

9. UNITATEA DE I/E

Pe lângă unitatea centrală și un set de module de memorie, un alt element important al unui sistem de calcul este sistemul de I/E. O *unitate de I/E* (UIE) este componenta sistemului de calcul care efectuează transferul informațiilor între memoria principală a sistemului de calcul și mediul exterior al acestuia: suporturile externe de informație și alte sisteme de calcul cu care se interconectează.

9.1. Structura unității de I/E

În Figura 9.1 se prezintă structura unui sistem de calcul împreună cu unitatea de I/E.

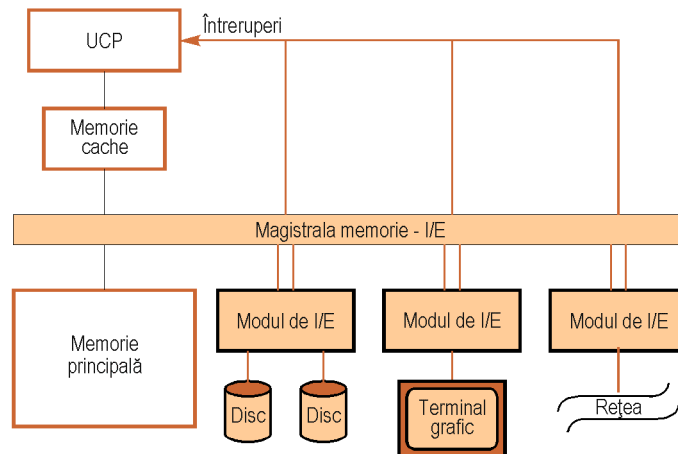


Figura 9.1. Structura unui sistem de calcul și a unei unități tipice de I/E.

Componentele principale ale unei unități de I/E sunt următoarele:

- Modulele de I/E;
- Echipamentele periferice;
- Interconexiunile dintre memorie și echipamentele periferice (magistralele);
- Interfețele dintre aceste componente.

Motivele pentru care perifericele nu se conectează direct la magistrala sistemului și se utilizează module de I/E sunt următoarele:

- Există o mare varietate de periferice cu diferite metode de funcționare. Nu ar fi practică încorporarea logicii necesare în cadrul UCP pentru controlul diferitelor echipamente.
- Rata de transfer a perifericelor este adesea mult mai mică decât cea a memoriei și a UCP. Deci, nu este eficientă utilizarea magistralei sistem cu viteză ridicată pentru comunicarea directă cu un periferic.
- Perifericele utilizează de multe ori formate diferite ale datelor și cuvinte de lungimi diferite față de calculatorul la care sunt conectate.

9.2. Tipuri de echipamente periferice

Există trei categorii de echipamente periferice, după utilizarea acestora:

1. Pentru transportul datelor între procesor și utilizator: echipamente de *achiziție și prezentare a datelor*.
2. Pentru transferul datelor între mai multe procesoare: echipamente de *rețea*.
3. Pentru memorarea informațiilor, ca parte a ierarhiei de memorii a procesorului: echipamente de *memorare*.

Anumite periferice au funcții multiple. În Tabelul 9.1 și Tabelul 9.2 se prezintă cerințele privind ratele de transfer pentru unele echipamente tradiționale de prezentare a datelor, respectiv de memorare.

Tabelul 9.1. Ratele de transfer ale unor echipamente tradiționale de prezentare a datelor.

Echipament	Rata de transfer
Senzori	1 B/s - 1 KB/s
Tastatură	10 B/s
Linie de comunicație	56 Kbit/s (linie analogică)
Display	2 KB/s
Imprimantă	1 - 5 KB/s

Tabelul 9.2. Parametrii unor echipamente tradiționale de memorare.

Echipament	Timp de acces	Rata de transfer	Capacitate
Disc	8 ms	80 MB/s	50 GB
Bandă	0 (sec)	3 - 6 MB/s	25 GB (pe casetă)

În timp ce echipamentele tradiționale de prezentare a datelor sunt relativ lente comparativ cu cele de memorare, situația este diferită în cazul celor utilizate pentru multimedia. Documentele multimedia constau din:

- Grafică;
- Video;
- Audio (sinteza și recunoașterea vocii).

În Tabelul 9.3 se prezintă cerințele principale pentru aceste medii.

Tabelul 9.3. Parametrii unor echipamente multimedia.

	Rata de transfer	Întârziere maximă
Grafică	1 MB/s	1 - 5 s
Video	400 MB/s	20 - 33 ms
Audio	8 KB/s	50 - 300 ms

Fiecare mediu necesită o putere considerabilă de calcul. Pentru prelucrarea *imaginilor grafice*, au fost create procesoare specializate (transformarea spațiului culorilor, modelare 3D, animație). În cazul *imaginilor video*, cerințele cresc substanțial, deoarece o nouă imagine este creată la fiecare 1/30 s (33 ms). Se pot utiliza tehnici de compresie și interpolare a datelor în scopul reducerii ratei de transfer necesare pentru procesarea video în timp real. Prelucrarea *semnalelor audio* cu o rată de 2-8 KB/s necesită o putere considerabilă de calcul, pentru crearea sau recunoașterea diferitelor foneme în timp real.

9.3. Module de I/E

9.3.1. Funcțiile modulelor de I/E

Un modul de I/E are rolul de a efectua controlul echipamentelor externe și transferul datelor între aceste echipamente și memoria principală și/sau registrele UCP. Deci, un modul de I/E trebuie să dispună de o interfață internă în cadrul calculatorului (cu UCP și memoria principală) și o interfață externă (cu echipamentul extern).

Funcțiile și cerințele principale ale unui modul de I/E se încadrează în următoarele categorii:

- Control și sincronizare;
- Comunicația cu UCP;
- Comunicația cu echipamentele externe;
- Bufferarea datelor;
- Detecția erorilor.

În orice moment de timp, UCP poate comunica cu unul sau mai multe echipamente externe. Resursele interne, ca memoria internă și magistrala sistem, trebuie par-

tajate între mai multe activități, inclusiv operațiile de I/E ale datelor. Funcția de I/E necesită deci o operație de *control și sincronizare*, pentru coordonarea fluxului de date între resursele interne și echipamentele externe. De exemplu, controlul transferului de date de la un echipament extern la UCP poate cuprinde următoarele etape:

1. UCP interoghează modulul de I/E pentru a testa starea echipamentului conectat.
2. Modulul de I/E returnează starea echipamentului.
3. Dacă echipamentul este operațional și este pregătit pentru transmisia datelor, UCP solicită transferul datelor, printr-o comandă adresată modulului de I/E.
4. Modulul de I/E preia un octet sau cuvânt de date de la echipamentul de I/E.
5. Datele sunt transferate de la modulul de I/E la UCP.

Dacă sistemul utilizează o magistrală, fiecare interacțiune dintre UCP și modulul de I/E implică una sau mai multe operații de arbitraj a magistralei.

Din exemplul precedent rezultă că modulul de I/E trebuie să permită comunicația cu UCP și cu echipamentul extern. *Comunicația cu UCP* cuprinde:

- *Decodificarea comenzilor*: Modulul de I/E acceptă comenzi de la UCP. Aceste comenzi sunt transmise de obicei ca semnale pe magistrala de control. De exemplu, un modul de I/E pentru o unitate de disc poate accepta următoarele comenzi: CITIRE SECTOR, SCRIERE SECTOR, CĂUTARE PISTĂ. Unele comenzi au un parametru care este transmis pe magistrala de date.
- *Transmiterea datelor*: Datele sunt transmise între UCP și modulul de I/E pe magistrala de date.
- *Raportarea stării*: Deoarece perifericele sunt mai lente, este importantă cunoașterea stării modulului de I/E. De exemplu, un modul poate fi ocupat cu execuția comenzii precedente de I/E, ceea ce poate fi raportat printr-un semnal de stare. Semnalele obișnuite de stare sunt *BUSY* și *READY*. Există de asemenea semnale pentru raportarea diferitelor condiții de eroare.
- *Recunoașterea adreselor*: Un modul de I/E trebuie să recunoască o adresă unică pentru fiecare periferic pe care îl controlează.

Comunicația cu echipamentele externe se realizează prin semnale de control, de stare și de date.

Bufferarea datelor este o funcție esențială a unui modul de I/E. În timp ce rata de transfer la sau de la memorie sau UCP este ridicată, pentru cele mai multe periferice această rată este cu câteva ordine de mărime mai mică. Datele transferate de la memoria principală sunt bufferate de modulul de I/E și apoi sunt transmise la periferic cu rata acestuia. Modulul de I/E trebuie deci să opereze atât la viteza UCP, cât și la cea a memoriei.

În final, un modul de I/E trebuie să efectueze *detecția erorilor* și raportarea acestora către UCP. O clasă a erorilor cuprinde disfuncționalități mecanice și electrice raportate de echipament. O altă clasă constă în modificarea accidentală a datelor

transmise de periferic la modulul de I/E. Se utilizează anumite coduri detectoare de erori pentru detectarea erorilor de transmisie, de exemplu bitul de paritate.

9.3.2. Structura unui modul de I/E

Modulele de I/E diferă considerabil ca și complexitate și număr de echipamente externe pe care le controlează. O schemă bloc generală a unui modul de I/E este prezentată în Figura 9.2.

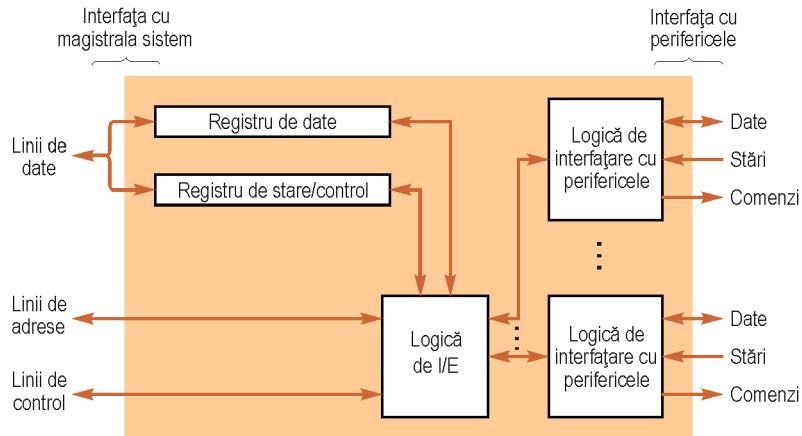


Figura 9.2. Structura unui modul de I/E.

Modulul se conectează cu restul sistemului de calcul printr-un set de linii (de exemplu, magistrala sistem). Datele transferate la și de la modul sunt bufferate într-unul sau mai multe *registre de date*. Pot exista de asemenea unul sau mai multe *registre de stare*. Un registru de stare poate funcționa și ca *registru de control*, pentru a primi informații de control detaliate de la UCP. Logica din cadrul modulului interacționează cu UCP printr-un set de *linii de control*. Acestea sunt utilizate de UCP pentru a transmite comenzi la modulul de I/E. Anumite linii de control pot fi utilizate de modulul de I/E, de exemplu ca semnale de arbitraj sau de stare.

Modulul trebuie de asemenea să recunoască și să genereze adresele asociate cu echipamentele pe care le controlează. Fiecare modul de I/E are o adresă unică, sau dacă acesta controlează mai multe echipamente externe, un set unic de adrese. În sfârșit, fiecare modul de I/E conține o logică specifică interfeței cu fiecare echipament pe care îl controlează.

Modulele de I/E pot avea complexități diferite. Unele module pot ascunde detalii ca sincronizarea datelor, formatul acestora și detaliile electromecanice ale unui echipament extern, astfel încât UCP poate executa numai comenzi simple de citire și scriere, eventual de deschidere și închidere pentru fișiere. În cazul altor module, cele mai multe operații de control ale perifericelor sunt vizibile de către UCP.

Un modul de I/E care are o interfață de nivel înalt cu UCP, eliberând UCP de sarcina controlului perifericelor, este numit *canal de I/E* sau *procesor de I/E*. Un modul de I/E care este simplu și necesită un control detaliat al UCP este numit de obicei *controler de I/E* sau *controler de dispozitiv*. Controlerele de I/E se utilizează de obicei la microcalculatoare, în timp ce canalele de I/E sunt utilizate la calculatoarele mari (*mainframe*).

Modulele de I/E complexe (canalele de I/E) sunt partiționate sub forma unor canale de I/E care controlează unul sau mai multe controlere de I/E.

9.4. Interfața externă

Interfața dintre un modul de I/E și un periferic trebuie să fie adaptată tipului de periferic și modului de funcționare al acestuia. Există două tipuri principale de interfețe. *Interfețele paralele* dispun de linii multiple care conectează modulul de I/E cu perifericul, fiind transferați simultan mai mulți biți (un octet sau cuvânt). În cazul *interfețelor seriale*, există o singură linie utilizată pentru transmiterea datelor, biții fiind transmiși succesiv. Interfețele paralele se utilizează în mod obișnuit pentru periferice mai rapide, ca discurile magnetice. Interfețele seriale sunt utilizate mai ales pentru terminale și unele tipuri de imprimante.

În ambele cazuri, modulul de I/E trebuie să execute un dialog cu perifericul. De exemplu, dialogul pentru o operație de scriere are următoarea formă generală:

1. Modulul de I/E activează un semnal de control, cerând permisiunea de a transmite datele.
2. Perifericul confirmă cererea.
3. Modulul de I/E transferă datele (un octet, un cuvânt sau un bloc).
4. Perifericul confirmă recepția datelor.

O operație de citire are loc în mod similar.

9.5. Metode de transfer a datelor

Există următoarele metode principale de realizare a transferurilor de date:

- Transferul programat;
- Transferul prin întreruperi;
- Transferul prin acces direct la memorie (DMA – *Direct Memory Access*);
- Transferul prin canale de I/E.

9.5.1. Transferul programat

9.5.1.1. Principiul transferului programat

În cazul transferului programat, datele sunt transferate între UCP și modulul de I/E. UCP execută un program prin care controlează direct operația de I/E: detectarea stării perifericului, transmiterea unei comenzi de citire sau scriere, și transferul datelor. După transmiterea unei comenzi modulului de I/E, UCP trebuie să aștepte până la terminarea operației de I/E. Testarea stării perifericului se realizează de obicei prin citirea stării, executând o buclă de instrucțiuni.

La execuția unei instrucțiuni de I/E, UCP transmite o comandă modulului de I/E corespunzător. Modulul de I/E execută acțiunea respectivă și setează biții corespunzători în registrul de stare. UCP trebuie să testeze periodic starea modulului de I/E pentru a detecta terminarea operației.

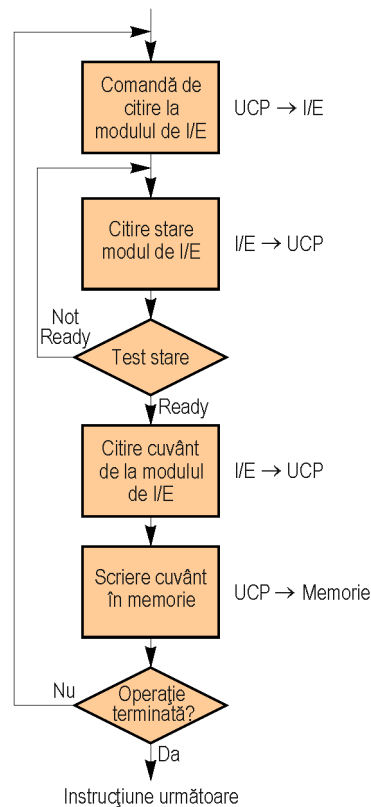


Figura 9.3. Utilizarea transferului programat pentru citirea unui bloc de date.

Figura 9.3 prezintă un exemplu de utilizare a transferului programat pentru citirea unui bloc de date de la un periferic (de exemplu, o înregistrare de pe o bandă) în memorie. Pentru fiecare cuvânt citit, UCP trebuie să rămână într-un ciclu de testare a stării pentru a determina dacă respectivul cuvânt este disponibil în registrul de date al modului de I/E.

9.5.1.2. Comenzi de I/E

Pentru execuția unei instrucțiuni de I/E, UCP transmite o adresă, specificând modulul de I/E și dispozitivul extern, și o comandă de I/E. Există patru tipuri de comenzi de I/E pe care le poate recepționa un modul de I/E atunci când este adresat de UCP. Acestea sunt clasificate ca fiind de *control*, de *test*, de *citire* și de *scriere*.

O comandă de *control* este utilizată pentru activarea unui periferic și pentru specificarea operației de executat. De exemplu, pentru o unitate de bandă magnetică, se poate specifica o operație de rebobinare sau poziționare pe următoarea înregistrare. Aceste comenzi sunt specifice diferitelor tipuri de periferice.

O comandă de *test* este utilizată pentru testarea diferitelor condiții de stare asociate cu un modul de I/E și perifericele acestuia. UCP trebuie să determine dacă:

- Perifericul este operațional și este disponibil;
- Ultima operație lansată este terminată;
- Au apărut sau nu erori la execuția operației.

O comandă de *citire* determină obținerea unui octet sau cuvânt de date de la periferic și depunerea acestuia într-un buffer intern (registrul de date al modului de I/E). UCP poate obține datele prin plasarea lor pe magistrala de date de către modulul de I/E.

O comandă de *scriere* determină preluarea de către modulul de I/E a unui octet sau cuvânt de pe magistrala de date și transmiterea acestuia perifericului.

9.5.1.3. Instrucțiuni de I/E

În general, există mai multe dispozitive conectate la sistem prin module de I/E. Fiecărui dispozitiv *i* se atribuie o adresă unică. Comanda de I/E transmisă de UCP conține adresa dispozitivului. Deci fiecare modul de I/E trebuie să interpreteze liniile de adrese pentru a determina dacă respectiva comandă îi este destinată.

Atunci când UCP, memoria principală și sistemul de I/E partajează aceeași magistrală, sunt posibile două moduri de adresare: adresare cu *mapare în memorie* și *adresare izolată*.

În cazul adresării cu *maparea în memorie*, există un singur spațiu de adrese pentru locațiile de memorie și dispozitivele de I/E. UCP tratează registrele de stare și de date ale modulelor de I/E ca locații de memorie și utilizează aceleași instrucțiuni atât pentru accesul la memorie cât și la dispozitivele de I/E. De exemplu, cu 10 linii de adresă pot exista în total 1024 locații de memorie și adrese de I/E, în orice combinație. În acest caz este necesar un singur semnal de citire și unul de scriere în memorie.

În cazul adresării *izolate*, spațiul adreselor de I/E este izolat de cel al memoriei. Magistrala trebuie să aibă linii de citire și scriere pentru memorie și linii de comandă pentru intrare și ieșire. Liniile de comandă specifică dacă o adresă se referă la o locație de memorie sau un dispozitiv de I/E. Pentru ambele este disponibilă toată gama adreselor. Pentru 10 linii de adresă, pot exista 1024 locații de memorie și 1024 adrese de I/E.

Se exemplifică cele două tehnici de I/E programate. Figura 9.4 prezintă interfața pentru tastatura unui terminal, din punctul de vedere al programatorului.

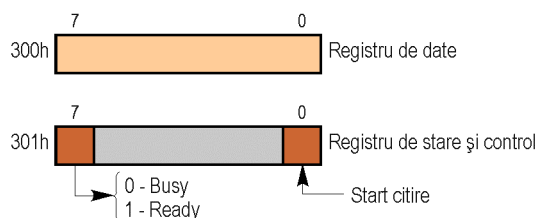


Figura 9.4. Interfața pentru tastatura unui terminal.

Pentru I/E mapate în memorie, se presupune că există 10 biți de adresă, spațiul de adrese fiind format din 512 locații de memorie (locațiile 0-1FFh) și până la 512 adrese de I/E (locațiile 200h-3FFh). Interfața cu tastatura utilizează două adrese. Adresa 300h este utilizată pentru registrul de date, iar adresa 301h pentru registrul de stare, care este utilizat și ca registru de control pentru recepționarea comenzilor UCP. Programul prezentat va citi un octet de la tastatură într-un registru acumulator al UCP. De notat că UCP execută o buclă de program până când octetul este disponibil.

Adresă	Instrucțiune	Operand	Comentariu
100h	Load AC	1	
101h	Store AC 301h		; Start citire tastatură
102h	Load AC	301h	; Citire octet de stare
103h	Jump if Sign = 0	102h	; Buclare
104h	Load AC	300h	; Citire octet de date

Pentru I/E izolate, presupunem că porturile de I/E sunt accesibile numai prin comenzi de I/E speciale, care activează liniile de comandă de I/E de pe magistrală.

Adresă	Instrucțiune	Operand	Comentariu
100h	Start I/E	5	; Start citire tastatură
101h	Test I/E	5	; Test sfârșit operație
102h	Jump Not Ready	101h	; Buclare
103h	In	5	; Citire octet de date

Pentru cele mai multe UCP, există un număr relativ mare de instrucțiuni diferite pentru referirea la memorie. Dacă se utilizează I/E izolate, există un număr mic de instrucțiuni de I/E. Deci, un avantaj al I/E mapate în memorie este că se poate utiliza un set extins de instrucțiuni, permițând o programare mai eficientă. Un dezavantaj este că se utilizează o parte din spațiul de adrese al memoriei.

Dezavantajul transferului programat, în varianta realizată prin citirea stării, este că UCP trebuie să aștepte până când perifericul devine disponibil, sau până la terminarea operației. Performanțele sistemului vor fi mult reduse. Pentru eliminarea acestui dezavantaj se poate utiliza transferul programat prin întreruperi.

9.5.2. Transferul prin întreruperi

Întreruperea reprezintă suspendarea execuției unui program de către un semnal extern UCP, indicând producerea unui eveniment extern, sau de către un eveniment intern, indicând apariția unei situații de excepție în funcționarea UCP. Întreruperea se produce la terminarea execuției instrucțiunii curente, deoarece execuția unei instrucțiuni nu este întreruptibilă.

Dacă sistemul de întreruperi este validat (activat), se apelează o *rutină de tratare a întreruperii*, asociată evenimentului care a generat întreruperea. După servirea întreruperii, se reia programul întrerupt, din starea existentă în momentul apariției întreruperii. În acest scop trebuie să se salveze contextul de lucru al UCP, și anume:

- Adresa de revenire, care reprezintă conținutul contorului de program din momentul apariției întreruperii;
- Indicatorii de condiții și de stare ai UCP;
- Registrele interne utilizate de rutina de tratare;
- Registrele de comandă de I/E care sunt modificate de rutina de tratare.

Sursele de întreruperi pot fi:

- Echipamentele periferice care solicită servicii (de exemplu, cereri de transfer).
- Elemente hardware de supraveghere a funcționării normale a sistemului: circuite de detectare a erorii de paritate a memoriei, de sesizare a căderii tensiunii de alimentare etc.
- Evenimente interne: depășiri la operațiile aritmetice, instrucțiuni inexistente etc.

Există două metode pentru alegerea adresei rutinei de tratare. Ele se numesc întreruperi *nevectorizate*, respectiv *vectorizate*. În cazul întreruperilor nevectorizate, rutina de tratare se află la o adresă fixă în memorie. La întreruperile vectorizate, adresa este furnizată procesorului de către sursa de întrerupere, sub forma unui *vector de întrerupere*. La unele sisteme, vectorul de întrerupere indică adresa de început a rutinei de tratare. La alte sisteme, vectorul de întrerupere reprezintă o referință la o locație de memorie unde este memorată adresa de început a rutinei de tratare.

În cazul transferului prin întreruperi, UCP transmite o comandă modulului de I/E, după care execută alte operații. Modulul de I/E va întrerupe UCP atunci când este pregătit pentru transferul datelor, UCP va executa transferul, după care va continua operațiile anterioare.

Pentru intrare, modulul de I/E primește o comandă de citire de la UCP. Modulul de I/E va începe citirea datelor de la un periferic asociat. După depunerea datelor în registrul de date al modulului, acesta generează o cerere de întrerupere, și apoi așteaptă

până când datele sunt solicitate de UCP. În acest moment, modulul plasează datele pe magistrala de date, fiind pregătit pentru o altă operație de I/E.

Din punctul de vedere al UCP, operațiile efectuate pentru intrare sunt următoarele. UCP transmite o comandă de citire, iar apoi execută alte operații (alte programe în cazul multiprelucrării). La sfârșitul fiecărei instrucțiuni, UCP testează dacă există cereri de întrerupere. La apariția unei cereri de la un modul de I/E, UCP salvează contextul programului și execută rutina de tratare a întreruperii. În această rutină UCP citește un cuvânt de date de la modulul de I/E și îl depune în memorie. Apoi reface contextul programului întrerupt și continuă execuția acestuia. Această procedură continuă până când se termină de transferat numărul de cuvinte necesare.

Figura 9.5 prezintă utilizarea transferului prin întreruperi pentru citirea unui bloc de date.

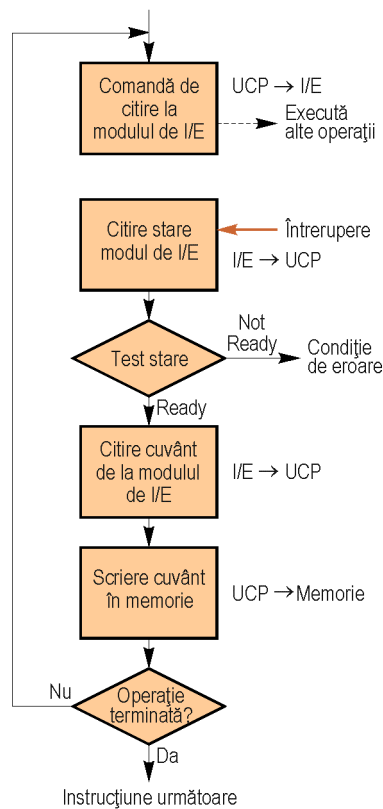


Figura 9.5. Utilizarea transferului prin întreruperi pentru citirea unui bloc de date.

9.5.3. Transferul prin acces direct la memorie

Transferul programat, chiar dacă se efectuează prin întreruperi, necesită intervenția UCP pentru transferul datelor între memorie și un modul de I/E, pentru fiecare octet sau cuvânt. Acest mod de transfer are deci două dezavantaje:

1. Rata de transfer este limitată de viteza cu care UCP poate testa și deservi un dispozitiv.
2. UCP este ocupată cu gestionarea transferului, fiind necesară execuția unui anumit număr de instrucțiuni pentru fiecare transfer.

Transferul prin DMA elimină aceste dezavantaje, prin executarea transferului direct între memoria internă și sistemul de I/E, participarea UCP fiind foarte redusă. Este necesar un modul suplimentar, numit *modul DMA* sau *controler DMA*, care poate prelua controlul asupra magistralelor sistemului, generând semnalele de comandă necesare. Structura generală a unui modul DMA este prezentată în Figura 9.6.

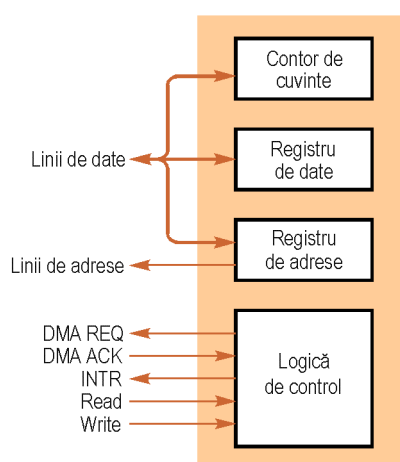


Figura 9.6. Structura generală a unui modul DMA.

Modulul DMA trebuie să conțină următoarele elemente principale:

- O logică pentru asigurarea legăturii cu memoria (semnale de citire și scriere);
- O linie de cerere DMA către memorie și o linie de acceptare DMA de la UCP;
- O logică de arbitraj a priorităților în cazul existenței mai multor canale;
- Un registru de comenzi pentru validarea modulului DMA, selectarea priorității canalelor, selectarea modului de lucru etc.

Pentru fiecare canal, trebuie să existe următoarele registre:

- *Registru de adrese*, pentru păstrarea adresei de început a transferului și a adresei curente;

- *Registru numărător*, pentru păstrarea numărului de cuvinte rămase de transferat;
- *Registru de stare*, care indică starea activată sau neactivată a canalului și condiția de terminare a transferului.

Controlerul DMA trebuie să preia controlul asupra magistralei pentru a efectua transferul datelor cu memoria. De aceea, controlerul DMA trebuie să utilizeze magistrala numai când aceasta nu este utilizată de UCP, sau să determine ca UCP să suspende temporar operațiile. Rezultă două metode de efectuare a transferului DMA:

1. Prin utilizarea intervalelor de timp în care UCP nu face acces la memorie; metoda se numește transfer prin *furt de ciclu* (“*cycle stealing*”), deoarece modulul DMA “fură” un ciclu de memorie de la UCP. Activitatea UCP nu este influențată de operațiile DMA, cu excepția cazului când se ajunge la un punct în care este necesară memoria.
2. Prin suspendarea operațiilor efectuate de UCP în timpul transferului și trecerea magistralei în starea de înaltă impedanță. Ciclul de instrucțiuni se va relua apoi din punctul în care a fost suspendat. Metoda se numește transfer *în rafală* (“*data break*”).

Figura 9.7 indică punctele din ciclul de instrucțiune în care activitatea UCP poate fi suspendată. În fiecare caz, activitatea UCP este suspendată înaintea momentului în care necesită utilizarea magistralei. Controlerul DMA transferă un cuvânt și returnează controlul UCP. De notat că aceasta nu este o întrerupere; nu se salvează contextul programului și nu se execută alte operații, ci UCP își suspendă activitatea pentru un ciclu de magistrală.

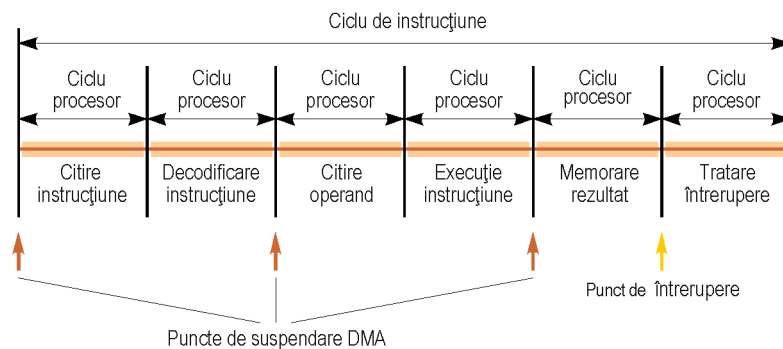


Figura 9.7. Puncte de suspendare a activității UCP pentru DMA și întreruperi.

Pentru citirea sau scrierea unui bloc de date, UCP transmite controlerului DMA o secvență de inițializare, care trebuie să conțină următoarele informații:

- Sensul transferului (citire sau scriere);
- Adresa echipamentului de I/E implicat;

- Adresa de început a zonei de memorie cu care se efectuează transferul;
- Numărul de octeți sau de cuvinte care trebuie transferate.

Pentru transmiterea acestor informații se utilizează transferul programat.

UCP va executa alte operații, iar controlerul DMA va solicita controlul asupra magistralei, generând adresele și semnalele de comandă pentru efectuarea transferului. Se incrementează registrul adresei curente și se decrementează numărătorul de cuvinte. După acest ciclu DMA, în care s-a transferat un cuvânt, se continuă cu alte cicluri sau se dă controlul UCP. La terminarea transferului, controlerul generează un semnal de întrerupere către UCP.

Astfel, interacțiunea dintre UCP și sistemul de I/E este la nivel de bloc, UCP intervenind numai la începutul și sfârșitul transferului (Figura 9.8). Eficiența este mult mai ridicată decât în cazul transferului prin întreruperi.

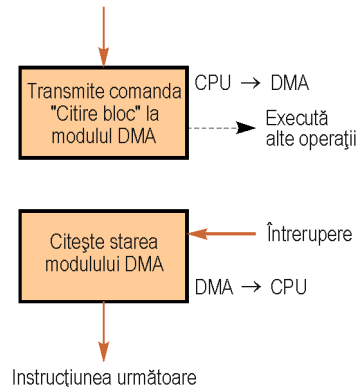


Figura 9.8. Utilizarea transferului prin DMA pentru citirea unui bloc de date.

Există mai multe configurații ale unui sistem de I/E care utilizează transferul prin DMA (Figura 9.9).

În primul exemplu (Figura 9.9(a)), toate modulele partajează aceeași magistrală sistem. Modulul DMA utilizează I/E programate pentru transferul datelor între memorie și un modul de I/E. Această configurație, deși poate fi ieftină, este ineficientă. Fiecare transfer a unui cuvânt necesită două cicluri de magistrală, ca și în cazul transferului programat prin intermediul UCP.

Numărul ciclurilor de magistrală poate fi redus prin integrarea funcțiilor DMA și a celor de I/E. În acest caz există o cale între modulul DMA și unul sau mai multe module de I/E, cale care nu include magistrala sistem (Figura 9.9(b)). Logica DMA poate fi o parte a unui modul de I/E, sau poate fi un modul separat care controlează unul sau mai multe module de I/E.

În ultimul exemplu (Figura 9.9(c)) modulele de I/E se conectează la modulul DMA printr-o magistrală de I/E. Astfel se reduce numărul interfețelor din modulul DMA la una singură și se permite extinderea sistemului. În ultimele două cazuri, magistrala sistem pe care modulul DMA o partajează cu UCP și memoria este utilizată de modulul

DMA numai pentru transferul datelor cu memoria. Transferul datelor între modulul DMA și modulele de I/E nu utilizează magistrala sistem.

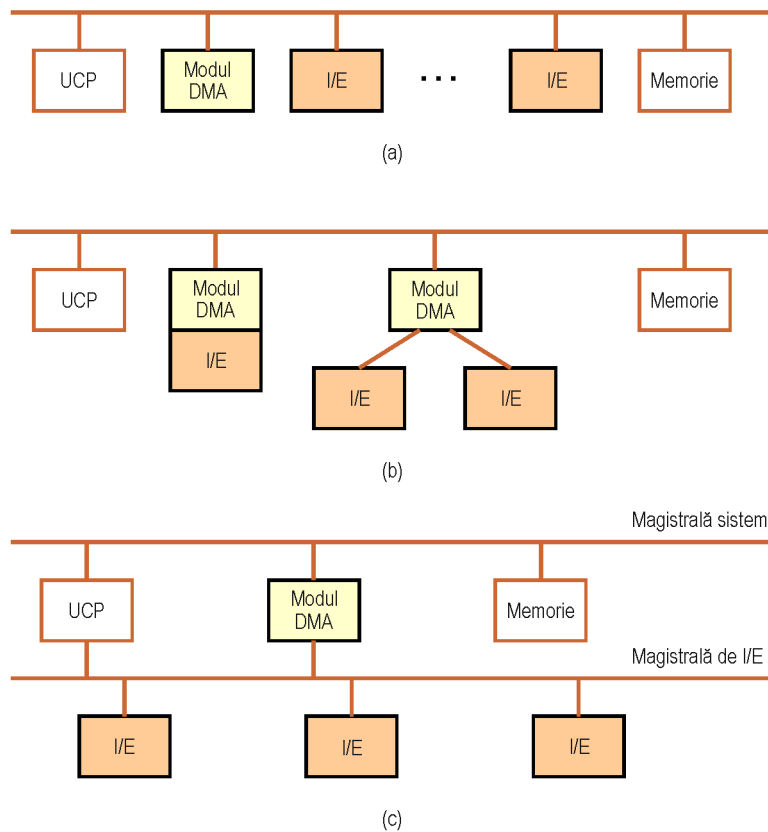


Figura 9.9. Diferite configurații ale unui sistem de I/E care utilizează transferul prin DMA.

9.5.4. Transferul prin canale de I/E

9.5.4.1. Principiul transferului prin canale de I/E

Deși accesul direct la memorie eliberează UCP de numeroase operații de I/E, în cazul perifericelor rapide vor fi necesare numeroase cicluri de magistrală pentru referințele la memorie. În timpul acestor cicluri, UCP trece în starea de așteptare. Furturile de ciclu vor satura magistrala, și chiar dacă se va genera o singură întrerupere la fiecare bloc transferat, se consumă un anumit timp pentru tratarea întreruperilor.

Pentru a se elimina aceste dezavantaje, modulele de I/E au fost îmbunătățite, devenind procesoare care dispun de un set de instrucțiuni specializat pentru I/E. UCP

transmite o comandă procesorului de I/E pentru a executa un program de I/E din memorie. Acesta execută instrucțiunile programului fără intervenția UCP. Astfel este posibilă specificarea de către UCP a unei secvențe de operații de I/E și întreruperea UCP numai la terminarea întregii secvențe.

Unele module de I/E au o memorie locală proprie, fiind de fapt calculatoare specializate care pot controla un set larg de dispozitive de I/E, cu o intervenție minimă a UCP. Modulele de I/E care pot executa programe sunt numite în unele cazuri *canale de I/E*, iar în alte cazuri *procesoare de I/E* (mai ales cele care dispun de o memorie locală proprie).

Structura unui calculator cu un canal de I/E este prezentată în Figura 9.10. Comunicația cu memoria se poate realiza prin DMA. Canalul de I/E asigură o cale pentru transferul datelor între diferitele periferice și memorie. UCP inițiază transferul de I/E, după care canalul de I/E funcționează independent de UCP și continuă transferul datelor între periferice și memorie.

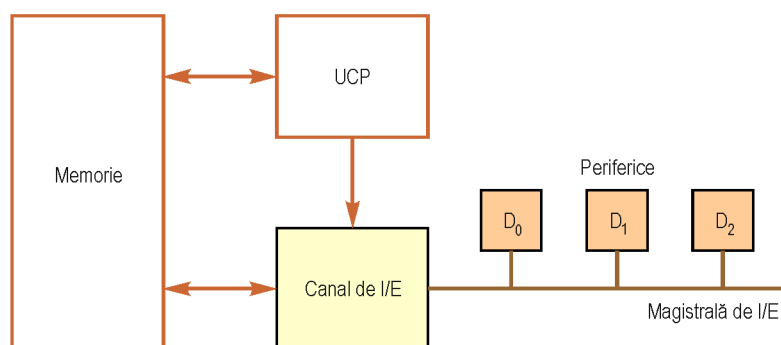


Figura 9.10. Structura unui calculator cu un canal de I/E.

Canalul poate restructura datele de la diferite periferice. De exemplu, poate fi necesară citirea mai multor caractere de la un dispozitiv de intrare și împachetarea lor într-un cuvânt înaintea transferului în memorie. Datele sunt preluate de canal la viteza dispozitivului în timp ce UCP execută un program. După asamblarea datelor de intrare într-un cuvânt de memorie, acesta se transferă de la canalul de I/E direct în memorie printr-un furt de ciclu. Similar, un cuvânt transferat din memorie la canalul de I/E va fi transmis la un dispozitiv de ieșire la viteza și numărul de biți ai acestuia.

De obicei, UCP are sarcina inițierii operațiilor de I/E, prin instrucțiuni de *start al unui transfer* și de *test al stării operațiilor*. Canalul solicită o intervenție a UCP printr-o cerere de întrerupere, și răspunde la cererile UCP prin plasarea unui cuvânt de stare într-o locație de memorie, cuvânt care poate fi examinat de UCP. Pentru execuția unei operații de I/E, UCP transferă o instrucțiune canalului, indicând perifericul care trebuie utilizat și adresa de început a instrucțiunilor suplimentare.

Instrucțiunile citite de canal din memorie se numesc *comenzi*, pentru a le deosebi de instrucțiunile transmise de UCP canalului. Instrucțiunile și comenzile au funcții similare. Cuvintele de comandă sunt pregătite și sunt depuse de UCP în memorie.

Acestea reprezintă un *program de canal*. UCP informează canalul asupra locațiilor în care va găsi cuvintele de comandă.

Există două tipuri de canale de I/E mai utilizate: *canale selectoare* și *canale multiplexoare*. Un *canal selector* controlează mai multe periferice rapide, la un moment dat fiind dedicat pentru transferul datelor cu un singur periferic. Canalul de I/E selectează un periferic și efectuează transferul datelor. Fiecare periferic, sau un set redus de periferice, este gestionat de un controler (modul de I/E). Astfel, canalul de I/E are rolul de a gestiona aceste controlere de I/E în locul UCP.

Un *canal multiplexor* poate gestiona transferuri simultane cu mai multe periferice, prin multiplexare în timp. Pentru periferice lente, un *multiplexor la nivel de octet* acceptă sau transmite caractere la dispozitive multiple. Pentru periferice rapide, un *multiplexor la nivel de bloc* transferă blocuri de date între diferite dispozitive.

9.5.4.2. Comunicația dintre UCP și canalul de I/E

În cele mai multe cazuri, această comunicație se realizează prin *mesaje* depuse în memorie. Se va prezenta printr-un exemplu simplificat modul în care poate avea loc comunicația dintre UCP și canal.

UCP inițiază o operație de I/E printr-o instrucțiune de I/E adresată canalului. Un format posibil al acestei instrucțiuni este prezentat în Figura 9.11.



<u>Cod de operație</u>	<u>Operație</u>
001	Test I/E
010	Start I/E
100	Citire stare I/E

Figura 9.11. Formatul unei instrucțiuni tipice pentru inițierea unei operații de I/E.

Codul instrucțiunii are următoarele câmpuri:

1. Un cod de operație care specifică operația de executat;
2. O adresă de dispozitiv care specifică adresa unui anumit dispozitiv de I/E conectat la canal;
3. O adresă de memorie care va fi utilizată de canal atunci când acesta răspunde la instrucțiunea UCP.

Secvența de operații care se execută este ilustrată în Figura 9.12. UCP transmite o instrucțiune de *test I/E* către canal. Canalul răspunde prin depunerea unui cuvânt de stare în memorie, la adresa specificată de câmpul de adresă al instrucțiunii, care indică starea canalului și a dispozitivului de I/E.

UCP testează cuvântul de stare din memorie și decide operația următoare. UCP transmite apoi instrucțiunea de lansare a transferului de I/E. Această instrucțiune specifică adresa primului cuvânt de comandă pentru operația de I/E.

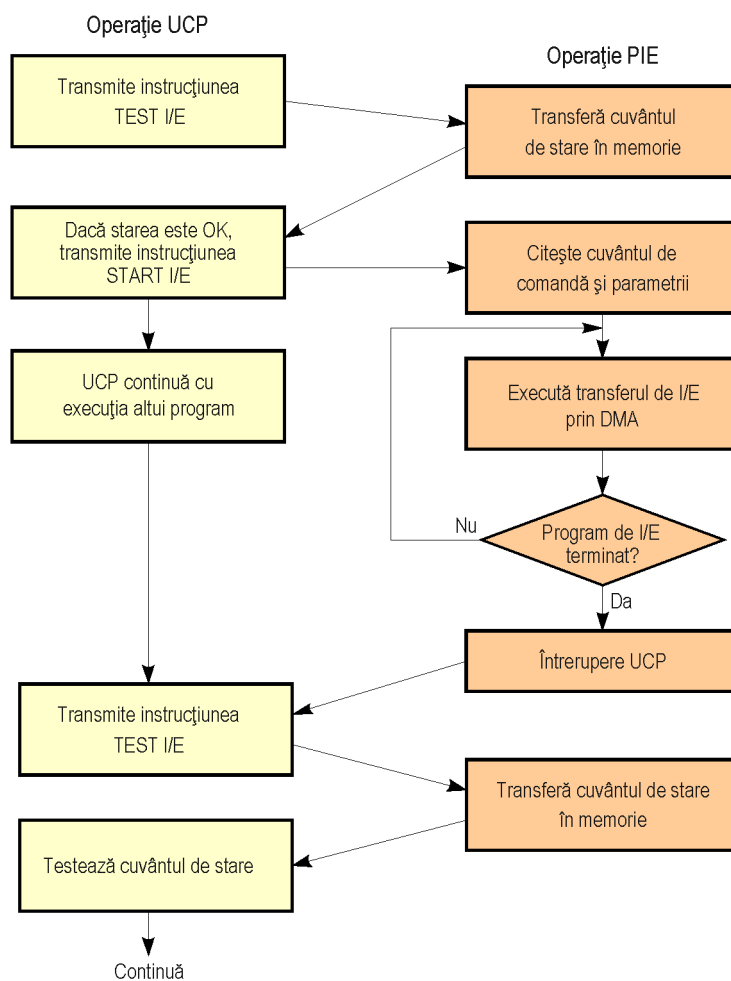


Figura 9.12. Operațiile executate pentru transferul prin canale de I/E.

UCP poate continua apoi execuția unui alt program în timp ce canalul este ocupat cu transferul. La terminarea transferului, canalul transmite o cerere de întrerupere către UCP. UCP răspunde la această cerere prin transmiterea instrucțiunii de citire a stării canalului. Canalul va plasa un cuvânt de stare în locația de memorie specificată de instrucțiune, care indică terminarea normală a transferului sau apariția unor erori.

Cuvintele de comandă specifică programul care este executat de canal. Adresa cuvintelor de comandă este transmisă de UCP printr-o instrucțiune de *start I/E*. Formatul tipic al unui cuvânt de comandă este indicat în Figura 9.13.

Adresă buffer	Contor de cuvinte	Cod de operație	Cod special
<u>Cod de operație</u>		<u>Operație</u>	
001		Citire (intrare)	
010		Scriere (ieșire)	
011		Rebobinare (bandă)	
<u>Cod special</u>		<u>Operație</u>	
0000	1010	Translatare de cod	
0111	0010	Adresa unei înregistrări (bandă)	

Figura 9.13. Formatul tipic al unui cuvânt de comandă pentru un canal de I/E.

Câmpurile cuvântului de comandă sunt următoarele:

1. Adresa bufferului utilizat pentru transfer;
2. Contorul de cuvinte indică lungimea bufferului, deci numărul de cuvinte pe care le conține;
3. Câmpul de operație specifică operația care trebuie executată: o intrare, o ieșire sau alte operații;
4. Câmpul special indică o funcție specifică unui anumit echipament de I/E, câte-va din acestea fiind indicate în figură.

Canalul de I/E asigură transferul datelor între mai multe periferice și memorie. Canalul și UCP utilizează aceeași memorie, astfel că numărul de periferice este limitat de timpul de acces al memoriei. Unele unități rapide, ca discurile magnetice, pot utiliza un număr mare de cicluri de memorie. În acest caz, viteza UCP va fi micșorată, deoarece va trebui să aștepte terminarea unui acces la memorie de către canal înaintea utilizării memoriei.