ARHITECTURA CALCULATOARELOR 2003/2004
CURSUL 10

4.1.4 Ceasuri (semnale de tact)

În majoritatea circuitelor digitale ordinea în care au loc evenimentele este critică. Uneori un eveniment trebuie să preceadă un altul, uneori două evenimente trebuie să aibă loc simultan. Pentru a permite proiectanților să obțină relațiile de temporizare necesare, multe circuite digitale folosesc ceasuri pentru a oferi sincronizarea.

Un ceas în acest context este un circuit care emite o serie de impulsuri cu un interval de timp exact între impulsuri consecutive. Intervalul de timp între fronturile corespondente a două impulsuri consecutive este denumit ciclu de ceas (perioadă - "clock cycle time"). Pentru frecvențele impulsurilor cuprinse între 1 şi 500 MHZ, corespund perioade între 1000 nsec şi 2 nsec. La 1 GHz perioada este de 1 nsec ş.a.m.d. Pentru a atinge o acuratețe ridicată, frecvența de ceas este controlată de un oscilator cu cristal de cuarț.

Într-un computer, multe evenimente pot avea loc în timpul unei singure perioade de ceas. Dacă aceste evenimente trebuie să respecte o anumită ordine, ciclul de ceas trebuie să fie divizat în subcicluri. Un mod obişnuit de obținere a unei rezoluții mai fine decât ceasul de bază este să se deschidă linia ceasului primar şi să se insereze un circuit cu o intârziere cunoscută, generând astfel un semnal secundar de ceas care este defazat față de cel primar, după cum este prezentat în figura 4.6.

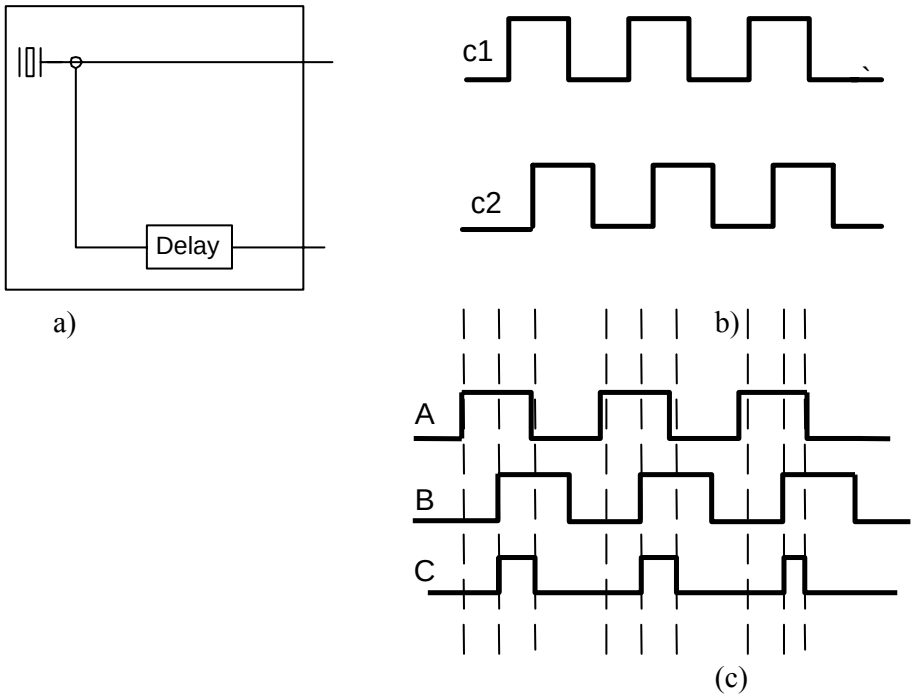

Figura 4.6 (a) Ceas (b) Diagrama de timp pentru un ceas. (c) Generarea unui tact asimetric.

Diagrama de timp din figura 4.6b prezintă 4 referințe de timp pentru evenimente discrete :

1. Frontul crescător al lui c_1,
2. Frontul descrescător al lui c_1,
3. Frontul crescător al lui c_2,
4. Frontul descrescător al lui c_2.

Asociind diferite evenimente la diferite fronturi se poate obține secvențierea necesară. Dacă e nevoie de mai mult de 4 referințe de timp într-un ciclu de ceas, pot fi extrase mai multe linii secundare din cea primară cu diferite intârzieri.

În unele circuite se poate să intereseze mai curând intervalele de timp decât momentele de timp discrete. De exemplu, unele evenimente pot fi acceptate oricând c_1 este în starea HIGH, nu în timpul frontului crescător. Un alt eveniment poate avea loc când c_2 este HIGH. Dacă este nevoie de mai mult de 2 intervale pot fi oferite mai multe linii de ceas sau stările HIGH ale celor 2 ceasuri pot fi făcute să se suprapună parțial în timp. În ultimul caz, patru intervale discrete pot fi deosebite: $\overline{C_1}$ AND $\overline{C_2}$, $\overline{C_1}$ AND C_2, C_1 AND $\overline{C_2}$, C_1 AND C_2.

Cum s-a putut remarca, ceasurile de mai sus sunt simetrice, cu timpul petrecut în starea HIGH egal cu timpul în starea LOW după cum se vede în figura 4.6b. Pentru a genera un tren de impulsuri asimetrice, ceasul de bază este deplasat utilizând un circuit de întârziere şi apoi AND cu semnalul original, ca în figura 4.6c semnalul C.

Observatie:
În continuare ar trebui studiate circuitele de memorie: latch-uri (de exemplu RS sincrone, D sincrone), bistabile, registre, organizarea memoriei, cipuri de memorie, RAM, ROM. Unele noțiuni de bază au fost prezentate anterior şi sunt aprofundate la alte cursuri.

4.2 Circuitele CPU si magistrale

Înarmați cu informație despre cipurile numerice şi cipurile de memorie putem începe să asamblăm toate piesele. În această secțiune vom acorda atenție mai intâi aspectelor generale ale procesorului văzut din perspectiva nivelului logic numeric, incluzând pinii de ieşire (ce semnal, pe ce pin, cu ce semnificație). Din moment ce procesoarele sunt strâns legate de proiectarea magistralelor folosite, vom face şi introducerea magistralelor.

4.2.1 Circuitele CPU

Toate CPU-urile moderne conțin un singur cip. Acest lucru conduce la o comunicare bine definită cu restul sistemului. Fiecare procesor are un set de pini, prin care comunică cu restul sistemului. Uni pini dau semnale emise de procesor, alți pini primesc semnale emise de la celelalte componente ale sistemului, alți pini pot face amble sarcini. Înțelegând cum funcționează aceşti pini, putem învăța cum interacționează procesorul cu memoria şi unitățile de intrare şi ieşire.

Pinii de pe procesor pot fi împărțiti în trei categorii:

- Adrese,
- Date,
- Control.

Aceşti pini sunt conectați la pini similari de la memorie şi de la unitățile de intrare/ieşire printr-un ansamblu de fire şi trasee electrice numite magistrale. Pentru a prelucra o instrucțiune procesorul mai întâi pune adresa locației de memorie care conține instrucțiunea pe pinii săi de adresă. Atunci activează încă o linie de control pentru a informa memoria că vrea sa citească, de exemplu, un cuvânt. Memoria îi răspunde activând un semnal spunându-i că aceasta s-a facut. Când procesorul primeşte semnalul, accecptă cuvântul şi prelucrează instrucțiunea.

Instrucțiunea poate cere să scrie sau să citească mai multe cuvinte de date, caz în care se repetă intregul proces. Pentru moment, important este ca să înțelegem că procesorul comunică cu memoria şi unitățile de intrare/ieşire prezentând semnalele pe pinii săi şi acceptând semnale pe pinii săi. Nici o altă comunicare nu este posibilă.

Doi dintre parametrii care determină performanța unui procesor sunt numărul de pini de adresă şi numărul de pini de date. Un cip cu m pini de adresă poate adresa direct până la 2^m locații de memorie. Valorile obişnuite pentru m sunt 16, 20, 32 si 64. Similar un cip cu n pini de date poate citi sau scrie un n-cuvant într-o singură operație. Valorile obişnuite sunt 8, 16, 32, 64. Un procesor cu 8 pini de date va face 4 operații pentru a citi un cuvânt pe 32 de biți, pe când un procesor cu 32 pini de date poate face aceeaşi lucru într-o singură operație. Astfel cipul cu 32 pini de date este mult mai rapid, dar este bineînțeles mai scump. Fiecare procesor are câțiva pini de control. Aceşti pini controlează şi sincronizează fluxul de date de la şi spre procesor. Au şi alte utilizări. Toate procesoarele au pini pentru alimentare (de obicei +3.3 volți sau +5 volți), masă şi un semnal de ceas, dar ceilalți pini variază de la cip la cip. Totuşi pinii pot fi grupați în următoarele categorii, ca în figura 4.7:

1. Controlul magistralei,
2. Întreruperi,
3. Arbitrarea magistralei,
4. Semnalele coprocesorului,
5. Stare,
6. Diverse.

Majoritatea pinilor oferă ieşiri ale CPU către magistrală (intrăile memoriei ş ale dispozitivelor de I/O), adică semnale care transmit dacă CPU vrea să scrie sau să citească în sau din memorie sau să facă altceva. Unitatea CPU foloseşte aceşti pini pentru a controla restul sistemului şi pentru a-i comunica ceea ce doreşte să facă.

Pinii de întrerupere sunt intrări de la componentele I/O către CPU. În majoritatea sistemelor, unitatea CPU poate comunica unei componente I/O să pornească operațiunea şi apoi să iasă din funcțiune relativ la această operație şi să facă ceva folositor în timp ce componenta I/O, cu viteza de lucru mai mică, işi execută sarcina. Când operațiunea I/O a fost terminată, controlerul chipului I/O transmite un semnal unuia dintre aceşti pini să întrerupă unitatea CPU pentru ca aceasta să servească componenta I/O. De exemplu, să verifice dacă a intrervenit vreo eroare I/O. Unele procesoare au un pin de ieşire pentru a semnala acceptarea semnalului de întrerupere.

Pinii de arbitrare ai magistralei sunt folosiți pentru a reglementa traficul pe magistrală pentru a preveni două componente să încerce să o folosească în acelaşi timp. Din punct de vedere al acestei arbitrări, CPU este considerat componentă.

Unele CPU sunt proiectate să opereze cu coprocesoare, cum sunt cipurile în virgulă mobilă, şi uneori chiar şi cu cipuri grafice sau alte tipuri de cipuri specializate. Pentru a facilita comunicarea dintre procesor şi coprocesor, au fost proiectați pini speciali pentru a procesa diferite operațiuni. Pe lângă aceste semnale, există şi alți diverşi pini la procesoare. Unii dintre aceştia furnizează sau acceptă informații de stare, alții sunt folosiți pentru resetarea calculatorului, şi alții pentru asigurarea compatibilității cu alte cipuri I/O.

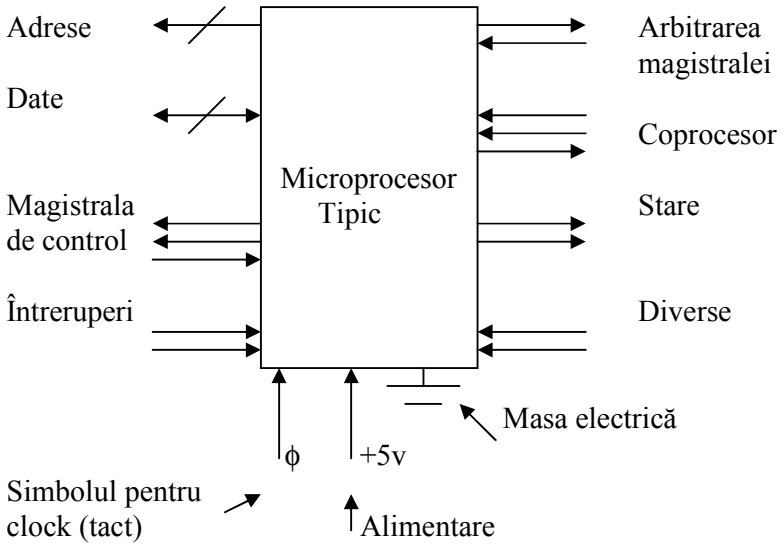

Figura 4.7 Clase de pini logici pentru un procesor tipic

În figura 4.7 săgețile indică semnalele de intrare şi semnalele de ieşire. Liniile diagonale mici indică, că sunt folosiți pinii multipli. Unui CPU specificat i se va da un număr pentru a spune câte linii există.

4.2.2 Magistrale de calculator

O magistrală este un traseu electric între mai multe componente. Magistralele pot fi categorisite după funcțiile pe care le îndeplinesc. Pot fi interne față de procesor pentru a transmite date către şi dinspre ALU, sau externe față de procesor, pentru a-l conecta cu memoria sau componentele I/O. Fiecare tip de bus are prorpiile sale necesități şi proprietăți. În această secțiune, cât şi în următoarele, ne vom concentra pe magistralele care fac legătura între procesor şi componentele I/O.

PC-urile mai vechi aveau o singură magistrală externă sau magistala de sistem. Aceasta consta din 50 până la 100 de fire paralele de cupru imprimate pe placa de bază, cu conectori dispuşi la intervale regulate pentru conectarea memoriei şi a plăcilor I/O. PC-urile moderne au în general o magistrală specială între CPU şi memorie, şi cel puțin o altă magistrală pentru componentele I/O. Un sistem minim, cu o magistrală de memorie şi o magistrală I/O, e ilustrat în figura 4.8.

În literatură, magistralele sunt câteodată desenate cu săgeți groase, precum în figura 4.8. Diferența, dintre o săgeată groasă şi o linie simplă cu o linie diagonală peste ea şi un număr de biți, lângă este subtilă. Când toți biții sunt de acelaşi tip, toți biții de adresă sau toți biții de date, atunci linia diagonală scurtă de acces este des folosită. Când sunt şi adrese şi date şi linii de control implicate, o sageată groasă este des folosită.

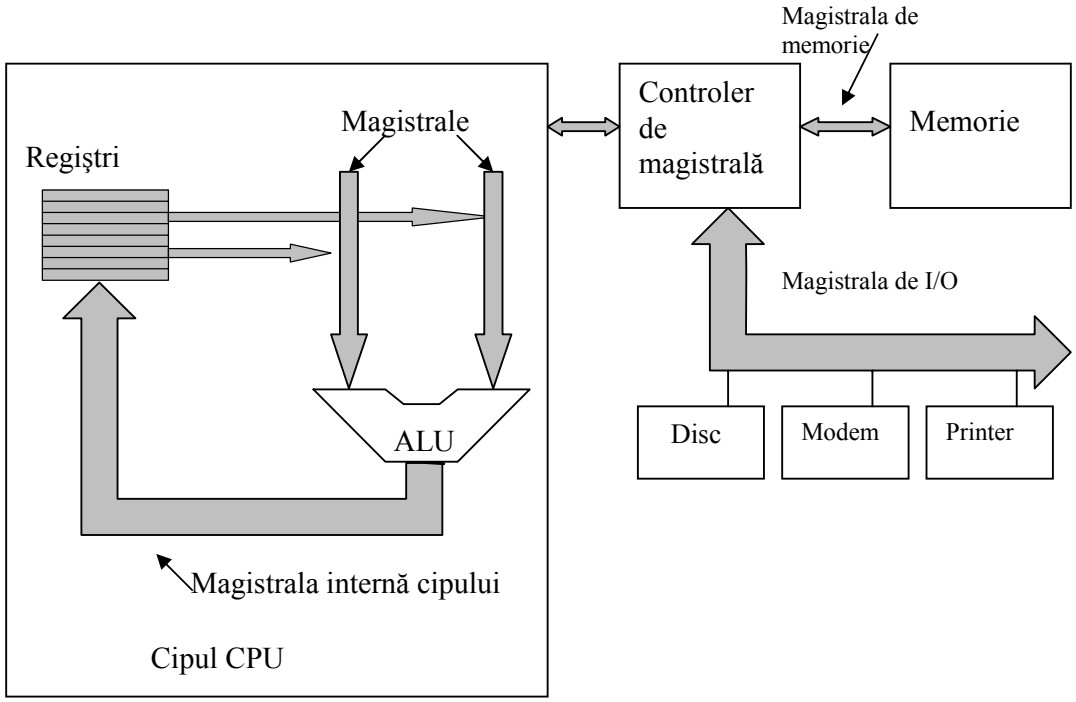

Figura 4.8 Un computer cu mai multe magistrale

Proiectanții unității CPU sunt liberi să folosească orice tip de magistrală doresc în interiorul cipului. Pentru a face posibil ca plăci proiectate de o a treia parte să fie ataşate la magistrala de sistem, trebuie elaborate reguli bine definite despre cum lucrează magistrala şi care trebuie respectate de toate componentele ataşate.

Aceste reguli sunt numite protocol de magistrală. În plus, trebuie să aibă şi specificații electrice şi mecanice, astfel încât plăcile altor fabricanți să se potrivească în carcasă şi să aibă conectori care se potrivesc cu cei de pe placa de bază din punct de vedere mecanic, de voltaj şi de sincronizare în timp.

Un numar de magistrale sunt în uz larg răspândit în lumea computerelor. Câteva mai bine cunoscute, curente şi istorice sunt Omnibus (PDP-8), Unibus (PDP-11), Multibus (8086), IBM PC bus (PC/XT), ISA bus (PC/AT), EISA bus (80386), Microchannel (PS/2), PCI bus (multe PC-uri), SCSI bus (multe PC-uri si statii de lucru), Nubus (Macintosh), Universal Serial Bus (PC-uri moderne), FireWire (electronice de consum), VME bus (echipamente pentru laboratoare de fizică), şi Camac bus (fizică energetică). Probabil ca lumea ar fi un loc mai bun dacă toate în afară de una ar dispare de pe fața pământului (bine, perfect, măcar toate în afară de doua?). Din păcate, standardizarea în acest domeniu pare foarte nepotrivită deoarece deja s-a investit prea mult în toate aceste sisteme incompatibile.

Să începem studiul asupra felului de funcţionare al magistralelor. Unele componente care se ataşează la o magistrală sunt active şi pot iniţia transferuri pe magistrală, în timp ce altele sunt pasive şi aşteaptă o apelare. Cele active sunt numite "master", iar cele pasive sunt numite "slave".

Când unitatea CPU cere unui controler de disc să citească sau să scrie un bloc de informaţie, unitatea CPU funcţionează ca "master" şi controlerul de disc funcţionează ca "slave". Oricum, mai târziu controlerul de disc poate funcţiona ca "master" atunci când comandă memoria să accepte datele pe care le citeşte de pe disk . Câteva combinaţii tipice de "master" şi "slave" sunt listate în figura 4.9. Memoria nu poate fi în niciun fel de condiţii "master" vreodată.

MASTER	SLAVE	EXEMPLE
CPU	Memorie	Prelucrarea (fetching) instrucţiunilor şi datelor
CPU	Dispozitiv I/O	Iniţializarea transferului de date
CPU	Coprocesor	CPU transmite instrucţiunile pentru coprocesor
I/O	Memorie	DMA (acces direct la memorie)
Coprocesor	CPU	Coprocesorul prelucrează operanzii de la CPU

Figura 4.9 Exemple de “master” şi “slave” pe magistrale.

Semnalele binare pe care componentele computerului le emit sunt frecvent insuficient de puternice, din punct de vedere electric, să alimenteze o magistrală, mai ales dacă aceasta este destul de lungă sau deserveşte multe componente. Din acest motiv, majoritatea magistralelor "master" sunt conectate la magistrala printr-un cip numit "bus driver", care în esenţă este un amplificator numeric. Similar, majoritatea magistralelor "slave" sunt conectate la magistrală printr-un cip numit "bus receiver". Pentru componentele care pot funcţiona şi ca "master" şi ca "slave" un cip combinat numit "bus transceiver" este folosit pentru a face legatura între magistrale şi magistrala principală. Cipurile interfaţă de magistrală sunt deseori componente cu trei stări (tri state), pentru a le permite deconectarea atunci când nu sunt utilizate, sau sunt blocate într-un mod, numit colector deschis (open collector), cu efect similar. Când două sau mai multe componente pe un linia unui colector deschis activează linia în acelaşi timp, rezultatul este operaţiunea logică "OR" a tuturor semnalelor. Acest aranjament este deseori numit “SAU cablat”. Pe majoritatea magistralelor, unele linii sunt cu trei stări, iar altele, care necesită proprietatea "SAU cablat" sunt cu colector deschis.

Ca şi CPU, magistrala are deasemenea adrese, date şi linii de control. Oricum, nu este necesară o corespondenţă unu la unu între pinii CPU şi semnalele magistralei. De exemplu, unele unităţi CPU au trei pini care codează oricare din operaţiile de citire sau scriere din/în memorie, scriere sau citire la/de la I/O, sau o altă operaţie. O magistrală obişnuită poate avea o linie pentru citire din memorie, o a doua linie pentru scriere în memorie, o a treia pentru citire I/O, o a patra pentru scriere I/O, şi aşa mai departe. Un decodificator ar fi atunci necesar între CPU şi o astfel de magistrală pentru a face legatura între cele două părţi, adică să convertească semnalul codat pe 3 biţi în semnale separate care pot conduce liniile magistralei.

Caracteristicile majore în proiectarea magistralelor sunt:

- lăţimea magistralei (bus width),
- sincronizarea magistralei (bus clocking),
- arbitrarea magistralei (bus arbitration),
- operaţiile magistralei (bus operations).

Fiecare dintre acestea are un impact substanțial asupra vitezei (speed) şi lățimii de bandă (bandwidth) a magistralelor. Le vom examina pe toate acestea în continuare.

4.2.3 Lățimea de bandă a magistralei

Lățimea magistralei este cel mai evident parametru de proiectare. Cu cât mai multe linii de adresă are magistrala, cu atât mai multă memorie poate adresa direct unitatea CPU. Dacă o magistrală are *n* linii de adresă, atunci un CPU poate folosi până la 2^n locații de memorie diferite. Pentru a permite utilizarea de memorii mari, magistralele necesită multe linii de adrese.

Problema constă în faptul că magistralele mari necesită mai multe fire decât cele mici. De asemenea, acestea necesită mai mult spațiu fizic (de exemplu pe placa de bază) şi conectori mai mari. Toți aceşti factori cresc prețul unei magistrale. Astfel avem o relație directă între marimea maximă de memorie şi prețul sistemului. Un sistem cu o magistrală cu 64 de linii de adresă şi 2^{32} octeți de memorie va costa mai mult decât una cu 32 de linii de adrese şi aceiaşi 2^{32} octeşi de memorie. Posibilitatea de mărire ulterioară a capacității memoriei va fi şi ea costisitoare. Rezultatul acestui fapt este că mulți proiectanți de sisteme tind să rămâna la o viziune îngustă asupra problemei, ceea ce duce la consecințe nedorite ulterior. PC-ul original IBM avea un CPU 8088 şi o magistrală de adrese pe 20 de biți, precum în figura 4.10a. Având 20 de biți permitea PC-ului să adreseze direct 1 MB de memorie.

Când CPU-ul următor (80286) a apărut, INTEL a decis să mărească spațiul de memorie adresabil la 16 MB, aşa că încă 4 linii magistrale au fost adăugate (fără a modifica pe celelalte 20 din motive de compatibilitate cu modelele anterioare), precum în figura 4.10b. Din nefericire, mai multe linii de control au trebuit adăugate pentru noile linii de adrese. La apariția lui 80386 alte 8 linii au fost adăugate, împreună cu alte noi linii de control precum în figura 4.10c. Rezultatul proiectării a fost că magistrala EISA a fost mult mai complicată decât dacă i s-ar fi dat cele 32 de linii de la început. Nu numai numărul de linii de adresă tinde să crească în timp, ci şi numărul de linii de date, dar dintr-un alt motiv.

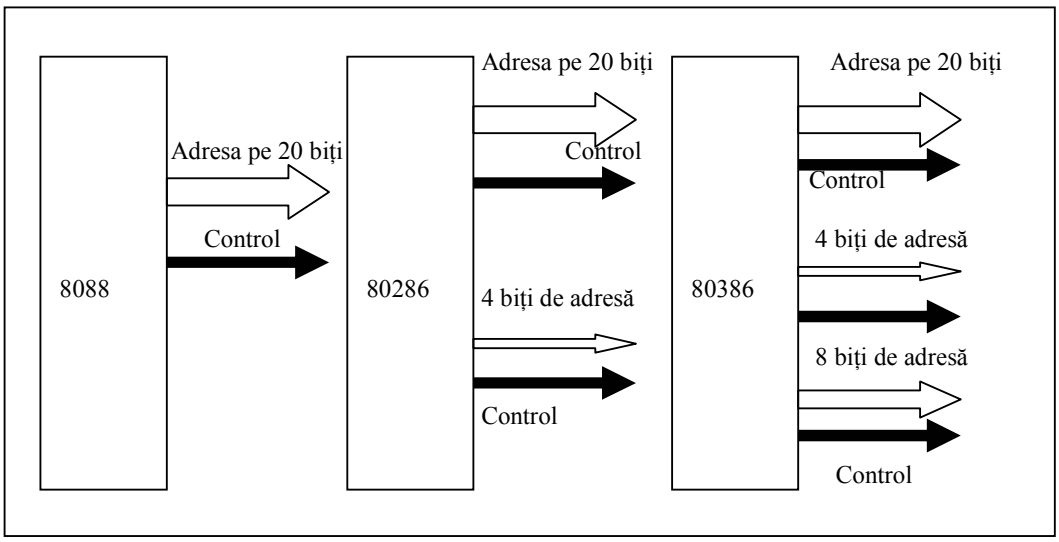

Figura 4.10 Creşterea adreselor magistralelor de-alungul timpului

Sunt două moduri de a mări lățimea de bandă a unei magistrale de date:

- micşorarea timpului de ciclu (mai multe transferuri într-o secundă) sau
- mărirea lățimii magistralei de date (mai mulți biți în cadrul unui singur transfer).

Nu numai numărul de linii de adresă tinde să crească în timp, ci şi numărul de linii de date, dar dintr-un alt motiv.

Mărirea vitezei magistralei este posibilă, dar dificilă deoarece semnalele pe linii diferite au vitezei diferite, problemă cunoscută sub numele de "desincronizarea magistralei". Cu cât magistrala este mai rapidă, cu atât este mai mare posibilitatea de desincronizare.

O altă problemă este că, odată cu marirea vitezei magistralei, se pierde compatibilitatea cu modelele anterioare. Plăcile vechi proiectate pentru magistrale cu viteza mică nu vor funcționa cu cele noi. Invalidând plăcile vechi face ca atât deținătorii de plăci vechi, cât şi producătorii acestor plăci să fie puşi într-o situație delicată. De aceea cel mai uşor mod de a îmbunătăți performanța magistralei este de a mări numărul de linii de adrese, ca în figura 4.10. Cum ne aşteptam, oricum această mărire a performanței nu duce la o proiectare simplă. IBM PC şi urmaşii săi, de exemplu, au pornit de la 8 linii şi au trecut la 16 şi apoi la 32 de linii de adrese pe acelaşi tip de magistrală.

Pentru a rezolva problema magistralelor foarte mari, uneori proiectanții au optat pentru magistralele multiplexate. În acest proiect, înloc de a avea adresa şi liniile de date separate, sunt 32 de linii de adrese şi date împreună. La începutul unei operațiuni pe magistrală, liniile sunt folosite pentru adrese. Apoi, la alt moment de timp, aceleaşi linii sunt folosite pentru date. Pentru o operațiune de scriere în memorie, de exemplu, înseamnă că liniile de adresă trebuie să fie setate şi apoi propagate către memorie înainte ca datele să poată fi puse pe magistrală. Cu linii separate, adresa şi datele pot fi puse în acelaşi timp. Multiplexând liniile se reduce lățimea de banda (şi costurile), dar rezultă un sistem mai lent. Proiectanții de magistrale trebuie să cântărească cu grijă toate aceste opțiuni înainte de a face o alegere.