

Familia de microcontrolere 80C51

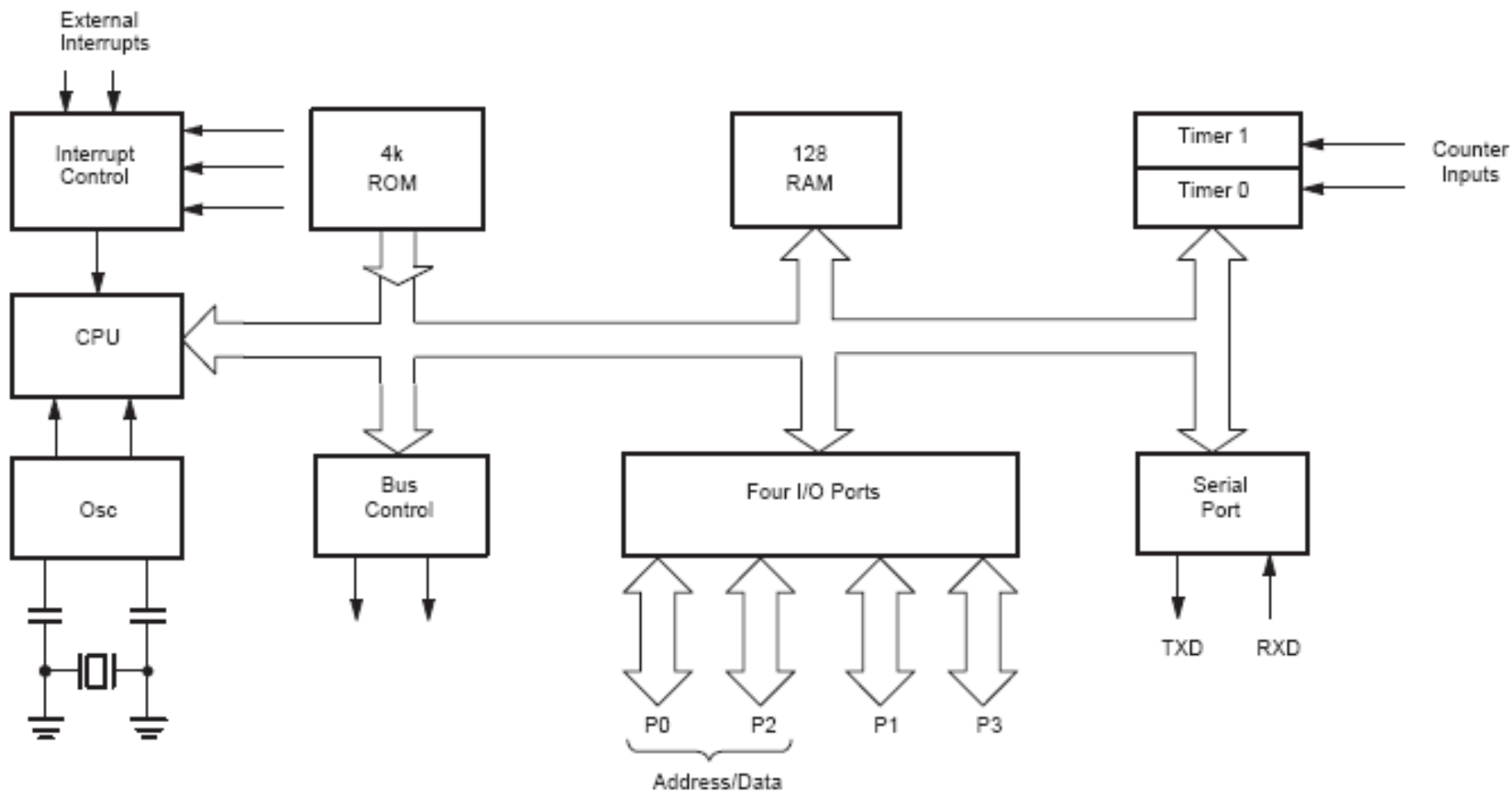
- Introducere**
- Adresarea memoriei**
- Sistemul de întreruperi**
- Regiștrii sistemului de timere**
- Comunicația serială**

Introducere

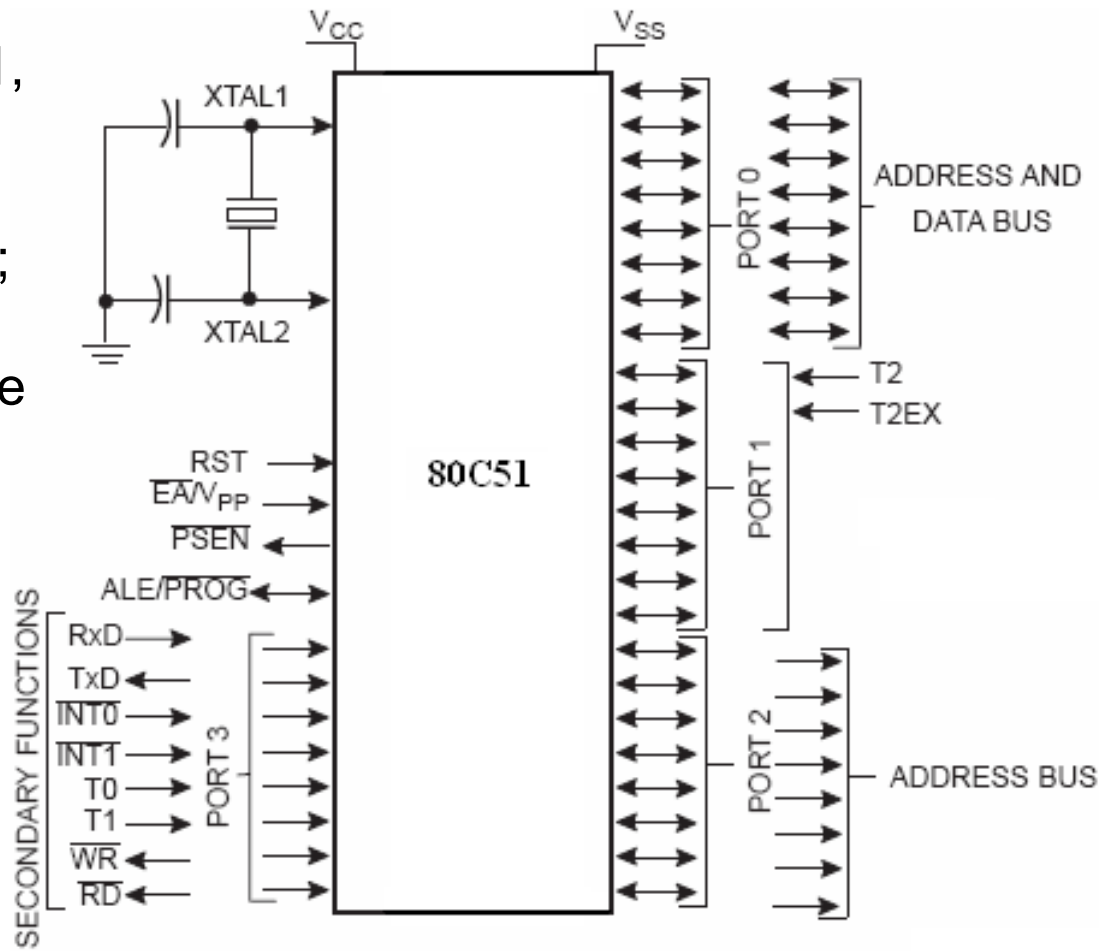
- Microcontrollerele înglobează diferite circuite necesare unui sistem de calcul:
 - processor boolean
 - memorie
 - numărătoare/timere
 - sistem de întreruperi
 - porturi de intrare/iesire

- Caracteristicile familiei de microcontrolere 80C51:
 - Unitate centrala de procesare 8051
 - 4k*8 ROM
 - 128*8 RAM
 - 3*16-biti numărătoare/timere
 - procesor boolean
 - Capabilitate de adresare a memoriei externe
 - 64k*8 ROM (program)
 - 64k*8 RAM (data)
 - 6 întreruperi cu 2 nivele de prioritate
 - 4*8-biti porturi I/O
 - UART full–duplex
 - port asincron de reset

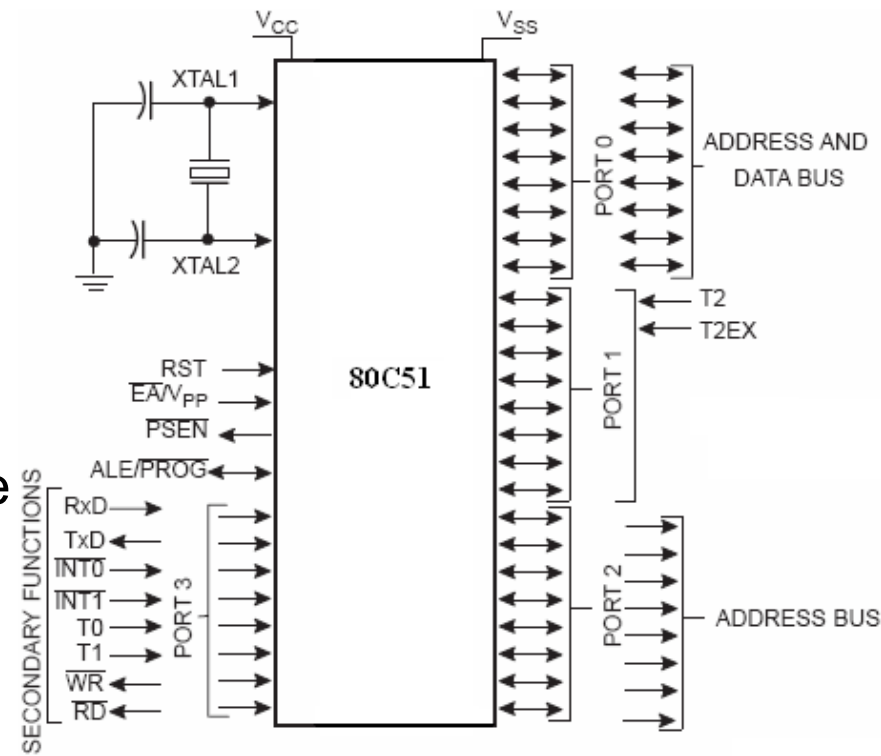
- Circuitele integrate în microcontroller comunica prin intermediul unor magistrale interne pe care se pot vehicula adrese, date sau semnale de control



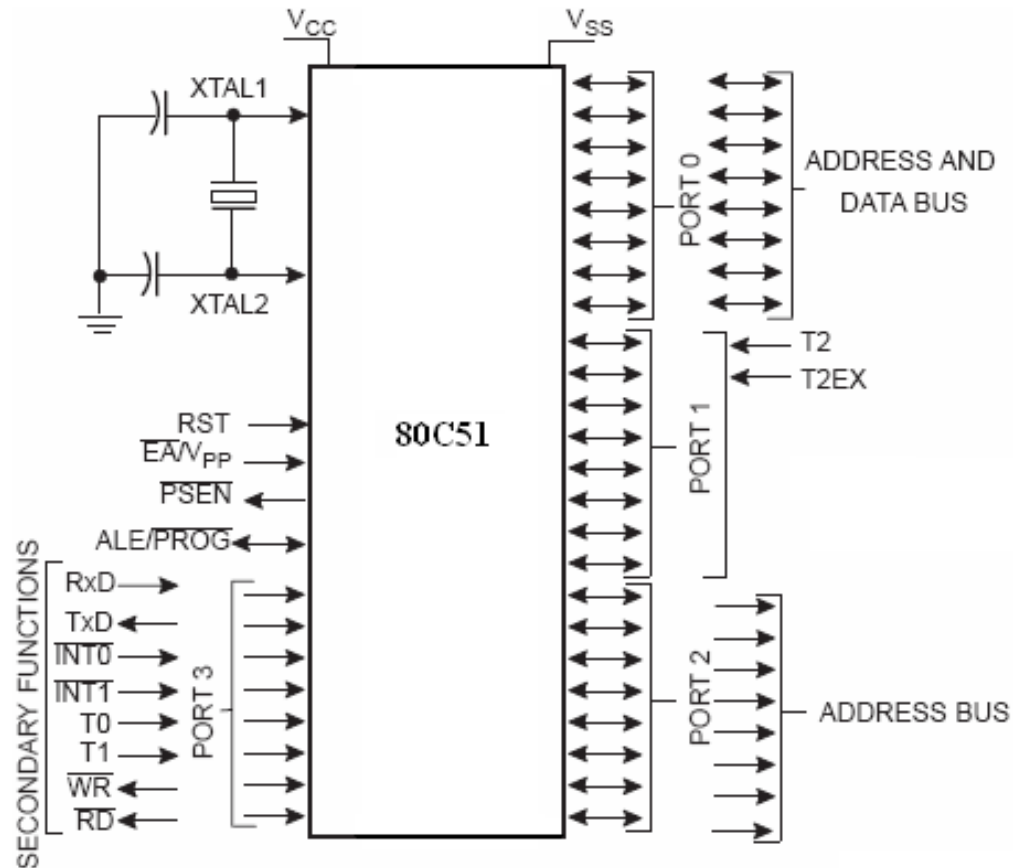
- **Simbolul logic și descrierea pinilor**
- **VSS**, intrare, masă
- **VCC**, intrare, alimentare
- **P0.0–0.7**, intrare/ieșire, port 0, funcție de port bidirecțional de 8 biți open-drain cu intrări trigger Schmitt; pinii care au 1 înscris sunt flotați și pot fi folosiți ca intrări cu înaltă impedanță; funcție de magistrală de date și magistrală de adrese mai puțin semnificativă multiplexate pe perioada accesului la memoria externă de date și program folosind rezistențe de ridicare interne
- **P1.0–P1.7**, intrare/ieșire, port 1, funcție de port bidirecțional de 8 biți cu rezistențe de ridicare interne și intrări trigger Schmitt; pinii care au 1 înscris sunt ridicați în 1 de către rezistențele interne de ridicare și pot fi folosiți ca intrări; doi pini pot avea și funcții alternative
 - **T2**, intrare/ieșire, (P1.0), Timer/Numărător 2
 - **T2EX**, intrare, (P1.1), Timer/Numarator 2 cu funcție de captură



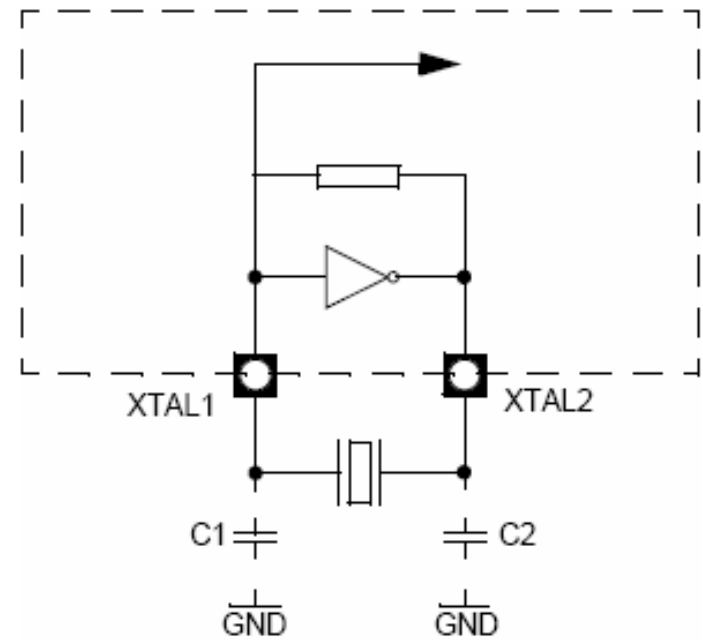
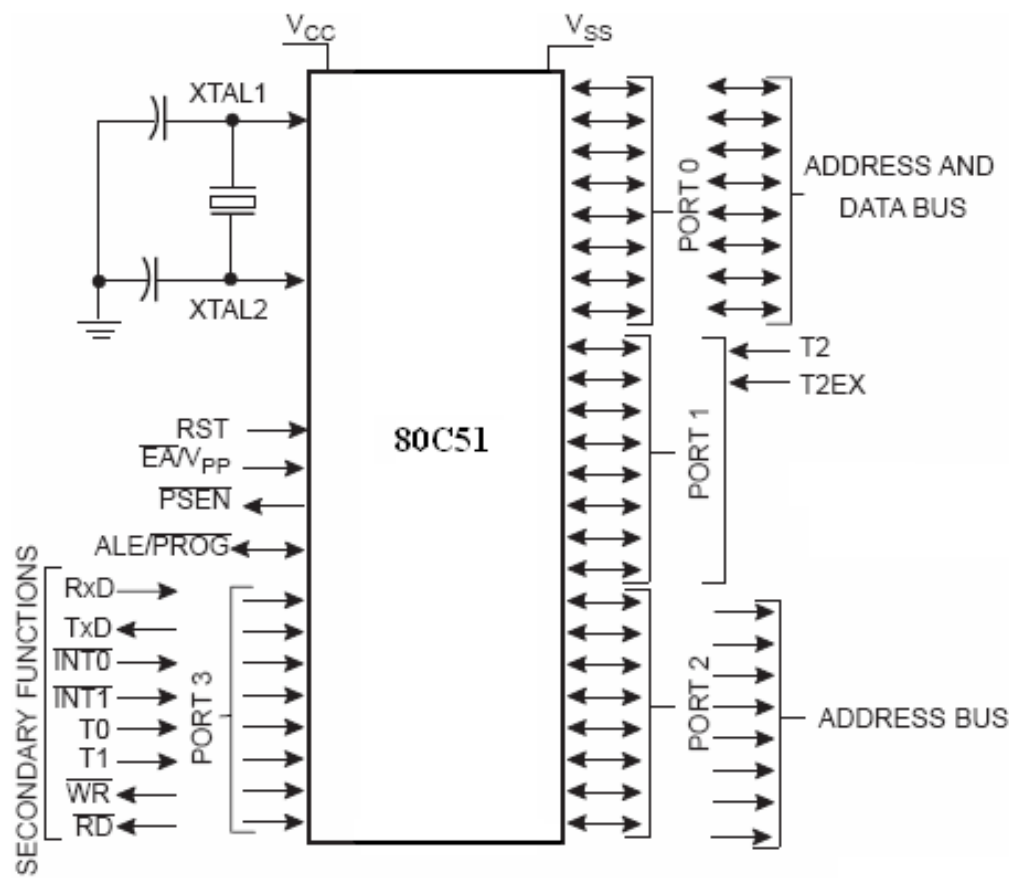
- **P2.0–P2.7**, intrare/ieșire, port 2, funcție de port bidirecțional de 8 biți cu rezistențe de ridicare interne și intrări trigger Schmitt; pinii care au 1 înscris sunt ridicați în 1 de către rezistențele interne de ridicare și pot fi folosiți ca intrări; ca funcție alternativă emite octetul mai semnificativ de adresă pe perioada accesului la memoria externă de program și date
- **P3.0–P3.7**, intrare/ieșire, port 3; funcție de port bidirecțional de 8 biți cu rezistențe de ridicare interne și intrări trigger Schmitt; pinii care au 1 înscris sunt ridicați în 1 de către rezistențele interne de ridicare și pot fi folosiți ca intrări; fiecare pin poate avea și funcție alternativă
- **RxD**, intrare, (P3.0), portul de intrare serial
- **TxD**, ieșire, (P3.1), portul de ieșire serial
- **INT0**, intrare, (P3.2), întreruperea externă 0
- **INT1**, intrare, (P3.3), întreruperea externă 1
- **T0**, intrare, (P3.4), intrarea externă pentru Timer 0
- **T1**, intrare, (P3.5), intrarea externă pentru Timer 1
- **WR**, ieșire, (P3.6), semnalul de activare a scrierii memoriei de date externe
- **RD**, ieșire, (P3.7), semnalul de activare a citirii memoriei de date externe



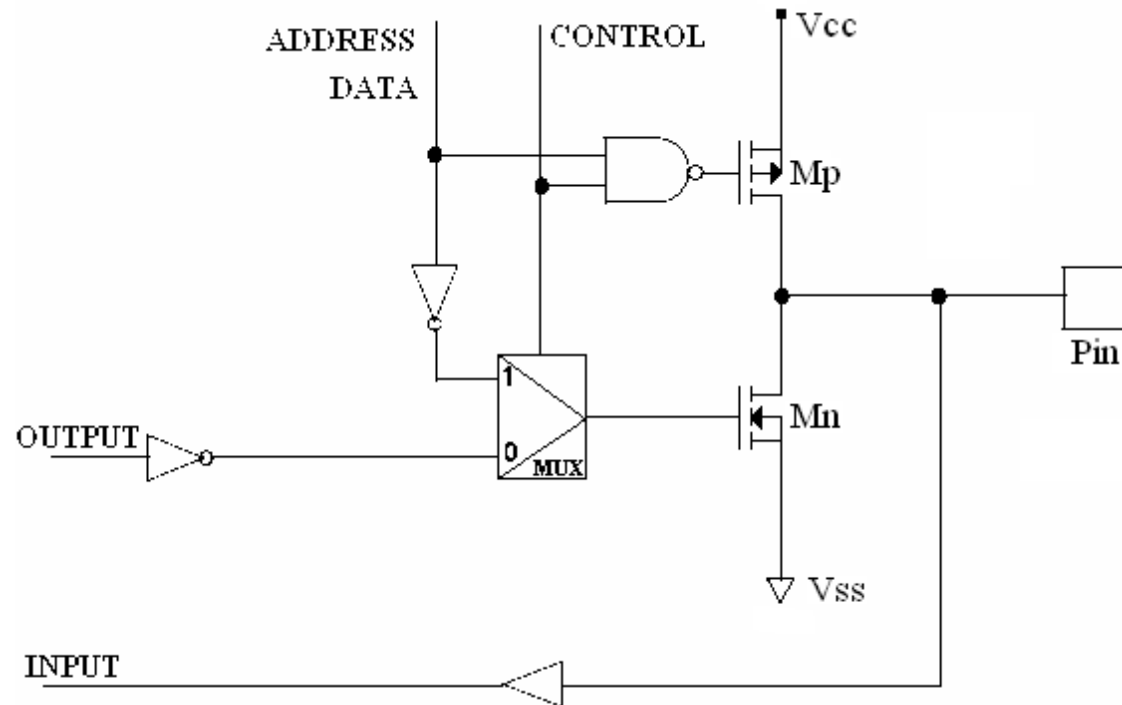
- **RST**, intrare, reset
- **ALE/PROG**, intrare/ieșire, Address Latch Enable/Program Pulse, pulsul de ieșire ALE este folosit pentru memorarea octetului mai puțin semnificativ de adresă pe perioada unui acces la memoria externă, pinul PROG este intrarea pulsului de program pe perioada programării EPROM-ului
- **PSEN**, ieșire, Program Store Enable, semnalul de activare a citirii memoriei de program externe
- **EA** / **VPP**, intrare, External Access Enable/Programming Supply Voltage, dacă EA este ținut din exterior în 0, microcontrollerul execută întregul cod din memoria de program externă; dacă EA este ținut din exterior în 1, microcontrollerul execută codul aflat la locațiile de memorie 0000H până la 0FFFH din memoria ROM internă; pinul VPP primește tensiunea de alimentare pentru programare pe perioada programării EPROM-ului



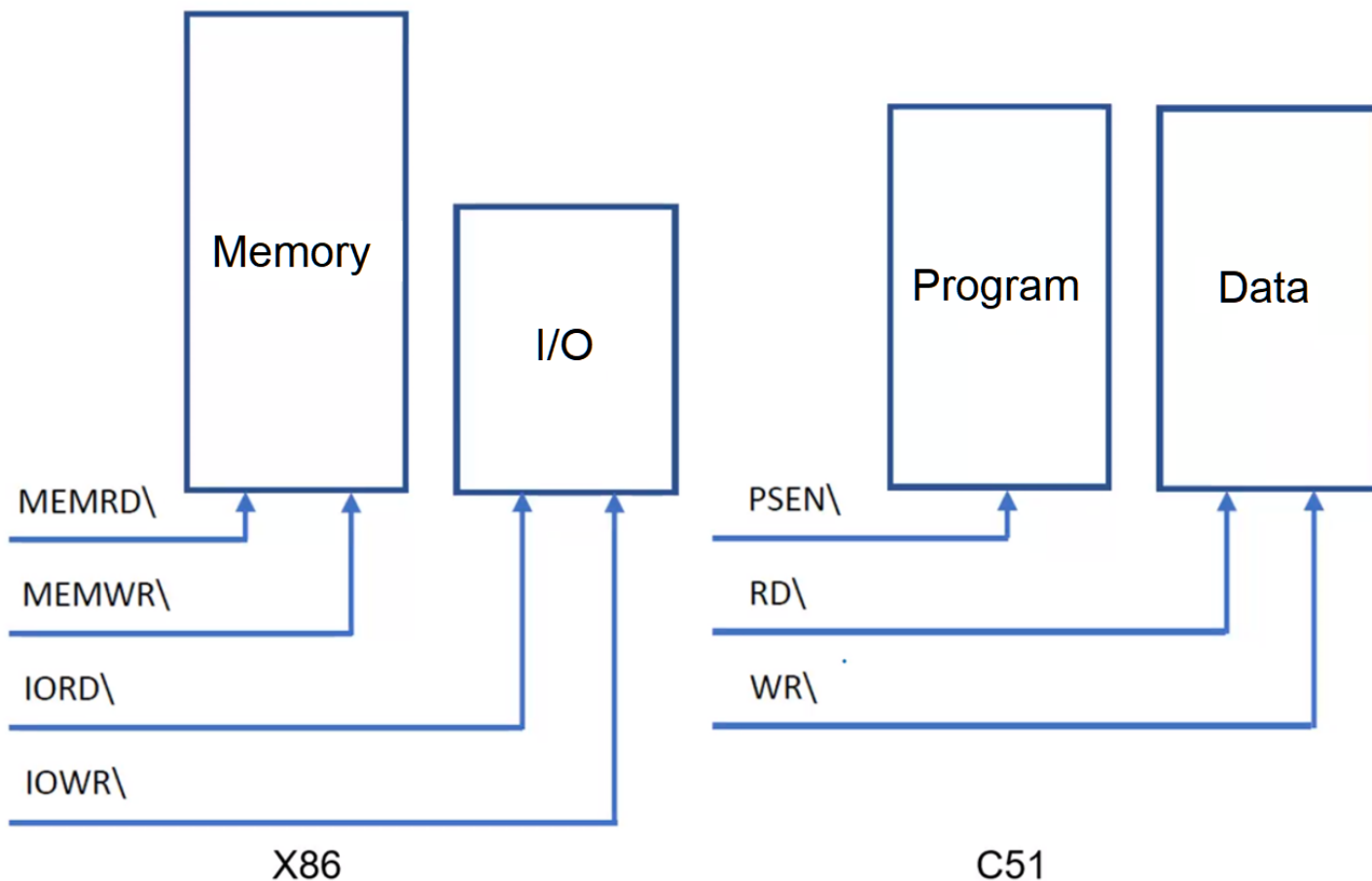
- **XTAL1**, intrare, Crystal 1, intrarea pentru amplificatorul inversor al oscilatorului și circuitul generator de tact
- **XTAL2**, ieșire, Crystal 2, ieșirea de la amplificatorul inversor al oscilatorului



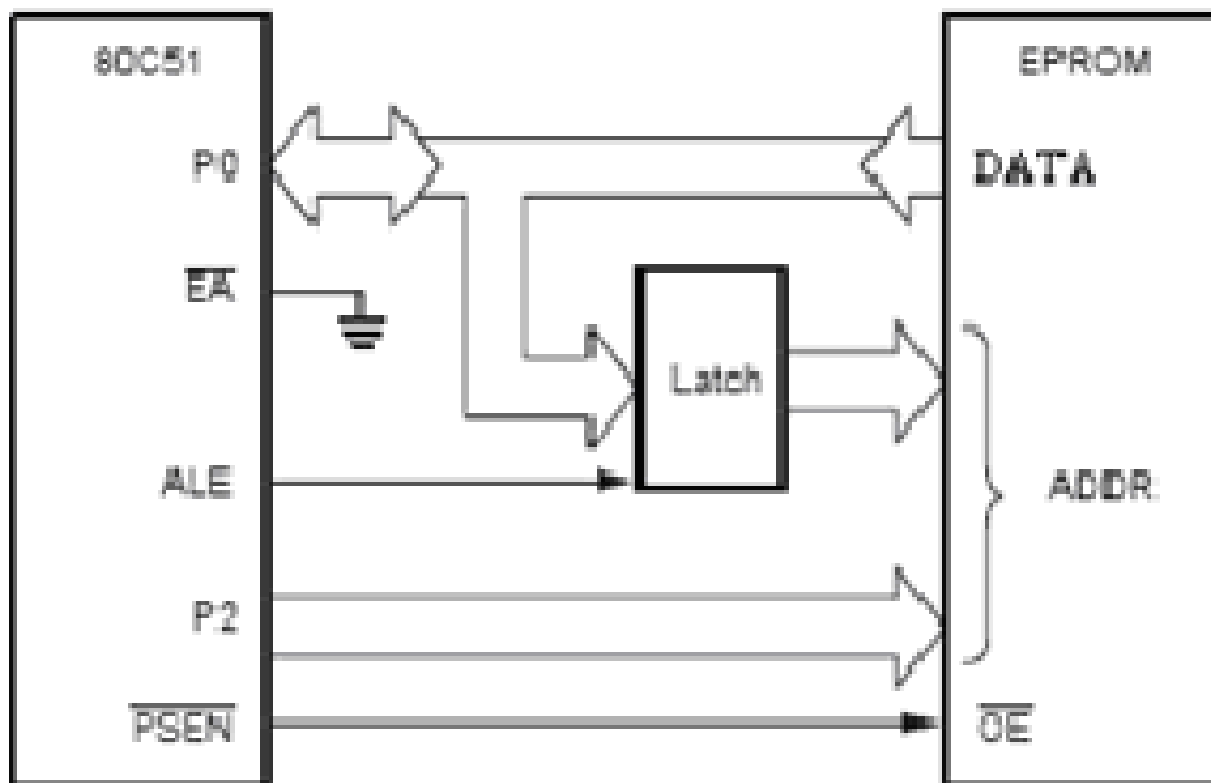
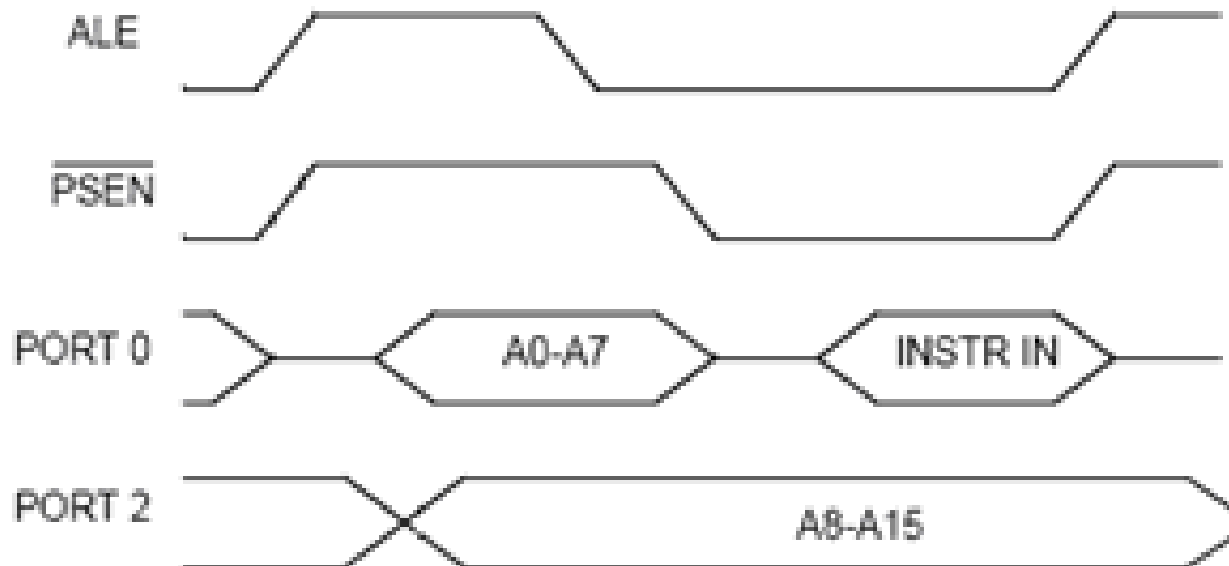
- **Schema internă a portului 0**
- Semnalul CONTROL comandă multiplexorul astfel încât portul 0 să poată genera adrese sau date sau să poată fi folosit ca și port de uz general.
- Dacă semnalul CONTROL are valoarea logică 1, portul 0 este folosit pentru magistrala de adrese și date folosind rezistențe de ridicare interne. Ieșirea porții ȘI-NU nu este influențată de semnalul CONTROL, deci, starea tranzistorului Mp este determinată doar de starea logică a adreselor sau datelor transmise.
- Dacă semnalul CONTROL are valoarea logică 0, portul 0 este un port bidirecțional open-drain. Ieșirea porții ȘI-NU este 1, deci, tranzistorul Mp este în starea blocată, determinând configurația open-drain a portului. Așadar, dacă portul este folosit ca și ieșire, se folosesc rezistențe de ridicare externe. Dacă portul este folosit ca și port de intrare, valoarea logică 1 este setată prin program la pin, deci, tranzistorul Mn este în starea blocată și intrarea este flotantă.



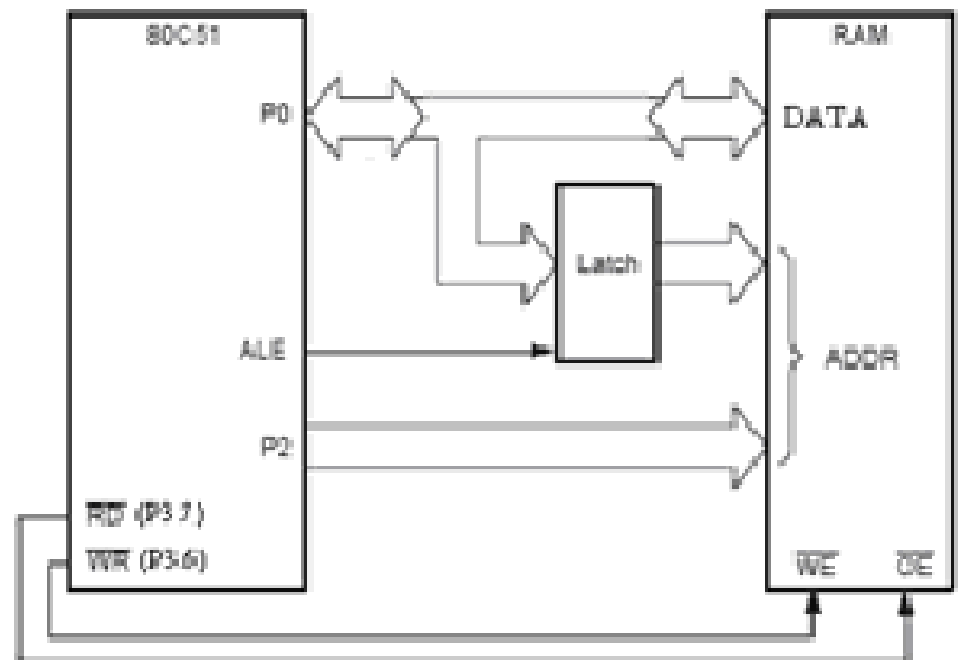
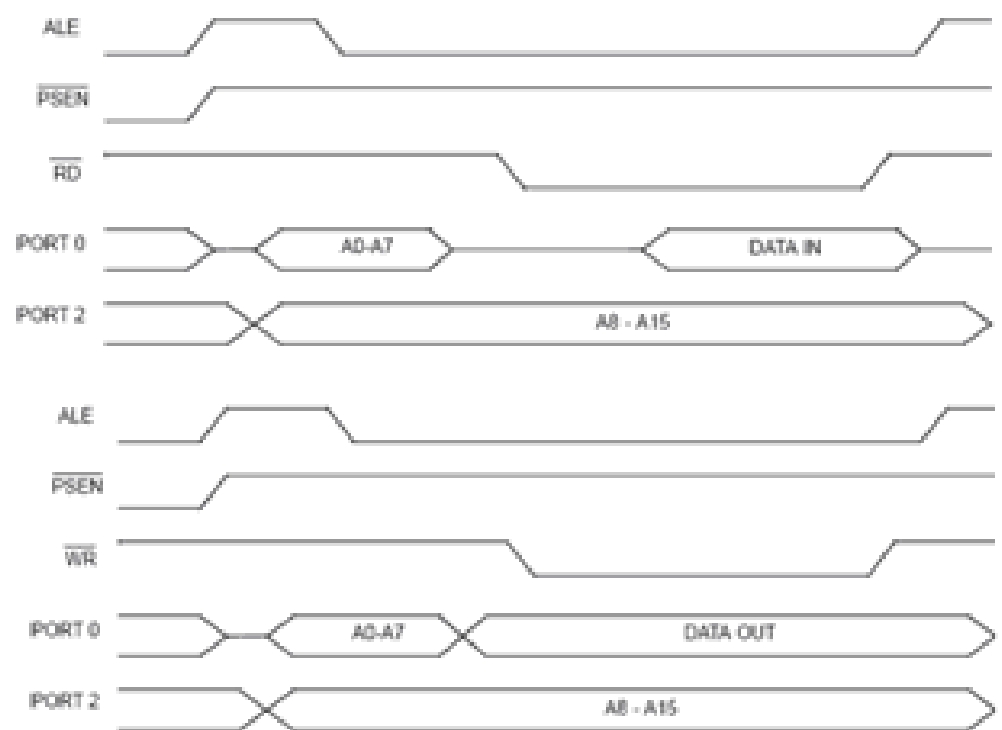
Adresarea memoriei



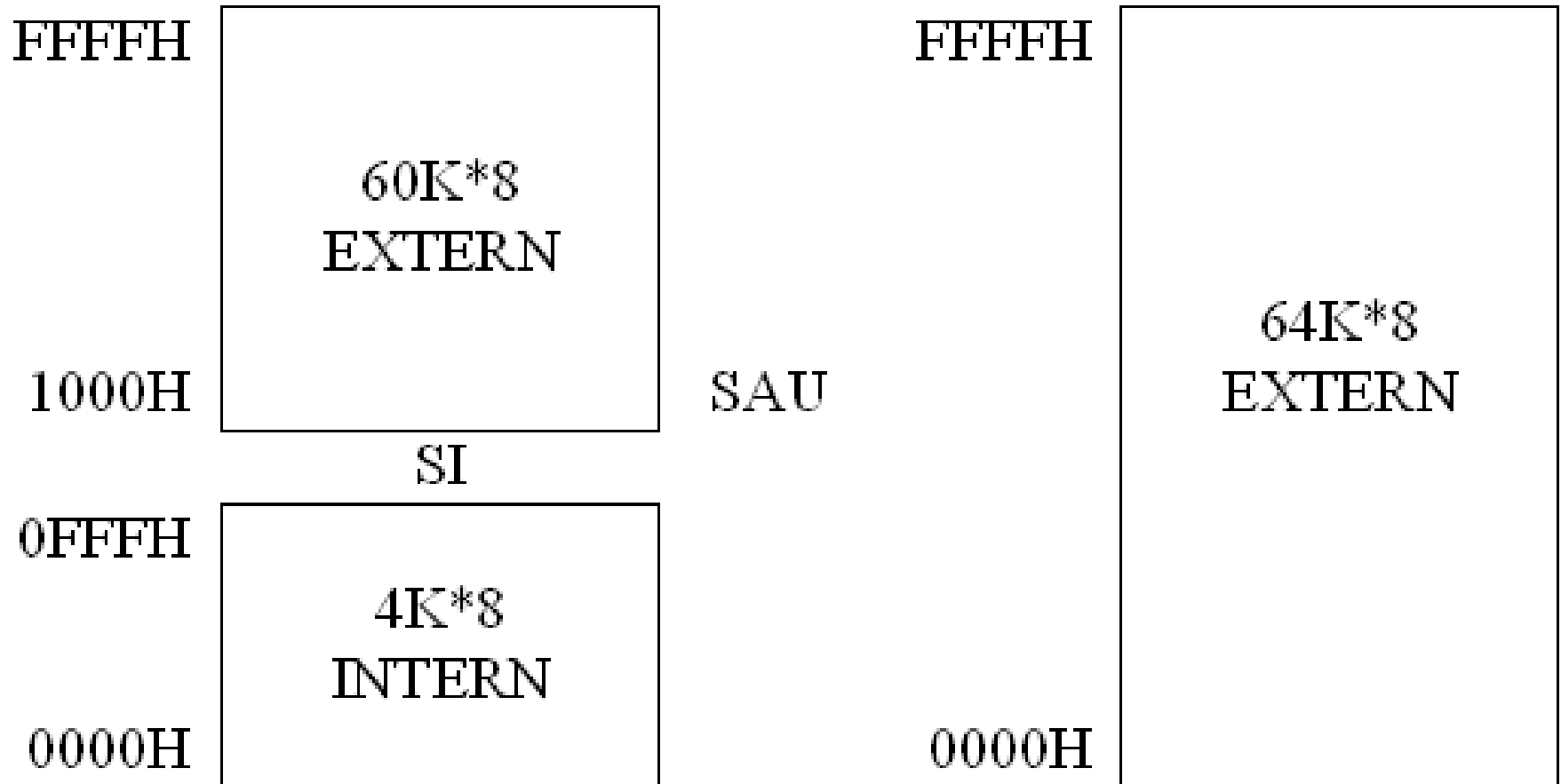
Ciclul de citire și configurația memoriei de program externe



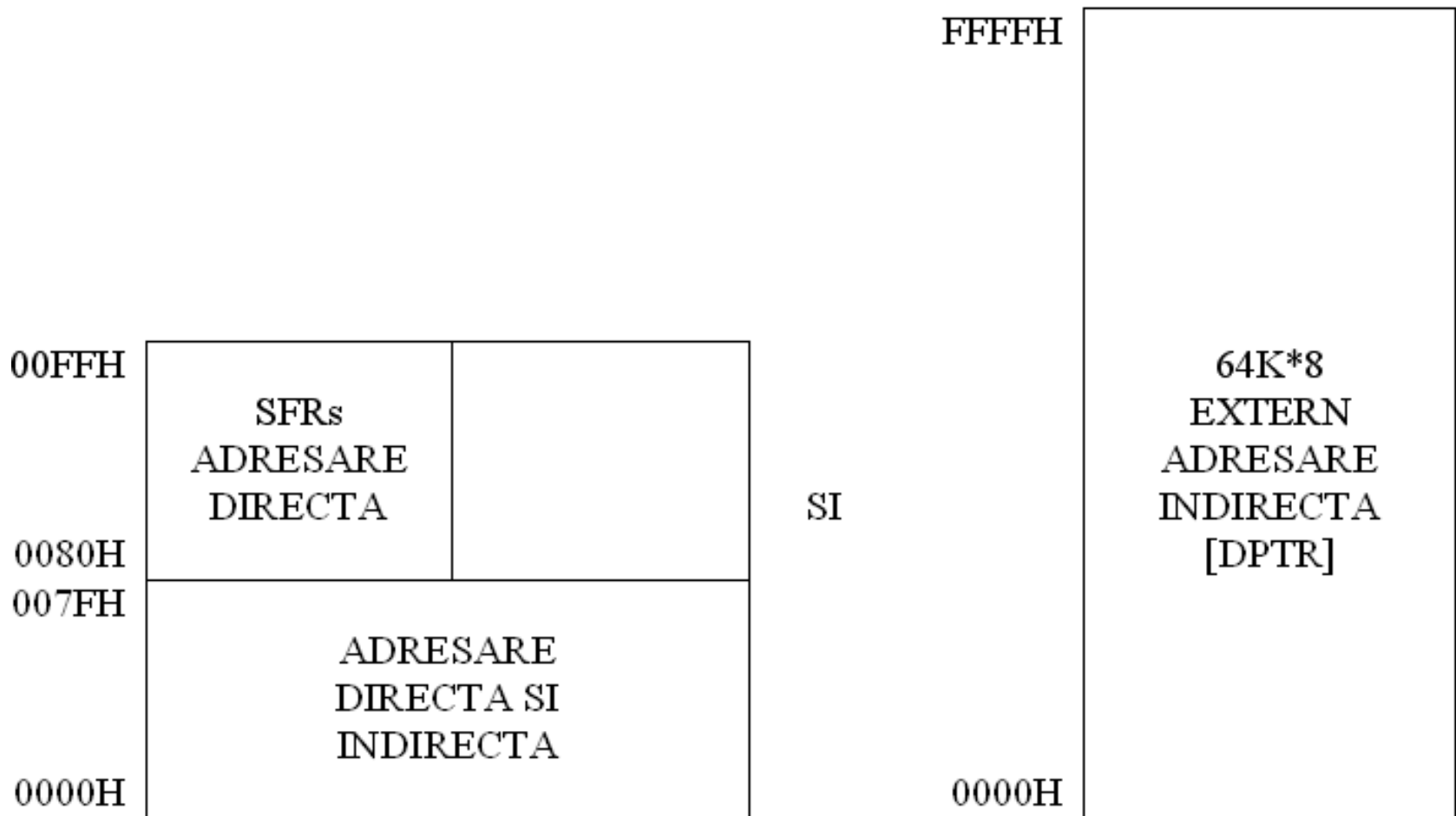
Cicli de citire/scriere și configurarea memoriei de date externe



- **Organizarea memoriei**
- Spații de adresă separate pentru memoria de program și date.
- Memoria de program este o memorie nevolatilă având o lungime de până la 64K*8.
- Rolul pinului EA\



- Acceseaza atât de memorie de date internă cât și externă.
- Memoria de date internă este de 128*8 RAM plus un număr de Regiștri cu Funcțiuni Speciale (Special Function Registers SFRs).
- Memoria de date externă poate avea o lungime de până la 64K*8, fiind o memorie de tip RAM.



- Cele mai mici 128*8 adrese pot fi împărțite în 3 segmente:

- Register Banks 0-3
- Bit Addressable Area
- Scratch Pad Area

← 8 OCTETI →			
78H		7FH	SCRATCH PAD AREA
70H		77H	
68H		6FH	
60H		67H	
58H		5FH	
50H		57H	
48H		4FH	
40H		47H	
38H		3FH	
30H		37H	
28H	...7FH	2FH	BIT ADDRESSABLE SEGMENT
20H	0...	27H	
18H	3	1FH	REGISTER BANKS
10H	2	17H	
08H	1	0FH	
00H	0	07H	

- Regiștrii cu funcțiuni speciale marcați cu * sunt adresabili atât pe bit cât și pe octet. Ceilalți regiștri sunt adresabili doar pe octet. După reset, fiecare registru este încărcat cu o valoare care nu interferă cu posibile valori utilizator. Regiștrii adresabili atât pe octet cât și pe bit se află pe prima coloană.

8 octeti							
F8H							FFH
F0H	B						F7H
E8H							EFH
E0H	ACC						E7H
D8H							DFH
D0H	PSW						D7H
C8H							CFH
C0H							C7H
B8H	IP						BFH
B0H	P3						B7H
A8H	IE						AFH
A0H	P2						A7H
98H	SCON	SBUF					9FH
90H	P1						97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1	8FH
80H	P0	SP	DPL	DPH			87H
						PCON	

Simbol	Descriere	Adresa	Valoare reset
ACC*	Accumulator	E0H	00000000
B*	B Register	F0H	00000000
PSW*	Program Status Word	D0H	00000000
SP	Stack Pointer	81H	00000111
DPTR	Data Pointer 2 Bytes		
DPL	Low Byte	82H	00000000
DPH	High Byte	83H	00000000
P0*	Port 0	80H	11111111
P1*	Port 1	90H	11111111
P2*	Port 2	A0H	11111111
P3*	Port 3	B0H	11111111
IP*	Interrupt Priority Control	B8H	xxx00000
IE*	Interrupt Enable Control	A8H	0xx00000
TMOD	Timer/Counter Mode Control	89H	00000000
TCON*	Timer/Counter Control	88H	00000000
TH0	Timer/Counter 0 High Byte	8CH	00000000
TL0	Timer/Counter 0 Low Byte	8AH	00000000
TH1	Timer/Counter 1 High Byte	8DH	00000000
TL1	Timer/Counter 1 Low Byte	8BH	00000000
SCON*	Serial Control	98H	00000000
SBUF	Serial Data Buffer	99H	xxxxxxxx
PCON	Power Control	87H	00xx0000

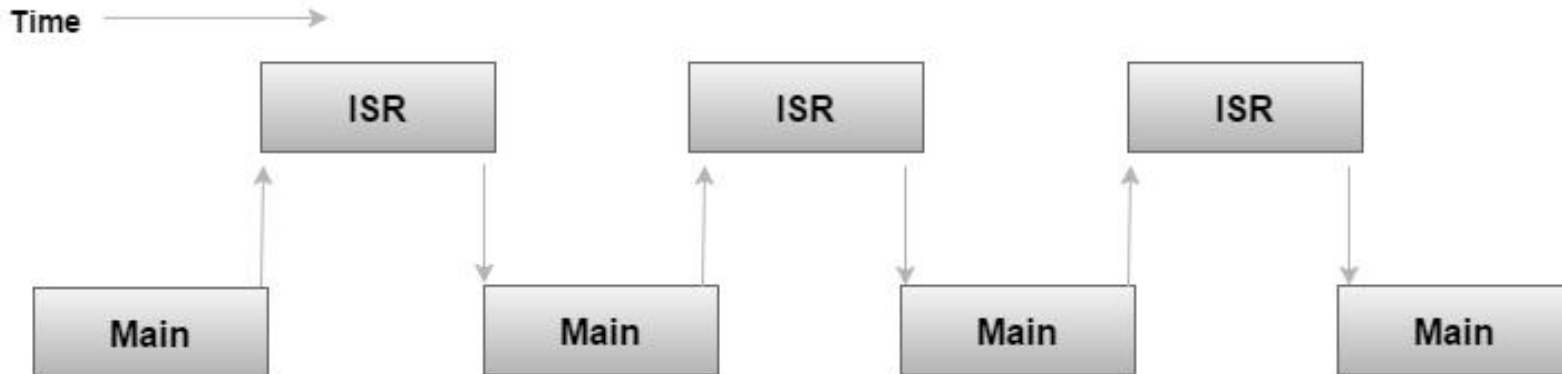
Sistemul de întreruperi

- La apariția unei întreruperi, microcontrollerul suspendă temporar execuția programului și execută rutina de tratare a întreruperii care deservește întreruperea. După aceea, continuă execuția programului.

Program Execution without Interrupts

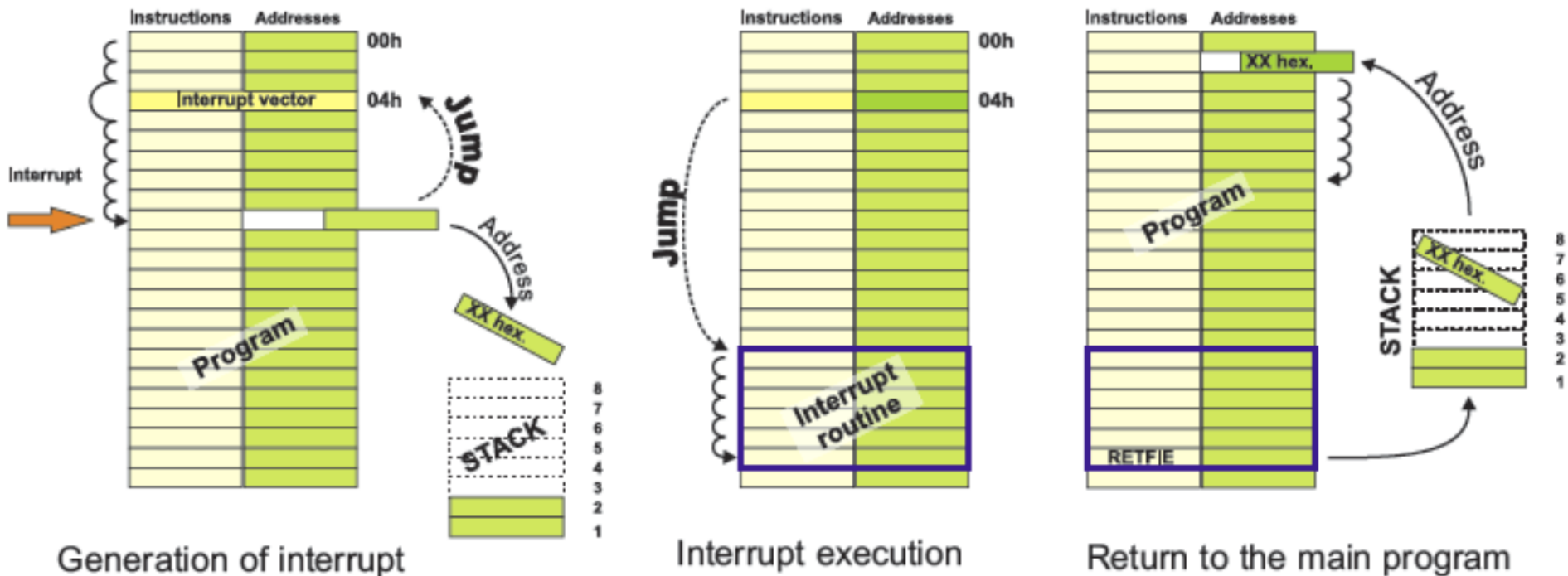


Program Execution with Interrupts

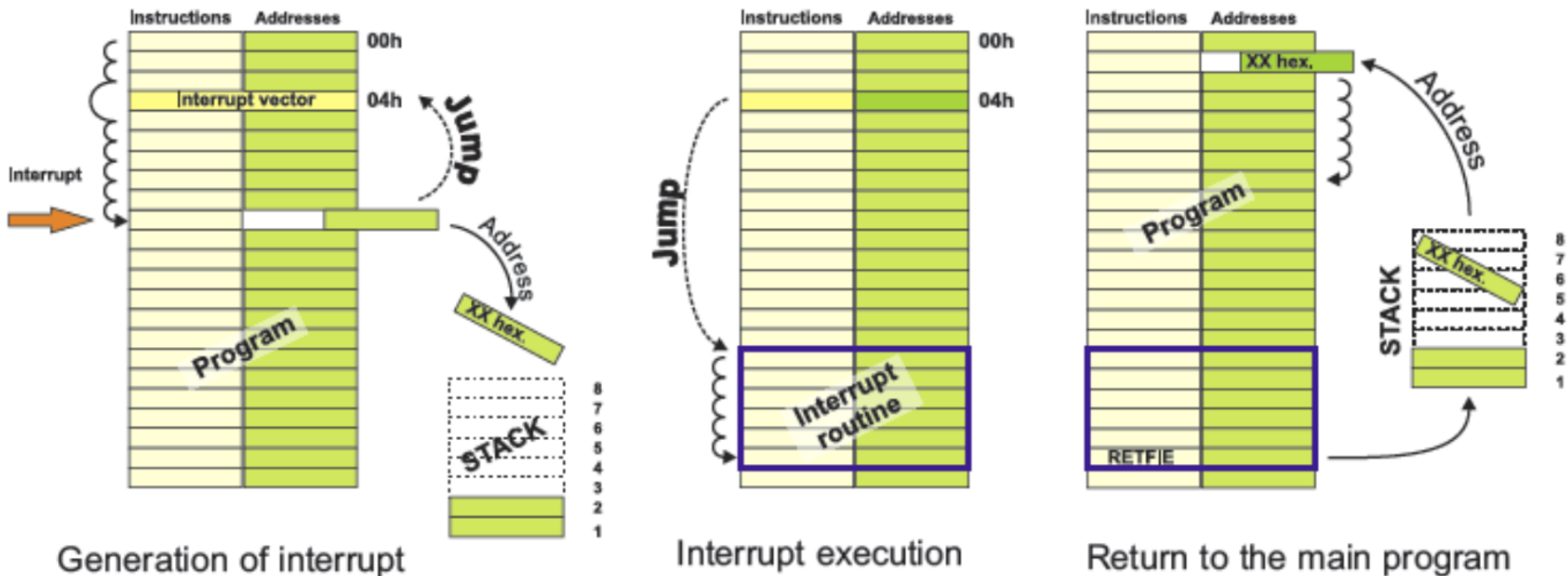


ISR : Interrupt Service Routine

- Pentru a executa rutina de tratare a întreruperii microcontrollerul parcurge următorii pași:
 - Salvează pe stivă locația următoarei instrucțiuni (2 octeți) și PSW (2 octeți).
 - Determină sursa de întrerupere (numărul întreruperii). Fiecare sursă de întrerupere are un vector de întrerupere care este încărcat cu o instrucțiune de salt la adresa rutinei de tratare a întreruperii. Adresa vectorului este calculată automat de către microcontroller. Vectorii de întrerupere (8 octeți) se află în tabela vectorilor de întrerupere.



- Pentru a executa rutina de tratare a întreruperii microcontrollerul parcurge urmatorii pași: (continuare)
 - Accesează tabela vectorilor de întrerupere folosind vectorul de întrerupere ca index pentru a determina adresa rutinei de tratare a întreruperii.
 - Execută rutina de tratare a întreruperii.
 - Întreruperea se termină cu instrucțiunea IRET care încarcă de pe stivă locația următoarei instrucțiuni (2 octeți) și PSW (2 octeți)



- Pentru implementarea întreruperilor, trebuie parcurși următorii pași:
 - Se setază bitul EA din registrul IE la valoarea logică 1. Această setare permite activarea întreruperilor.
 - Se seteaza la valoarea logică 1 în registrul IE biții de activare a întreruperii pentru întreruperile care vor fi folosite.
 - Adresa de început a rutinei de tratare a întreruperii va corespunde adresei vectorului întreruperii respective. Fiecare vector va fi încărcat cu o instrucțiune long jump la adresa rutinei de tratare a întreruperii.
 - In plus, pentru întreruperile externe, pinii INT0\ (P3.2) și INT1\ (P3.3) trebuie setați la valoarea logică 1, și biții corespunzatori (IT0 și IT1) din registrul TCON trebuie resetați sau setați pentru activarea întreruperii pe nivel sau pe front.
- Întreruperile microcontrollerelor din familia 80C51

INTERRUPT SOURCE	DESCRIPTION	VECTOR ADDRESS
IE0	External interrupt 0	0003H
TF0	Timer 0 overflow	000BH
IE1	External interrupt 1	0013H
TF1	Timer 1 overflow	001BH
RI&TI	Serial interrupts	0023H
TF2&EXF2	Timer 2 overflow or external interrupts	002BH

- Structura registrului IE
- Adresabil atât pe octet cât și pe bit. Dacă EA este 0, toate întreruperile sunt dezactivate. Dacă EA este 1, o întrerupere este activată prin setarea bitului corespunzător la 1. Dacă bitul corespunzător este 0 întreruperea este dezactivată. Funcțiile biților:
 - EA, IE.7: dacă EA=0, nici o întrerupere nu va fi achitată. Dacă EA=1, fiecare sursă de întrerupere poate fi activată sau dezactivată individual.
 - ET2, IE.5: timer 2 overflow sau întrerupere de captură (doar 8052)
 - ES, IE.4: întrerupere port serial
 - ET1, IE.3: întrerupere Timer 1 overflow
 - EX1, IE.2: întrerupere externă 1
 - ET0, IE.1: întrerupere Timer 0 overflow
 - EX0, IE.0: întrerupere externă 0

EA	-	ET2	ES	ET1	EX1	ET0	EX0
----	---	-----	----	-----	-----	-----	-----

- Sistemul de priorități al întreruperilor
- Două nivele de prioritate.
 - O prioritate mai mare poate întrerupe o prioritate mai mică.
 - O prioritate mai mică nu poate întrerupe o prioritate mai mare.
- Pentru asignarea unei priorități mai mari sau mai mici unei întreruperi bitul corespunzător din registrul IP trebuie setat la 1 sau 0.
- În același nivel de prioritate sunt mai multe priorități.
 - Prioritățile din același nivel nu pot fi întrerupte de alte priorități din același nivel chiar dacă în interiorul nivelului de prioritate acele priorități au un nivel mai mare.
 - Prioritățile în interiorul unui nivel se folosesc doar pentru rezolvarea cererilor simultane ale aceluiași nivel de prioritate.
- Prioritățile de la mare la mic: IE0, TF0, IE1, TF1, RI sau TI și TF2 sau EXF2.

- Structura registrului IP
- Adresabil atât pe octet cât și pe bit. Dacă bitul este 0, întreruperea corespunzătoare are o prioritate mai mică. Dacă bitul este 1, întreruperea corespunzătoare are o prioritate mai mare. Funcțiile biților:
 - PT2, IP.5: nivelul de prioritate al întreruperii Timer 2 (doar 8052)
 - PS, IP.4: nivelul de prioritate al întreruperii portului serial
 - PT1, IP.3: nivelul de prioritate al întreruperii Timer 1
 - PX1, IP.2: nivelul de prioritate al întreruperii externe 1
 - PT0, IP.1: nivelul de prioritate al întreruperii Timer 0
 - PX0, IP.0: nivelul de prioritate al întreruperii externe 0

-	-	PT2	PS	PT1	PX1	PT0	PX0
---	---	-----	----	-----	-----	-----	-----

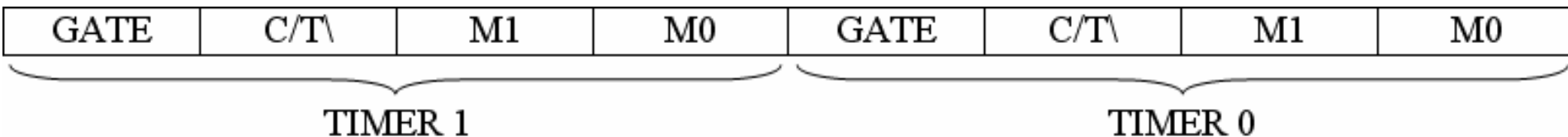
Regiștrii sistemului de timere

- Structura registrului TCON
- Adresabil atât pe octet cât și pe bit. Funcțiile biților:
 - TF1, TCON.7: Flag Timer 1 overflow. Setat hardware la overflow Timer 1. Șters hardware când procesorul execută rutina de tratare a întreruperii.
 - TR1, TCON.6: Bit de control funcționare Timer 1. Dacă TR1=1, Timer 1 ON. Dacă TR1=0, Timer 1 OFF.
 - TF0, TCON.5: Flag Timer 0 overflow. Setat hardware la overflow Timer 0. Șters hardware când procesorul execută rutina de tratare a întreruperii.
 - TR0, TCON.4: Bit de control funcționare Timer 0. Dacă TR0=1, Timer 0 ON. Dacă TR0=0, Timer 0 OFF.
 - IE1, TCON.3: Flag de front întrerupere externă 1. Setat hardware când este detectat frontul la întreruperea externă 1, șters hardware când întreruperea este procesată.
 - IT1, TCON.2: Bit de control tip întrerupere 1. Dacă IT1=1, întreruperea 1 este declanșată de un front căzător. Dacă IT1=0, întreruperea 1 este declanșată de nivelul logic 0.
 - IE0, TCON.1: Flag de front întrerupere externă 0. Setat hardware când este detectat frontul la întreruperea externă 0, șters hardware când întreruperea este procesată.
 - IT0, TCON.0: Bit de control tip întrerupere 0. Dacă IT0=1, întreruperea 0 este declanșată de un front căzător. Dacă IT0=0, întreruperea 0 este declanșată de nivelul logic 0.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

- Structura registrului TMOD
- Adresabil pe octet. Funcțiile biților:
- GATE: Dacă GATE=1, TIMERx va rula doar atât timp cât TRx=1 și INTx=1 (control hardware). Dacă GATE=0, TIMERx va rula doar atât timp cât TRx=1 (control software).
- C/T\, Selector Timer sau Counter. Dacă C/T\=0, operare Timer (intrare de la ceasul sistem intern). Dacă C/T\=1, operare Counter (intrare de la pinul de intrare Tx).
- M1: Bit de selectie mod.
- M0: Bit de selectie mod.

M1	M0	MOD DE OPERARE
0	0	Timer 13-biți
0	1	Timer/Counter 16-biți
1	0	Timer/Counter 8-biți cu Auto-Reload
1	1	(Timer 0) TL0 Timer/Counter 8-biți controlat prin biții de control standard corespunzători Timer 0. TH0 Timer 8 biți controlat prin biții de control corespunzători Timer 1.
1	1	(Timer 1) Timer/Counter 1 OFF.



Comunicația serială

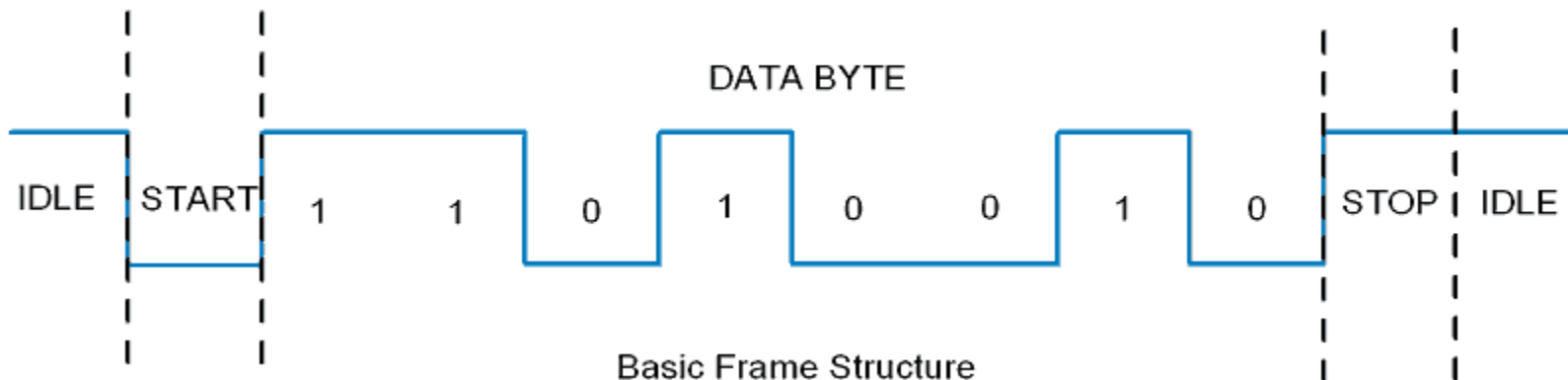
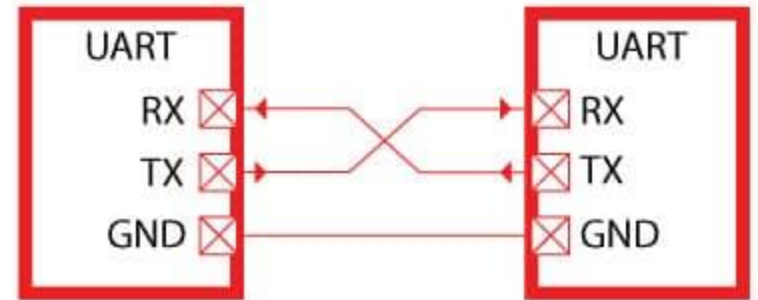
- Structura registrului SCON
- Adresabil atât pe octet cât și pe bit. Funcțiile biților:
 - SM0: Bit 0 mod Port Serial (MSB).
 - SM1: Bit 1 mod Port Serial (LSB).
 - SM2: Activează caracteristica de comunicare multiprocesor în modurile 2 și 3. În modurile 2 sau 3, dacă SM2=1, RI va fi activat (setat la 1) doar dacă al 9-lea bit de date recepționat (RB8) este 1. În modul 1, dacă SM2=1, RI va fi activat doar dacă un bit de stop valid a fost recepționat. În modul 0, SM2=0.
 - REN: Setat/șters prin software pentru a Activa/Dezactiva recepția.
 - TB8: Al 9-lea bit care va fi transmis în modurile 2 și 3.
 - RB8: În modurile 2 și 3, este al 9-lea bit de date recepționat. În modul 1, dacă SM2=0, RB8 este bitul de stop recepționat. În modul 0, RB8 nu este folosit.
 - TI: Flag de întrerupere la transmisie. Setat prin hardware. Trebuie șters prin software.
 - RI: Flag de întrerupere la recepție. Setat prin hardware. Trebuie șters prin software.

SM0	SM1	Descriere
0	0	Registru de shiftare
0	1	UART pe 8 biți
1	0	UART pe 9 biți
1	1	UART pe 9 biți

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

- În **modul 0** portul serial comunică la rata baud fixă.
- $\text{Rata Baud} = \text{Osc Freq}/12$
- Pentru configurarea acestui mod trebuie definit doar registrul SCON. Nu este necesară setarea unui Timer/Counter.
- În acest mod portul de comportă ca și un registru de shiftare de opt biți transmițând și recepționând date la rata baud. Datele sunt transmise cu LSB primul bit și intră și ies din UART prin pinul RXD. În consecință acest mod nu suportă comunicația full duplex. Transmisia începe când registrul SBUF este încărcat cu date. Octetul este transmis și bitul TI este setat când transmisia este completă. Recepția începe când bitul REN din registrul SCON este setat. Bitul RI este setat când octetul este recepționat.

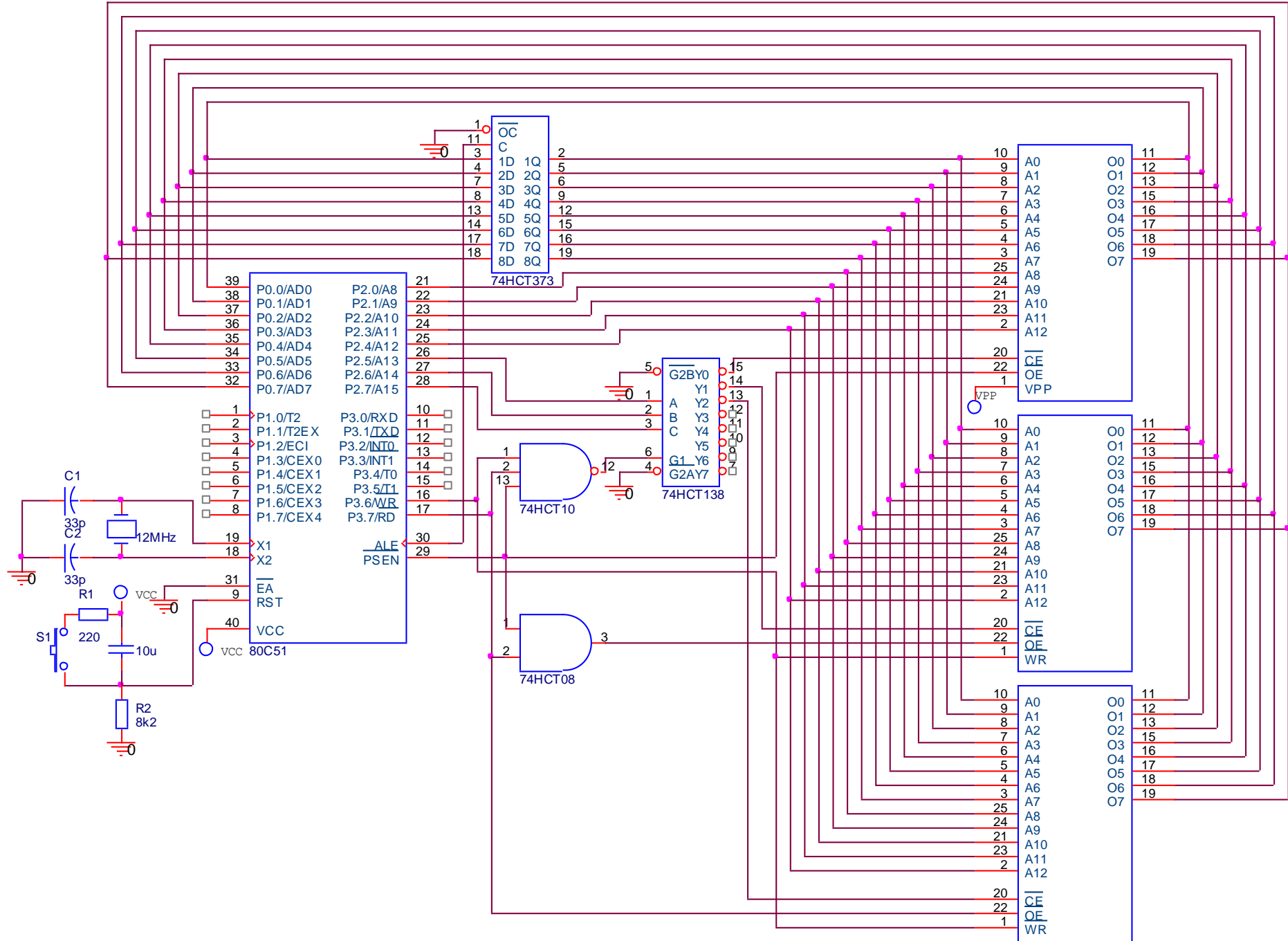
- În **modul 1** rata baud este variabilă fiind generată de Timer 1 care este folosit în modul 2 (Auto-Reload).
- $\text{Rata Baud} = (K * \text{Osc Freq}) / \{32 * 12 * [256 - (\text{TH1})]\}$
- Bitul SMOD din registrul PCON determină valoarea parametrului K. Dacă $\text{SMOD} = 0$, $K = 1$, dacă $\text{SMOD} = 1$, $K = 2$.
- $\text{TH1} = 256 - (K * \text{Osc Freq}) / (384 * \text{Rata Baud})$
- Formatul cadrului: 1 bit de start, 8 biți de date, 1 bit de stop.
- Semnalele de întrerupere TI și RI sunt activate atunci când un cadru a fost transmis sau recepționat.

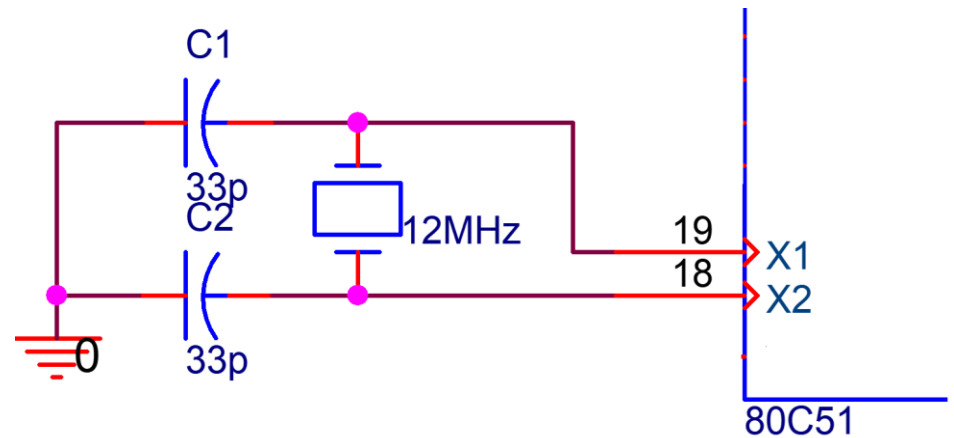
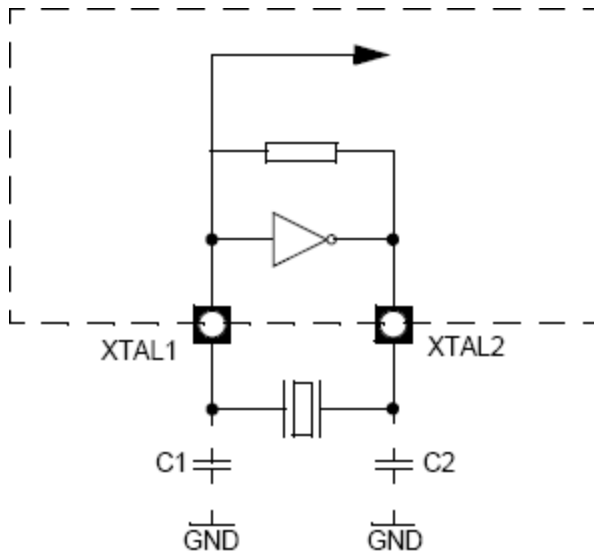


- In **modul 2** portul serial comunică la rata baud fixă.
- $SMOD = 1$, Rata Baud = $1/32 \cdot \text{Osc Freq}$.
- $SMOD = 0$, Rata Baud = $1/64 \cdot \text{Osc Freq}$.
- Pentru configurarea acestui mod trebuie definit doar registrul SCON. Nu este necesară setarea unui Timer/Counter.
- Formatul cadrului: 1 bit de start, 8 biți de date, al 9-lea (stick) bit, și 1 bit de stop. Valoarea celui de-al 9-lea bit este determinată de bitul TB8 la transmitător și este recepționată în bitul RB8. Acest bit este folosit pentru comunicații interprocesor. UART-ul poate fi inițializat prin setarea bitului SM2 să genereze o întrerupere la recepție doar atunci când bitul al 9-lea este setat. Un octet de adresă sau comandă poate fi transmis cu al 9-lea bit setat. Toate procesoarele sunt întrerupte și bitul recepționat este procesat pentru a se vedea dacă este necesară recepționarea mesajului. Dacă microcontrollerul trebuie să recepționeze mesajul, bitul SM2 este șters și restul mesajului este recepționat. Dacă microcontrollerul nu trebuie să recepționeze mesajul, bitul SM2 este lăsat setat și mesajul nu va genera o întrerupere.
- **Modul 3** este similar modului 2 în termenii formatului cadrului și folosirii celui de-al 9-lea bit. Ratele baud sunt similare modului 1.

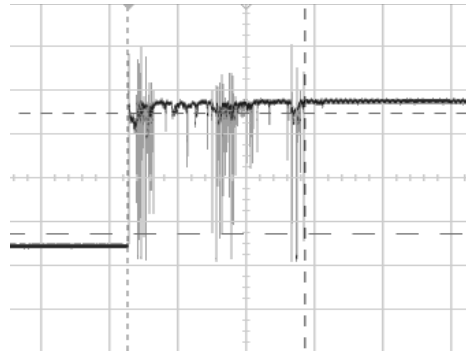
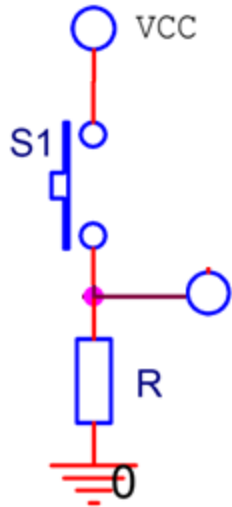
Probleme rezolvate

- Sa se proiecteze un sistem cu microcontroller 80C51 avand urmatoarele blocuri:
 - oscilator cu quartz avand frecventa de 12MHz
 - circuit de reset
 - 8k*8 memorie de program externa avand adresa de baza 0000H
 - 8k*8 memorie de date si program externa avand adresa de baza 2000H
 - 8k*8 memorie de date externa avand adresa de baza 4000H



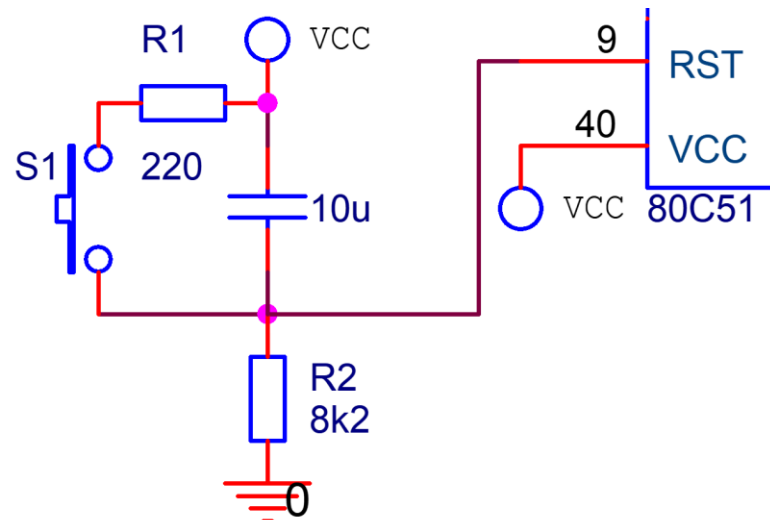


- oscilator cu quartz avand frecventa de 12MHz



- semnal obtinut la apasarea unei taste

- circuit de reset

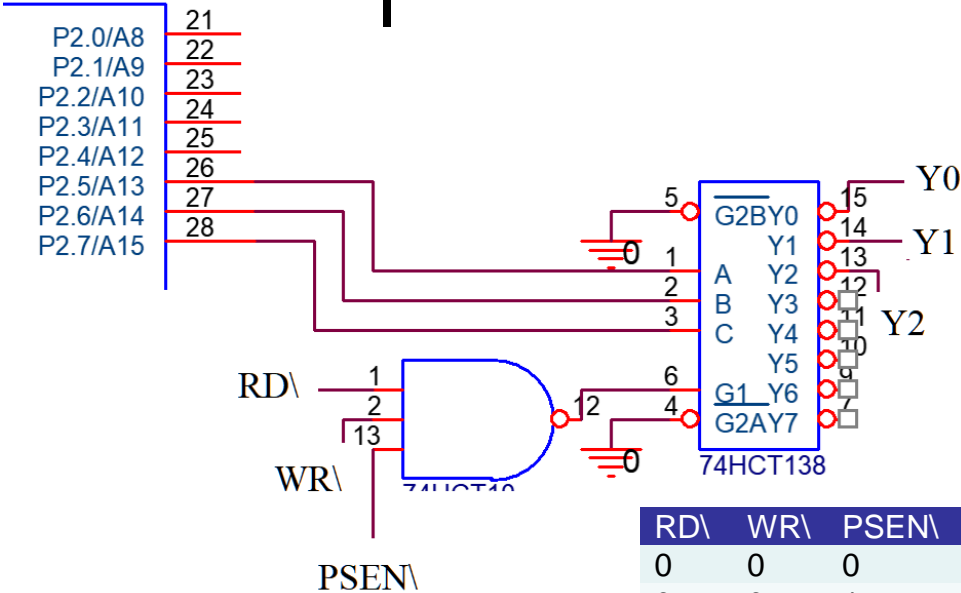


- 8k*8 memorie de program externa avand adresa de baza 0000H
- 8k*8 memorie de date si program externa avand adresa de baza 2000H
- 8k*8 memorie de date externa avand adresa de baza 4000H

0000H: **0000**.....0b
 2000H: **0010**.....0b
 4000H: **0100**.....0b

64k*8	A15	A14	A13
8k*8	0	0	0
8k*8	0	0	1
8k*8	0	1	0
8k*8	0	1	1
8k*8	1	0	0
8k*8	1	0	1
8k*8	1	1	0
8k*8	1	1	1

A15 A14 A13 | A12 A11... A0
 Selectie bloc | 13 linii adresa (8k*8)



RD\	WR\	PSEN\	NAND
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

- 8k*8 memorie de program externa avand adresa de baza 0000H
- mag. adrese: P0 si P2
- mag. date: P0
- mag. control: ALE, PSEN\ si EA\

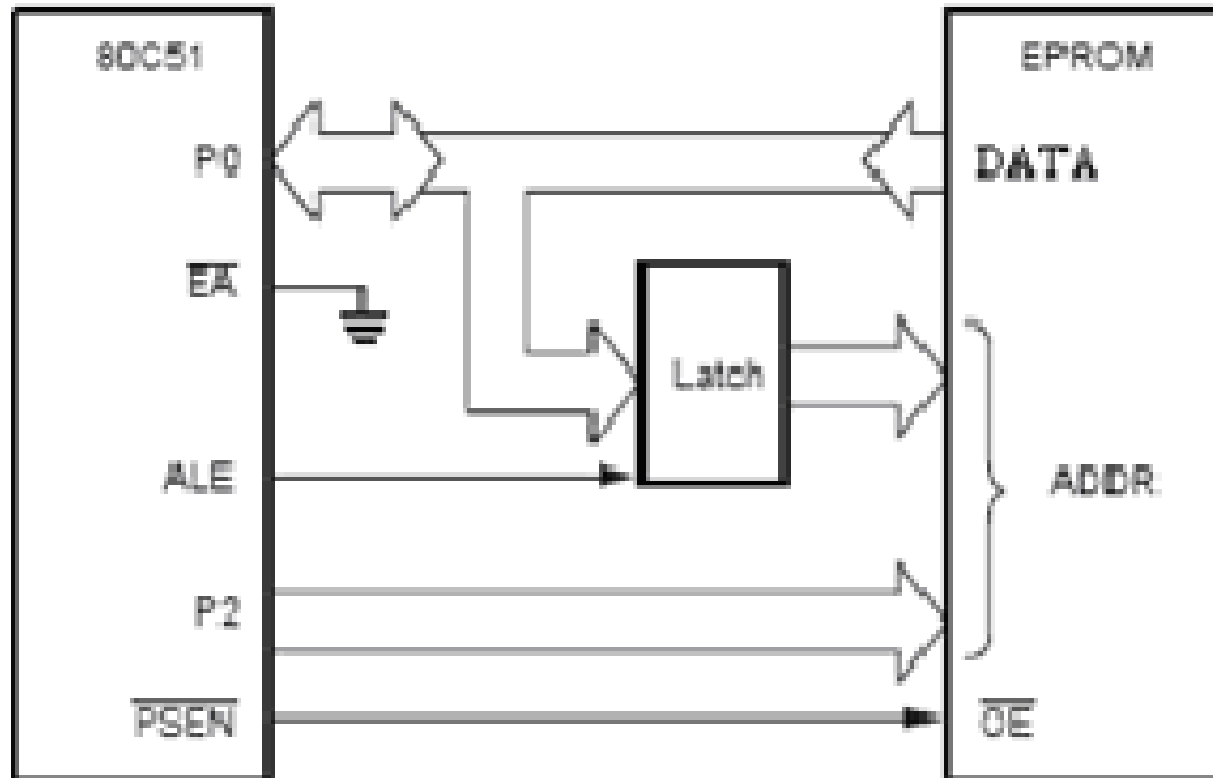
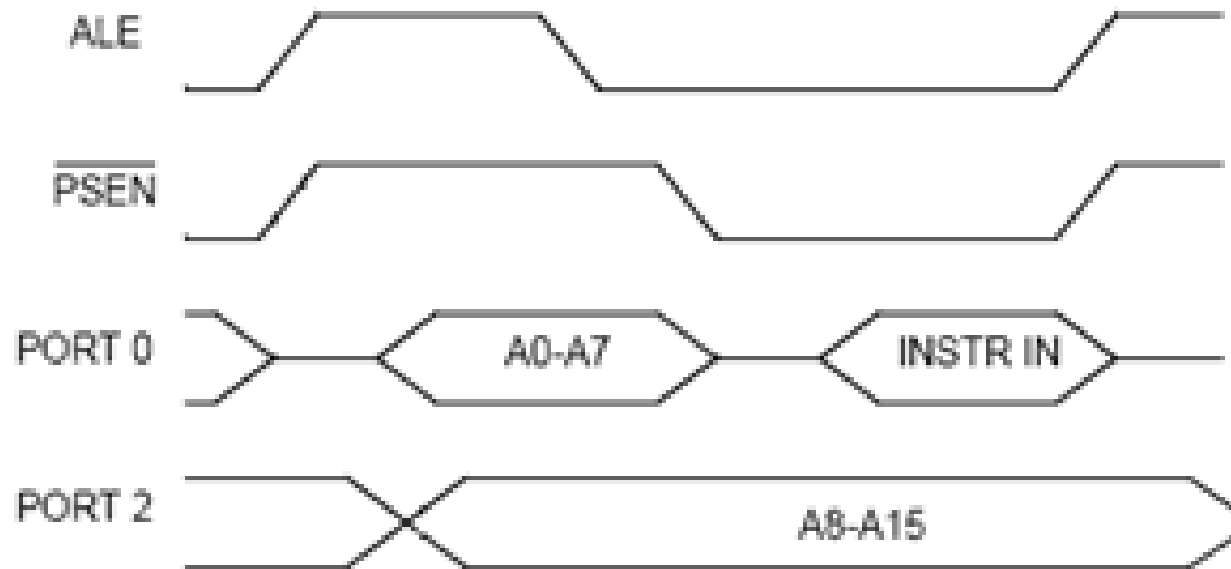
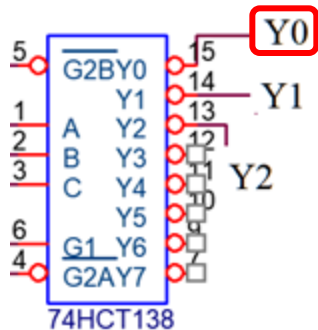
0000H: **0000**.....0b

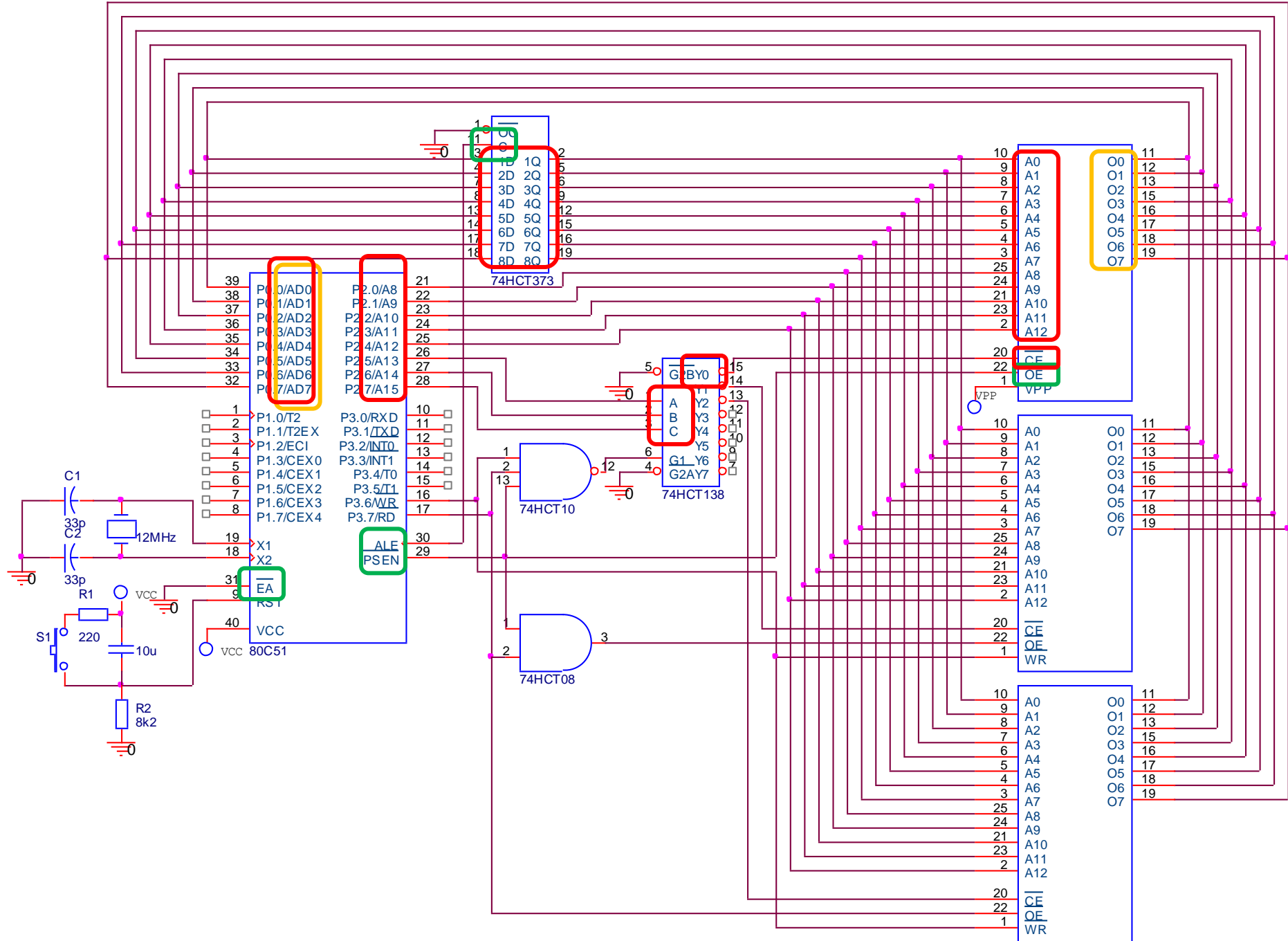
2000H: **0010**.....0b

4000H: **0100**.....0b

A15 A14 A13 A12 A11... A0

Selectie bloc 13 linii adresa





- 8k*8 memorie de date externa avand adresa de baza 4000H
- mag. adrese: P0 si P2
- mag. date: P0
- mag. control: ALE, RD\ si WR\

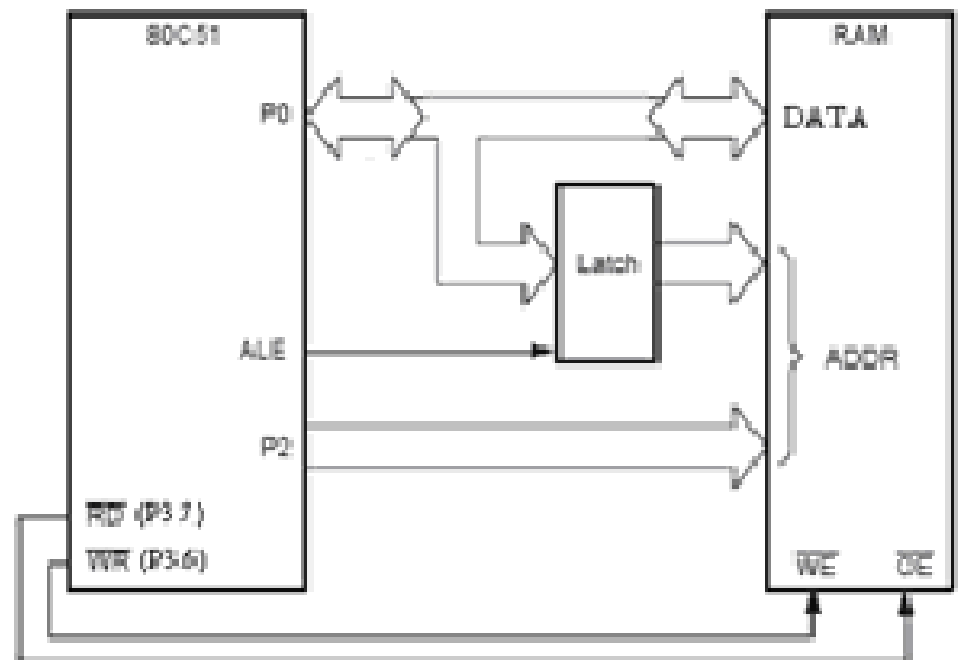
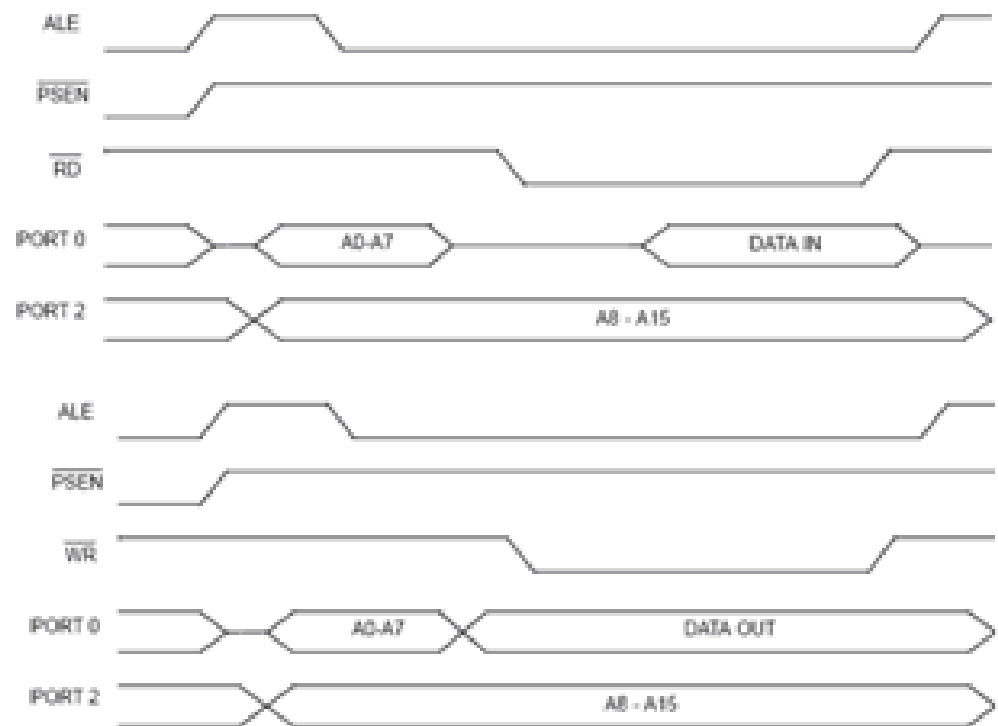
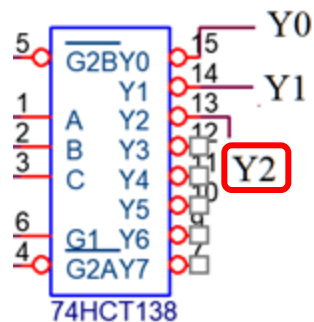
0000H: **0000**.....0b

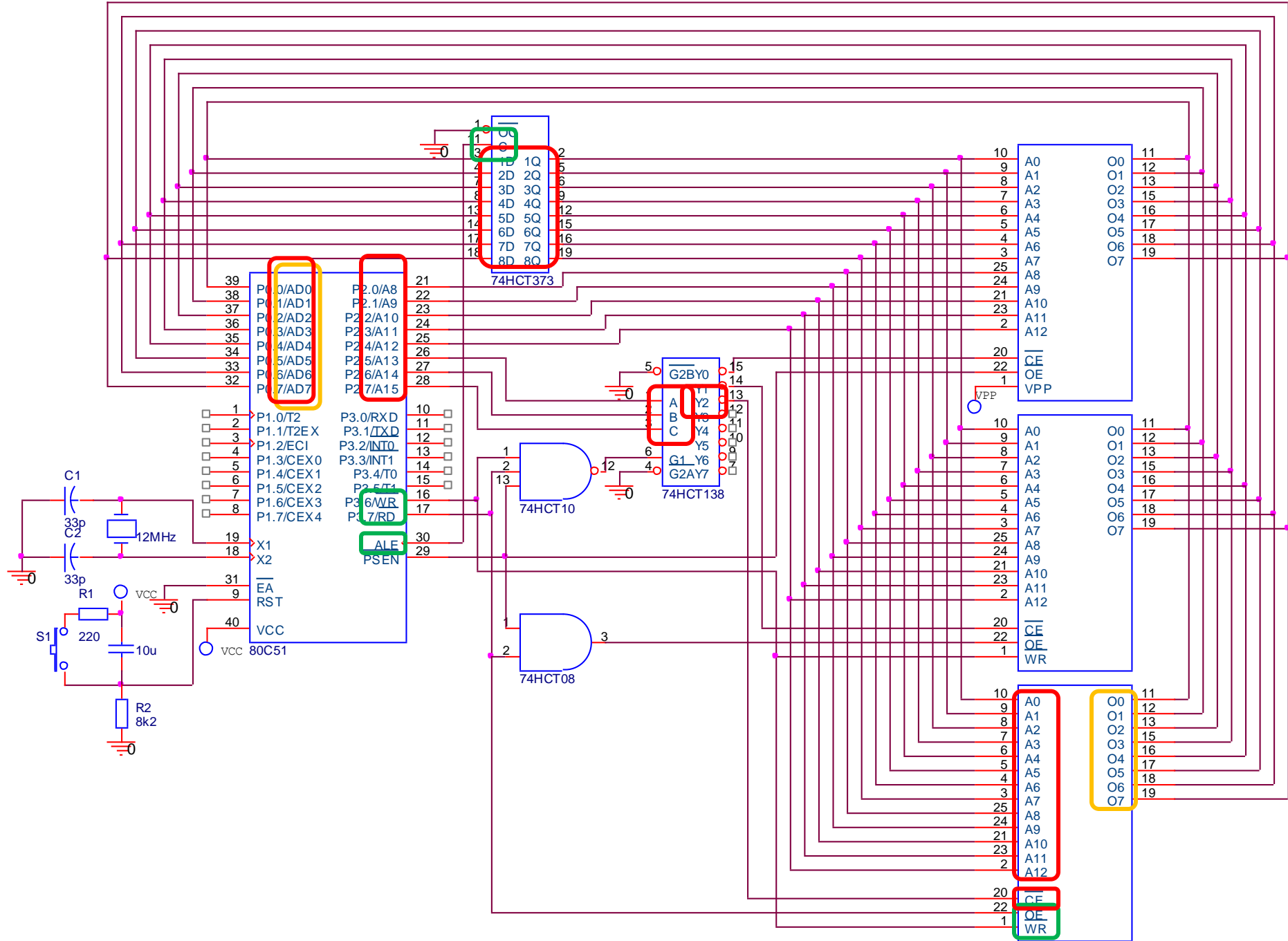
2000H: **0010**.....0b

4000H: **0100**.....0b

A15 A14 A13 A12 A11... A0

Selectie bloc 13 linii adresa





- 8k*8 memorie de date si program externa avand adresa de baza 2000H
- mag. adrese: P0 si P2
- mag. date: P0
- mag. control: ALE, RD\, PSEN\ si WR\

Citire: memorie de date si program (RD\ sau PSEN\)

RD\ PSEN\ | **OE** (functie SI)

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

0000H: **0000**.....0b

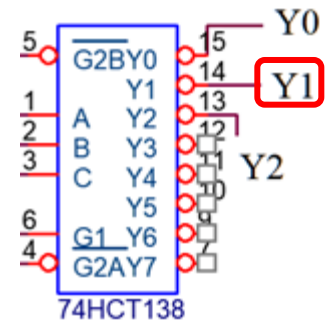
2000H: **0010**.....0b

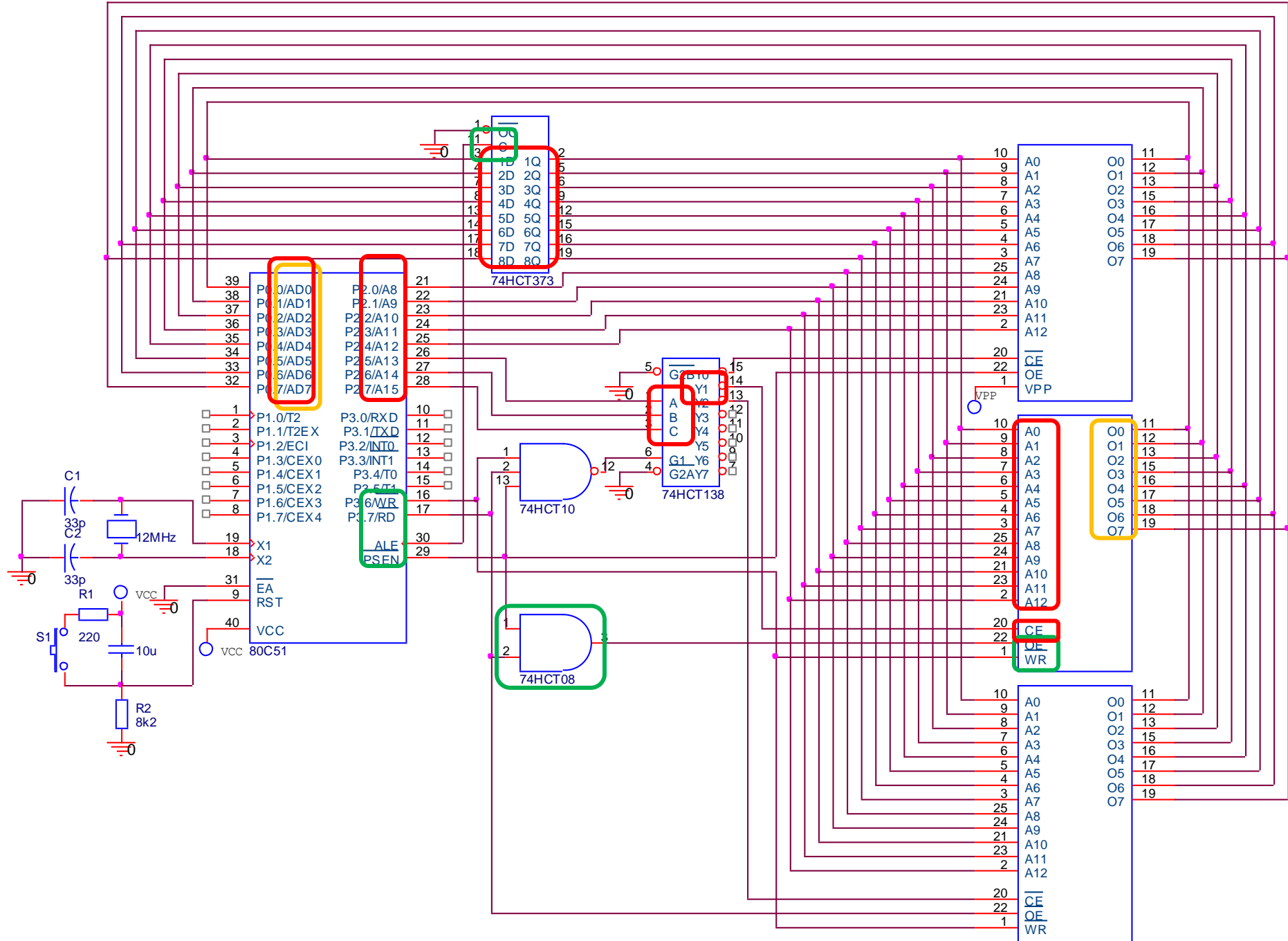
4000H: **0100**.....0b

A15 A14 A13 A12 A11... A0

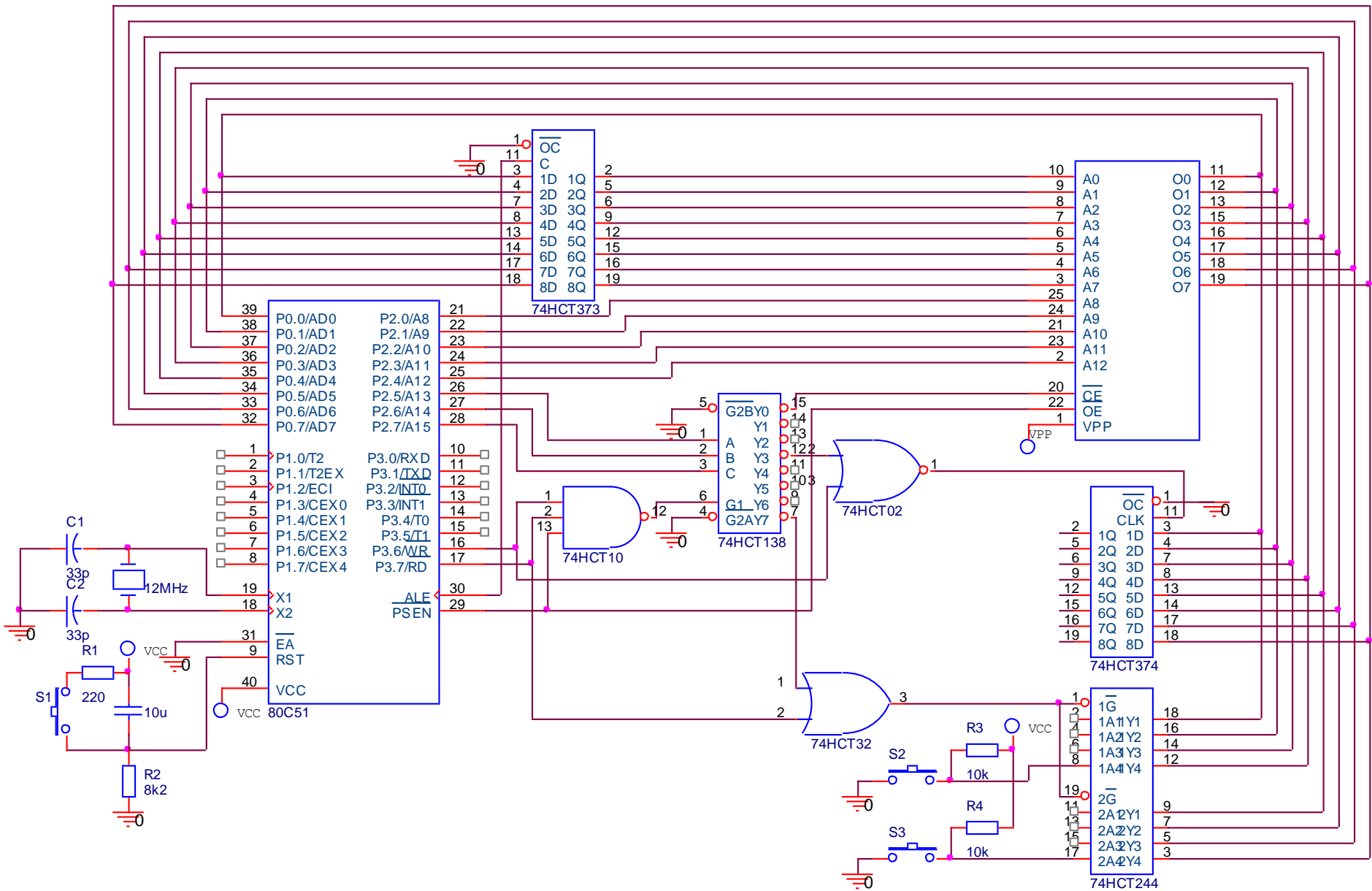
Selectie bloc 13 linii adresa

Sciere: memorie de date (WR\)





- Sa se proiecteze un sistem cu microcontroller 80C51 avand urmatoarele blocuri:
 - oscilator cu quartz avand frecventa de 12MHz
 - circuit de reset
 - 8k*8 memorie de program externa avand adresa de baza 0000H
 - port de iesire la adresa 011XXXXXXXXXXXXXb
 - port de intrare la adresa 111XXXXXXXXXXXXXb la care sunt conectate 2 taste



- 8k*8 memorie de program externa avand adresa de baza 0000H
- port de iesire la adresa 011XXXXXXXXXXXXXXXXXb
- port de intrare la adresa 111XXXXXXXXXXXXXXXXXb la care sunt conectate 2 taste

00000.....0b
011XX.....Xb
111XX.....Xb

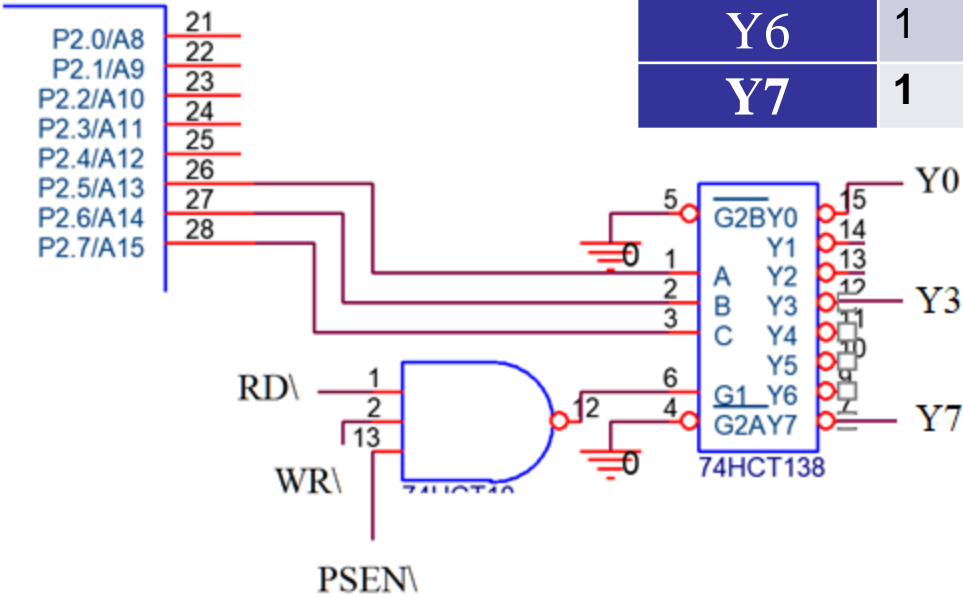
memorie de program externa 0000H
port de iesire, selectie partiala
port de intrare, selectie partiala

A15 A14 A13
Selectie bloc memorie
Selectie porturi

A12 A11... A0
13 linii adresa (8k*8)
selectie partiala

DCD	A15	A14	A13
Y0	0	0	0
Y1	0	0	1
Y2	0	1	0
Y3	0	1	1
Y4	1	0	0
Y5	1	0	1
Y6	1	1	0
Y7	1	1	1

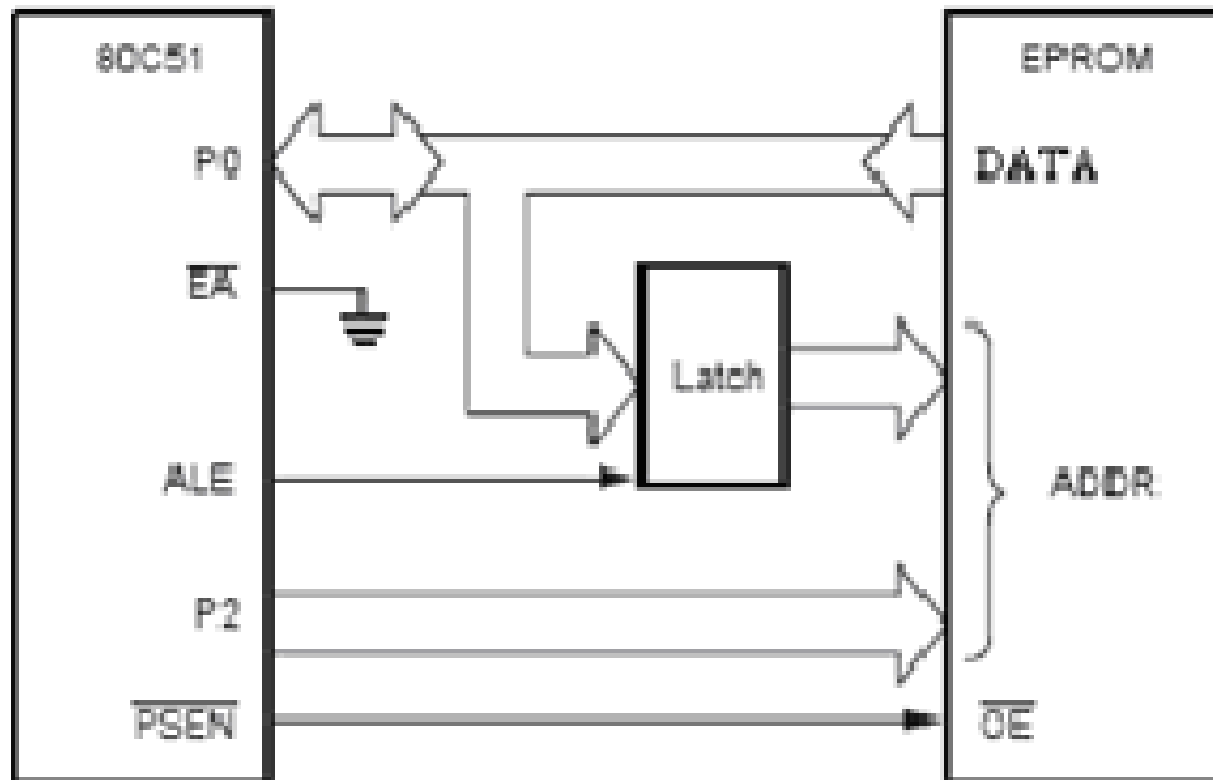
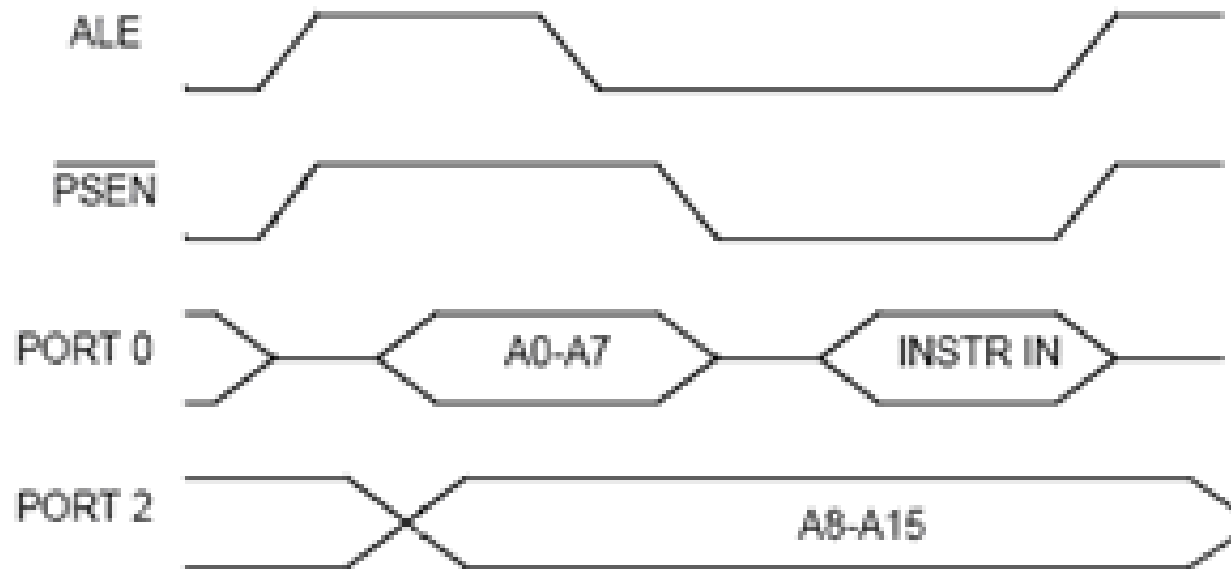
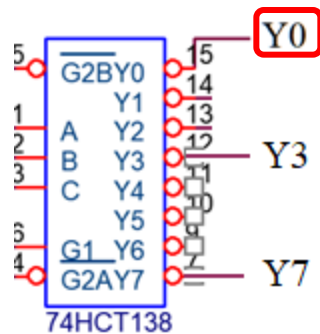
RD\	WR\	PSEN\	NAND
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

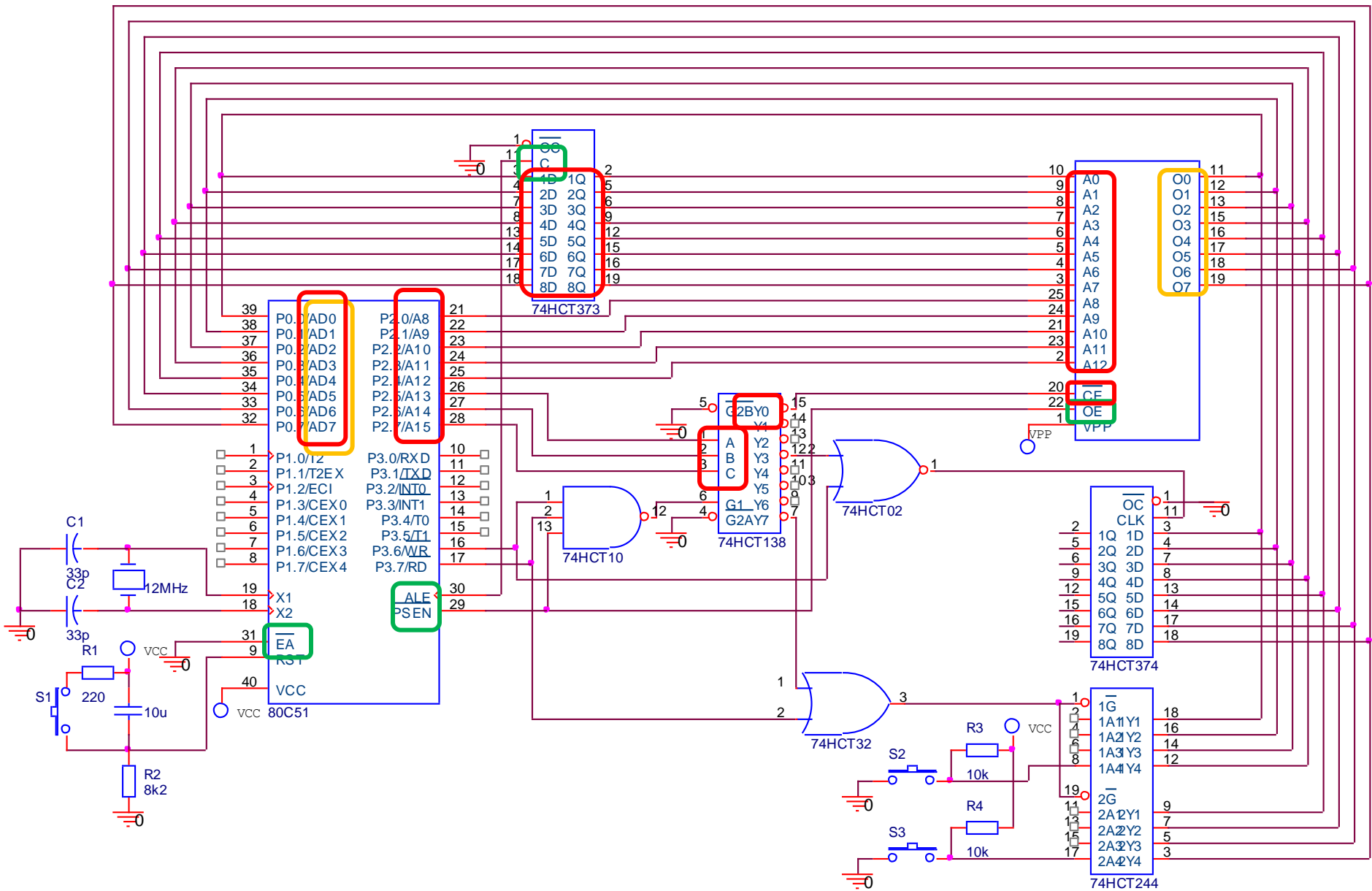


- 8k*8 memorie de program externa avand adresa de baza 0000H
- mag. adrese: P0 si P2
- mag. date: P0
- mag. control: ALE, PSEN\ si EA\

0000H: **0000**.....0b

A15 A14 A13 | A12 A11... A0
 Selectie bloc | 13 linii adresa
 memorie

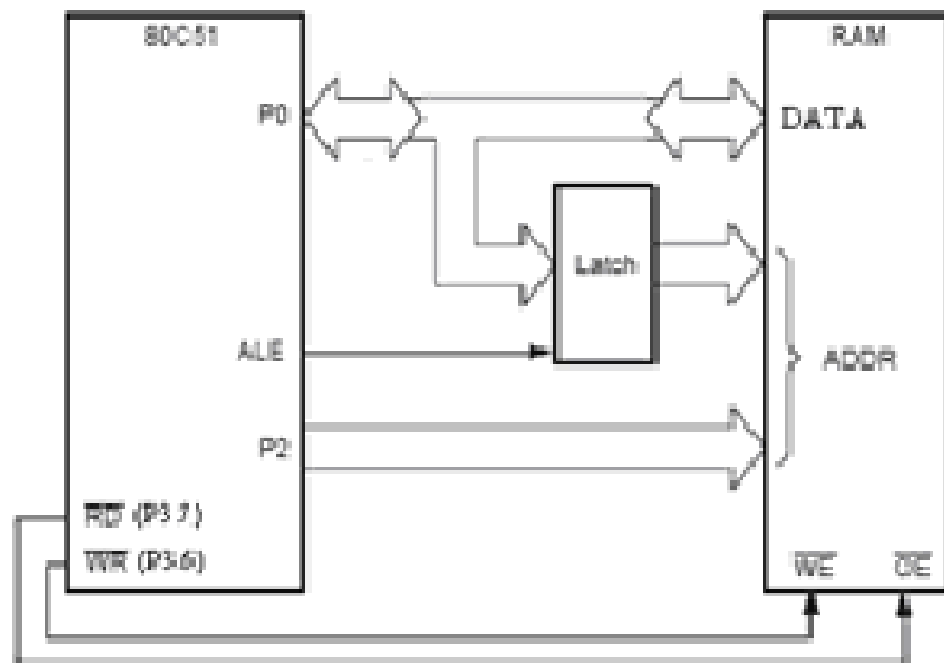
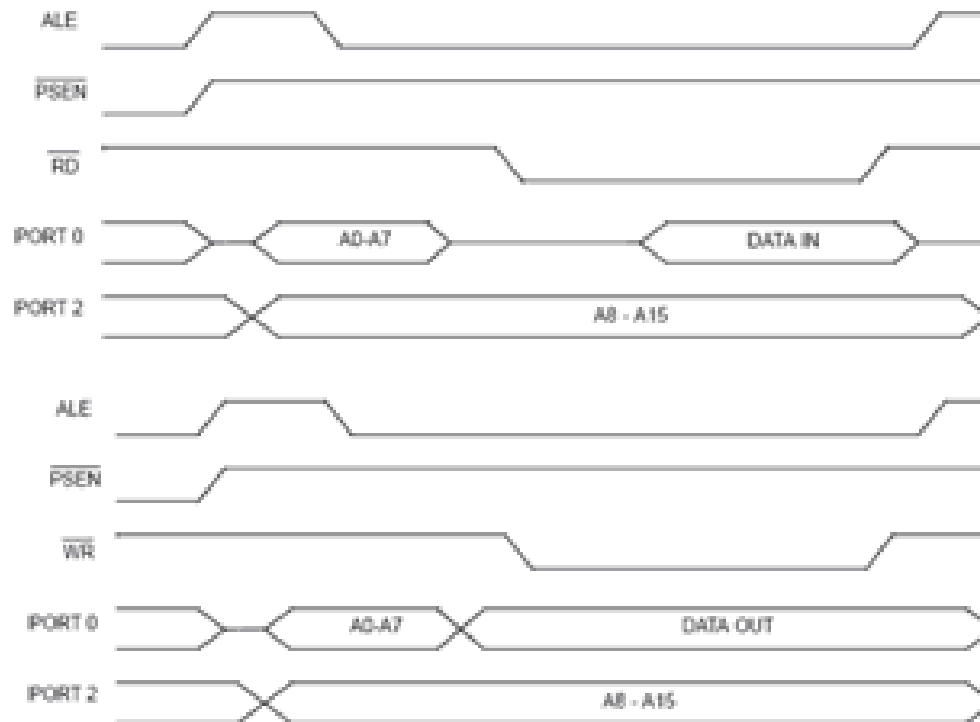
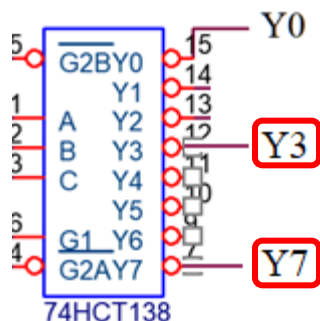




- port de iesire la adresa 011XXXXXXXXXXXXXXXXXb
- port de intrare la adresa 111XXXXXXXXXXXXXXXXXb la care sunt conectate 2 taste
- mag. adrese: P0 si P2
- mag. date: P0
- mag. control: ALE, RD\ si WR\

011XX.....Xb port de iesire
111XX.....Xb port de intrare

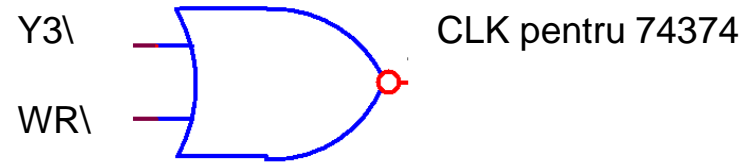
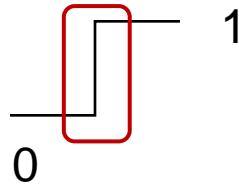
A15 A14 A13 | A12 A11... A0
Selectie porturi | selectie partiala



Port de iesire la adresa 011XXXXXXXXXXXXXb – registru 74374

Scrierea datelor in registrul 74374 se face pe frontul pozitiv al semnalului CLK atunci cand $Y3\backslash = 0$ si $WR\backslash = 0$

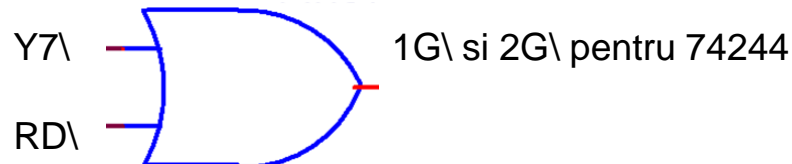
Y3\	WR\	CLK
0	0	1
0	1	0
1	0	0
1	1	0

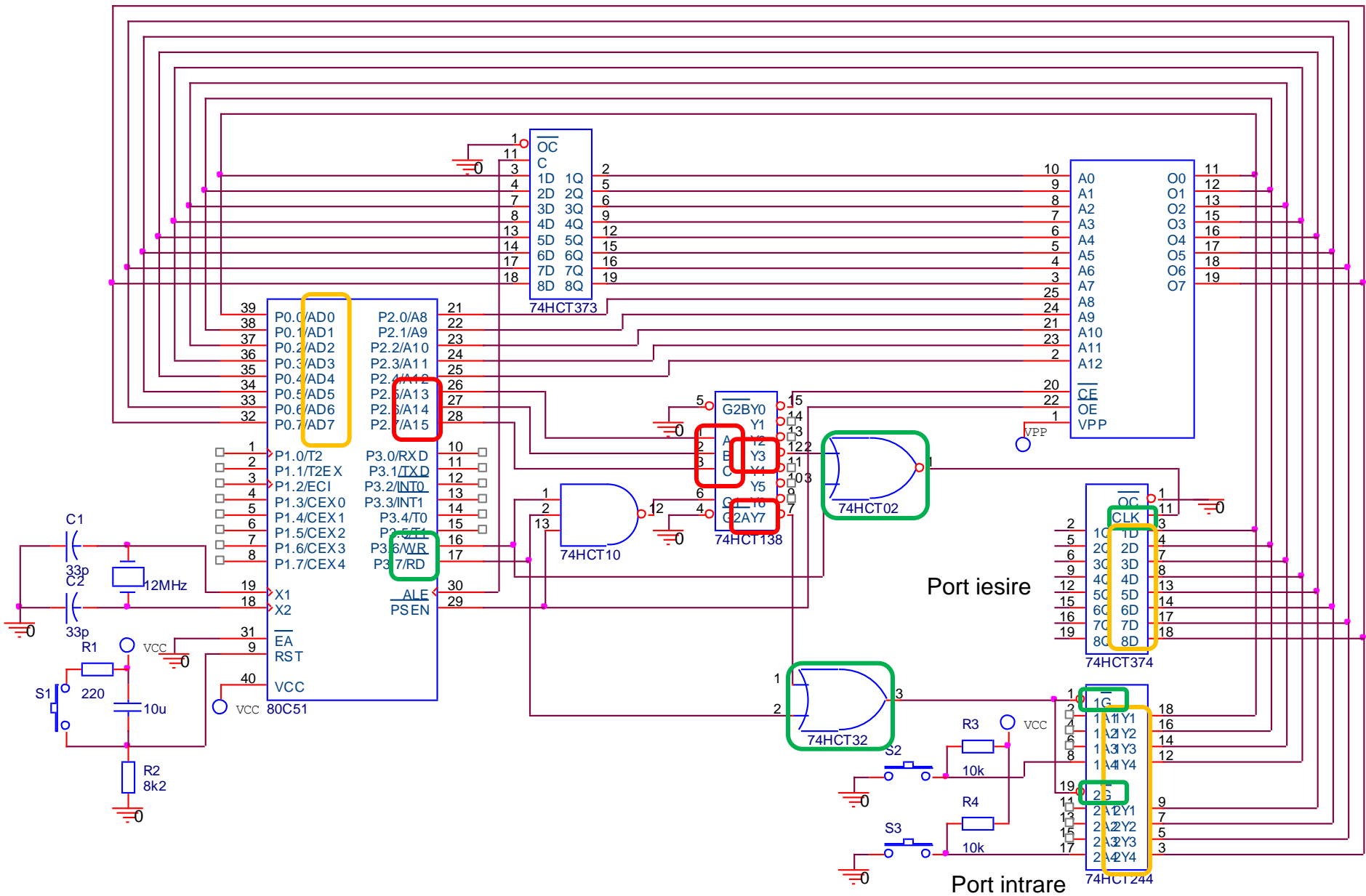


Port de intrare la adresa 111XXXXXXXXXXXXXb la care sunt conectate 2 taste - amplificator unidirectional 74244

Citirea starii tastelor, realizata prin citirea datelor de la amplificatorul 74244, se face activand semnalele 1G\ si 2G\, atunci cand $Y7\backslash = 0$ si $RD\backslash = 0$

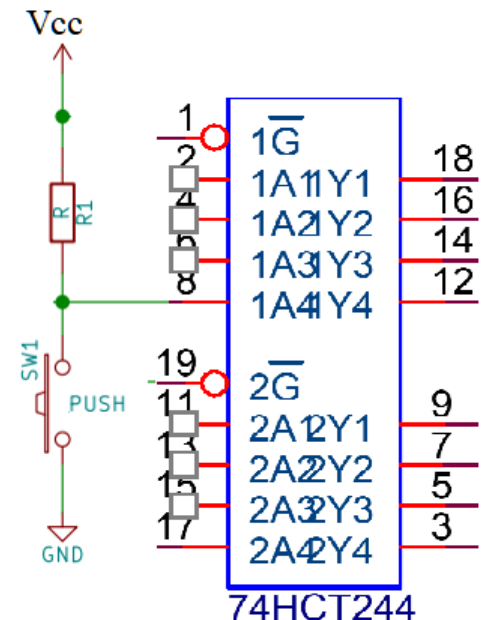
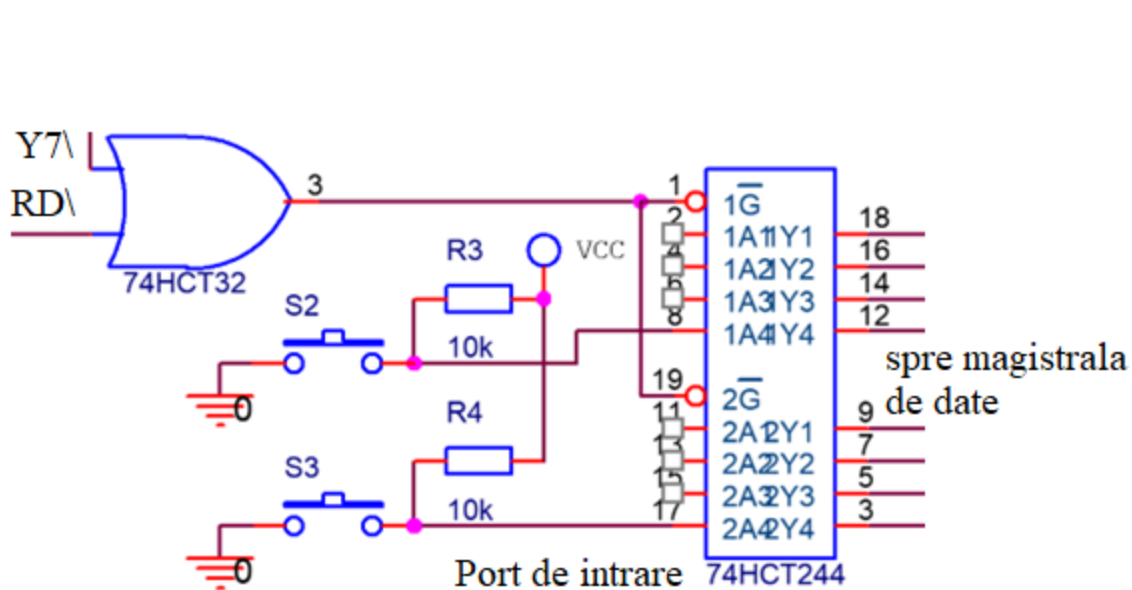
Y7\	RD\	out
0	0	0
0	1	1
1	0	1
1	1	1

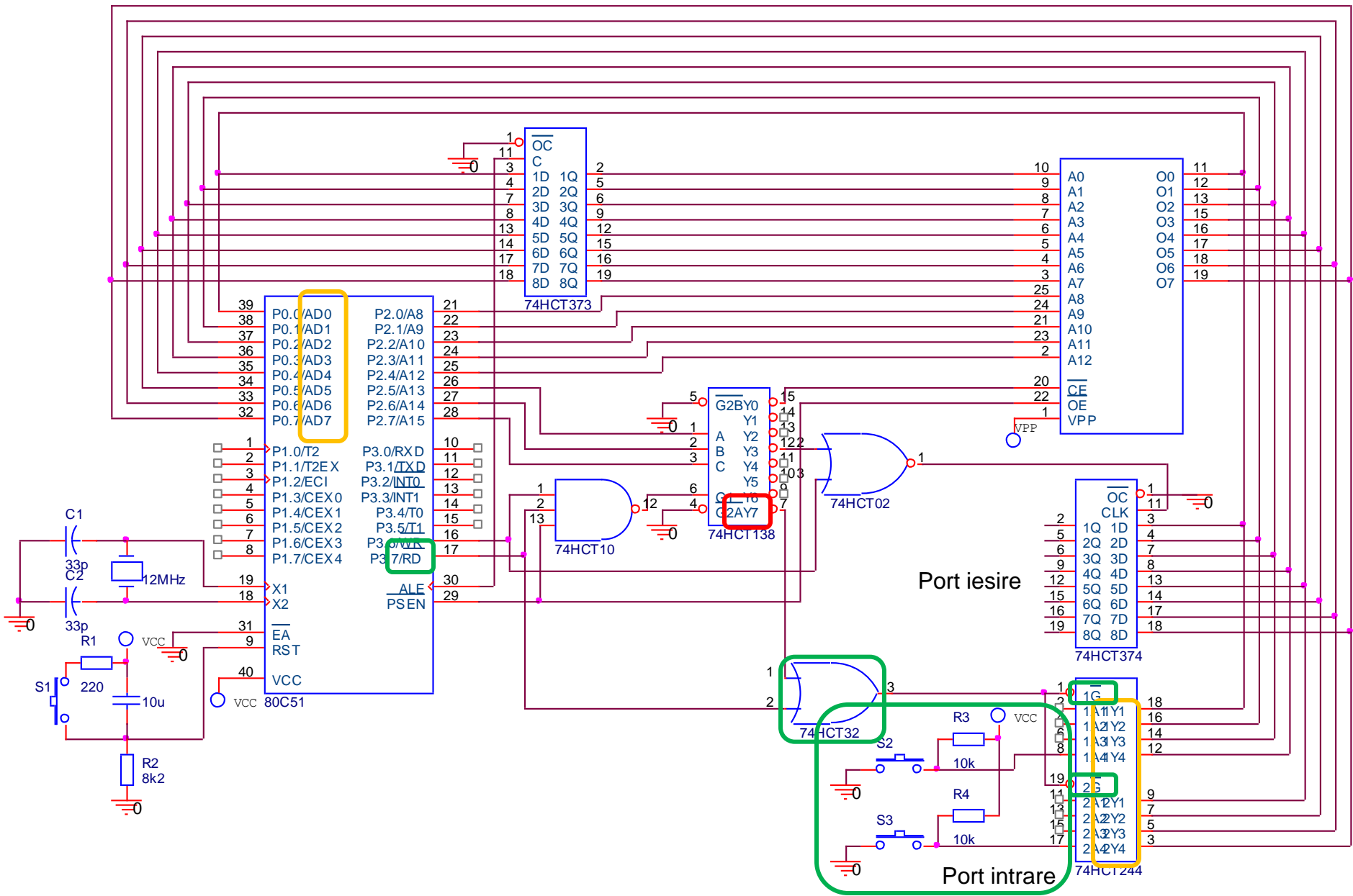




Taste – citirea datelor de la taste se face folosind portul de intrare prin amplificatorul 74244, atunci cand $Y7\backslash = 0$ si $RD\backslash = 0$

- daca tasta nu este apasata, se citeste “1” deoarece circuitul format din tasta si rezistenta este astfel dimensionat incat tensiunea la pinul de intrare al amplificatorului corespunde nivelului logic 1
- daca tasta este apasata, se citeste “0” deoarece circuitul format din tasta si rezistenta este astfel dimensionat incat tensiunea la pinul de intrare al amplificatorului corespunde nivelului logic 0





- Pentru sistemul proiectat in problema precedenta, sa se scrie un program in limbajul C care:
 - defineste variabila input_port in memoria externa de date, pentru citirea portului de intrare;
 - defineste variabila output_port in memoria externa de date, pentru scrierea portului de iesire;
 - defineste variabila temp;
 - genereaza o intrerupere Timer0 la fiecare 10ms in care citeste portul de intrare in variabila temp si scrie variabila temp in portul de iesire.
- Deoarece portul de intrare se afla la adresa 111XXXXXXXXXXXXb, putem defini variabila input_port in memoria externa de date la adresa 0xE000;
- Deoarece portul de iesire se afla la adresa 011XXXXXXXXXXXXb, putem defini variabila output_port in memoria externa de date la adresa 0x6000.

- Registrul TMOD
- GATE: Dacă GATE=1, TIMERx va rula doar atât timp cât TRx=1 și INTx=1. Dacă GATE=0, TIMERx va rula doar atât timp cât TRx=1.
- C/T\, Selector Timer sau Counter. Dacă C/T\=0, operare Timer. Dacă C/T\=1, operare Counter.
- M1: Bit de selectie mod.
- M0: Bit de selectie mod.

M1	M0	MOD DE OPERARE
0	0	Timer 13-biți
0	1	Timer/Counter 16-biți
1	0	Timer/Counter 8-biți cu Auto-Reload
1	1	(Timer 0) TL0 Timer/Counter 8-biți controlat prin biții de control standard corespunzători Timer 0. TH0 Timer 8 biți controlat prin biții de control corespunzători Timer 1.
1	1	(Timer 1) Timer/Counter 1 OFF.

GATE	C/T\	M1	M0	GATE	C/T\	M1	M0
TIMER 1				TIMER 0			

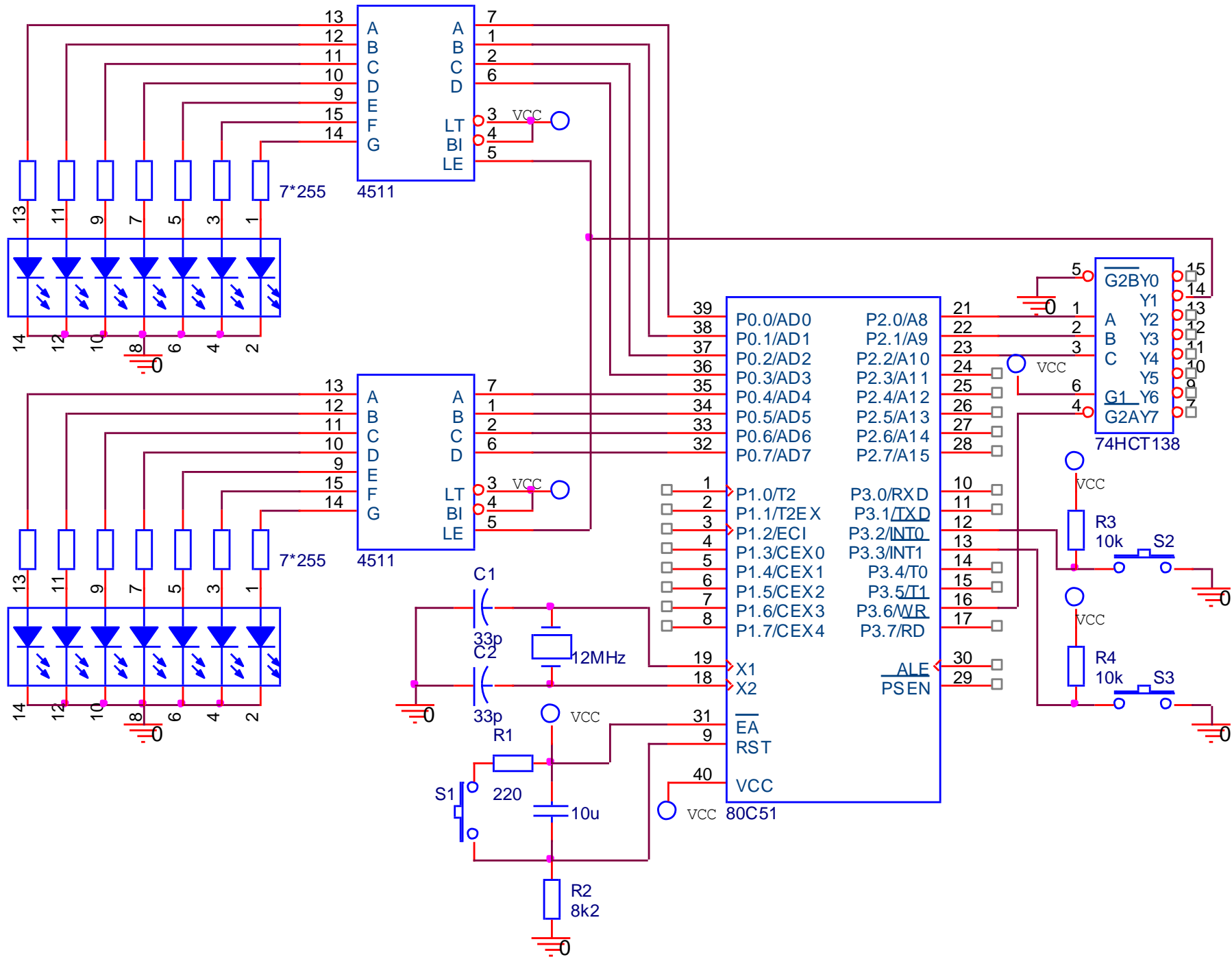
- Registrul IE
- EA, IE.7: dacă EA=0, nici o întrerupere nu va fi achitată. Dacă EA=1, fiecare sursă de întrerupere poate fi activată sau dezactivată individual.
- ET0, IE.1: întrerupere Timer 0 overflow

EA	-	ET2	ES	ET1	EX1	ET0	EX0
----	---	-----	----	-----	-----	-----	-----

- Registrul TCON
- TR0, TCON.4: Bit de control funcționare Timer 0. Dacă TR0=1, Timer 0 ON. Dacă TR0=0, Timer 0 OFF.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

- Sa se proiecteze un sistem cu microcontroller 80C51 avand urmatoarele blocuri:
 - oscilator cu quartz avand frecventa de 12MHz
 - circuit de reset
 - port de iesire la adresa XXXXX001XXXXXXXXXb la care sunt conectate doua display-uri cu LED-uri 7-segmente catod comun pentru care $V_{LED}=1,7V$ si $I_{LED}=10mA$
 - 2 taste conectate la intrarile de intrerupere
 - microcontrollerul foloseste doar memoria interna de program

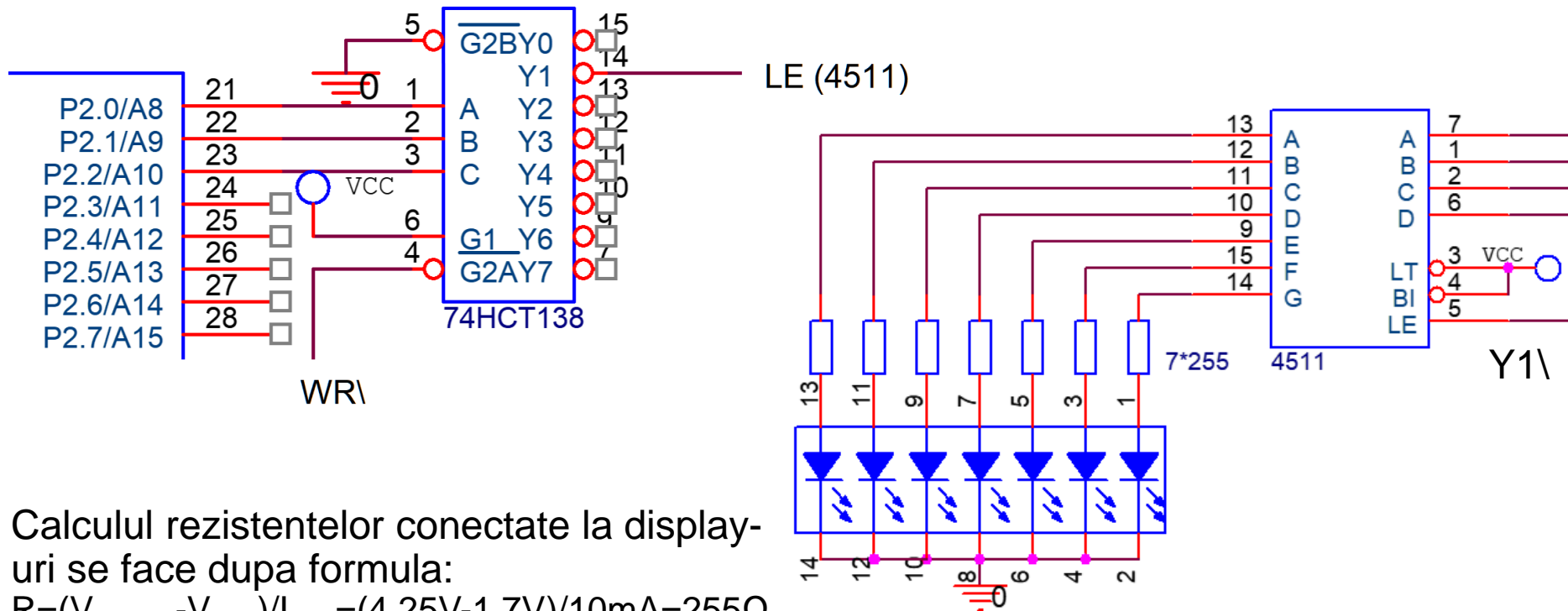


Port de iesire la adresa XXXXX001XXXXXXXXXb la care sunt conectate doua display-uri cu LED-uri 7-segmente catod comun pentru care $V_{LED}=1,7V$ si $I_{LED}=10mA$

P2								P0							
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
X	X	X	X	X	0	0	1	X	X	X	X	X	X	X	X



Y1 este activ daca $A10A9A8=001$ si $WR\backslash=0$



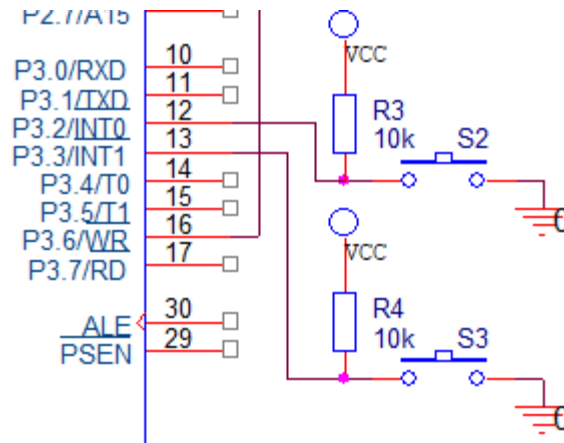
Calculul rezistentelor conectate la display-uri se face dupa formula:
 $R=(V_{OHTYP}-V_{LED})/I_{LED}=(4,25V-1,7V)/10mA=255\Omega$

2 taste conectate la intrarile de intrerupere

Tastele se vor conecta la pinii INT0\ (P3.2) și INT1\ (P3.3)

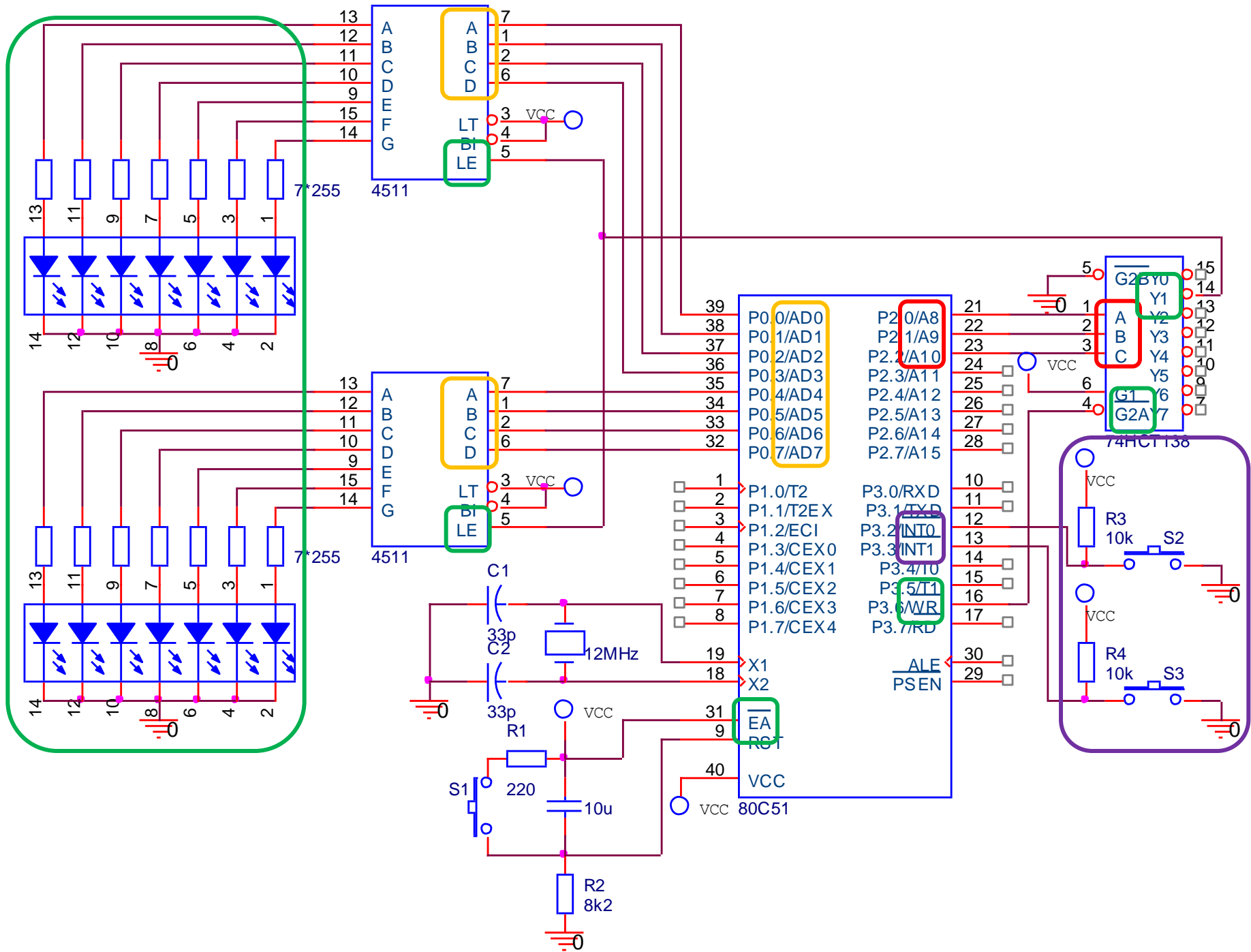
- daca tasta nu este apasata, se citeste "1" deoarece circuitul format din tasta si rezistenta este astfel dimensionat incat tensiunea la pinul INTX\ corespunde nivelului logic 1

- daca tasta este apasata, se citeste "0" deoarece circuitul format din tasta si rezistenta este astfel dimensionat incat tensiunea la pinul INTX\ corespunde nivelului logic 0



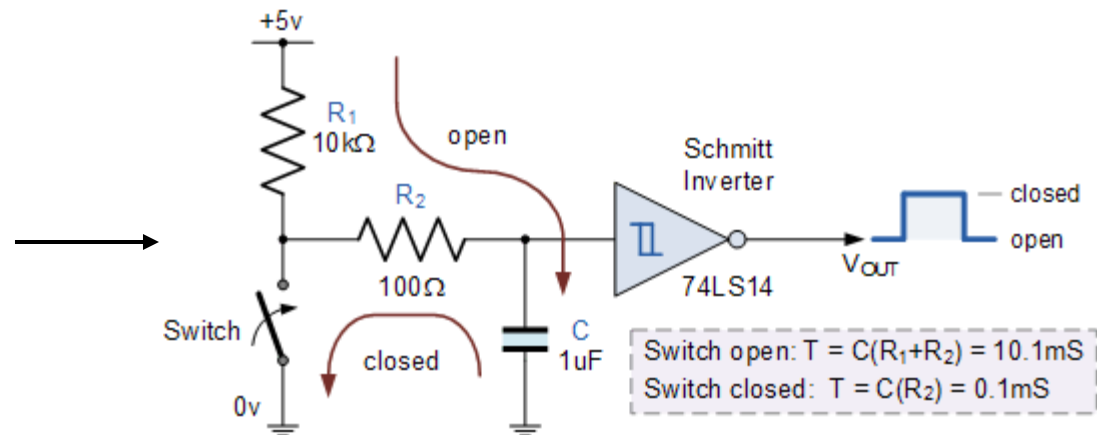
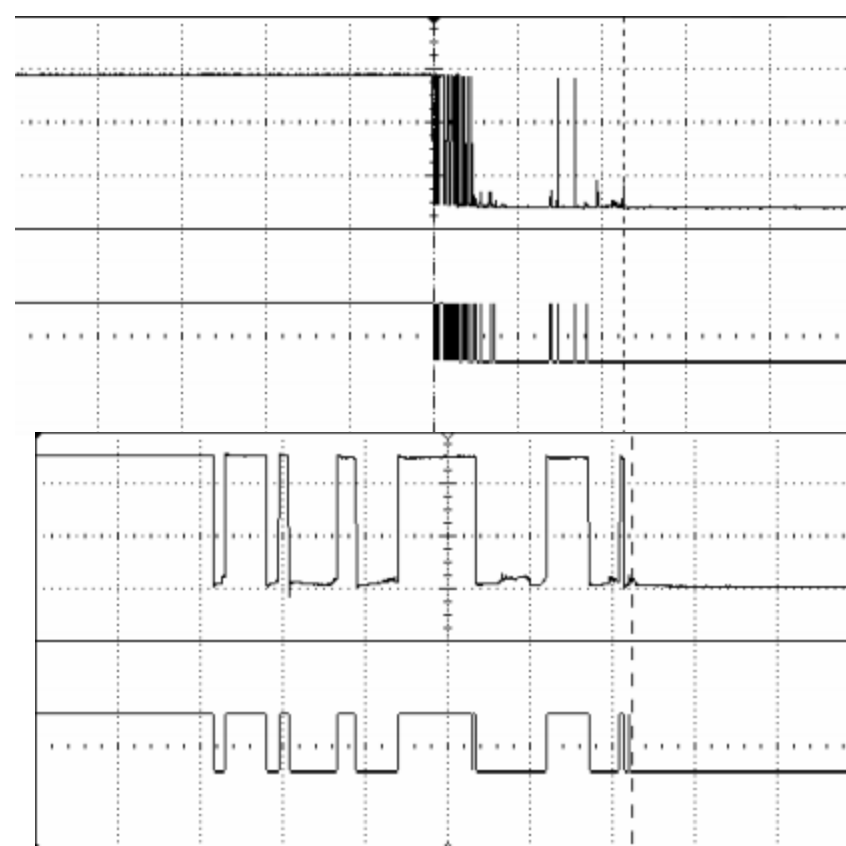
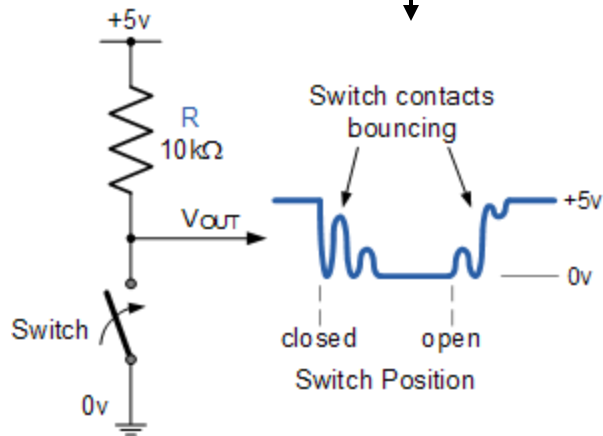
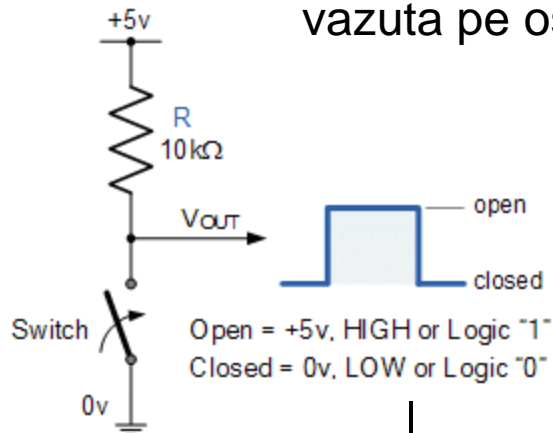
microcontrollerul foloseste doar memoria interna de program

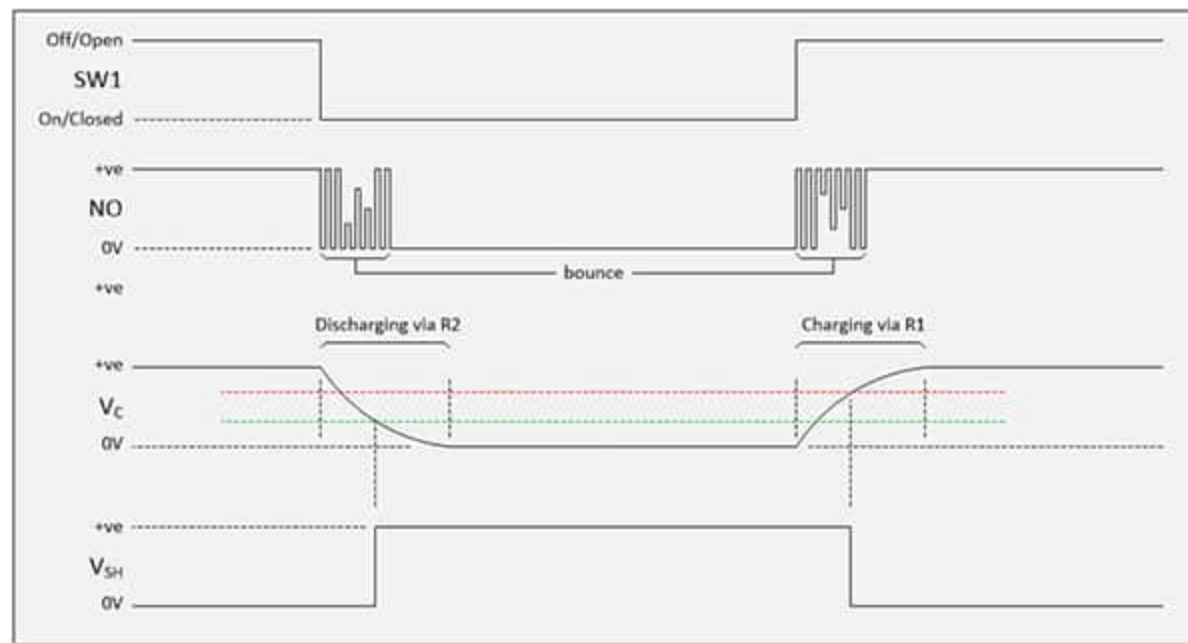
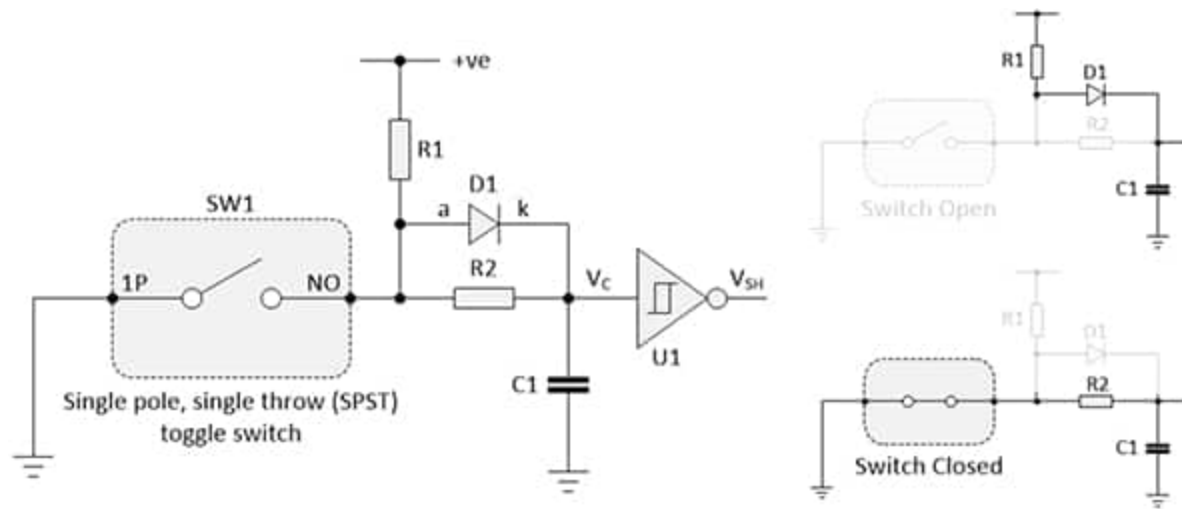




Circuit pentru filtrarea salturilor de tensiune la apasarea unei taste (comutator)

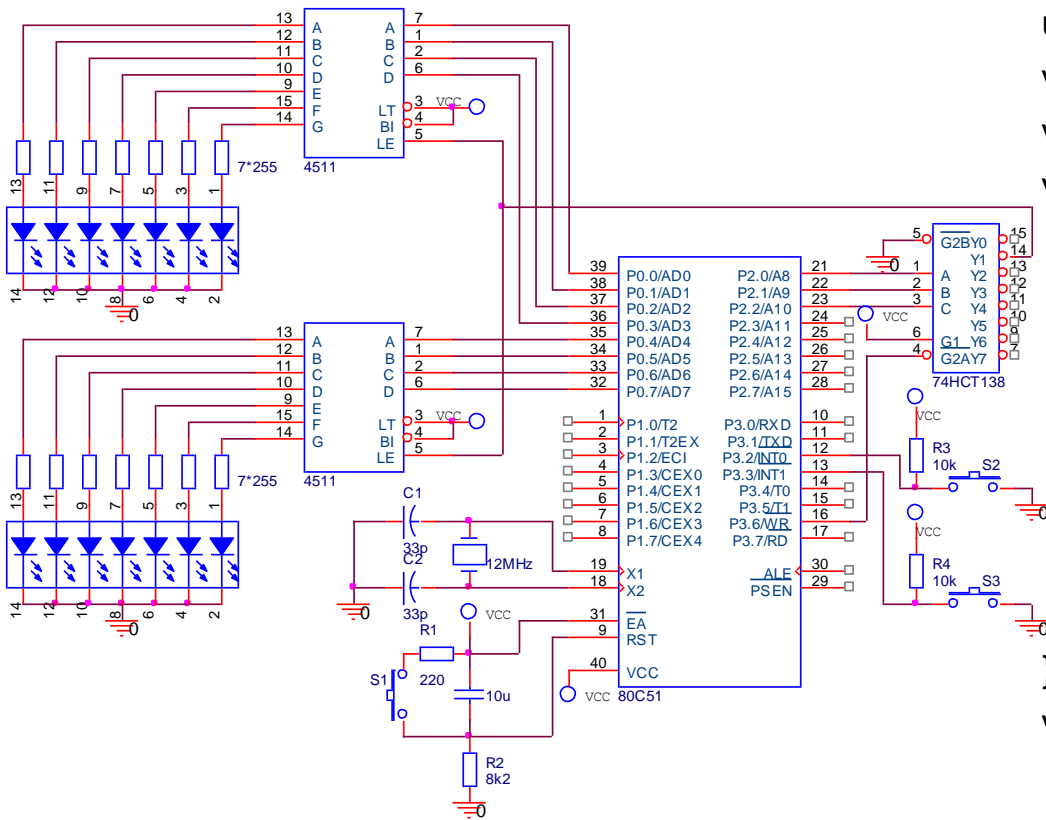
Apasarea unei taste
vazuta pe osciloscop





<https://www.digikey.ro/ro/articles/how-to-implement-hardware-debounce-for-switches-and-relays>

- Pentru sistemul proiectat in problema precedenta, sa se scrie un program in limbajul C care:
 - defineste variabila afisare in memoria externa de date, pentru afisarea informatiei la display-uri;
 - defineste variabila parametru;
 - la generarea intreruperii INT0\ incrementeaza variabila parametru si o afiseaza la display-uri;
 - la generarea intreruperii INT1\ decrementeaza variabila parametru si o afiseaza la display-uri;
- Deoarece portul de iesire la care sunt conecatare cele doua display-uri cu LED-uri 7-segmente se afla la adresa XXXXX001XXXXXXXXb, putem defini variabila afisare in memoria externa de date la adresa 0x0100.



Informatii despre registrii folositi se gasesc pe slide-ul urmator

```
#include <reg51.h>
```

```
xdata unsigned char afisare _at_ 0x0100;
```

```
unsigned char parametru = 0x20;
```

```
void external_int0_ISR(void) interrupt 0;
```

```
void external_int1_ISR(void) interrupt 2;
```

```
void main(void) {
```

```
    afisare = parametru;
```

```
    IT0 = 1; // INT0\ declansata de un front cazator
```

```
    IT1 = 1; // INT1\ declansata de un front cazator
```

```
    EX0 = 1; // activare intrerupere INT0\
```

```
    EX1 = 1; // activare intrerupere INT1\
```

```
    EA = 1; // activare globala intreruperi
```

```
    while (1) {
```

```
    }
```

```
}
```

```
void external_int0_ISR(void) interrupt 0 {
```

```
    parametru++;
```

```
    afisare = parametru;
```

```
}
```

```
void external_int1_ISR(void) interrupt 2 {
```

```
    parametru--;
```

```
    afisare = parametru;
```

```
}
```

```
}
```

- Registrul TCON
- IT1, TCON.2: Bit de control tip întrerupere 1. Dacă IT1=1, întreruperea 1 este declanșată de un front căzător. Dacă IT1=0, întreruperea 1 este declanșată de nivelul logic 0.
- IT0, TCON.0: Bit de control tip întrerupere 0. Dacă IT0=1, întreruperea 0 este declanșată de un front căzător. Dacă IT0=0, întreruperea 0 este declanșată de nivelul logic 0.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

- Registrul IE
- EA, IE.7: dacă EA=0, nici o întrerupere nu va fi achitată. Dacă EA=1, fiecare sursă de întrerupere poate fi activată sau dezactivată individual.
- EX1, IE.2: întrerupere externă 1
- EX0, IE.0: întrerupere externă 0

EA	-	ET2	ES	ET1	EX1	ET0	EX0
----	---	-----	----	-----	-----	-----	-----

Probleme propuse

- Sa se proiecteze un sistem cu microcontroller 80C51 avand urmatoarele blocuri:
 - oscilator cu quartz avand frecventa de 12MHz
 - circuit de reset
 - 32k*8 memorie de date si program externa avand adresa de baza 8000H
 - 2 taste conectate la intrarile de intrerupere

- Sa se proiecteze un sistem cu microcontroller 80C51 avand urmatoarele blocuri:
 - oscilator cu quartz avand frecventa de 12MHz
 - circuit de reset
 - port de iesire la adresa XXX101XXXXXXXXXXb la care sunt conectate doua display-uri cu LED-uri 7-segmente anod comun pentru care $V_{LED}=1,6V$ si $I_{LED}=30mA$
 - pentru tranzistoare se considera $\beta=100$
 - microcontrollerul foloseste doar memoria interna de program

- Sa se proiecteze un sistem cu microcontroller 80C51 avand urmatoarele blocuri:
 - oscilator cu quartz avand frecventa de 12MHz
 - circuit de reset
 - 64k*8 memorie de program externa
 - 32k*8 memorie de date externa avand adresa de baza 8000H
 - tastatura matriciala cu 16 taste conectata la Port 1
 - microcontrollerul foloseste doar memoria externa de program