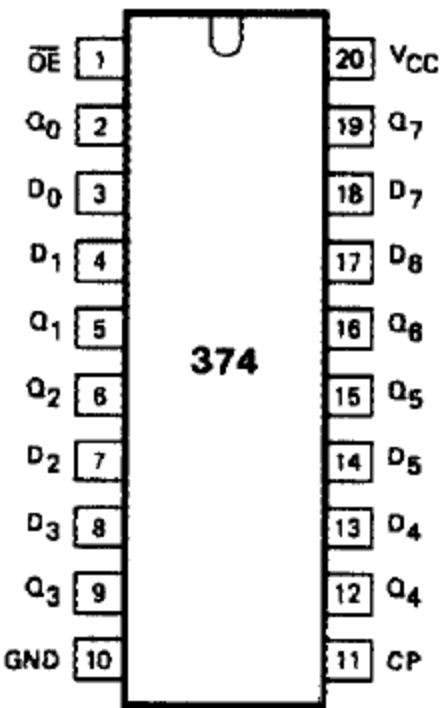
- CPU transmite adresa pe magistrala de adrese
- CPU furnizeaza datele pe magistrala de date
- Semnalul de control WR\ este comutat in 0 logic de catre CPU pentru a semnaliza dispozitivului extern ca datele sunt pregatite pentru a fi preluate de pe magistrala de date
- Dispozitivul extern preia datele de pe magistrala de date



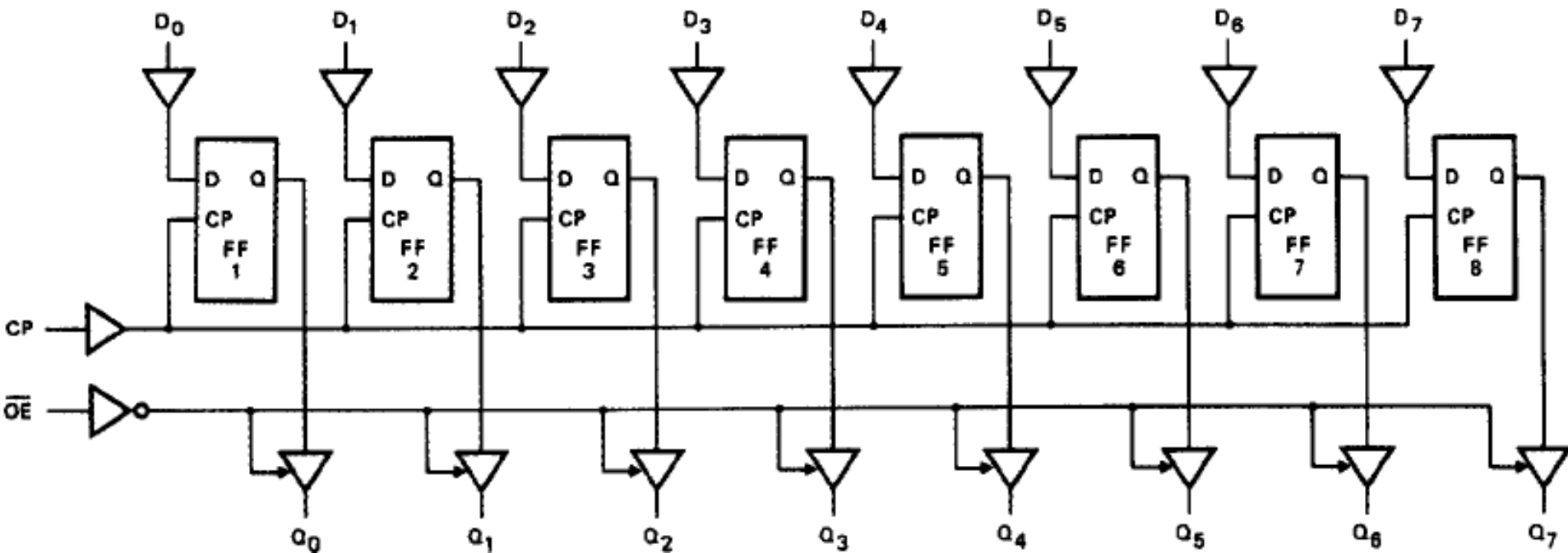
# TRANSFERUL ÎNTRE REGISTRE ȘI LOGICA TREI STĂRI

- Formarea magistralelor este posibilă prin folosirea logicii cu trei stări (*Three State Logic*) la realizarea ieșirilor registrelor
- Pentru transferul informației pe liniile magistralelor se folosesc registre cu trei stări, amplificatoare de magistrală cu trei stări (unidirectionale sau bidirectionale) precum și decodificatoare

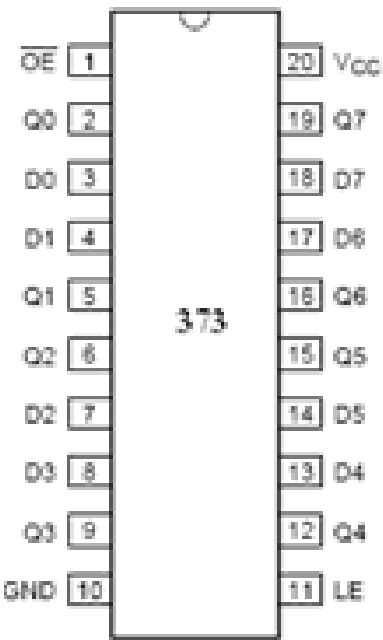
# Registrul 74374



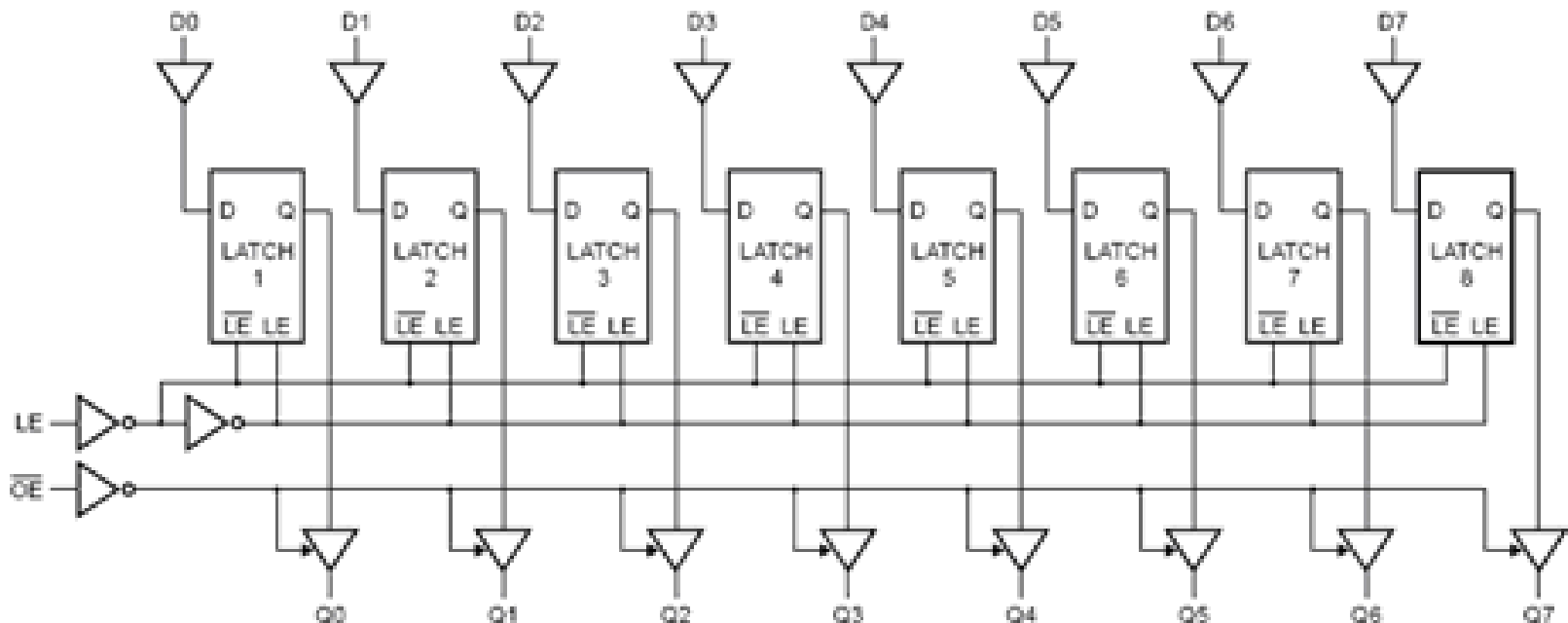
- 8 bistabile D ale căror ieșiri sunt bufferate cu circuite trei stări, activate prin intermediul intrării  $\overline{OE}$  activă pe nivel cobarat
- Datele vor fi valide la iesire doar daca  $\overline{OE}$  este '0'
- Scrierea datelor in registru se face pe frontul pozitiv al semnalului CP



# Registrul 74373



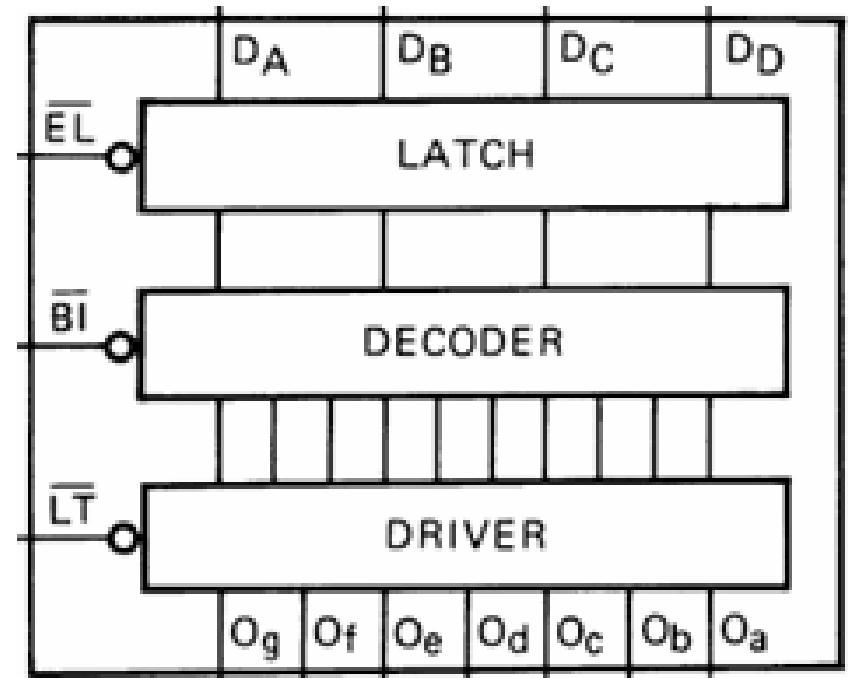
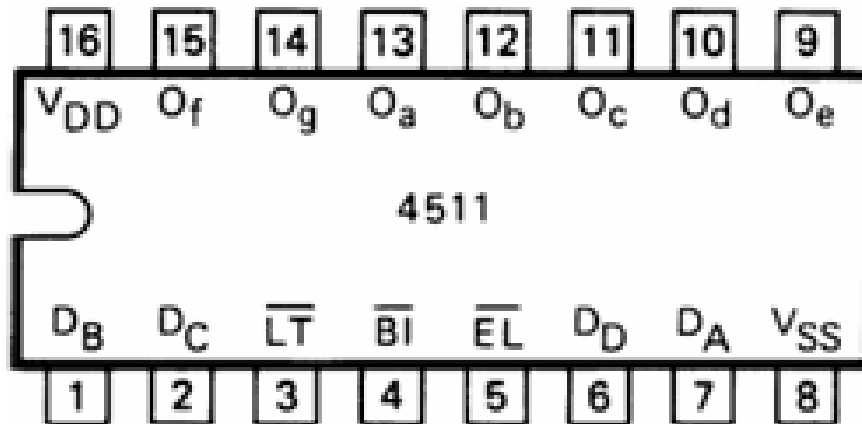
- 8 bistabile D ale căror ieșiri sunt bufferate cu circuite trei stări, activate prin intermediul intrării OE\ activă pe nivel coborat
- Datele vor fi valide la ieșire doar dacă OE\ este '0'.
- Datele sunt transmise la ieșirea bistabilelor atât timp cat semnalul LE se afla în '1'.
- La trecerea semnalului LE din '1' în '0' datele sunt memorate în latch.



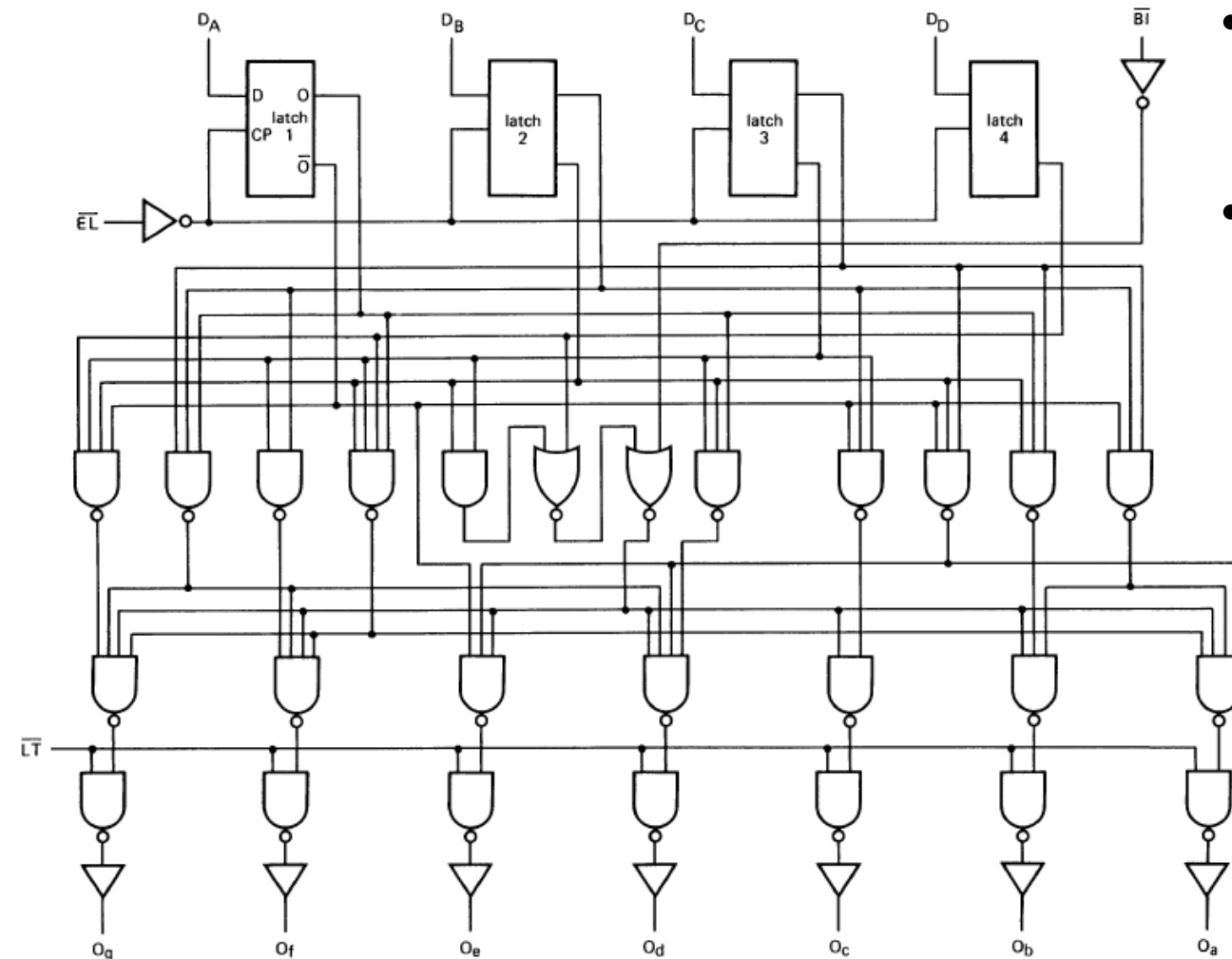


# 4511 - latch, decodificator, driver

- Comanda unor afișoare cu LED-uri 7-segmente
- Functii: latch, decodificator BCD la 7 segmente și driver
- 4 intrări de date ( $D_A$ - $D_D$ )
- 1 intrare de activare a latch-ului activă '0' ( $\overline{EL}$ )
- 1 intrare de blanking (stingerea segmentelor) activă '0' ( $\overline{BI}$ )
- 1 intrare de test lamp (aprinderea segmentelor) activă '0' ( $\overline{LT}$ )
- 7 ieșiri pentru segmente active '1' ( $O_a$ - $O_g$ )



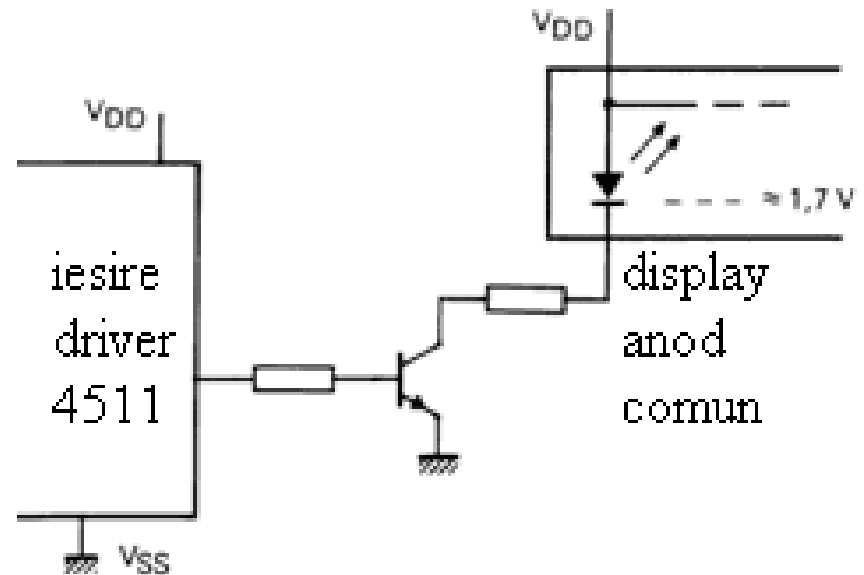
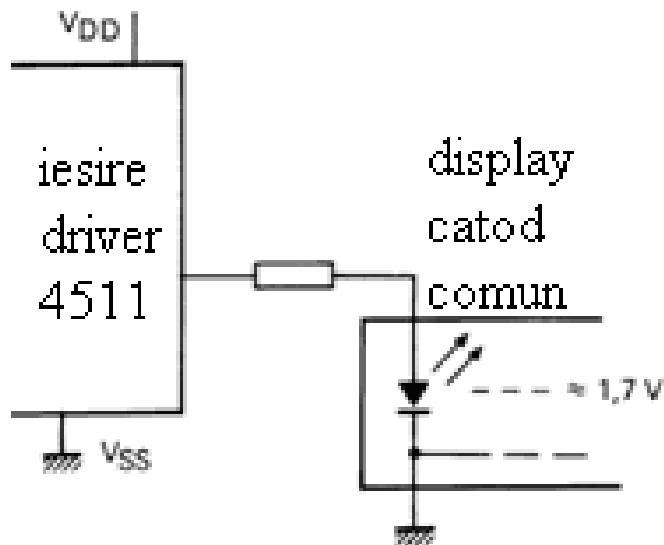
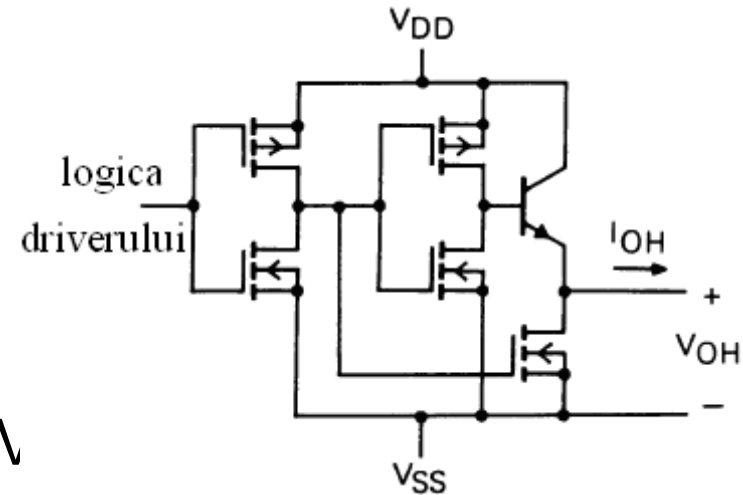
- EL\ este '0',  $O_a-O_g$  sunt determinate de către  $D_A-D_D$
- EL\ este '1', ultimele date la  $D_A-D_D$  sunt memorate în latch-uri și  $O_a-O_g$  sunt stabile
- LT\ este '0',  $O_a-O_g$  sunt '1' indiferent de celelalte condiții de intrare



- LT\ este '1', '0' la BI\ forțează  $O_a-O_g$  în '0'
- LT\ și BI\ sunt '1',  $O_a-O_g$  depinde de comenzile decodicatorului

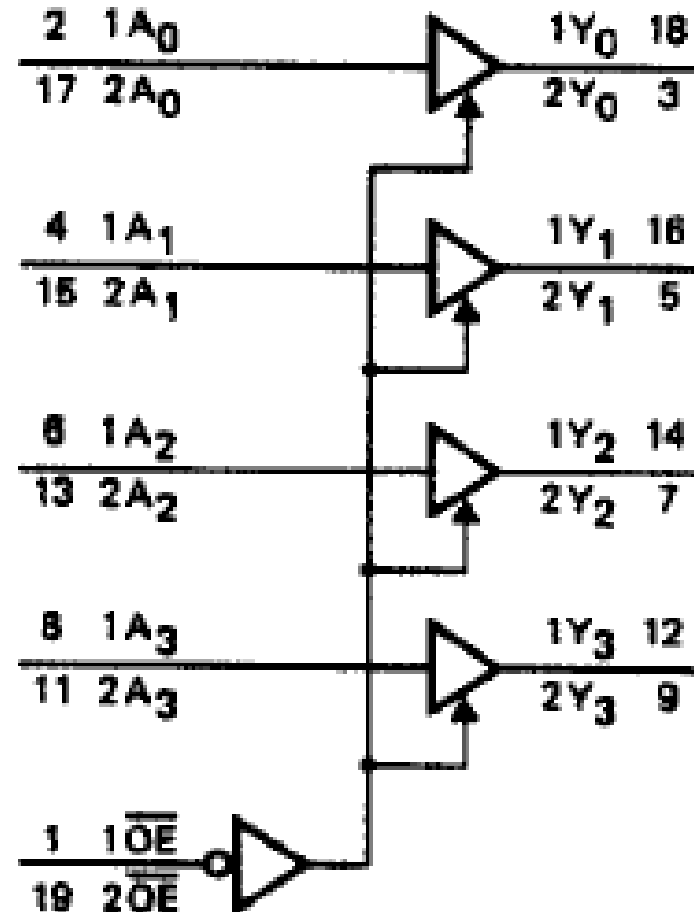
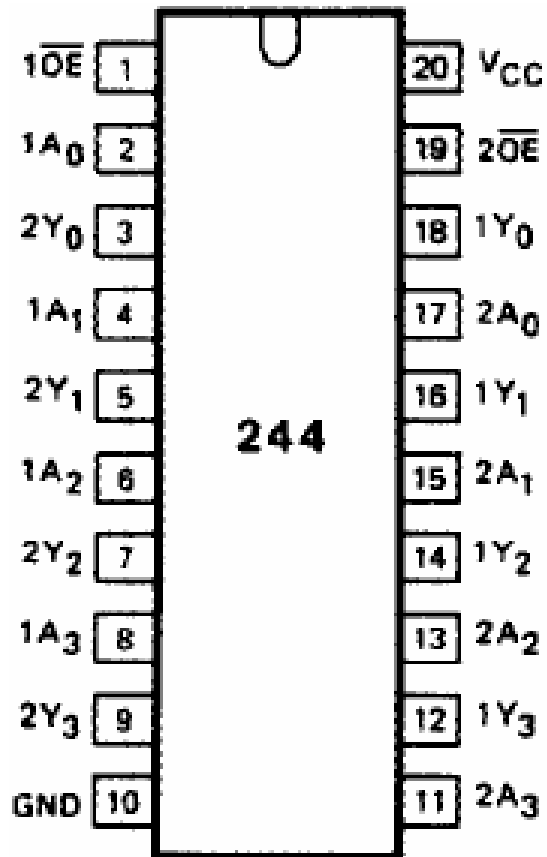


- Pentru a asigura un curent de ieșire mare dispozitivul de ieșire este un tranzistor bipolar
- Tensiunea de ieșire este dependentă de valoarea curentului generat
- Pentru  $V_{DD}=5V$ :
- $I_{OH}=0mA \rightarrow V_{OHmin}=4,10V, V_{OHtyp}=4,40V$
- $I_{OH}=10mA \rightarrow V_{OHmin}=3,60V, V_{OHtyp}=4,25V$
- $I_{OH}=20mA \rightarrow V_{OHmin}=2,80V, V_{OHtyp}=4,20V$



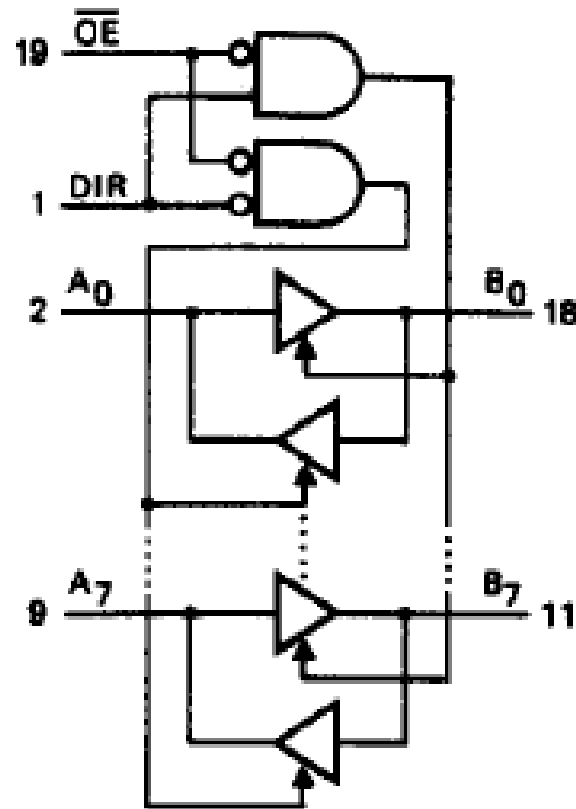
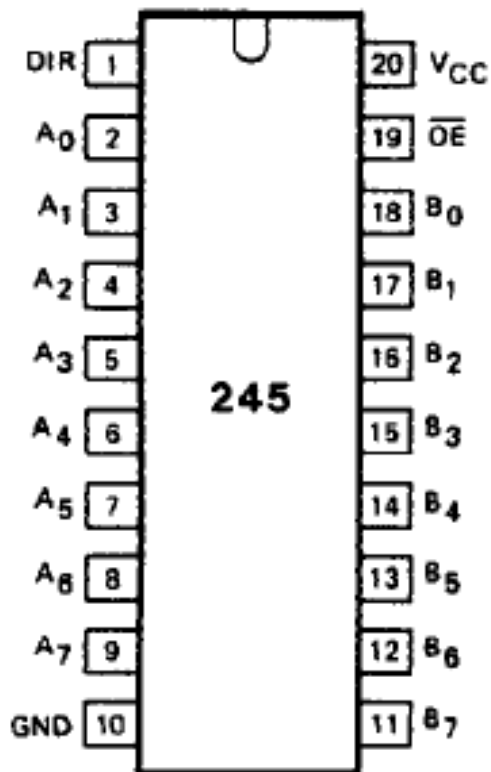
# Amplificatorul unidirectional 74244

- 8 amplificatoare cu trei stări ale caror iesiri sunt activate prin intermediul intrărilor  $1OE$  și  $2OE$  active pe '0'
- Intrările de activare permit activarea independentă a câte 4 iesiri:  $1Y_0 - 1Y_3$  activate de către  $1OE$  iar  $2Y_0 - 2Y_3$  activate de către  $2OE$

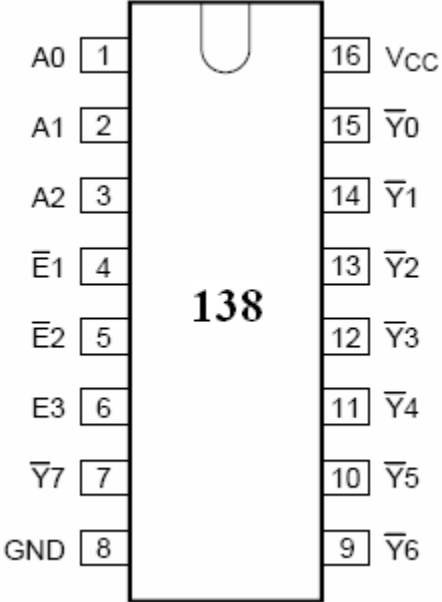


# Amplificatorul bidirectional 74245

- 16 amplificatoare cu trei stări, cate 8 pentru fiecare sens
- Starea logica a intrarii DIR permite selectia grupului de 8 amplificatoare ale caror iesiri vor fi activate: DIR='1' datele sunt transmise de la  $A_i$  la  $B_i$ , DIR='0' datele sunt transmise de la  $B_i$  la  $A_i$
- $\overline{OE}$ ='1' permite dezactivarea tuturor iesirilor,  $\overline{OE}$ ='0' iesirile vor fi activate in functie de starea logica a intrarii DIR

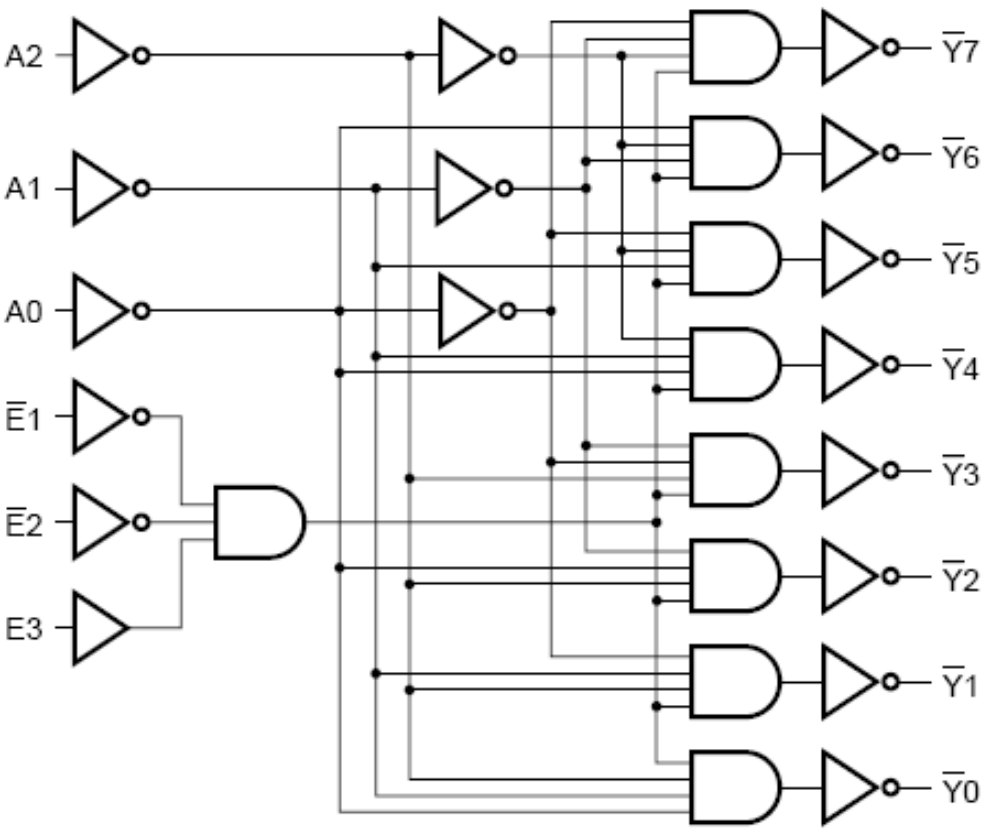


# Decodificatorul 74138



3 intrari de adresa, 8 iesiri active '0' si 3 semnale de validare

Daca  $E1 \neq '0'$ ,  $E2 \neq '0'$  si  $E3 = '1'$  iesirea  $Y_i$  corespunzatoare combinatiei binare de la intrarile de adresa A0, A1 si A2 va avea valoarea '0' restul iesirilor avand valoarea '1'



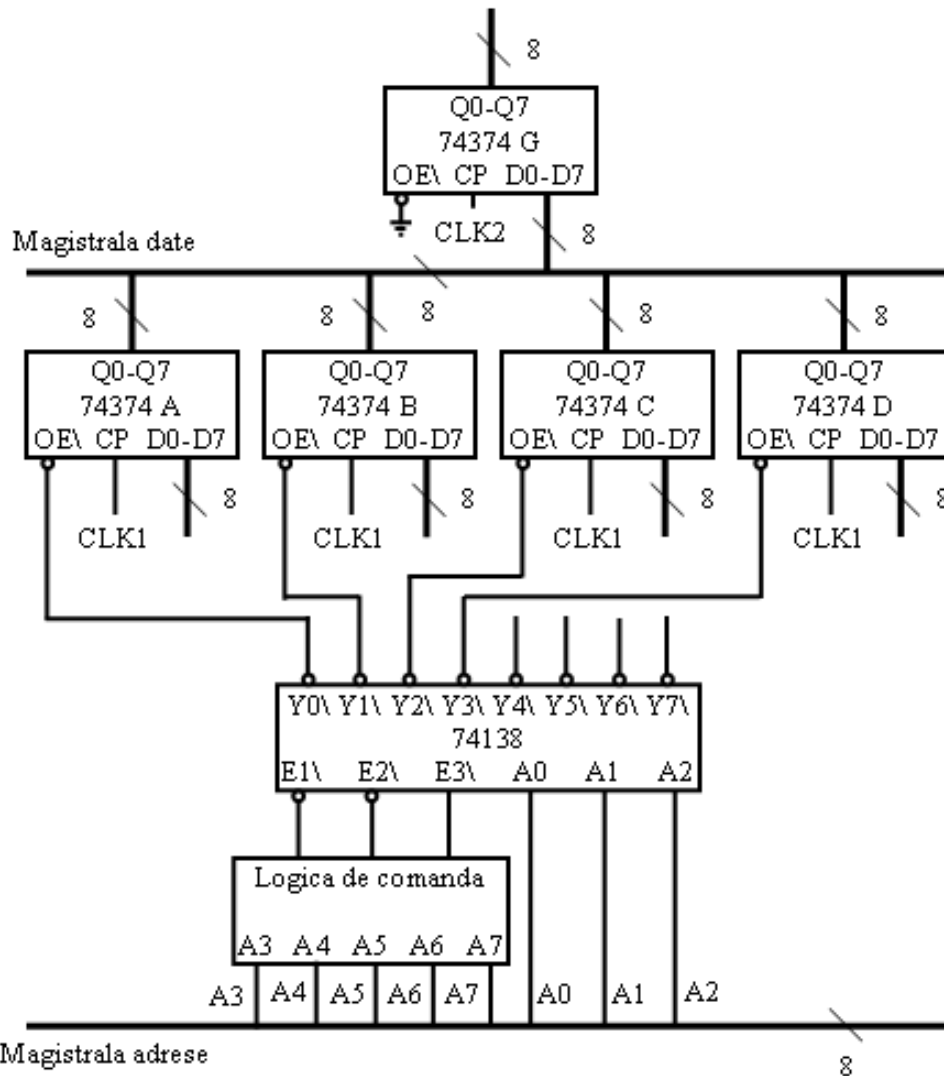
Pentru orice alte valori ale intrarilor de validare toate iesirile vor avea valoarea '1'

Invalidarea circuitului nu presupune trecerea iesirilor in starea de inalta impedanta ci setarea la valoarea '1' a acestora astfel incat iesirile nu vor selecta alte circuite

# TRANSFERUL ÎNTRE REGISTRE

Informația din unul dintre cele patru registre A, B, C sau D, este trecută într-un al cincilea registru, G

Magistrala comuna de 8 linii de date, fie date de ieșire ale registrelor A, B, C sau D, fie date de intrare pentru registrul G



Activarea unuia dintre cele patru registre se face prin decodificarea unor linii de adresă ale magistralei de adrese care face posibilă activarea liniei OE\ corespunzătoare

Pentru selectia registrelor se folosesc toate liniile de adresa (selectie totala)

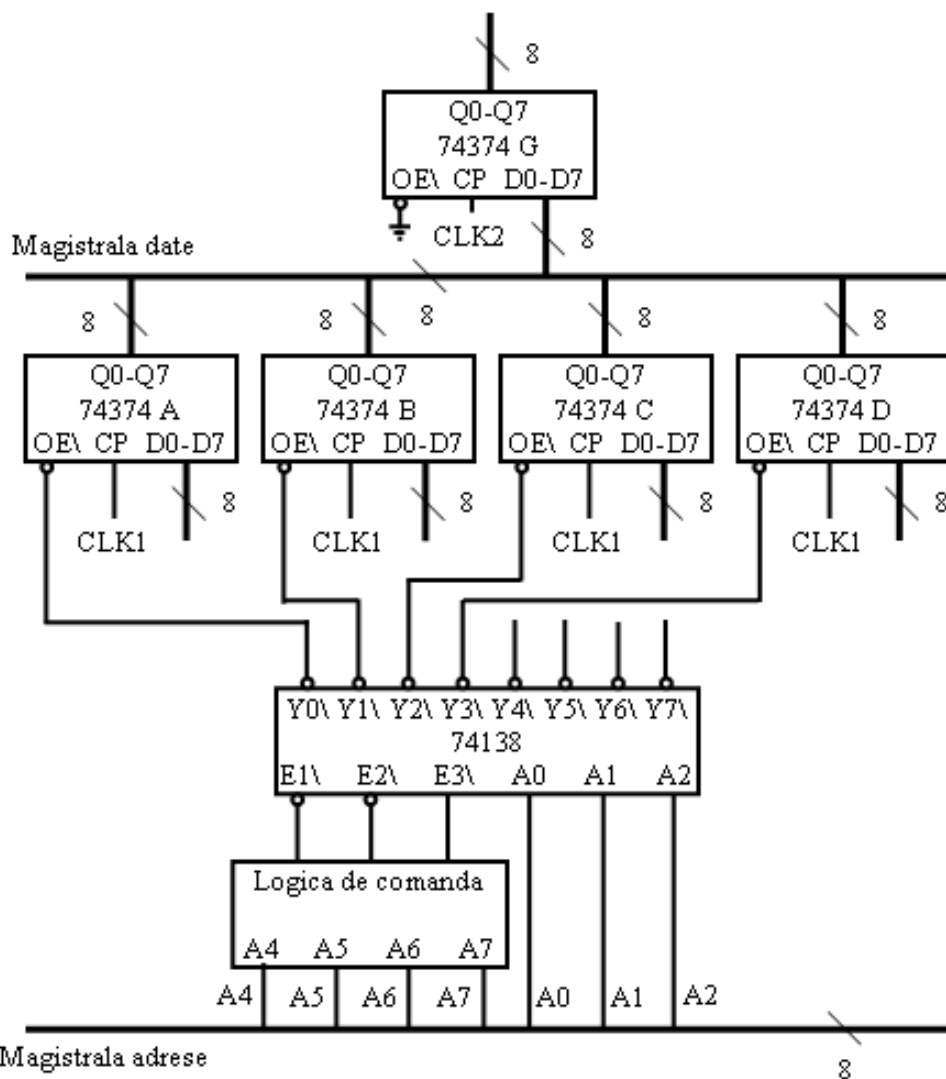
Avantaj - se poate folosi tot spatiul de adrese

Dezavantaj - complexitatea circuitului de selectie

Scrierea datelor in registre se realizeaza pe frontul crescator al semnalelor CLK1 si CLK2

Registrul G este activat tot timpul





Pentru selectia registrelor se folosesc o parte din liniile de adresa (selectie partiala)

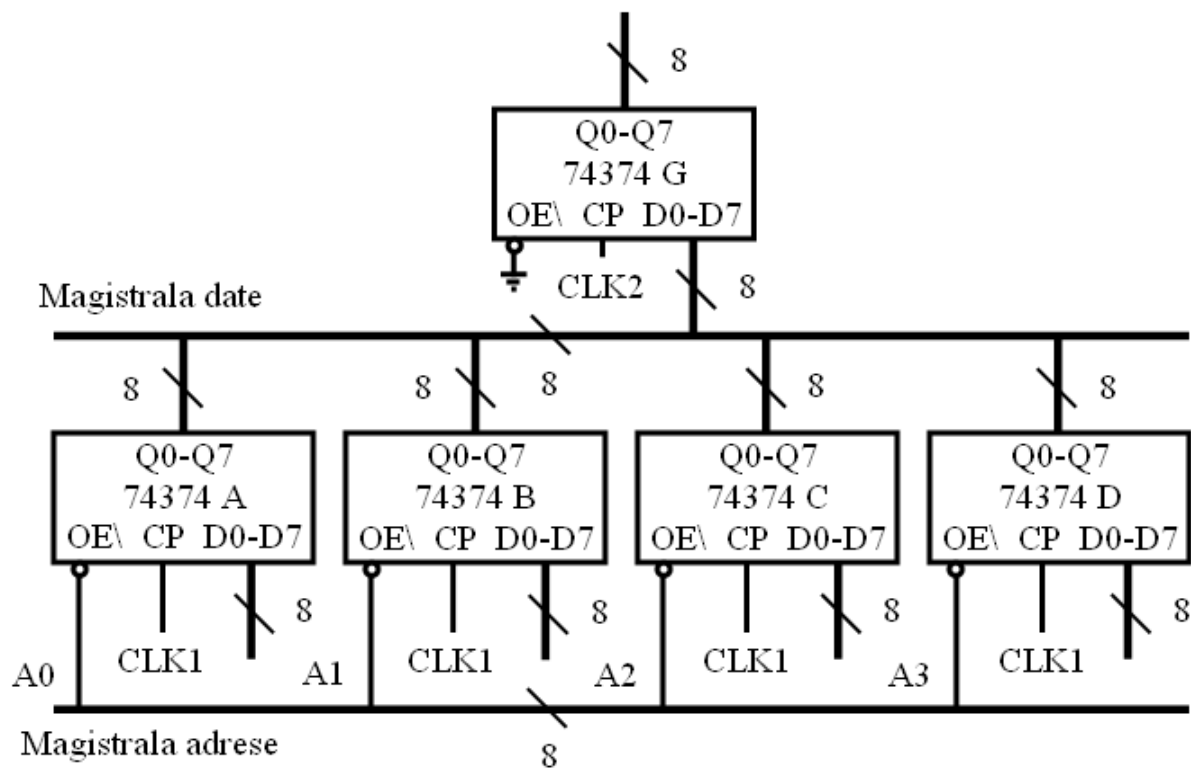
Avantaj – complexitatea circuitului de selectie scade

Dezavantaj – numarul de adrese care poate fi folosit pentru selectia registrelor scade

Pentru selectia fiecarui registru se foloseste cate o linie de adresa (selectie liniara)

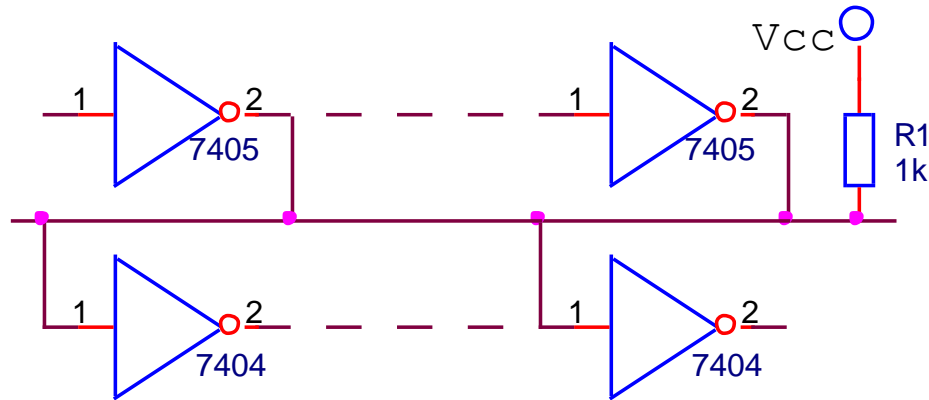
Avantaj – complexitatea circuitului de selectie este minima

Dezavantaj – numarul de adrese care poate fi folosit pentru selectia registrelor este minim



# Probleme propuse

- Sa se proiecteze o linie neadaptata de magistrala avand 5 transmitatori OC si 5 receptori TTL standard.



$$R_{1\max} = \frac{V_{CC\min} - V_{OH\min}}{5 I_{OH} + 5 I_{IH}} \approx 1,6k\Omega$$

$$R_{1\min} = \frac{V_{CC\max} - V_{OL\max}}{I_{OL} + 4 I_{OH} - 5 I_{IL}} \approx 0,5k\Omega$$

$$R_1 = 1k\Omega$$

- Sa se proiecteze o linie adaptata de magistrala avand 10 transmitatori OC si 2 receptori TTL standard. Linia de magistrala are impedanta caracteristica de  $250\Omega$ .

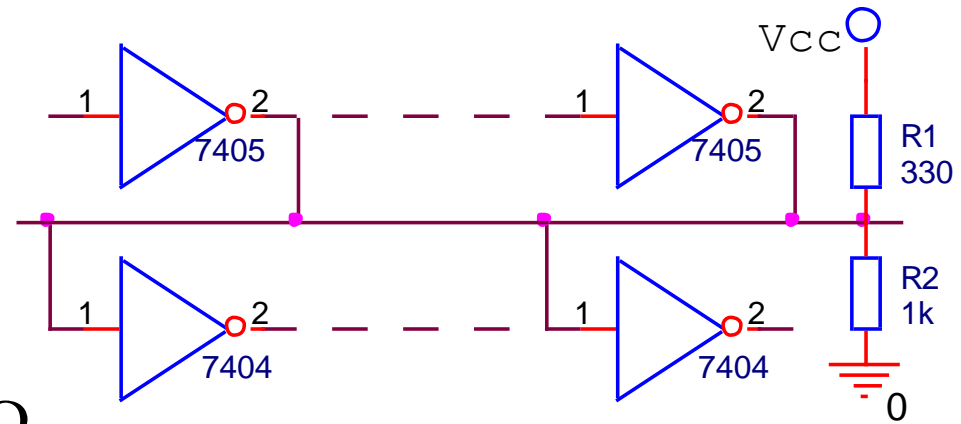
$$R_{1\max} = \frac{V_{CC\min} - V_{OH\min}}{10 I_{OH} + 2 I_{IH}} \approx 910\Omega$$

$$R_{1\min} = \frac{V_{CC\max} - V_{OL\max}}{I_{OL} + 9 I_{OH} - 2 I_{IL}} \approx 322\Omega$$

$$R_1 = 330\Omega$$

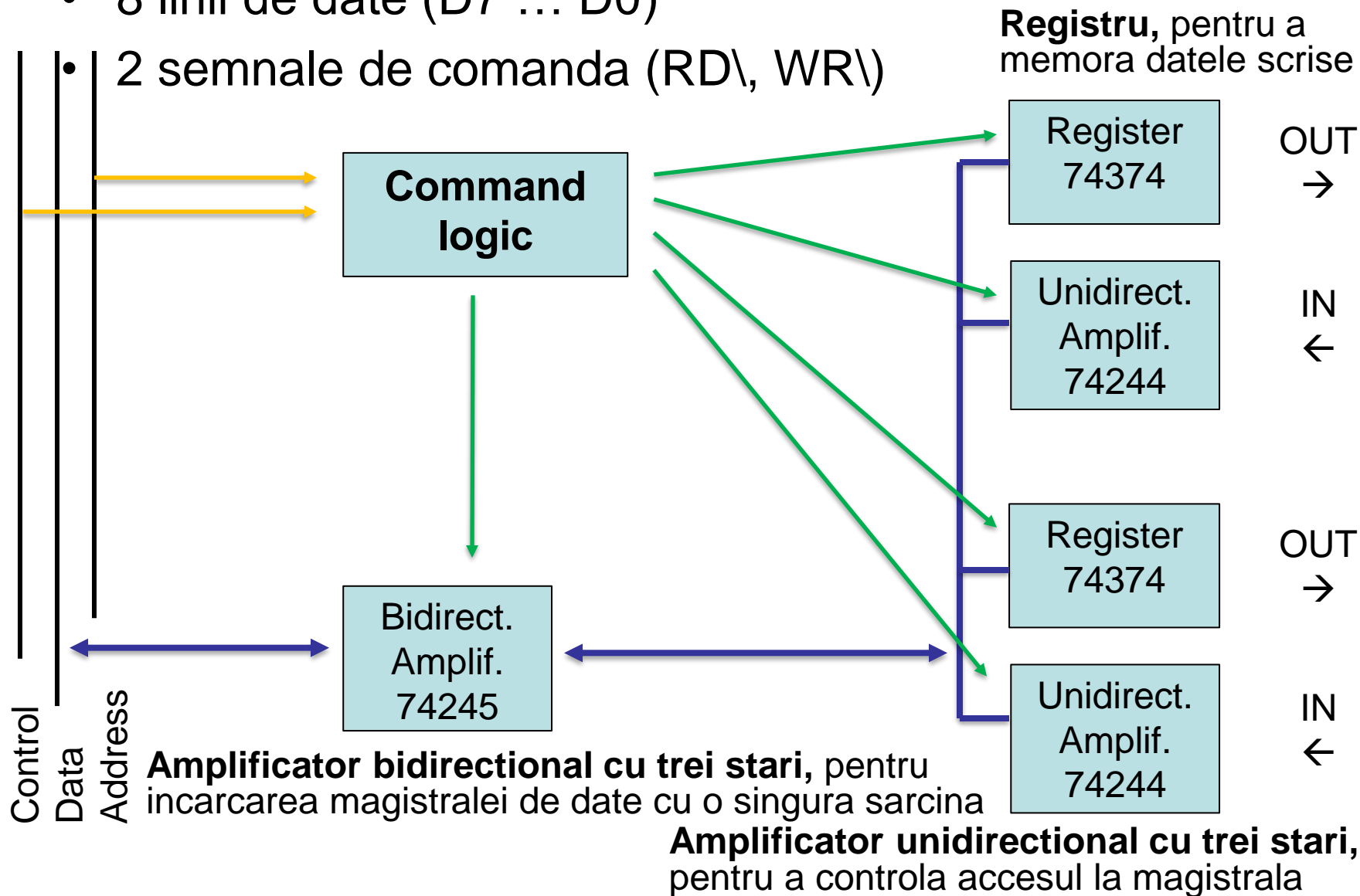
$$Z_0 = \frac{R_1 R_2}{R_1 + R_2}; R_2 = \frac{Z_0 R_1}{R_1 - Z_0} \approx 1k\Omega$$

$$V_{OH} = \frac{V_{CC\min} R_2}{R_1 + R_2} \approx 3,5V$$



- Sa se proiecteze o interfata de intrare/iesire care incarca magistrala unui microsystem cu o sarcina HCT si are urmatoarele porturi: 1 port intrare si un port iesire la adresa FEH, 1 port intrare si un port iesire la adresa FDH. Pentru activarea porturilor se foloseste selectia liniara. Magistrala microsystemului prezinta 8 linii de adresa, 8 linii de date si doua semnale de comanda pentru citire si respectiv scriere (RD\, WR\).

- 2 porturi de intrare + 2 porturi de iesire
- 8 linii de adresa (A7 ... A0)
- 8 linii de date (D7 ... D0)
- 2 semnale de comanda (RD\, WR\)

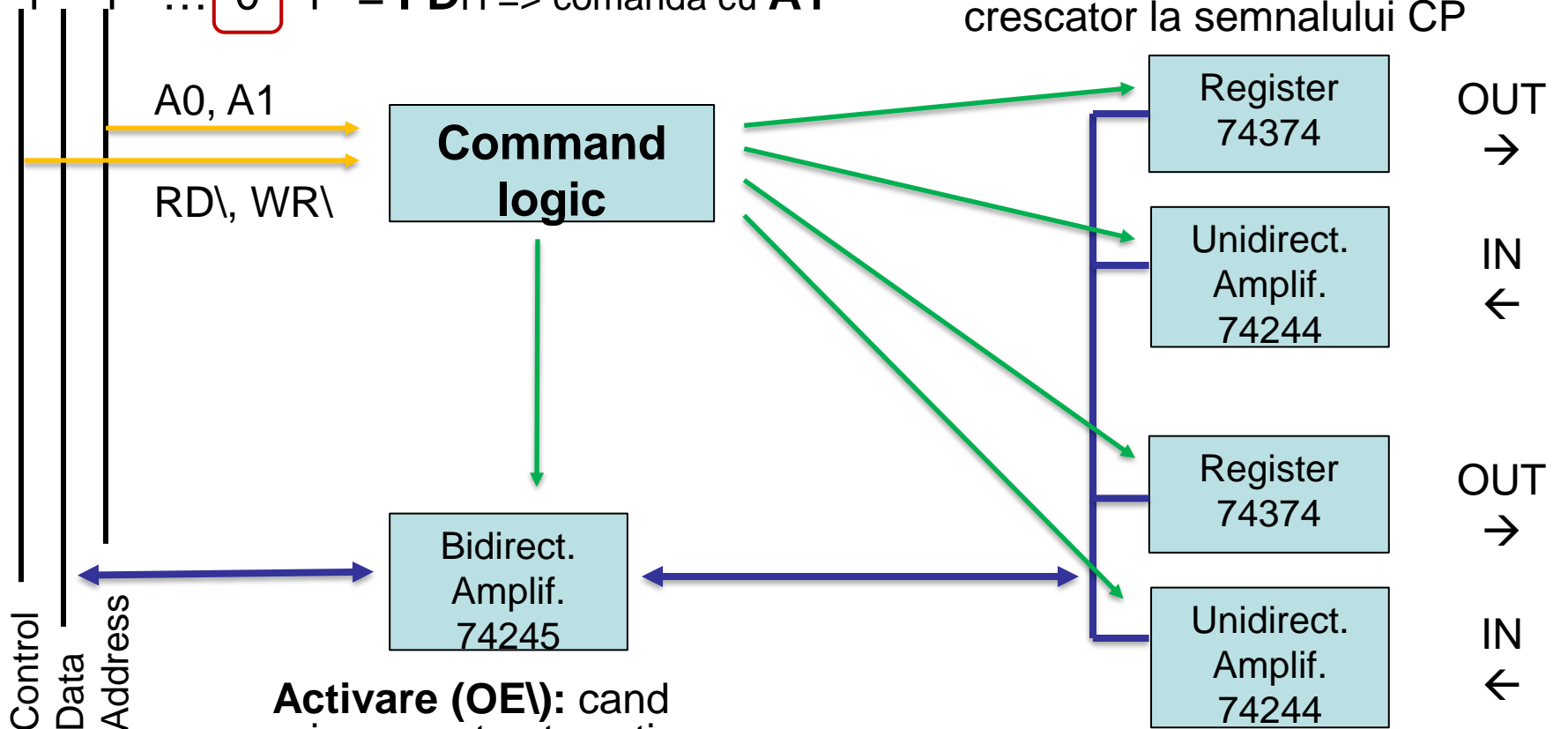


- 1 port de intrare + 1 port de iesire FEH
- 1 port de intrare + 1 port de iesire FDH

A7 A6 ... A1 A0

1 1 ... 1 0 = FEH => comanda cu A0

1 1 ... 0 1 = FDH => comanda cu A1



**Scrierea** datelor in registru:  
cand AX = 0 si WR\ = 0 -> front  
crescator la semnalului CP

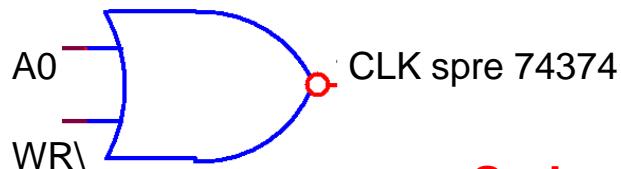
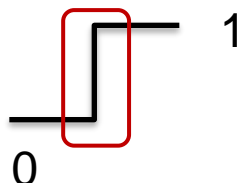
**Activare (OE\):** cand  
oricare port este activ  
**Setare directie (DIR):**  
pentru citire sau scriere

**Citirea** datelor de la amplificator:  
cand AX = 0 and RD\ = 0 ->  
activare iesire cu 0 la 1OE\ si 2OE\

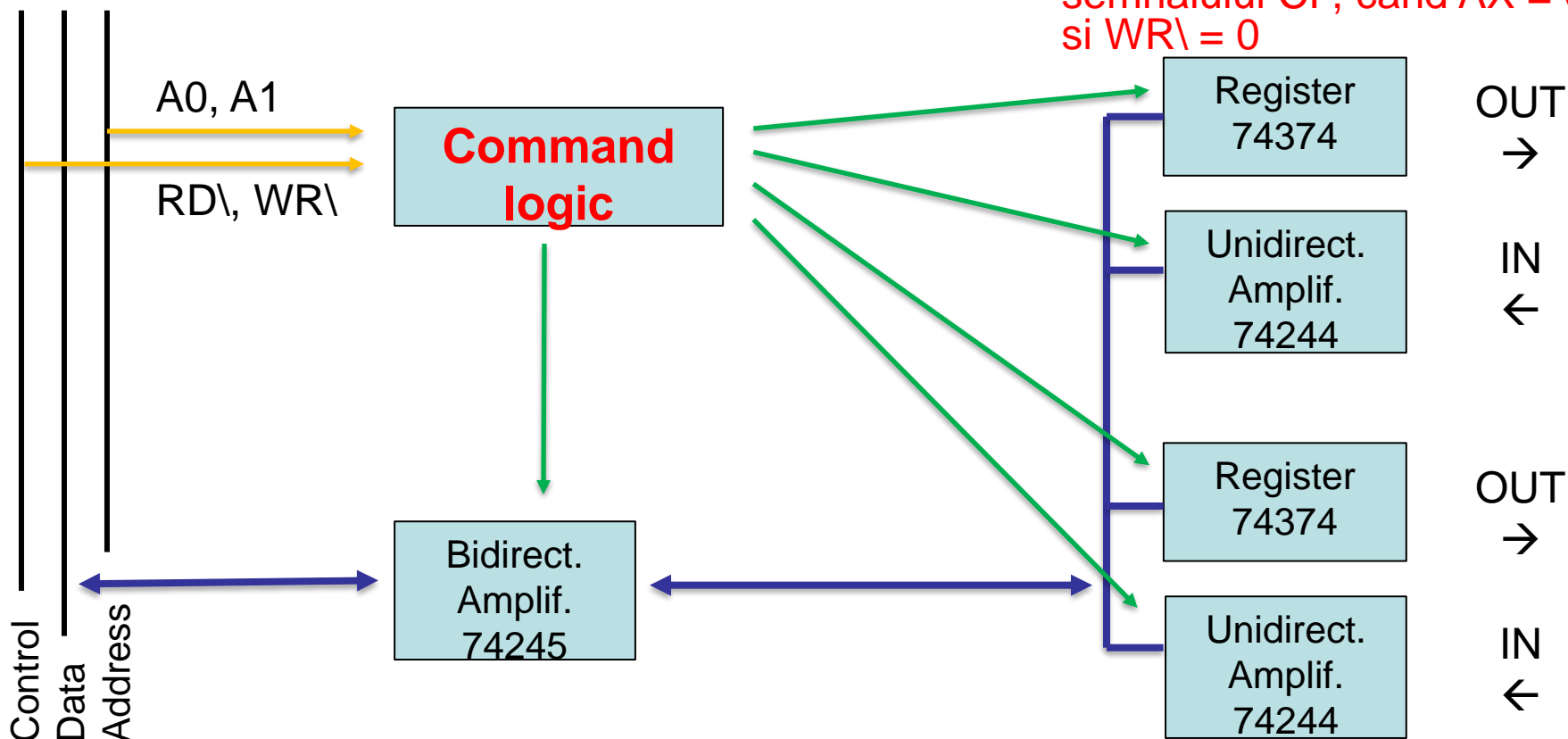
Scriere (primul port):  $A0 = 0$ ,  $WR\backslash = 0$ , front crescator

Poarta **NOR** furnizeaza tranzitia necesara de la 0 la 1

A0	WR\	out
0	0	1
0	1	0
1	0	0
1	1	0



**Scrierea** datelor in registru:  
pe front crescator al  
semnalului CP, cand  $AX = 0$   
si  $WR\backslash = 0$

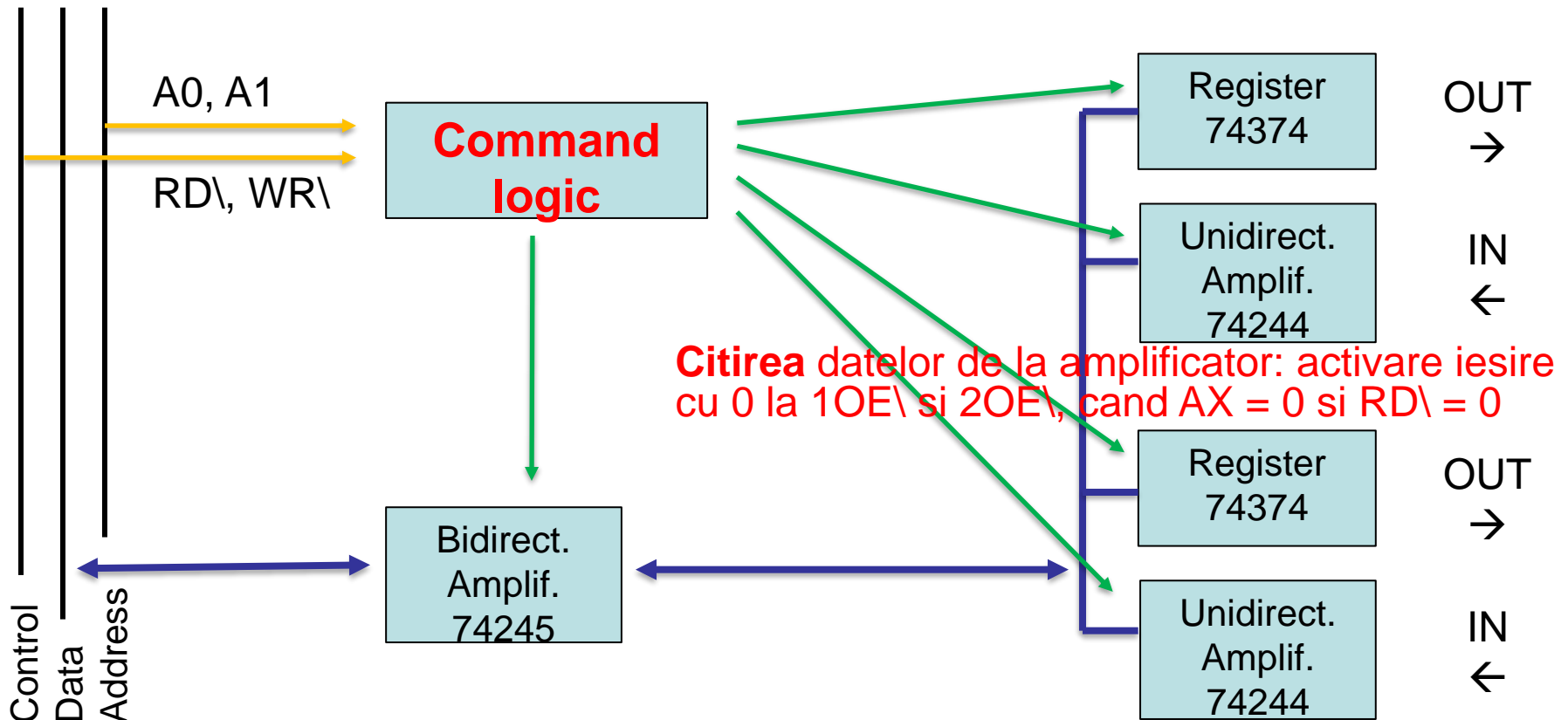
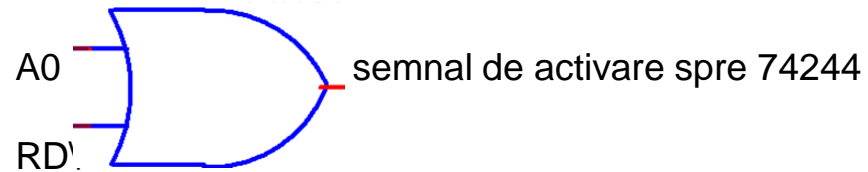




Citire (primul port):  $A0 = 0$ ,  $RD\backslash = 0$

Poarta **OR** furnizeaza iesirea necesara doar cand ambele semnale sunt 0

A0	RD\	out
0	0	0
0	1	1
1	0	1
1	1	1



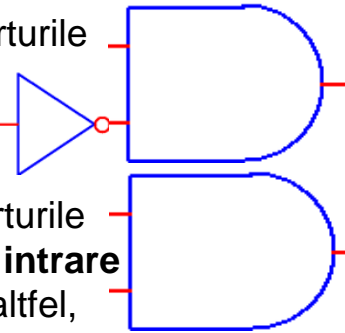
# Activare 74245 (OE\=0) si setare directie (DIR)

- Daca DIR = 0, directia este B -> A (citire port intrare)
- Daca DIR = 1, directia este A -> B (scriere port de iesire)

## AND gate

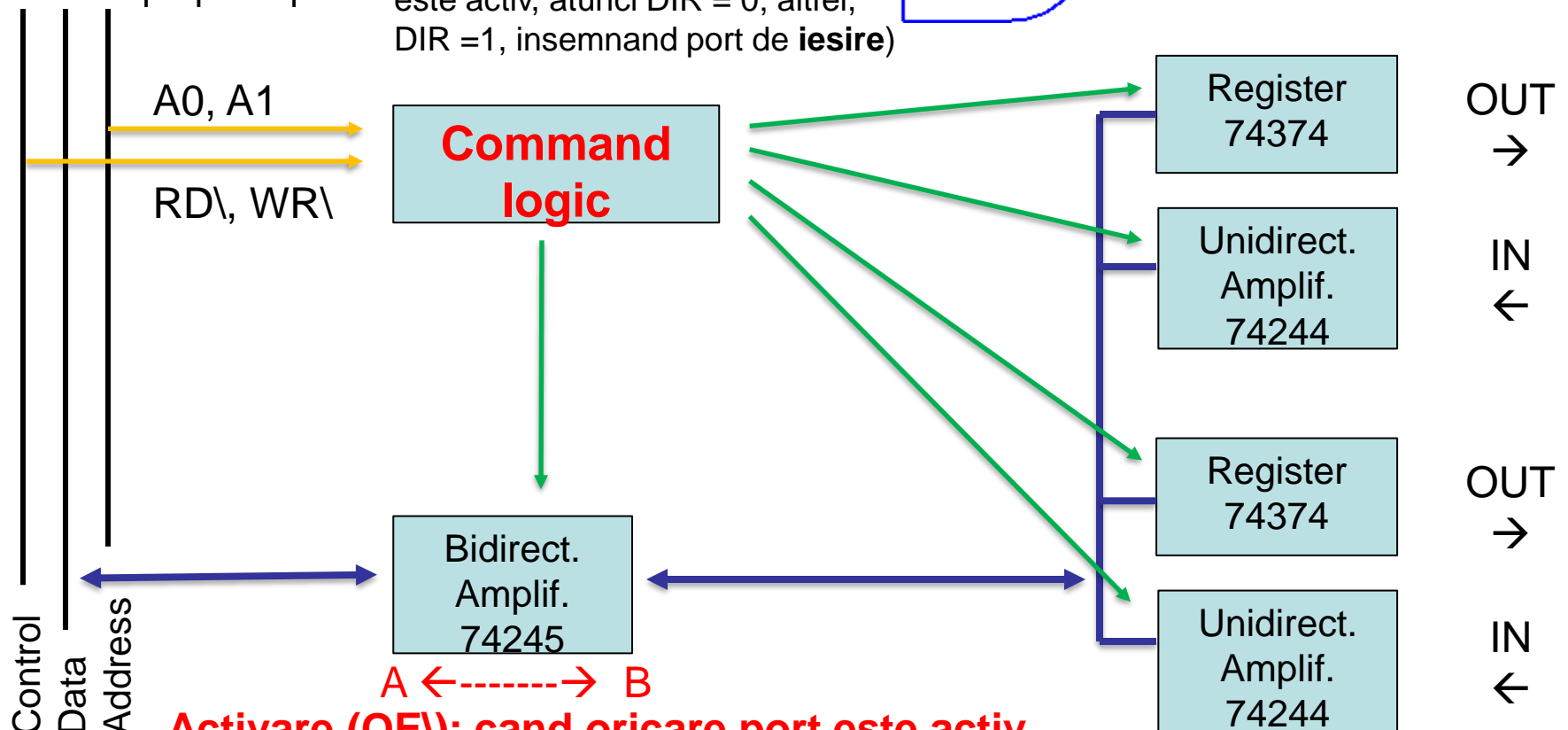
0	0	0
0	1	0
1	0	0
1	1	1

porti OR care comanda porturile de intrare  
porti NOR care comanda porturile de iesire  
porti OR care comanda porturile de intrare (daca un port de **intrare** este activ, atunci DIR = 0; altfel, DIR =1, insemnand port de **iesire**)



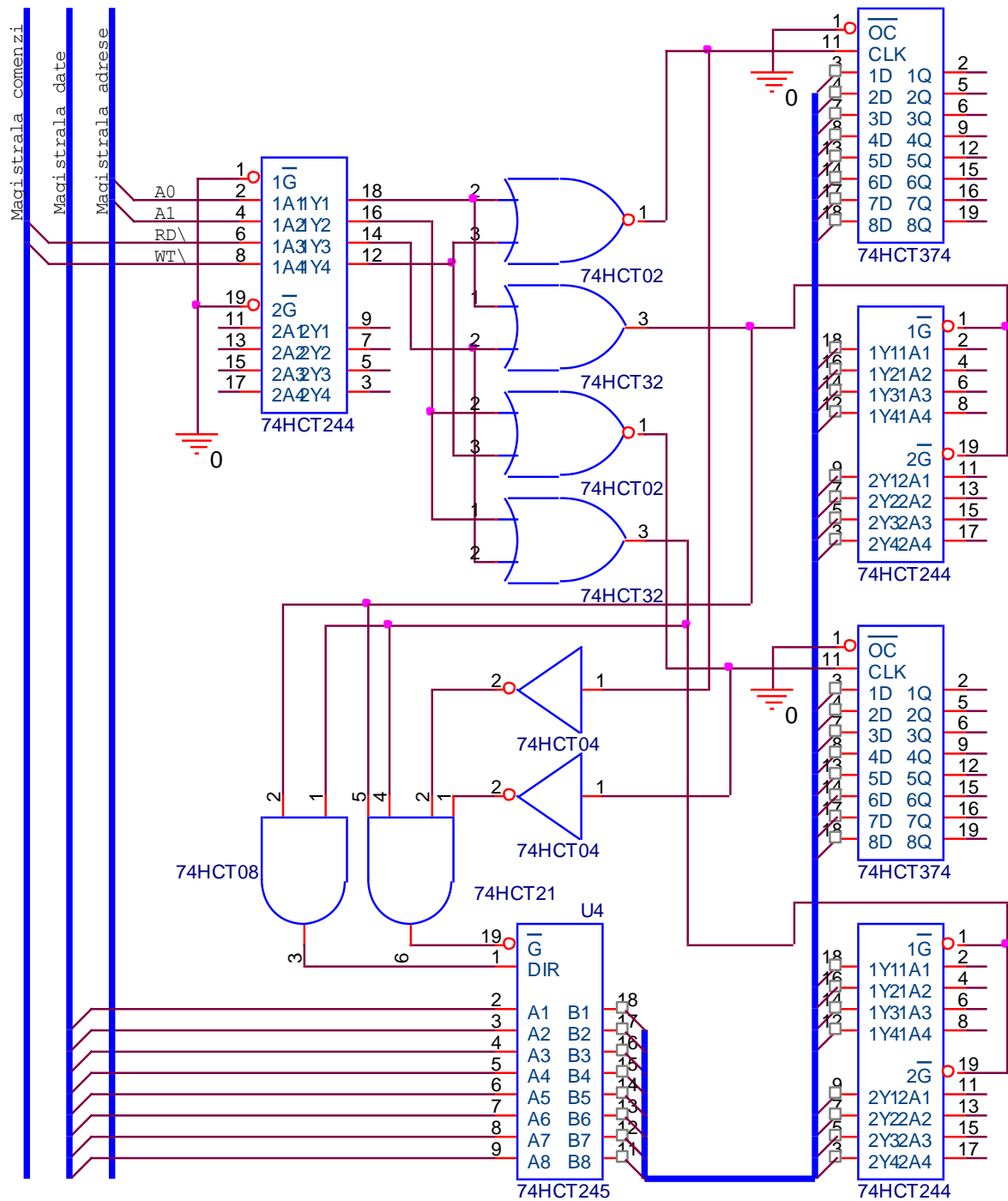
semnal **enable** spre 74245

semnal **DIR** spre 74245



A  $\longleftrightarrow$  B

**Activare (OE\): cand oricare port este activ**  
**Setare directie (DIR): pentru citire sau scriere**



- Sa se proiecteze o interfata de intrare/iesire care incarca magistrala unui microsistem cu o sarcina HCT si are urmatoarele porturi: 1 port intrare si un port iesire la adresa X0H, 1 port intrare si un port iesire la adresa X1H. Pentru activarea porturilor se foloseste selectia partiala. Magistrala microsistemului prezinta 8 linii de adresa, 8 linii de date si doua semnale de comanda pentru citire si respectiv scriere (RD\, WR\). La adresa X1H se conecteaza doua afisoare cu LED-uri 7-segmente anod comun si 8 taste. Pentru LED-uri se considera  $I_{LED}=10mA$  si  $V_{LED}=1,7V$ . Pentru tranzistoare se considera  $\beta=100$ .
- Sa se proiecteze o interfata de intrare/iesire care incarca magistrala unui microsistem cu o sarcina HCT si are urmatoarele porturi: 1 port intrare si un port iesire la adresa 70H, 1 port intrare si un port iesire la adresa 71H. Pentru activarea porturilor se foloseste selectia totala. Magistrala microsistemului prezinta 8 linii de adresa, 8 linii de date si doua semnale de comanda pentru citire si respectiv scriere (RD\, WR\). La adresa 70H se conecteaza doua afisoare cu LED-uri 7-segmente catod comun si 8 taste. Pentru LED-uri se considera  $I_{LED}=10mA$  si  $V_{LED}=1,7V$ .