



# *Sisteme cu MicroProcesoare*

## **Curs 07**

### **Sisteme de comunicații**

**Gigel Măceşanu**



# Cuprins

- Interfața SPI
- Interfața I2C



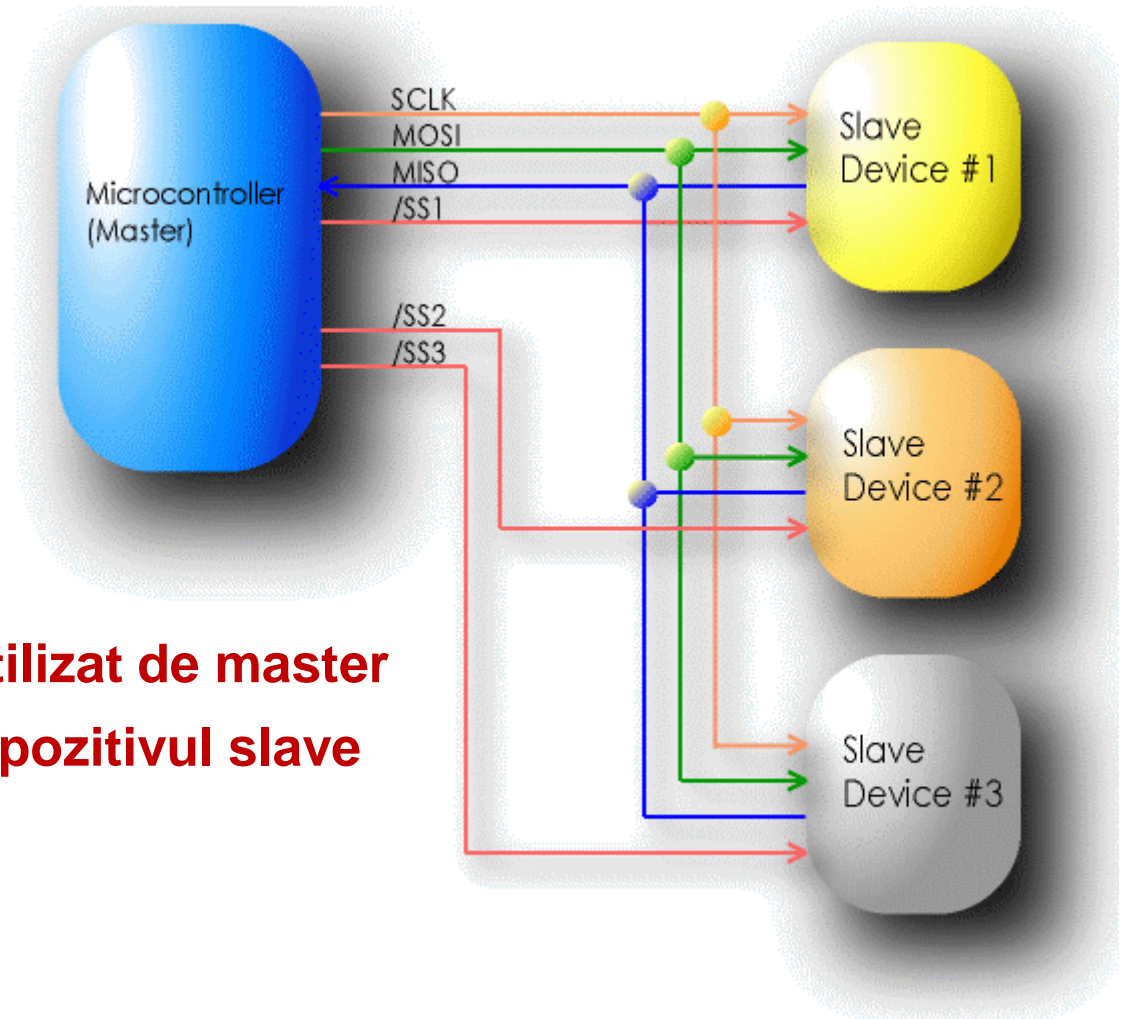
# Interfața SPI

- **SPI (*Serial Peripheral Interface*)**
- **Comunicație serială:**
  - **Sincronă, de tip punct la punct**
  - **Utilizează principiul master-slave**
  - **Full-duplex, master (MC), slave (dispozitive periferice)**
  - **Sincronizarea se poate face pe ambele fronturi ale clock-ului**
  - **Utilizează 4 linii de comunicație:**
    - **MOSI (Master Out, Slave In):** utilizată de master pentru a transmite date către slave.
    - **MISO (Master In, Slave Out):** utilizată de slave pentru a transmite date către master.



# SPI – linii de date și control

- **SCK (System Clock):** utilizată de master pentru a transmite semnalul de clock

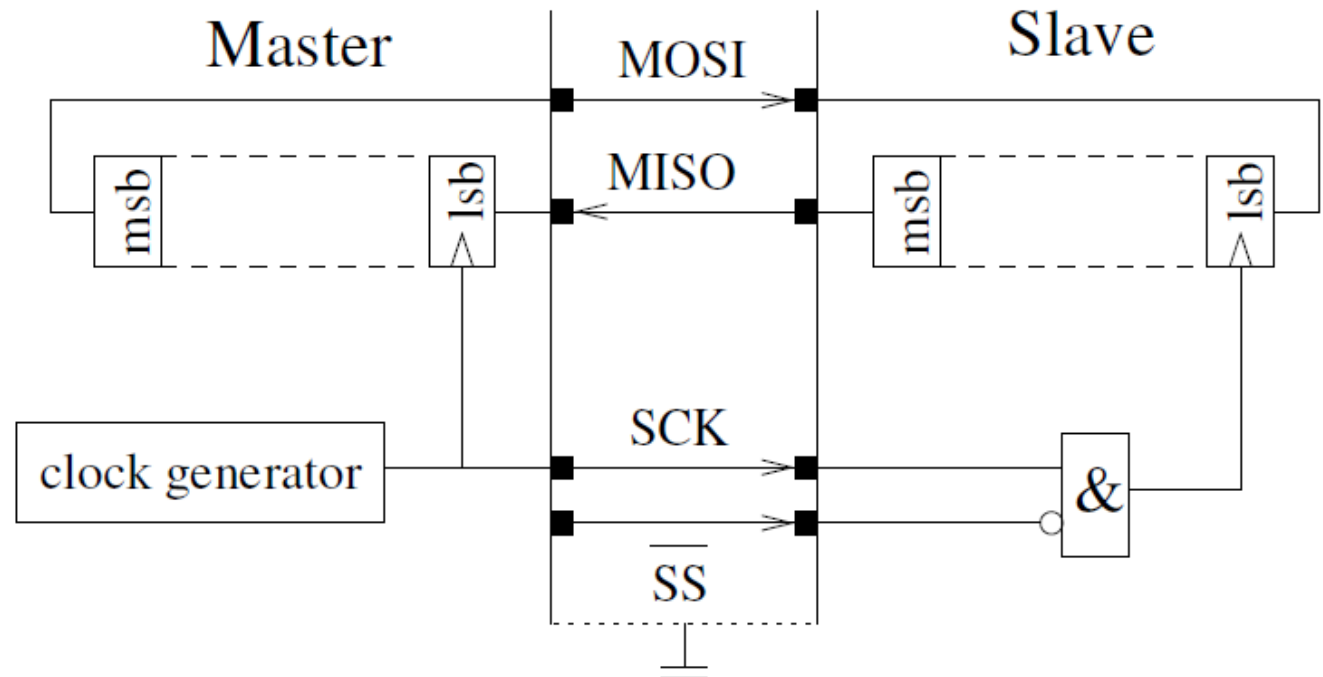


- **$\overline{SS}$  (Slave Select):** utilizat de master pentru a selecta dispozitivul slave



# SPI – principiu de funcționare

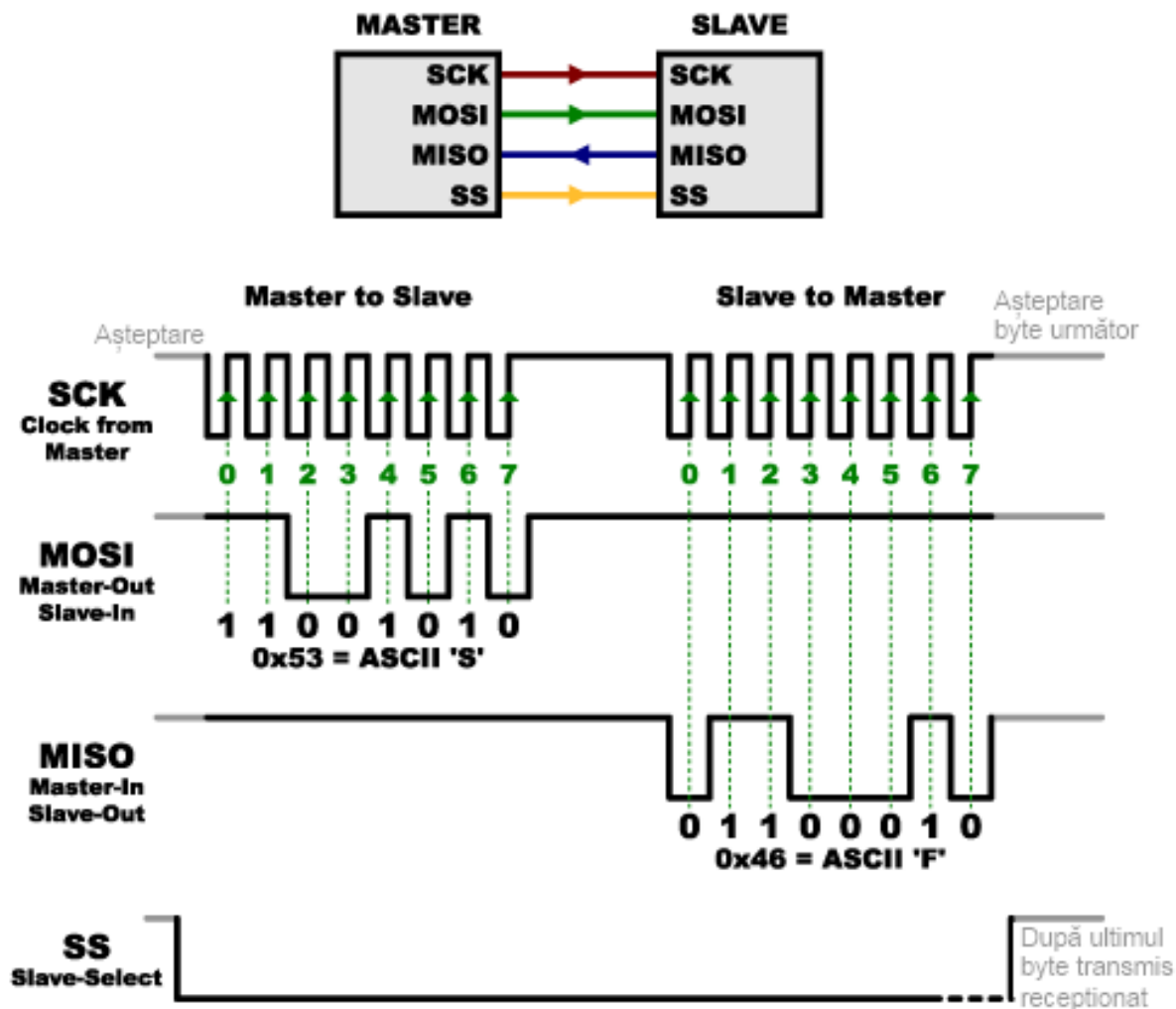
- Master/Slave au un registru cu deplasare (operat de SCK)
- La fiecare clock, master msb (sau lsb) (MOSI) → slave lsb
- În același timp, slave msb (MISO) → master lsb
- După 8 cicluri de clock, master-ul și slave-ul au schimbat între ele datele (8 biți)





# SPI – principiu de funcționare

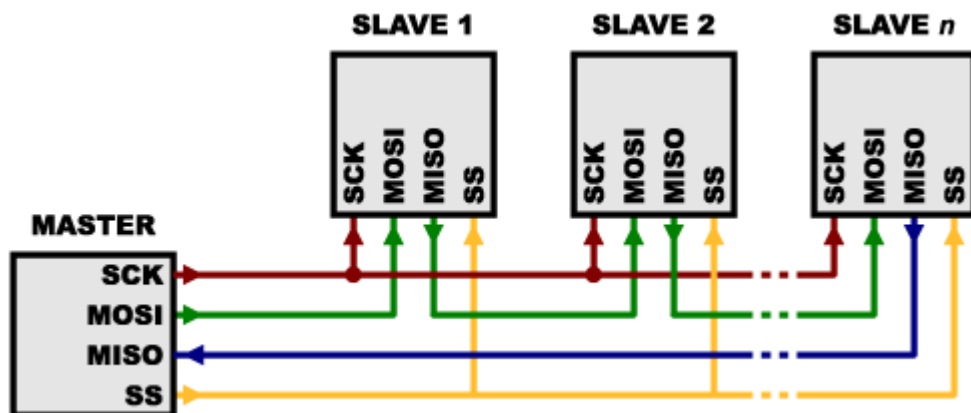
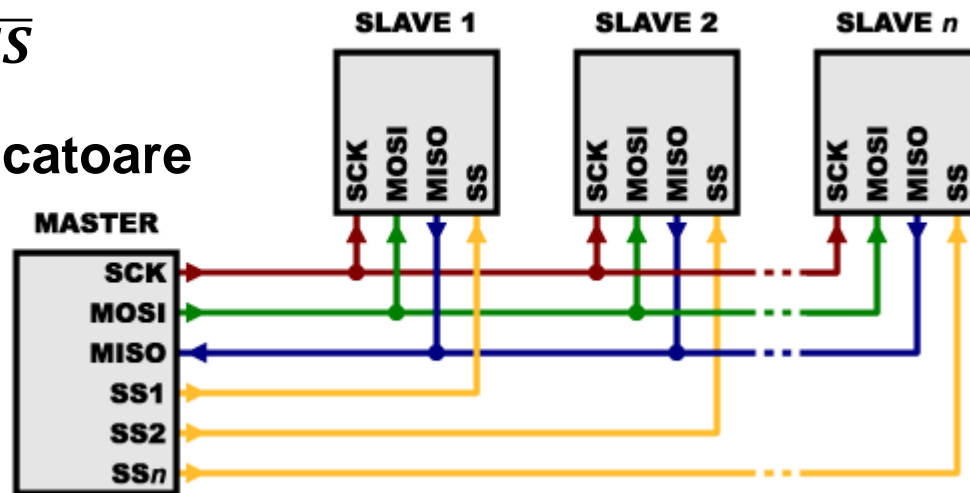
- Masterul adresează explicit dispozitivul slave prin setarea liniei  $\overline{SS}$  la nivel low





# SPI – conectare dispozitive slave

- Pentru fiecare dispozitive slave este necesară utilizarea unei linii  $\overline{SS}$
- Se pot utiliza circuite decodificatoare (M74HC238)



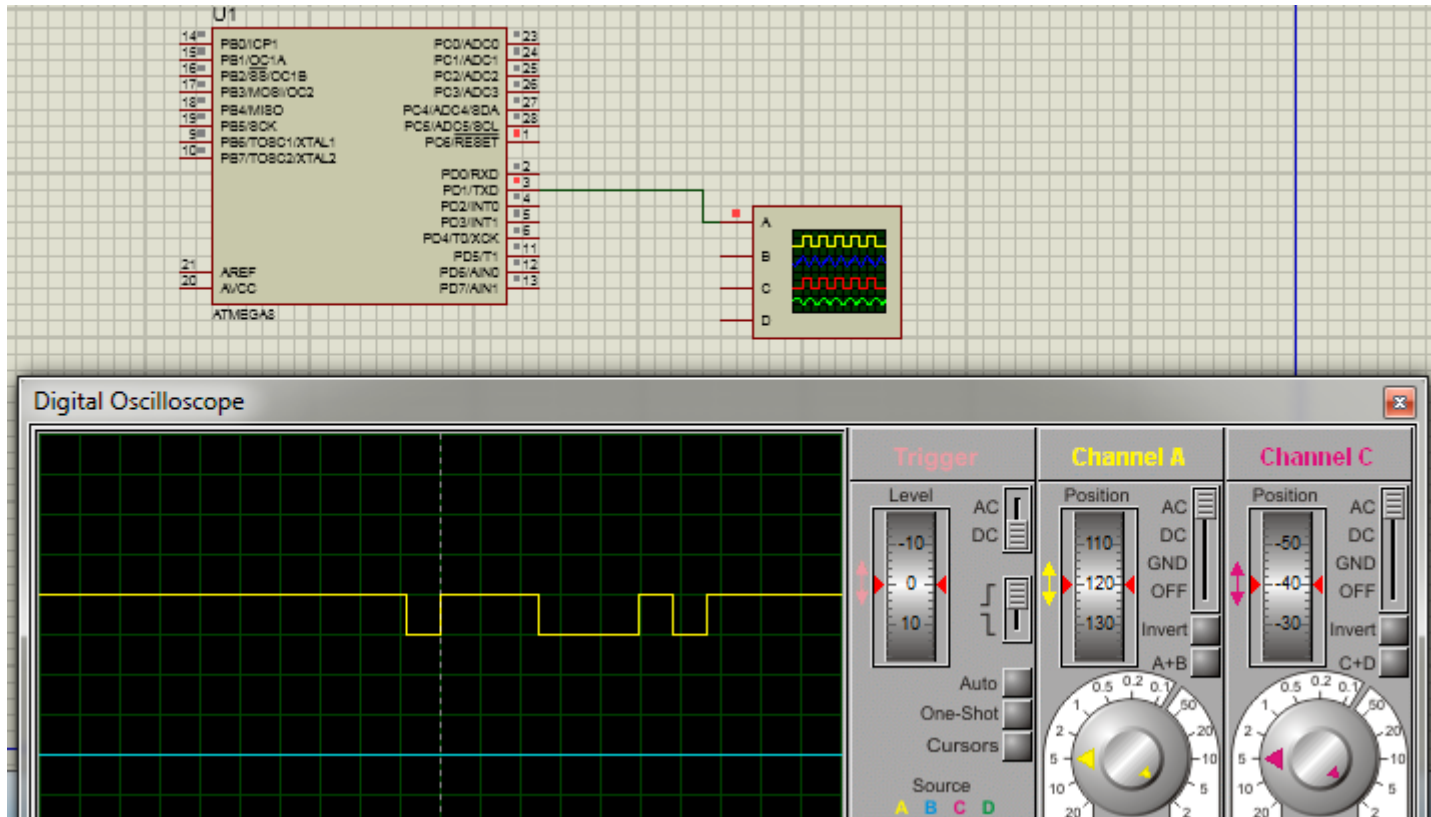
- Dacă nu este necesar transferul slave → master se poate folosi o legare înlănțuită (ex. driver adresă LED)



# Aplicație - SPI

- Exemplu de utilizare a modului SPI

- **aaaa**





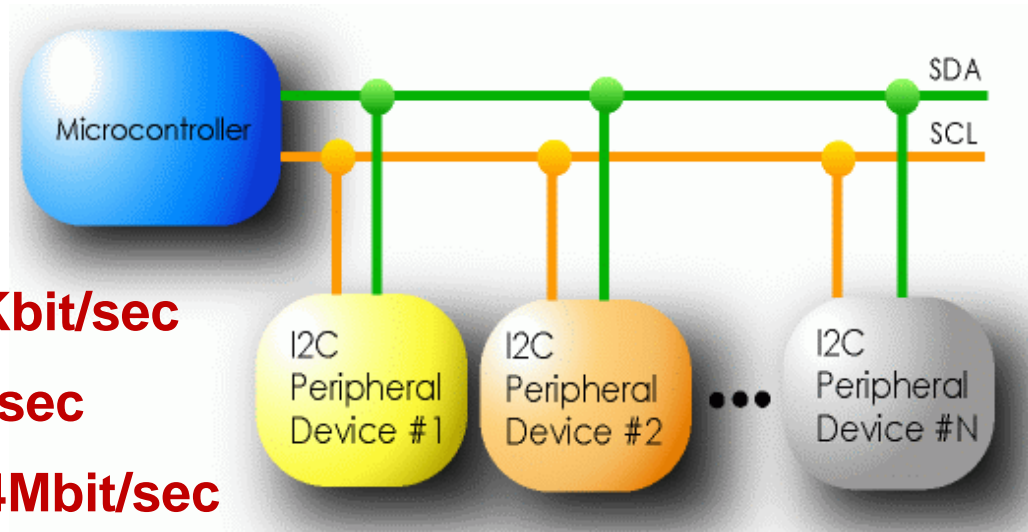


# Interfața I2C

- I2C - *Inter-integrated Circuit* (dezvoltat/patentat de Philips)
- Interfață sincronă, master-slave, half-duplex
- Utilizează două linii pentru comunicație (master - slave):
  - **SCL (Serial Clock Line)** pentru semnalul de clock (sincronizare)
  - **SDA (Serial Data Line)** pentru transferul de date

- Se pot seta trei moduri de transmisie de date:

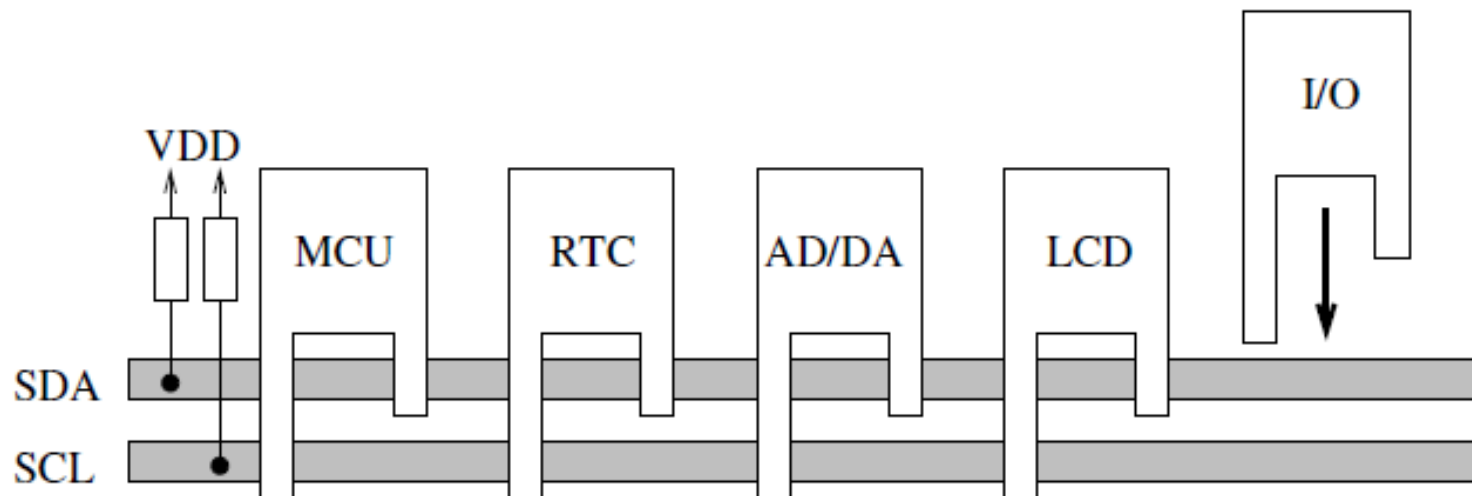
- **Modul standard** → 100Kbit/sec
- **Modul rapid** → 400Kbit/sec
- **Modul extra-rapid** → 3.4Mbit/sec





# Interfața I2C

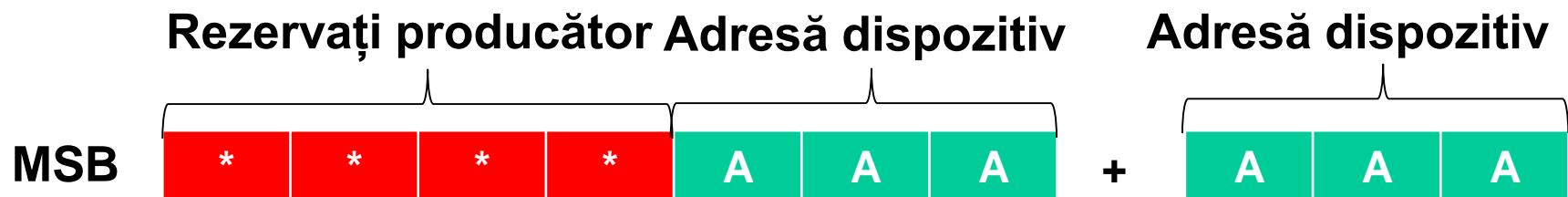
- Protocolul conține un mecanism de administrare a modulelor pe magistrala I2C → pot exista mai multe dispozitive master
  - **Ex: sistem cu mai multe MC**
- Avantajul I2C este extensibilitatea
  - **Limită: Capacitatea maximă a magistralei nu trebuie să depășească 400pF**





# Interfața I2C – moduri de adresare

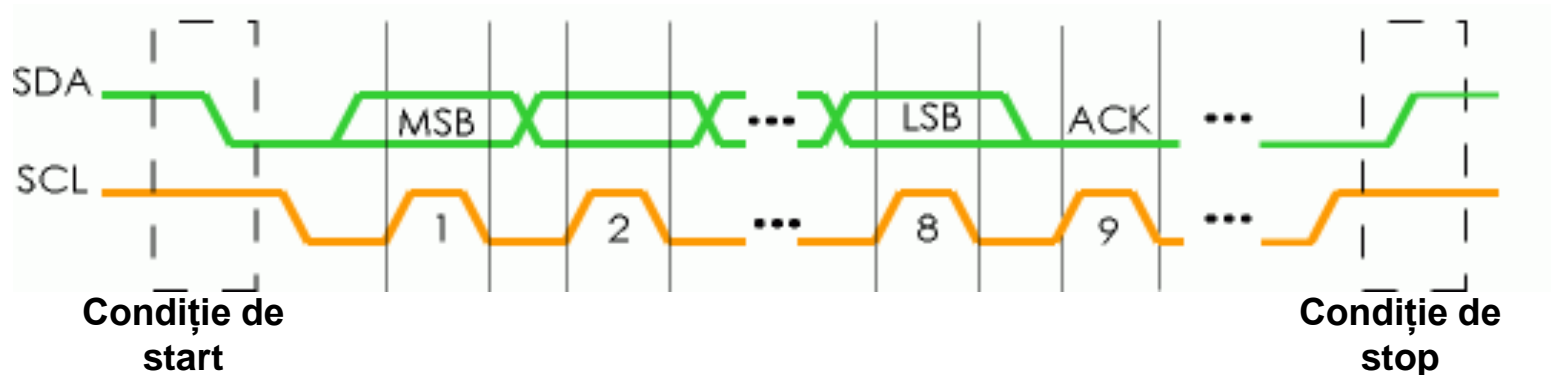
- I2C suportă adresare pe 7 sau 10 biți
- La adresarea pe 7 biți:
  - **Dispozitivele de pe bus sunt identificate după adresă (7 biți)**
    - 4 MSB sunt predefiniți de producător
    - 3 LSB pot fi definiți de programator
  - **Adresele: 0000XXX și 1111XXX sunt rezervate**
- La adresarea pe 10 biți:
  - **adresarea pe 7 biți + 3 noi biți**





# Interfața I2C – Transmiterea datelor

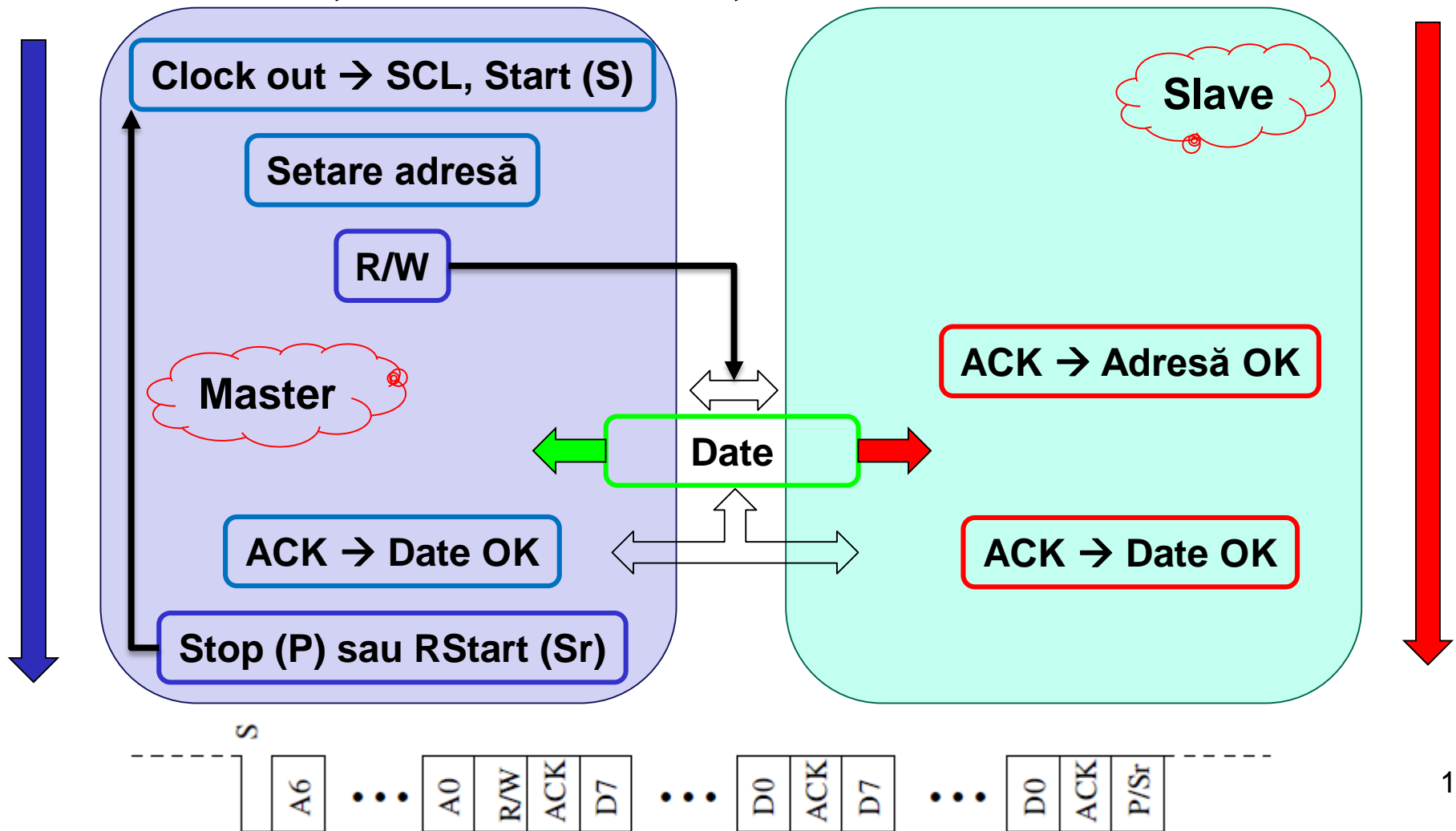
- Nivele tensiune:
  - **Low:  $-0.5 \div 0.3V_{dd}$**
  - **High:  $0.7V_{dd} \div V_{dd} + 0.5$**
- Linile au caracteristici dominante și recesive
  - **Un dispozitiv out = 0 → linia = 0 chiar dacă alte dispozitive generează out = 1**
  - **0 este dominant, 1 este recesiv (0 câștiga tot timpul)**
  - **Mai poartă numele de "wired AND" (toti E out = 1 → line = 1)**
- Un pachet I2C are forma următoare:





# Interfața I2C – Transmiterea datelor

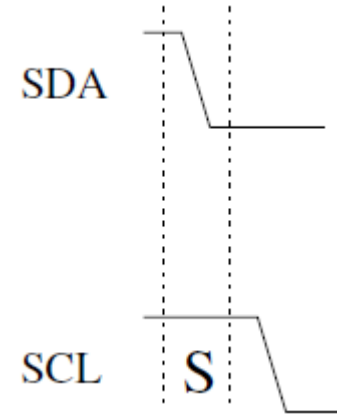
- În starea de așteptare liniile SDA și SCL sunt high





# Interfața I2C – Transmiterea datelor

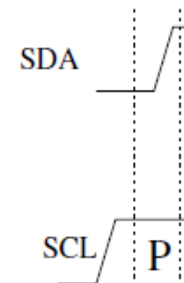
- Condiția de Start sau Repetare start:
  - **SDA – front descrescător (high → low)**
  - **SCL – nivel high**
  - **RS – înlocuiește perechea Start/Stop**
- $R/\overline{W}$  indică direcția transmiterii datelor
  - $R/\overline{W}$  - high: master-ul citește date de la slave (primește și Ack)
  - $R/\overline{W}$  - low: master-ul trimite date la slave
- Confirmarea primirii datelor este realizată cu  $\overline{ACK}$ 
  - Receptorul trimite  $\overline{ACK}$  după fiecare 8 biți primiți
  - $\overline{ACK}$  presupune trecerea SDA în starea low





# Interfața I2C – Transmiterea datelor

- Condiția de STOP (P):
  - SDA – front crescător (low → high)
  - SCL – nivel high
- Extinderea adreselor – 10 biți
  - Masterul trimite 11110XX către slave, și  $R/\overline{W} = 0$ 
    - XX sunt MSB din adresarea pe 10 biți
  - Fiecare slave compară adresa proprie cu MSB XX primit, și trimite ACK dacă corespunde
  - Masterul trimite următorii 8 biți ai adresei





# Interfața I2C – Transmiterea datelor

- Extinderea adreselor – 10 biți
- **Citirea de la slave presupune:**
  - Masterul trimite 11110XX către slave, și  $R/\overline{W} = 0$
  - Slave-ul trimite ACK
  - Masterul trimite următorii 8 biți ai adresei
  - Slave-ul trimite ACK
  - Masterul trimite 11110XX către slave, și  $R/\overline{W} = 1$  (slave este trecut în modul emitor – transmite date)
  - Slave-ul transmite datele







**Contact:**  
**Email: [gigel.macesanu@unitbv.ro](mailto:gigel.macesanu@unitbv.ro)**  
**Web: [rovis.unitbv.ro](http://rovis.unitbv.ro)**