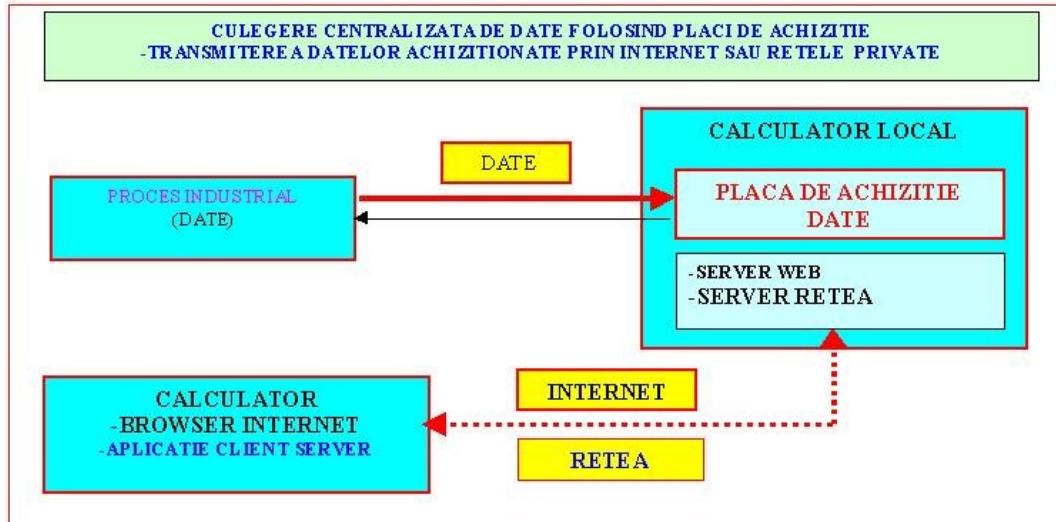


Sisteme de achizitie date bazate pe calculator

Culegerea de date cu ajutorul calculatorului PC presupune introducerea unei placi de achizitie, intr-un slot al placii de baza a calculatorului

Transferul de date ce are loc intre placa de achizitie si calculator trebuie efectuat la viteza si cantitatea de date impuse de sistemul monitorizat. Ideal ar fi ca viteza si cantitatea maxima de date pe care vrem sa le culegem din exterior sa depinda numai de parametrii functionali ai placii de achizitie. Din pacate comunicarea dintre placa de achizitie si memoria calculatorului are limitari serioase impuse de arhitectura PC-ului.



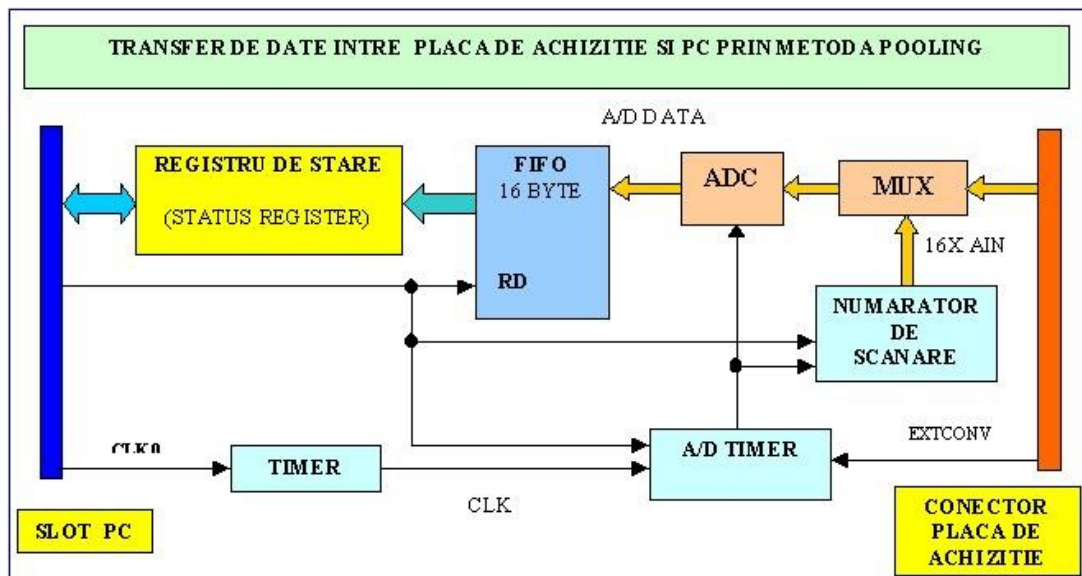
Exista trei posibilitati de comunicare intre placa de achizitie si calculator.

- **Pooling** –registrul de stare al placii de achizitie este citit periodic dictat de un timer al CPU
- **Intreruperi** –Placa de achizitie are capabilitatea de a intrerupe CPU de fiecare data cind trebuiesc transferate date
- **DMA** –controlerul DMA (Direct Memory Acces) efectueaza transferul de date in paralel cu activitatea CPU

Pooling

Citirea periodica a datelor achizitionate (monitorizarea repetata) porta numele de Pooling. Acest tip de transfer este potrivit pentru culegerea de date folosind placa de achizitie instalata in calculator, fiind destul de rapid si usor de implementat.

In figura de jos este reprezentat schematic principiul de achizitie prin aceasta metoda .



În figura de sus este reprezentat modul de achiziție pentru semnale de tip analogic. Partea de intrare analogică conține convertorul analogic-digital (ADC), un timer A/D timer ce generează pulsurile de trigger ale fiecărei citiri, o memorie FIFO pe post de buffer care are rol de tampon de date între două citiri ale calculatorului și un registru de stare ce păstrează informații referitoare la operația de citire.

Secvența de operații ce are loc pentru fiecare citire este următoarea:

1. Timerul generează un puls ce este trimis ADC-ului
2. ADC-ul execută o conversie
3. Valoarea digitală rezultată din conversie este depusă în buffer-ul FIFO
4. Bitul de stare numit DVAL din registrul de stare este activat

Programul din calculator care inițiază un transfer trebuie să conțină următoarea secvență de instrucțiuni:

1. Se citește registrul de stare
2. Dacă DVAL este activ se copiază conținutul FIFO în memoria calculatorului altfel dacă DVAL nu este activ se repetă algoritmul începând cu punctul 1

Perioada de timp dintre inițierea unei conversii și momentul când DVAL este activ este singurul moment care poate fi precizat în transferul POOLING. Operațiile efectuate de CPU au durate variabile, în funcție de ceasul CPU, gradul de încărcare al CPU etc. convertorul ADC generează valorile digitale, la frecvența dorită, și le depune în FIFO. Plăcile de achiziție pot avea buffere FIFO cu mărimi între 16 și 2048 locații.

Dacă bufferul este plin, și o nouă conversie are loc înainte ca bufferul să fie citit de CPU în registrul de stare se poziționează un bit de eroare. Această eroare nu este fatală, adică achiziția de date poate continua.

Principala limitare a transferului de date de tip pooling, este faptul că nu poate fi făcut la intervale precise de timp. CPU execută instrucțiunile în ordine secvențială dar cu întreruperi generate de sistemul de întreruperi în vederea deservirii și altor procese. Folosind deci pooling nu se poate conta pe transferul datelor din FIFO în memorie la intervale precizate de timp, ceea ce poate duce la pierderea de valori din semnalul cules.

Transferul de tip pooling, este ușor de implementat la nivelul registrilor plăcii, și poate fi folosit în operații de achiziție lente (<30 KHz), în care operațiile executate asupra datelor culese nu sunt critice, timpul culegerii de date nefiind important, iar punctele de semnal sunt culese și procesate individual.

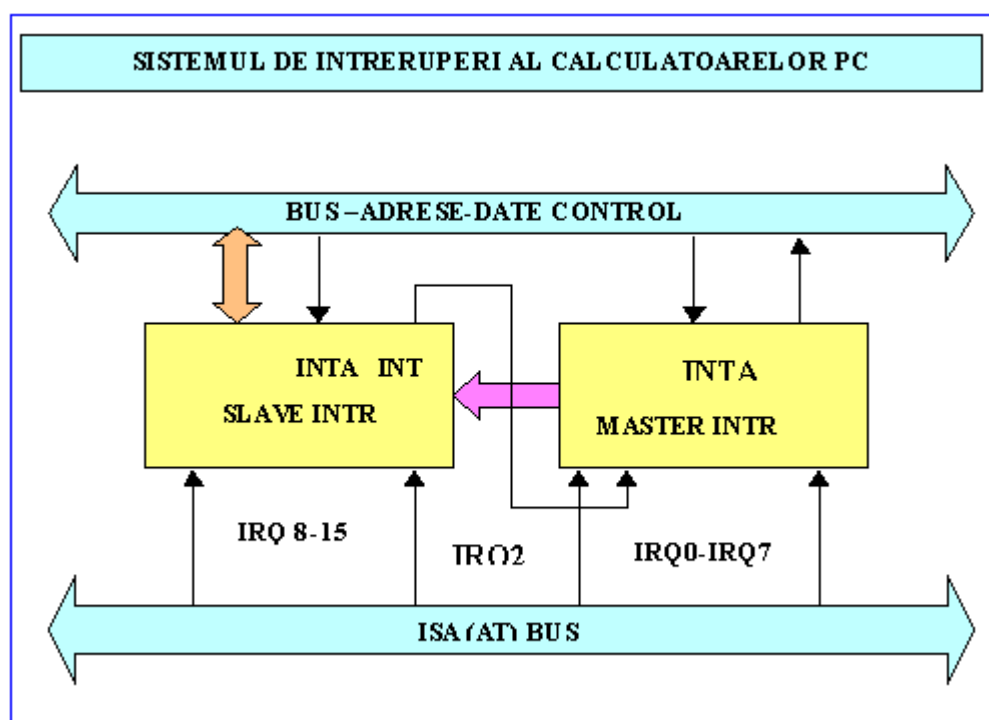
Intreruperi

Arhitectura calculatorului PC este astfel concepută încât procesorul să poată deservi toate resursele calculatorului. Acest tip de arhitectură se bazează pe utilizarea întreruperilor. Întreruperile sunt semnale hardware care provin de la periferice și care instiintează procesorul că trebuie să deservească un periferic (o anumită resursă).

Calculatoarele PC dispun de un controler de întrerupere programabil care furnizează 16 linii de întrerupere IRQ0-IRQ15. Întreruperile IRQ8-IRQ15 sunt prioritare liniilor IRQ3-IRQ7. În urma activării unei linii de întrerupere, controlerul de întreruperi generează spre procesor un semnal numit INT. În momentul când procesorul poate lua în considerare întreruperea, își salvează starea curentă pe stivă după care răspunde cu un semnal INTA pentru a anunța controlerul de întreruperi că este gata să trateze întreruperea în curs. Controlerul de întreruperi furnizează vectorul de întrerupere corespunzător întreruperii, iar procesorul pe baza vectorului furnizat găsește în tabela de întreruperi adresa rutinei de tratare a întreruperii și lansează în

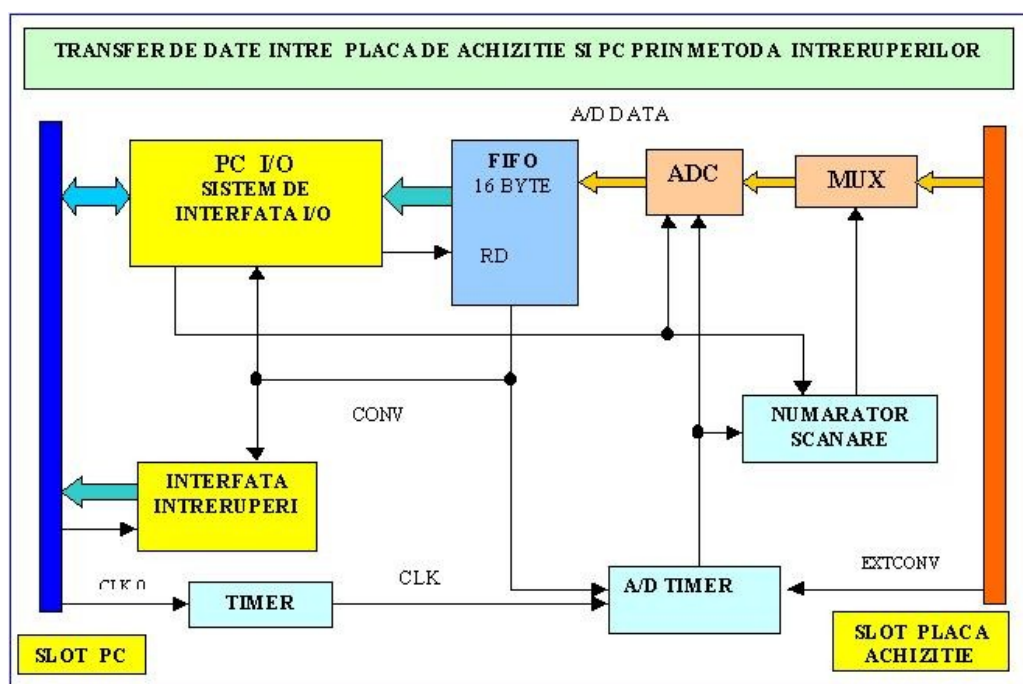
executie rutina respectiva, dupa care revine in procesul initial prin citirea starii de pe stiva si reluarea procesului exact din momentul in care a fost intrerupt.

In figura de jos este schitat sitemul hardware de intreruperi al unui PC:



Bazindu-se pe sistemul de intreruperi s-au conceput sisteme de achizitie care stabilesc legatura cu calculatorul prin intermediul intreruperilor. Sistemul de achizitie initiaza o achizitie de date. Dupa ce datele au fost achizitionate si stocate in buffere se lanseaza o cerere de intrerupere pe una din liniile de intrerupere. Procesorul trateaza aceasta intrerupere rulind o rutina specifica sistemului de achizitie prin care datele sunt citite din bufferul sistemului de achizitie si transfereate in memorie.

Schema bloc a unui astfel de sistem de achizitie ce utilizeaza intreruperile este prezentata mai jos:



Ori de cite ori convertorul ADC efectueaza o conversie si o depune in memoria FIFO se activeaza semnalul CONV care prin intermediul interfetei de intreruperi activeaza o linie de intreruperi a controlerului de intreruperi. In momentul cind intrerupera este acceptata de procesor acesta executa rutina de tratare a acestei intreruperi respectiv rutina de tratare a sistemului de achizitie care citeste defapt continutul FIFO in memorie prin intermediul sistemului de I/O.

In urma citirii, numaratorul de scanare este incrementat, furnizind o noua adresa pentru multiplexorul de intrare care va selecta o alta intrare analogica in vederea conversiei. Se reteaza in acelasi timp timerul A/D care initiaza un nou ciclu de conversie analogica.

In concluzie secventa de operatii pentru fiecare citire este urmatoarea:

1. Timerul A/D genereaza un puls ce determina convertorul A/D sa inceapa un ciclu de conversie
2. Convertorul A/D efectueaza conversia
3. Valoarea digitala este transferata in FIFO
4. Se lanseaza o intrerupere prin intermediul interfetei de intrerupere
5. Se accepta intreruperea si este lansata de procesor rutina de intrerupere
6. Rutina de intrerupere citeste datele prin I/O si le depune in memorie
7. Se revine din rutina de intrerupere
8. Se incrementeaza numaratorul scanner si Timerul ADC trimite un nou semnal ce declanseaza o noua operatie de conversie
9. Se reia ciclul

Prin intermediul sistemelor de achizitie bazate pe intreruperi viteza maxima ramane sub 50 KHz chiar si cu sisteme de achizitie performante care dispun de convertoare rapide din cauza timpului relativ mare intre doua acceptari de intrerupere ale procesorului. Se cistiga insa timp prin eliminarea sau cel putin reducerea factorului de intirziere din operatiile ce preced copierea bufferului FIFO in memoria RAM.

Intreruperea hardware, emisa de placa de achizitie, nu are prioritate maxima, deci acceptarea intreruperii poate sa intirzie daca alte periferice mai prioritare fac la rindul lor intreruperi. Cu toate acestea intreruperea sosita de la placa de achizitie in general este tratata imediat de procesor.

Transferul de date folosind intreruperi poate crea probleme daca executia rutinei de intrerupere ia prea mult timp sau daca procesorul consuma prea mult timp pentru acceptarea intreruperii.

In aceasta situatie se poate intimpla ca memoria FIFO sa genereze overflow, deci sa avem pierderi de date din cauza ca ritmul de citire al datelor este mai mic decit ritmul de conversie a datelor

Penrtu eliminarea acestor deficiente se utilizeaza o alta metoda de transfer, mult mai rapida si anume metoda de traqnsfer date prin transfer direct in memorie DMA-Direct Memory Acces .

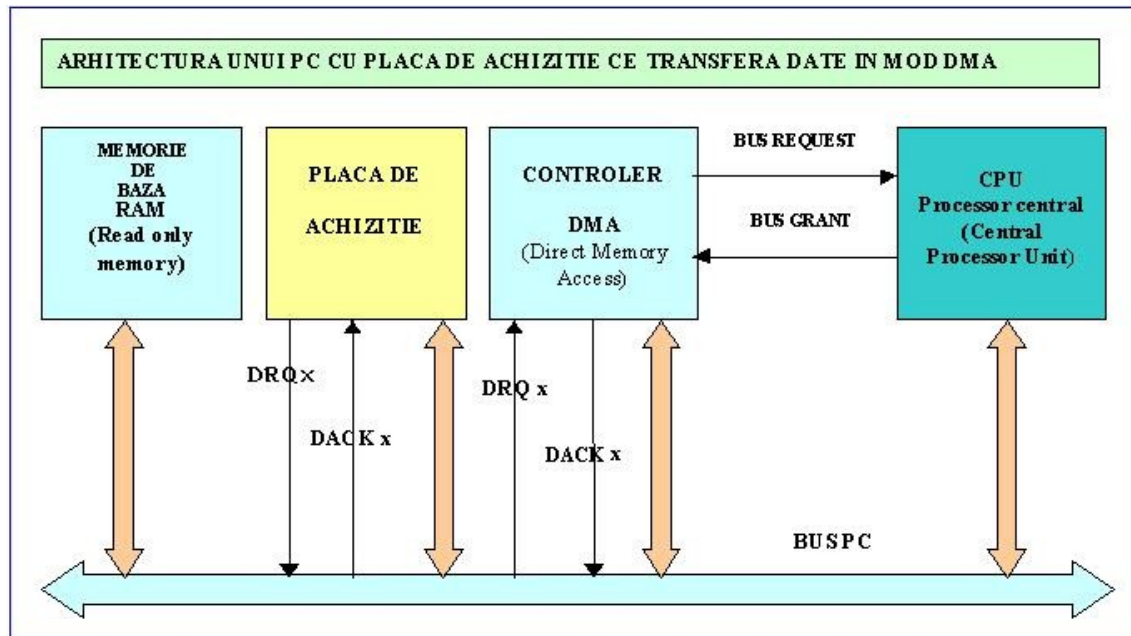
DMA-(Direct Memory Acces)

Mecanismul cel mai eficient, pentru transferul datelor intre placile de achizitie si memoria calculatorului il reprezinta sistemul DMA.

Metoda de transfer DMA este mai rapida decit metodele tratate anterior (poolong, intreruperi) deaoarece procesorul CPU nu este implicat in transferul de

date. Transferul este efectuat controlerul DMA al calculatorului in paralel cu activitatile curente ale procesorului CPU. Transferul de tip DMA este folosit in aplicatiile in care blocuri intregi de date trebuiesc transferate in memorie la viteze foarte mari.

In principiu arhitectura unui calculator care dispune de DMA si placa de achizitie date ce transfera datele prin intermediul DMA arata ca in schema bloc de mai jos.



Transferul de date bazat pe DMA, consta in programarea sistemului de achizitie in vederea utilizarii capabilitatilor hardware de a cere transfer DMA si programarea controerului DMA pentru a accepta cererile DMA de la sistemul de achizitie si de a transfera datele in zona de memorie programata.

Pentru a realiza un transfer DMA intre placa de achizitie si memorie se parcurg urmatoorii pasi.

1. Se programeaza controlerul DMA cu adresa de start ablocului de memorie urmeaza a fi umplut cu date, precum si numarul de locatii care vor fi transferate
2. Se programeaza placa de achizitie sa trimita semnalele de cerere de transfer, folosind linia DRQ x, ori de cite ori FIFO are date ce trebuiesc transferate.
3. Placa de achizitie trimite controolerului DMA un semnal DRQ x, semnal de cerere de transfer.
4. Controlerul DMA activeaza semmnalul Bus Request pentru a cere procesorului sa elibereze magistrala in vederea efectuarii transferului de date intre DMA si memorie prin intermediul BUS-ului comun.
5. Procesorul CPU cedeaza controlul asupra magistralei in favoarea sistemului DMA
6. Sistemul DMA activeaza semnalul DACK (acknowledge) pentru a anunta sistemul de achizitie ca cererea DMA a fost acceptata si poate incepe un transfer de date.DMA plaseaza pe bus-ul de adrese adresa de memorie din cadrul zonei de memorie in care se vor transfera datele,astfel incit octetul de date sa poata fi transferat de sistemul de achizitie direct in memorie.
7. Placa de achizitie scrie datele in memorie prin intermediul bus-ului PC si invalideaza semnalul DRQ

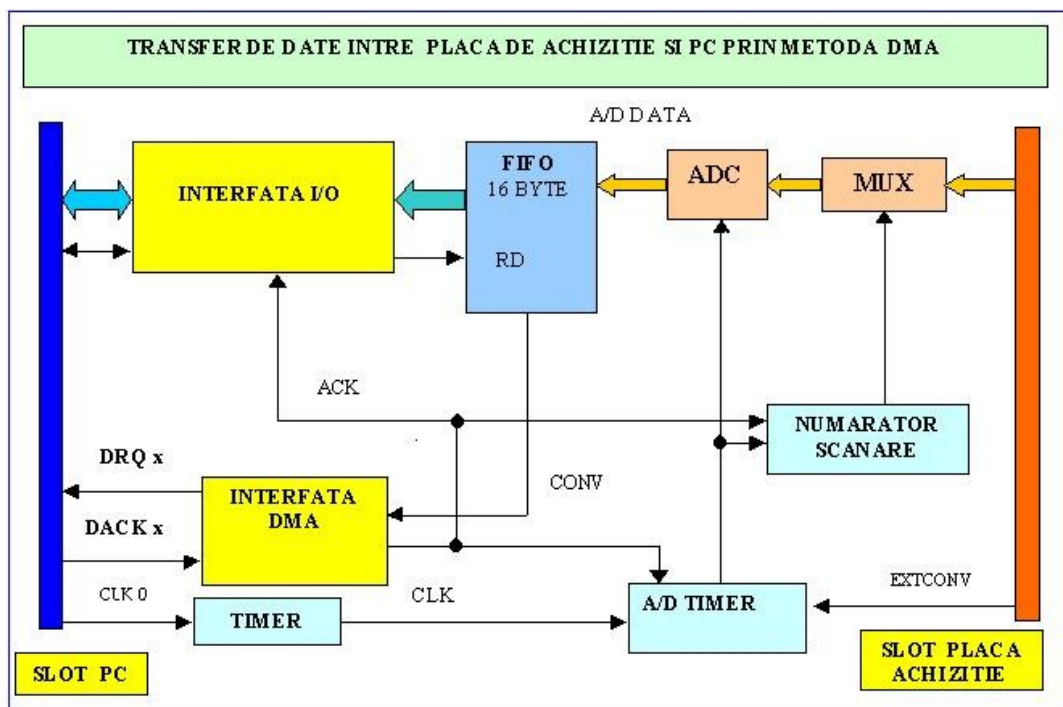
8. DMA formeaza o noua adresa pentru urmatorul octet de transferat si totodata numara octetii transferati din buff-erul FIFO al placii de achizitie, decrementind counterul care tine numarul de octeti care trebuie transferat
9. Se repeta algoritmul incepind cu punctul 3, pina cind counterul cu nr de octeti atinge valoarea 0, adica numarul de octeti programat la punctul 1 a fost deja transferat.

Exista mai multe moduri in care placa de achizitie activeaza semnalul DRQ . Astfel placa de achizitie poate cere transfer pentru fiecare octet de date din FIFO, caz in car avem de-a face cu un transfer DMA *-single mode*

Sistemul de achizitie poate fi programat in asa fel incit sa activeze semnalul DRQ pentru o perioada mai lunga pentru ca DMA sa poata face mai multe cicluri (3-9) deci sa transfere mai multe locatii pentru un singur DRQ. Acest mod de lucru cu DMA-ul se numeste DMA-*demand mode*.

Alta metoda de lucru este metoda DMA *-bloc transfer* prin care odata cererea de transfer acceptata, DRQ este pastrat activ pana la transferarea tuturor datelor din buff-er obligind deci DMA-ul sa tina bus-ul ocupat pana la transferul integral al datelor din FIFO

Sistemul de achizitie si transfer prin DMA poate fi reprezentat la nivel de schema bloc astfel:



Secventa de operatii care are loc pentru fiecare transfer DMA este urmatoarea:

1. Se programeaza canalul DMA corespunzator si placa de achizitie pentru a stabili numarul de octeti transferati si adresa de memorie unde vor fi transferate datele.
2. Un urma unei conversii datele ajung in FIFO, care activeaza semnalul CONV, iar prin interfata DMA se activeaza semnalul DRQ pe magistrala calculatorului. Controlerul de DMA face o cerere de acordare a magistralei spre procesorul CPU prin activarea semnalului BUSREQUEST. Procesorul acorda magistrala si semnalizeaza prin activarea semnalului BUSGRANT.
3. Controlerul de DMA activeaza semnalul DACK x pentru a semnaliza faptul ca sunt indeplinite coditiile pentru a se face transferul de date.
4. Se face transferul numarului stabilit de octeti la adresa stabilita.

5. Semnalul DACK x prin interfata DMA genereaza un puls ACK care este trimis interfetei I/O pentru a permite transmiterea datelor pe magistrala .Tot semnalul ACK incrementeaza counterul de scanare care stabileste o noua adresa pentru multiplexorul analogic de la intrare De asemenea ACK reseteaza timerul A/D care genereaza noi pulsuri pentru convertorul A/D
6. Convertorul efectueaza noi conversii
7. Valoarea digitala rezultata din conversie rezultata din conversie este depusa in FIFO
8. Se lanseaza din nou o cerere spre DMA
9. Se reia ciclul de la punctul 2 sau de la punctul 1 daca se schimba numarul sau locul de memorare al datelor.

Viteza de culegere si transfer a datelor in acest mod poate atinge frecventa de 300-500 KHz in functie de ceasul procesorului.Este unul din cele mai rapide procedee de achizitie date si se preteaza in locuri unde e nevoie de cantitate mare de date de achizitionat in timp real.

Un alt avantaj al folosirii procedeului DMA il reprezinta faptul ca in paralel cu transferul datelor procesorul poate rula anumite instructiuni care nu necesita magistrala,deci eficienta ansamblului calculator sistem de achizitie creste.Se pot crea aplicatii in care in care sa se execute operatii simultane de genul achizitie transfer prelucrare prezentare, cu alte cuvinte chizitie,analiza si prezentare in timp real.

Lungimea maxima a unui bloc de date transferat nu poate depasi 64 Ko din cauza limitarilor sistemului DMA.Dupa un astfel de transfer DMA-ul trebuie reprogramat cu o noua adresa de ransfer date si cu noua dimensiune a blocului de transferat.

Din cauza faptului ca memoria unui calculator trebuie refresh-ata la fiecare 15 microsecunde si deci trebuie eliberat bus-ul la fiecare 15 microsecunde, metodele de transfer DMA de tip demand sau blok sunt aproape imposibil de implementat.