

4.5 Subsistemul de intrare / iesire

Subsistemul de intrare / iesire, reprezentand unitatea de intrare si unitatea de iesire, este format din echipamentele periferice impreuna cu interfețele acestora. Componenta subsistemului de intrare / iesire este determinata in mare masura de domeniul aplicatiilor in care este utilizat sistemul de calcul.

Transferurile între echipamentele periferice si unitatea centrala a calculatorului se pot face in diferite moduri, cu performante si costuri diferite. Principalele moduri de transfer de I/E sunt:

- transfer programat de I/E;
- transfer prin DMA;
- transfer prin canal de I/E;
- transfer prin procesor de I/E.

1) *Transfer programat de I/E.* Acesta este cel mai simplu mod de transfer, cu un cost scazut, dar si cu performante modeste. Transferul este controlat chiar de procesor prin executia de instructiuni, in special instructiuni de citire si scriere porturi de I/E. In cadrul transferului programat exista doua submoduri:

- a) cu citirea ciclica a starii;
- b) in intreruperi.

Avantajul acestui mod este simplitatea interfetei si usurinta programarii, dar eficienta este scazuta. Se considera structura unei interfete generale de I/E in transfer programat (fig.4.5.1).

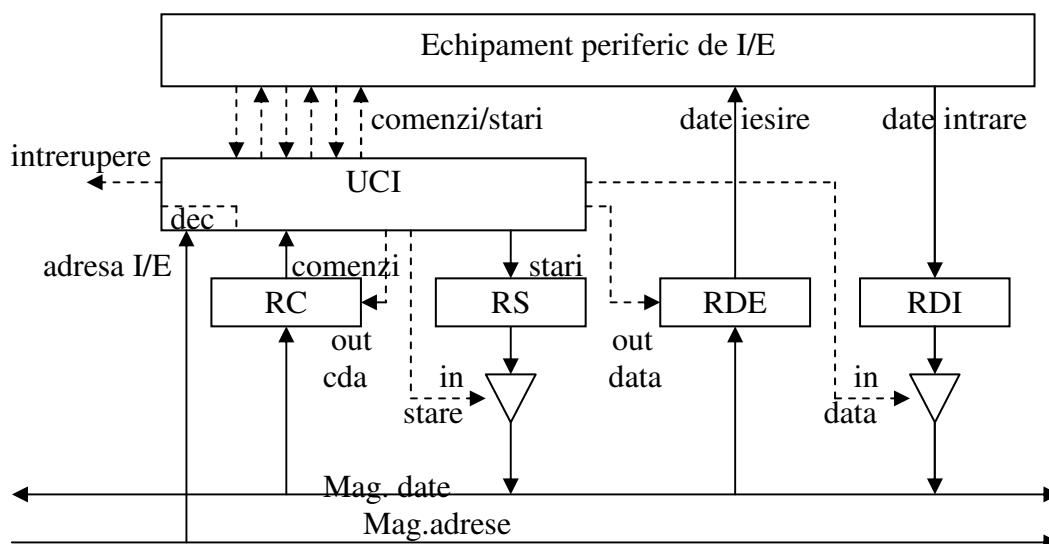


Fig.4.5.1 Interfața de I/E in transfer programat.

Resursele interfetei, prezentate in schema bloc sunt:

-echipamentul de intrare / iesire (poate fi numai de intrare, sau numai de iesire, sau de intrare / iesire);

-UCI reprezinta unitatea de comanda a interfetei;

-RC este registrul de comanda, in care programatorul inscrie printr-o instructiune **out** un cuvânt de comanda care determina operatiile realizate la nivelul interfetei. Incarcarea cuvântului de comanda se face prin activarea de catre UCI a unui semnal *out comanda*, la decodificarea adresei corespunzatoare portului de iesire asociat cu acest registru (adresa de port este preluata de pe magistrala de adrese de catre UCI si decodificata, pentru recunoasterea adreselor porturilor din cadrul interfetei);

-RS este registrul de stare, in care UCI inscrie informatiile de stare primite de la echipamentul periferic, dar si informatiile de stare de la nivelul interfetei. Fiecare bit din registru are o semnificatie precisa. Programatorul poate sa citeasca continutul acestui registru prin executia unei instructiuni **in**, cu adresa corespunzatoare portului de intrare asociat registrului. Citirea cuvântului de stare se face prin activarea semnalului de comanda *in stare* de catre UCI, cand continutul registrului este plasat prin intermediul circuitelor tampon cu trei stari pe magistrala de date si este transferat apoi la procesor.

-RDE este registrul de date de iesire in care programatorul inscrie un cuvânt de date printr-o instructiune **out** (UCI activand semnalul corespunzator *out data*). Data va fi transferata apoi la echipamentul periferic si scoasa pe suport extern.

-RDI este registrul de date de iesire in care un cuvânt trimis de echipamentul periferic este incarcat si apoi citit prin program cu o instructiune **in** (la activarea semnalului *in data*).

In schema bloc de mai sus este reprezentat si semnalul de intrerupere, care poate fi activat de catre UCI si care este transmis la sistemul de intreruperi si apoi la procesor, in cazul in care transferul de date cu echipamentul periferic se face in intreruperi.

a) *Transfer programat de I/E cu citirea ciclica a starii*. Toate fazele transferului sunt controlate de catre procesor, prin executia unei secvente de program. Cea mai mare parte din timp procesorul asteapta intr-o bucla de program incheierea unui transfer elementar (transferul unui cuvânt de date), datorita diferentei mari de viteza intre componentele mecanice ale echipamentului periferic si componentele integral electronice ale unitatii centrale de prelucrare. Pentru a urmări cum se desfasoara transferul de date in acest mod, se va considera scrierea unui bloc de cuvinte la un

echipament periferic de iesire. Organigrama operatiilor elementare necesare este prezentata in continuare (fig.4.5.2):

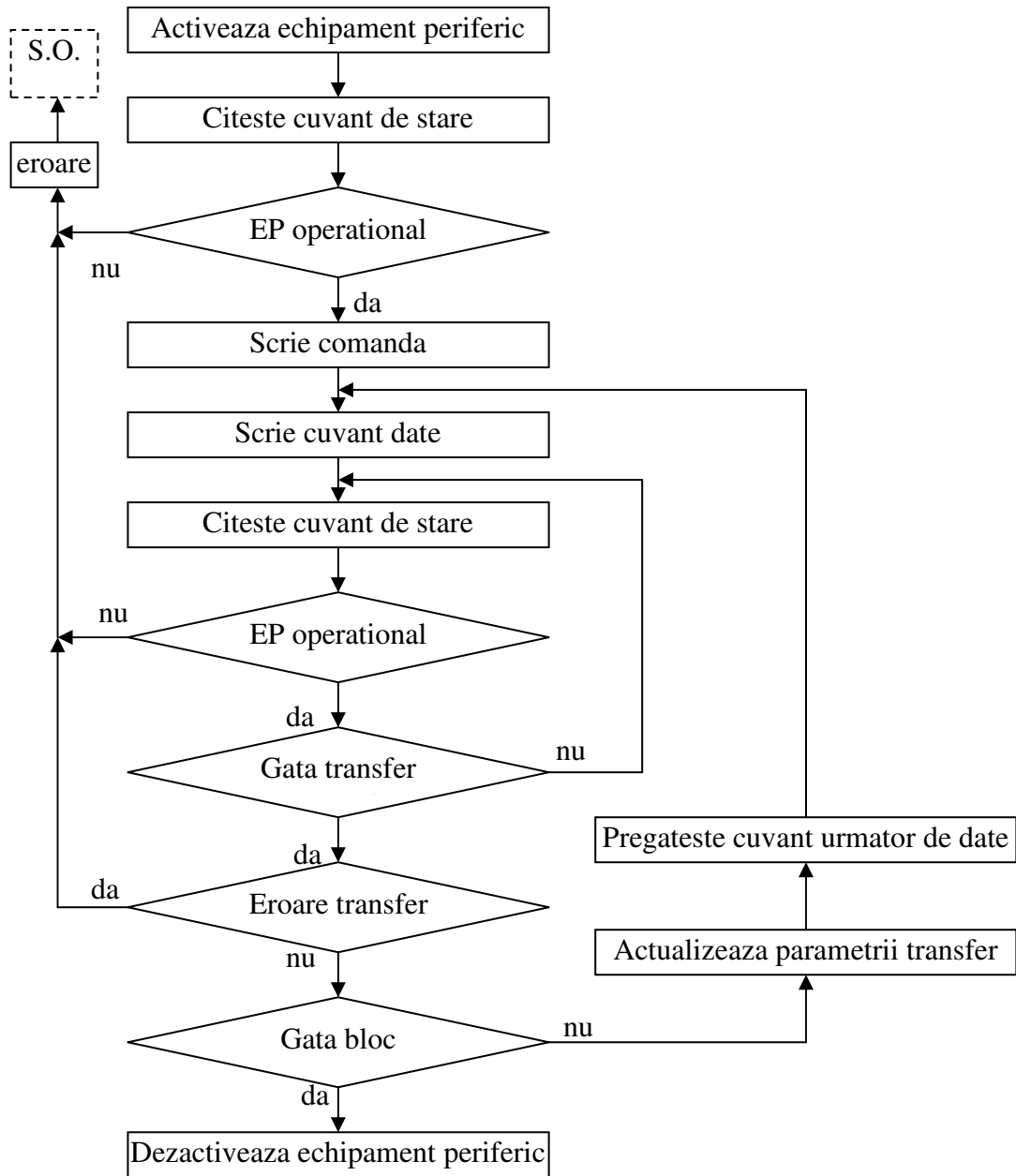


Fig.4.5.2 Transfer programat de I/E cu citirea ciclica a starii.

b) *Transfer programat de intrare / iesire in intreruperi.* Este o solutie mai performanta decat cea precedenta, deoarece procesorul comanda un transfer elementar pentru un cuvânt de date, iar in continuare procesorul nu mai asteapta incheierea transferului, realizand prelucrari utile. La incheierea transferului interfata echipamentului periferic genereaza o intrerupere, catre sistemul de intreruperi, care la randul sau

genereaza o intrerupere catre procesor. Procesorul isi suspenda prelucrarile in curs si trece la executia unei secvente speciale de program, procedura de tratare a intreruperii. In aceasta procedura procesorul executa o serie de operatii legate de transfer si poate comanda un nou transfer elementar, reluandu-si apoi prelucrarile suspendate.

Pentru exemplificare se considera organigrama operatiilor efectuate de catre procesor pentru scrierea unui bloc de cuvinte la un echipament periferic de iesire in intreruperi (fig.4.5.3).

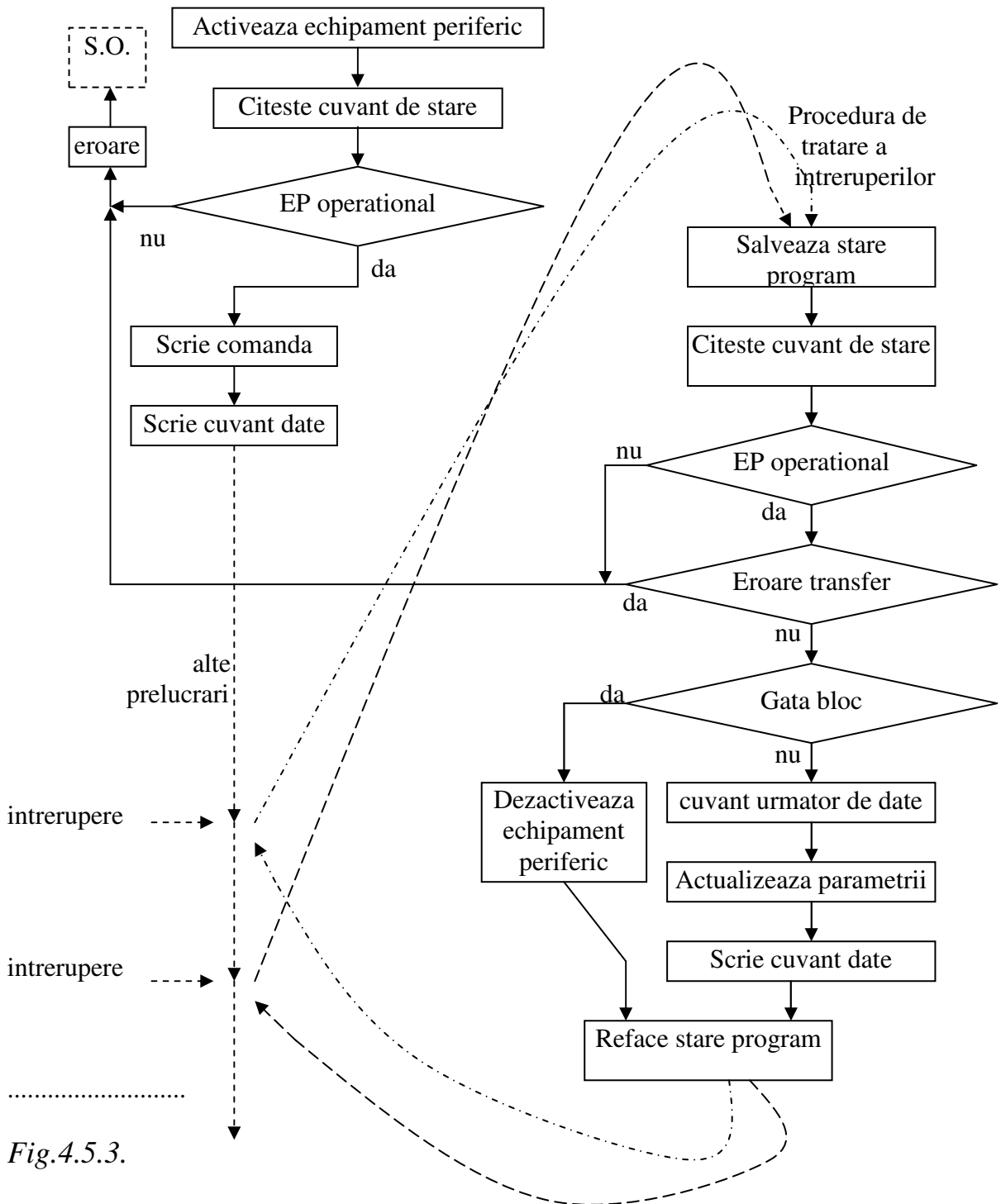


Fig.4.5.3.

2) *Transfer prin DMA*. DMA (Direct Memory Access) este o solutie perfectionata de transfer de I/E. Acest modul permite realizarea transferurilor direct intre memoria interna a calculatorului si echipamentele periferice. Procesorul executa numai initializarea procesului de transfer, in continuare toate operatiile elementare legate de transfer sunt realizate de catre modulul DMA. In timpul transferului procesorul poate executa alte prelucrari utile. Modulul DMA dispune de resurse proprii care ii permit sa preia controlul magistralei si sa efectueze transferul.

Procesorul furnizeaza modulului DMA parametrii de transfer, reprezentati de adresa de inceput a blocului de date din memorie implicat in transfer si lungimea acestuia. Actualizarea parametrilor se face automat de catre DMA dupa fiecare transfer elementar. Structura modulului DMA va fi studiata mai in detaliu in cadrul acestui capitol.

3) *Transfer prin canal de intrare / iesire*. Canalul este un dispozitiv mai perfectionat decat modulul DMA. Acesta poate transfera chiar o inlantuire de blocuri de date aflate in memorie la adrese diferite si de lungimi diferite. De asemenea, procesorul specifica canalului parametrii de transfer, dupa care toate operatiile necesare sunt efectuate de catre canal. In timpul transferului procesorul poate efectua prelucrari utile. In plus fata de modulul DMA, canalul de I/E poate lua decizii la aparitia unor evenimente neasteptate, ca de exemplu la aparitia unei erori in timpul transferului.

Exista doua tipuri de canale de I/E:

-canal selector, este mai simplu, cu mai putine resurse. Acest tip de canal realizeaza transferul cu un singur echipament de intrare / iesire la un moment dat. Dupa incheierea completa a transferului cu echipamentul respectiv se poate incepe transferul cu un nou echipament periferic.

-canal multiplexor, are o structura mai complexa, dispunand de resurse care ii permit gestionarea in paralele a mai multe transferuri cu echipamentele periferice. Evident, accesele efective la memorie pentru citire sau scriere sunt secventializate.

4) *Transfer prin procesor de intrare / iesire*. Procesorul de I/E este un procesor suplimentar in cadrul sistemului de calcul. Acesta preia de la procesorul central toate sarcinile legate de transferurile de intrare / iesire, astfel ca procesorul central poate efectua numai prelucrari utile. Principalele avantaje ale acestei solutii sunt:

- realizarea transferurilor la nivel logic (transfer de fisiere);
- preprocesari, filtrari de date care se transfera;
- cautari de date la nivelul echipamentelor periferice.

Interfata seriala

Se considera un exemplu de interfata in transfer programat si anume interfata seriala. Aceasta permite realizarea de transferuri de date intre sistemul de calcul si un dispozitiv numeric in modul serial (bit cu bit, transferandu-se un singur bit la un moment dat). Principalul avantaj il reprezinta simplitatea si costul scazut, dar rata de transfer este mai mica in raport cu transferurile paralele.

Pentru comunicatia seriala (standardul RS 232-C) intre doua dispozitive numerice se utilizeaza urmatoarele semnale (fig.4.5.4):

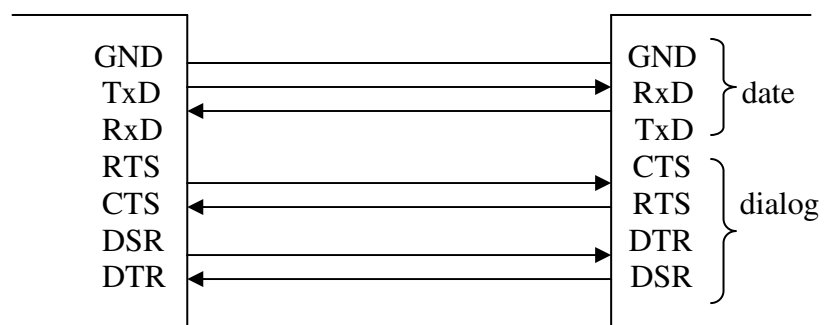


Fig.4.5.4 Schimbul de semnale in comunicatia seriala.

unde,

GND (ground) = masa electrica;

TxD (Transmitter Data) = transmisie de date;

RxD (Receiver Data) = receptie de date;

RTS (Request to Send) = cerere pentru emisie;

CTS (Clear to Send) = anulare in vederea transmisiei;

DSR (Data Set Ready) = semnifica conectare la linie (echipament operational);

DTR (Data Terminal Ready) = dispozitivul cu care se face transferul de date este conectat la linie.

Datele transmise pe linia seriala sunt organizate conform diagramei de semnale urmatoare (fig.4.5.5):

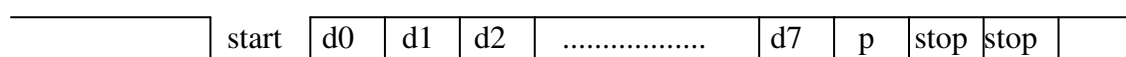


Fig.4.5.5 Diagrama de semnal pentru transmisia unui cuvnt pe linia seriala.

RC = registrul de comanda (la fel ca la interfata generala prezentata anterior). Anumiti biti au o importanta deosebita pentru functionarea interfetei seriale. Astfel,

.....	DTR	L1	L0	STOP	PAR
-------	-----	----	----	------	-----	-------

DTR : comanda linia de dialog cu acelasi nume, pentru conectarea la linie a dispozitivului;

L1,L0 : determina lungimea cuvintului transmis pe linia seriala;

STOP : specifica numarul de biti de stop utilizati in transmisie si care se verifica la receptie;

PAR : determina paritatea utilizata pentru verificarea datelor transmise (paritate para sau impara).

RS = registrul de stare (la fel ca la interfata generala). Anumiti biti din acest registru sunt importanti pentru determinarea starii interfetei:

.....	DSR	TxRDY	TxEMPTY	RxRDY	PE	FE	OE
-------	-----	-------	---------	-------	----	----	----	-------

DSR : furnizeaza starea liniei de dialog cu acelasi nume, avand semnificatia de echipament operational;

TxRDY (Transmitter Ready) : specifica daca registrul RDE este liber pentru ca procesorul sa trimita un nou cuvint de date care sa fie apoi transmis pe linia seriala;

TxEMPTY (Transmitter Empty) : specifica daca ambele registre folosite la transmisie RDE si RTS, sunt libere, deci interfata nu are nimic de transmis la momentul respectiv;

RxRDY (Receiver Ready) : specifica daca s-a receptionat un cuvint de date de pe linia seriala si acesta se poate citi din registrul RDI printr-o instructiune **out**;

PE (Parity Error) : activarea acestui bit de stare indica faptul ca la receptia ultimului cuvint de date a aparut o eroare de paritate;

FE (Framming Error) : indica o eroare datorata diferentei mari de frecventa intre ceasul de la transmisie si ceasul de la receptie, cand in locul unui bit de stop se detecteaza 0 logic (posibil bitul de paritate de la cuvintul curent sau bitul de start de la cuvintul urmator);

OE (Overrun Error) : eroare aparuta din programarea interfetei seriale. Cuvintul de date receptionat anterior si aflat in RDI nu a fost citit in timp util de catre procesor (printr-o instructiune **in**), s-a receptionat de pe linia seriala un cuvint nou, care a fost inscris in RDI peste cuvintul precedent Astfel s-a pierdut un cuvint din cadrul blocului transferat.

Modulul DMA

Modulul DMA (Direct Memory Access) realizeaza transferul de blocuri de date direct intre memoria interna a sistemului si echipamentele periferice. Organizarea unui sistem de calcul cu DMA este prezentata in figura urmatoare (fig.4.5.7):

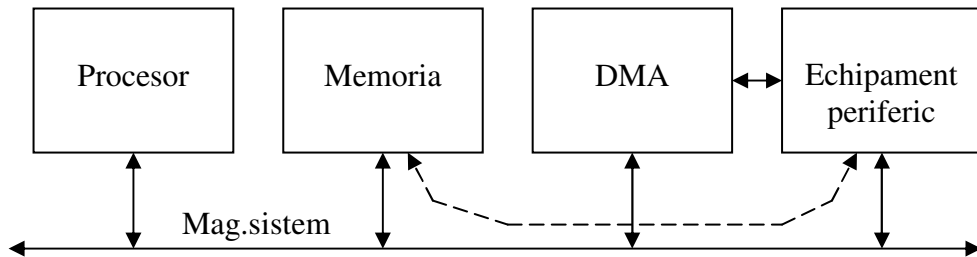


Fig.4.5.7 Organizarea unui sistem cu DMA.

Pentru transferul de date DMA-ul utilizeaza magistrala sistemului. Datele sunt transferate intre memoria principala si echipamentul periferic, fara interventia directa a procesorului, care poate realiza alte prelucrari, in paralel cu transferul, daca aceste prelucrari nu necesita magistrala, informatiile necesare gasindu-se in memoria cache.

Structura modulului DMA este prezentata in figura urmatoare (fig.4.5.8):

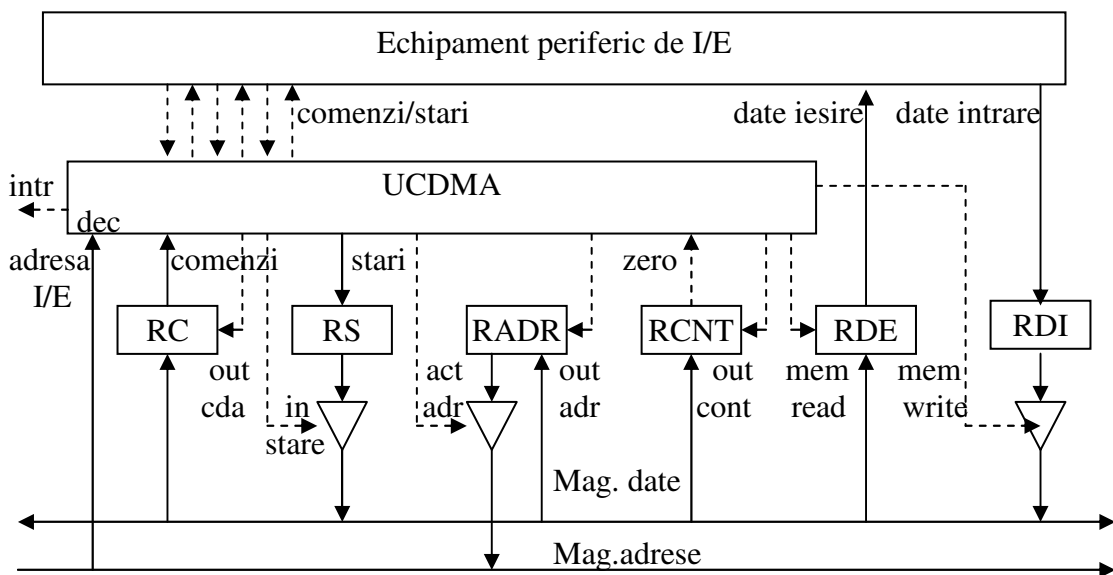


Fig.4.5.8 Structura unui modul DMA.

Resursele modulului DMA sunt:

UCDMA = unitatea de comanda a modulului DMA;

RC = registrul de comanda;

RS = registrul de stare, la fel ca la interfata generala in transfer programat. De remarcat faptul ca dialogul dintre procesor si DMA pentru stabilirea modului in care se va desfasura transferul si pentru testarea conditiilor de incheiere a transferului, se realizeaza prin instructiuni de intrare / iesire (**in** si **out**), la fel ca in transferul programat. Procesorul poate sa inscrie un cuvânt de comanda pentru DMA sau poate sa citeasca un cuvânt de stare de la DMA.

RDI, RDE = registrele pentru date de intrare si de iesire, prin intermediul carora se transfera blocul de date între echipamentul periferic si memoria interna a calculatorului.

RADR = registrul de adresa al DMA-ului. La inceputul transferului unui bloc de date, in acest registru procesorul inscrie adresa de inceput a blocului in memorie. Dupa fiecare transfer elementar continutul acestui registru se incrementeaza, pentru obtinerea adresei cuvântului urmator de date. La fiecare transfer elementar continutul acestui registru este plasat pe magistrala de adrese pentru furnizarea adresei curente la memoria interna.

RCNT = registrul contor. La inceputul transferului acest registru este incarcat cu lungimea blocului de cuvinte de transferat. Dupa fiecare transfer elementar, continutul registrului este decrementat. Registrul poate fi citit in orice moment prin program pentru a afla numarul de cuvinte ramase de transferat. Cand registrul ajunge la valoarea zero prin decrementari succesive, unitatea de comanda a modulului DMA este anuntata de incheierea transferului.

Modulul DMA, la fel ca si procesorul, este un modul master si deci poate intra in concurenta pentru preluarea controlului magistralei sistemului si executia transferului între memorie si echipamentul periferic. Cererile de preluare a controlului magistralei sunt transmise catre controlorul magistralei, care pentru cereri simultane va aplica un anumit algoritm de prioritati, oferind controlul unui anumit modul. In momentul in care DMA-ul primeste controlul magistralei, acesta va realiza transferul de date pe magistrala.

Sistemul de intreruperi

Intreruperile reprezinta cereri lansate de catre echipamentele periferice si de unele module interne ale calculatorului (sistemul de memorie, unitatea aritmetica-logica, ceasul de timp real, etc.) prin care se solicita atentia procesorului la aparitia unor evenimente in functionarea calculatorului. Intreruperile nu sunt asociate intotdeauna cu aparitia erorilor. Se genereaza intreruperi si in situatii normale de functionare, ca de exemplu la terminarea unui transfer de date cu un echipament periferic, la incheierea unui interval de timp la ceasul de timp real, etc. Luarea in considerare a unei cereri de intrerupere are ca efect suspendarea programului in curs de executie, salvarea starii programului intrerupt si executia de catre procesor a unei secvente speciale de program, numita procedura de tratare a intreruperii, cand se incearca rezolvarea situatiei aparute. Dupa incheierea procedurii, se reface starea programului intrerupt si se reia executia sa.

In general, intreruperile se pot clasifica in urmatoarele categorii:

- intreruperi interne, generate la nivelul unitatii centrale a calculatorului, de evenimente cum sunt: eroare de paritate la memorie, depasire in unitatea aritmetica – logica, executia unor instructiuni speciale (intreruperi software);

- intreruperi externe, generate din afara unitatii centrale a calculatorului. La randul lor intreruperile externe sunt reprezentate de:

- intrerupere externa nemascabila, care nu poate fi inhibata si este asociata cu un eveniment grav in functionarea sistemului (de exemplu caderea tensiunii de alimentare);

- intreruperi externe mascabile, generate de interfetele echipamentelor periferice. Aceste intreruperi pot fi mascate individual sau in bloc prin program.

Sistemul de intreruperi prezentat in continuare este un sistem vectorizat. Intr-o zona de memorie se gaseste un vector de locatii (vectorul de celule capcana), in care pentru fiecare tip de intrerupere se gaseste o locatie (celula) continand adresa procedurii de intrerupere respective. La

luarea in considerare a unei intreruperi, intr-un ciclu special, procesorul preia adresa procedurii din celula capcana corespunzatoare si trece apoi la executia acelei proceduri.

Resursele sistemului de intreruperi prezentate in schema bloc sunt urmatoarele (fig.4.5.9):

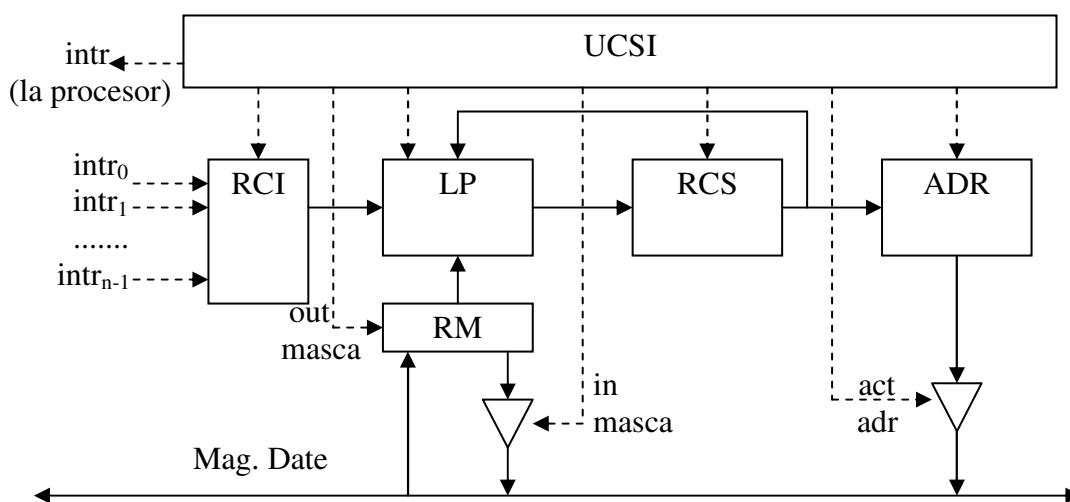


Fig.4.5.9 Structura unui sistem de intreruperi.

UCASI = unitatea de comanda a sistemului de intreruperi.

RCI = registrul cererilor de intrerupere. In acest registru se memoreaza toate cererile de intrerupere sosite din sistem. O cerere de intrerupere care intra in servire va fi stearsa din RCI.

RM = registrul masca a cererilor de intrerupere. Fiecarei cereri de intrerupere ii corespunde in acest registru cate un bit care permite mascarea (dezactivarea) sau activarea cererii respective. Continutul registrului poate fi in scris de catre procesor prin executia unei instructiuni **in** cu adresa de port corespunzatoare acestui registru.

LP = logica de prioritati, care determina cererea de intrerupere activa (nemascata) cu prioritatea cea mai mare din RCI.

RCS = registrul cererilor in servire memoreaza toate cererile de intrerupere pentru care a inceput executia procedurilor corespunzatoare de servire. Logica de prioritati LP va inscrie o cerere noua de intrerupere in RCS si o va sterge din RCI, daca aceasta are prioritate mai mare decat

cererea cea mai prioritara din RCS. In momentul inscrierii cererii in RCS, va incepe executia procedurii de tratare a intreruperii respective.

ADR = logica de adrese. La inscrierea unei cereri noi in RCS, aceasta logica va furniza pe magistrala in timpul unui ciclu special, numit ciclu de luare in considerare a intreruperii (interrupt acknowledge), adresa celulei capcana corespunzatoare cererii respective. In celula capcana se gaseste adresa rutinei de tratare a intreruperii, care va intra astfel in executie.