

Arhitectura calculatoarelor 2. Lucrarea 8,9.

Studiul și experimentarea interfeței paralele programabile Z80-PIO

1. Obiectivul lucrării

Lucrarea are ca obiectiv însușirea structurii și a modurilor de funcționare ale circuitului pentru comanda intrărilor/ieșirilor paralele, Z80-PIO. Se urmăresc atât aspecte hardware cât și software.

2. Structura circuitului și conexiunile externe

Schema bloc a circuitului este prezentată în Figura 1, iar conexiunile externe în figura 2.

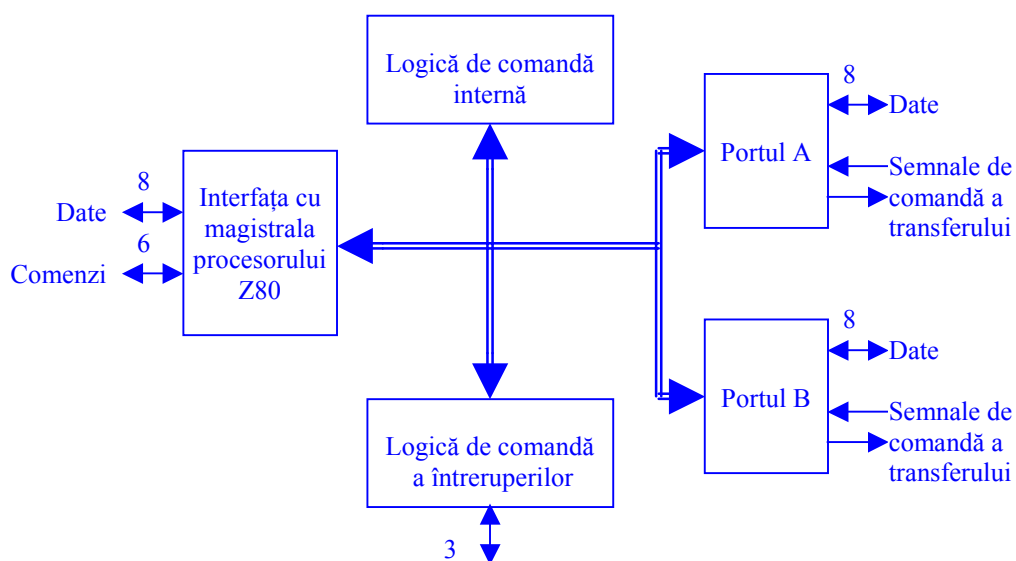


Fig. 1

Z80 – PIO dispune de două porturi independente de 8 biți, care pot opera ca porturi de intrare sau de ieșire. Fiecare port are asociate două semnale pentru comanda transferului: STROBE și READY. Ieșirea READY indică perifericului asociat faptul că portul este pregătit pentru un transfer de date. Prin intrarea STROBE, perifericul semnalează portului faptul că a apărut un transfer.

Interfața cu magistrala microprocesorului se face prin 8 linii de date D0–D7 și 6 linii de comandă: M1/, RD/, IORQ/, B/A/, C/D/, CE/. Semnalul B/A/ selectează portul (B sau A) iar C/D/ tipul informațiilor care se transferă (date sau comenzi). Aceste linii se leagă de obicei la liniile de adresă A1 și A0 ale microprocesorului. Astfel Z80-PIO ocupă patru octeți în spațiul de adrese de intrare/ieșire. Semnalul CE/ este o validare globală a circuitului.

Intern, se generează un semnal WR*/, activ atunci când IORQ/ și CE/ sînt active, iar RD/ este inactiv. Se remarcă lipsa unei linii de RESET datorită numărului limitat de pini ai capsulei. Semnalul M1/, prezent fără IORQ/ sau RD/ va produce inițializarea circuitului. În acest scop, la pinul M1/ se aduce semnalul (M1+RESET)/.

Z80-PIO poate lucra în întreruperi în modul vectorizat al procesorului Z80 și poate fi integrat într-un sistem de priorități serial. Linia INT/ (Interrupt Request) este o cerere de întrerupere adresată procesorului.

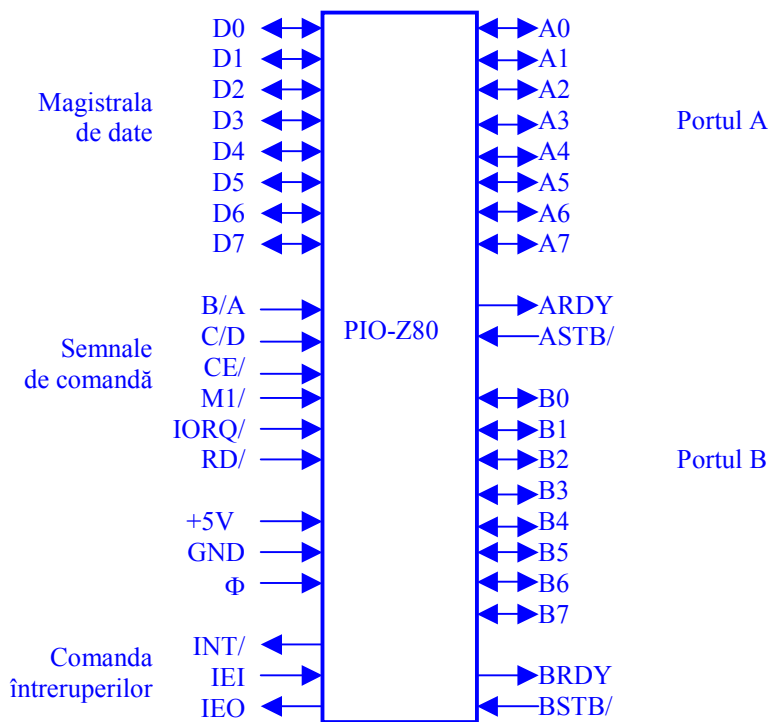


Fig.2

3. Modurile de lucru

3.1 Modul 0 (ieșire pe octet)

Inițierea unui ciclu de ieșire se face prin execuția de către procesor a unei instrucții OUT, la adresa portului. Aceasta va determina înscrierea datelor de pe magistrală în registrul de ieșire al portului adresat.

Comunicarea între port și periferic se face prin intermediul semnalelor de control al transferului. La sfârșitul execuției instrucțiunii OUT (semnalul WR*/ activ), PIO va activa ieșirea READY, informând perifericul că în port a fost înscris un nou octet. Ieșirea READY rămîne activă pînă la recepționarea unui front pozitiv pe linia STROBE/. Prin semnalul STROBE/ perifericul informează portul că a preluat octetul. Prin acest mecanism de sincronizare se asigură faptul că nu se pierde nici un octet și, deasemenea, că un octet nu este preluat de două ori la rînd.

Frontul semnalului STROBE/ poate genera și o cerere de întrerupere către procesor, dacă portul respectiv a fost programat și validat să lucreze cu întreruperi și dacă este prioritar. Diagrama de semnale este dată în Figura 3.

Semnalele de comandă a transferului nu sînt întotdeauna necesare (de exemplu în situația în

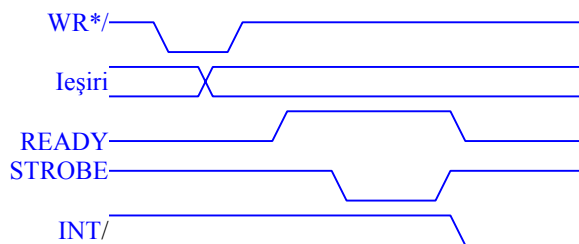


Fig. 3

care ieșirile comandă leduri, relee sau alte periferice neinteligente). În cazul în care perifericul nu emite semnalul STROBE/ această linie se va lega la '1' logic. În acest fel, portul primește o confirmare permanentă că poate modifica datele.

3.2. Modul 1 (intrare pe octet)

Un ciclu de intrare poate fi inițiat prin activarea de către periferic a liniei STROBE/. Cu frontul pozitiv al semnalului STROBE/ datele sînt memorate în port și se poate genera o cerere de intrerupere. După trecerea pe '1' a lui STROBE/ se va inactiva ieșirea READY, semnalînd perifericului că registrul de intrare al portului este plin. Microprocesorul preia datele din port printr-o instrucție IN (semnalul RD/ activ), ceea ce va reactiva semnalul READY. Diagrama de semnale este dată în Figura 4. Deoarece după inițializarea circuitului semnalul READY ramine pe '0', este necesară execuția unei instrucții IN înainte de primul transfer, pentru a determina activarea liniei READY.

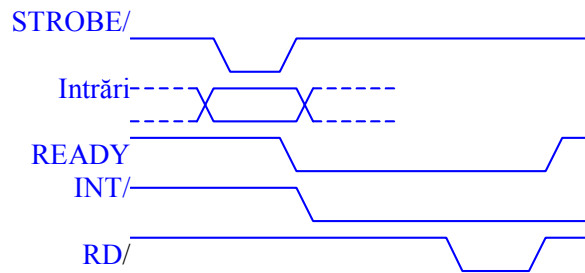


Fig. 4

În cazul în care linia STROBE/ nu este folosită, ea se va lega la '0'. În această situație, portul poate primi date permanent, dar microprocesorul nu mai primește nici o informație despre momentul în care datele se modifică.

3.2 Modul 2 (intrare/ieșire pe octet)

În acest mod poate fi programat doar portul A, care va lucra bidirecțional, utilizînd semnalele ASTB/ și ARDY pentru comanda ieșirii și semnalele BSTB/ și BRDY pentru comanda intrării, diagramele de timp fiind cele de la modulele 0 și 1.

În acest mod, portul funcționează ca port de ieșire doar pe durata cît ASTB/ este activ. În rest, portul funcționează ca intrare, memorarea datelor în registru făcîndu-se pe palierul activ al semnalului BSTB/. Este necesar ca perifericul să respecte acest protocol de transfer, în caz contrar fiind posibile conflicte pe pe liniile bidirecționale de date. În situația cînd portul A este programat în modul 2, portul B poate funcționa doar în modul 3.

3.3 Modul 3 (intrare/ieșire pe bit)

În acest mod semnalele de comandă a transferului nu sînt folosite. Fiecare bit al portului poate fi programat individual ca intrare sau ieșire.

Execuția unei instrucții OUT afectează doar biții programați ca ieșiri. Execuția unei instrucții IN citește, pentru biții programați ca intrari, starea liniei corespunzătoare în momentul execuției instrucției, iar pentru biții programați ca ieșiri, valoarea înscrisă prin ultima instrucție OUT.

Prin programare se specifică nivelul considerat activ al fiecărei intrări ('0' sau '1') și condiția logică ce va trebui să fie îndeplinită pentru a se genera o cerere de întreruperea (SAU: o singură intrare activă, ȘI: toate intrările active). De exemplu, dacă unul din porturi este programat să aibă intrări active pe 0 și condiția logică este ȘI, atunci PIO va genera o cerere de întrerupere cînd toate intrările acelu port trec pe '0'.

4. Programarea circuitului

Prin programare se specifică 3 grupe de parametri ce caracterizează funcționarea circuitului:

- vectorul de întrerupere
- modul de lucru
- cuvântul de comandă pentru întreruperi

În acest scop se trimite de către microprocesor către PIO mai multe cuvinte de comandă, notate CC, a căror semnificație este descrisă în cele ce urmează.

4.1. Vectorul de întrerupere

Vectorul de întrerupere este octetul plasat pe magistrala de date de către PIO în timpul unui ciclu de achitare a întreruperii.

| | | | | | | | | |
|------------|----|----|----|----|----|----|----|----|
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| CC0 | V7 | V6 | V5 | V4 | V3 | V2 | V1 | 0 |

4.2. Modul de lucru

| | | | | | | | | |
|------------|----|----|---|---|---|---|---|---|
| CC1 | M1 | M0 | X | X | 1 | 1 | 1 | 1 |
|------------|----|----|---|---|---|---|---|---|

Cuvântul **CC1** definește modul de lucru astfel:

| M1 | M0 | Modul |
|----|----|-----------------------------|
| 0 | 0 | 0 (ieșire pe octet) |
| 0 | 1 | 1 (intrare pe octet) |
| 1 | 0 | 2 (intrare/ieșire pe octet) |
| 1 | 1 | 3 (intrare/ieșire pe bit) |

Dacă s-a selectat modul 3, după **CC1** se va transmite **CC11**, care definește fiecare linie a portului ca intrare ($I/E_i=1$) sau ca ieșire ($I/E_i=0$).

| | | | | | | | | |
|-------------|------|------|------|------|------|------|------|------|
| CC11 | I/E7 | I/E6 | I/E5 | I/E4 | I/E3 | I/E2 | I/E1 | I/E0 |
|-------------|------|------|------|------|------|------|------|------|

4.3. Cuvântul de comandă a întreruperilor

| | | | | | | | | |
|------------|----------------------|--------|---------|-----------------|---|---|---|---|
| CC2 | Validare întreruperi | ȘI/SAU | '1'/'0' | Urmează măștile | 0 | 1 | 1 | 1 |
|------------|----------------------|--------|---------|-----------------|---|---|---|---|

D7 – validare întreruperi. Dacă $D7=1$ portul respectiv este validat să genereze o cerere de întrerupere în condițiile specifice fiecărui mod de funcționare.

Biții $D6-D4$ se referă la modul 3 de funcționare.

D6 – funcția logică. $D6=0$ - funcția ȘI, $D6=1$ – funcția SAU.

D5 – nivel activ. $D5=0$ – nivel '0' activ, $D5=1$ – nivel '1' activ.

D4 – urmează măștile. $D4=1$ – următorul cuvânt de programare este **CC21** pentru definirea măștilor. Numai liniile portului ai căror biți de mascare $MB_i = 0$ vor fi luați în considerare la generarea unei cereri de întrerupere.

| | | | | | | | | |
|------|-----|-----|-----|-----|-----|-----|-----|-----|
| CC21 | MB7 | MB6 | MB5 | MB4 | MB3 | MB2 | MB1 | MB0 |
|------|-----|-----|-----|-----|-----|-----|-----|-----|

Validarea/invalidarea întreruperilor pentru un anumit port se poate face și în timpul funcționării, fără a modifica acțiunea cuvântului CC2, prin transmiterea următoarei comenzi :

| | | | | | | | | |
|-----|-------------------------|---|---|---|---|---|---|---|
| CC3 | Validare întreruperi | X | X | X | 0 | 0 | 1 | 1 |
|-----|-------------------------|---|---|---|---|---|---|---|

5. Desfășurarea lucrării

Pe sistemul de dezvoltare existent în laborator și descris în prima lucrare există un circuit integrat de tip Z80-PIO. Adresele sale fizice sînt C8h, C9h pentru portul A, respectiv CAh, CBh pentru portul B. La liniile A0...A3 sînt legate LEDuri, așadar portul A se poate folosi ca ieșire. Portul B se poate folosi ca intrare, la linia B0 fiind legat un microswitch care aduce la această intrare fie 0V, fie +5V.

5.1 Se va programa portul B să funcționeze în modul 1. Într-o buclă infinită, se va citi portul și se va afișa conținutul său la display. În acest fel se poate afișa starea apăsăat/neapăsăat a microswitchului de la intrarea B0.

5.2. Se va programa portul A să funcționeze în modul 0. Se va modifica programul scris la lucrarea anterioară (ceas de 1 secundă) astfel încît valoarea ceasului, în loc de a fi afișată pe display, să fie trimisă la portul A, unde se poate urmări starea LEDurilor.

5.2 Se va programa portul B să funcționeze în modul 3, cu biții B0 și B1 ca intrări și B2...B7 ca ieșiri. Se va programa PIO să ceară o întrerupere atunci cînd B0 sau B1 au valoarea 1. În rutina de tratare a întreruperii se va afișa conținutul portului la display.