

## CURS 4

# FUNDAMENTE PRIVIND ARHITECTURA UNITĂȚII CENTRALE DE PROCESARE

### STRUCTURA DE PROCESOR

Unitatea Centrală de Prelucrare (UCP) a calculatorului numeric este o structură de procesor de uz general, cu set de instrucțiuni. UCP este un procesor *de uz general*, spre deosebire de alte procesoare cu set de instrucțiuni, cum ar fi procesoarele de I/O, sau procesoarele aritmetice, care îndeplinesc doar funcții limitate, specifice. De asemenea UCP este un procesor *cu set de instrucțiuni*, pentru că UCP recunoaște și execută un set specific de instrucțiuni, furnizate din exteriorul său, sub formă binară.

În cadrul unui sistem de calcul, UCP are responsabilitatea generală de *interpretare* și *execuție* a instrucțiunilor unui program.

În accepțiunea clasică (von Neumann) a unui sistem de calcul, mașina conține o singură Unitate Centrală de Prelucrare. Un astfel de calculator este numit *uniprocessor*, pentru a-l deosebi de calculatoarele *multiprocessor*, care conțin două sau mai multe UCP.

Cele mai importante considerații privind arhitectura și organizarea unei UCP se referă la următoarele:

- Tipurile de instrucțiuni, ce formează setul de instrucțiuni al UCP;
- Organizarea la nivel de registru a unității aritmetice și logice;
- Organizarea la nivel de registru a unității de control a UCP;
- Modul în care UCP comunică cu dispozitivele externe.

Ca structură generală, un procesor cuprinde o unitate de prelucrare a datelor (care conține o unitate aritmetică și logică și registrele folosite ca memorie locală) și o unitate de control. Conform modalității de clasificare a circuitelor digitale în ordine, fiecare dintre cele două componente amintite se constituie ca un automat cu ordinul cel puțin egal cu 2. Structura procesorului, presupune o buclă de reacție între cele două automate componente, deci procesorul este, conform clasificării în ordine descrisă în capitolul 1, un automat de ordinul 3.

Unitatea Aritmetică și Logică (ALU), efectuează operații cu datele de intrare (operanzi) și conține pe lângă structurile combinatoriale pentru efectuarea acestor operații, circuite de tip registru pentru memorarea locală, sau prelucrarea datelor și circuite necesare transferului de informație între registre și cu exteriorul ALU (circuite de decodificare, codificare, multiplexare, comparare, circuite tampon, etc.). Registrele locale folosite ca memorie de mare viteză, pot fi folosite și pentru efectuarea unor operații aritmetice sau logice. Principalele operații realizate de Unitatea Aritmetică și Logică (ALU) sunt:

- operații aritmetice și logice cu operanzi de intrare codificați pe  $n$  biți. Valoarea lui  $n$  este egală de obicei cu numărul de biți ai magistralei interne de date a procesorului, dar există și excepții de la aceasta regulă.
- furnizează indicații privind transportul, împrumutul, sau alte caracteristici privind rezultatul operațiilor efectuate, prin intermediul unor valori binare numite indicatori de condiții (F1 = flags).

În figura 3.1 datele de intrare în unitatea de prelucrare a datelor sunt intrările pentru operanzii citiți din exterior. Acești operanzi se citesc ca și cuvinte binare (deci cu  $n$  biți în paralel) și se stochează în registre temporare, interne unității de prelucrare a datelor. Rezultatul operațiilor efectuate, sau conținutul unor registre de stocare temporară, se poate furniza în exterior prin liniile notate "date ieșire".

Intrările în unitatea de prelucrare notate "comenzi interne procesorului", reprezintă comenzi către ALU generate de unitatea de control. Conform acestor comenzi ALU va prelucra datele de la intrare și va furniza datele la ieșire (registre interne sau externe). Aceste comenzi (în formă binară) pot reprezenta:

- codul operației pe care trebuie să o execute ALU;
- informații de adresare a registrelor interne unității de prelucrare, (de exemplu pentru a se indica registrele sursă și / sau destinație ale operațiilor efectuate de ALU, pentru a se selecta / valida registre tampon de memorare a valorilor binare de la "Intrare date" sau la "Ieșire date");
- semnale de sincronizare (destinată logicii secvențiale interne a unității de prelucrare); aceste semnale indică momentele de timp când se face transferul între registre, respectiv momentele de încărcare a unor registre.

*Indicatorii de condiții* sunt poziționați de unitatea de prelucrare în conformitate cu "evenimentele" ce s-au produs la efectuarea operației comandate anterior pentru ALU. Acești indicatori (numiți și fanioane) sunt memorați de obicei în bistabile, incluse ca celule ale unor registre de indicatori, registre interne UCP. Indicatorii de condiții pot reprezenta, de exemplu, semnul rezultatului, rezultat zero, paritatea rezultatului etc.

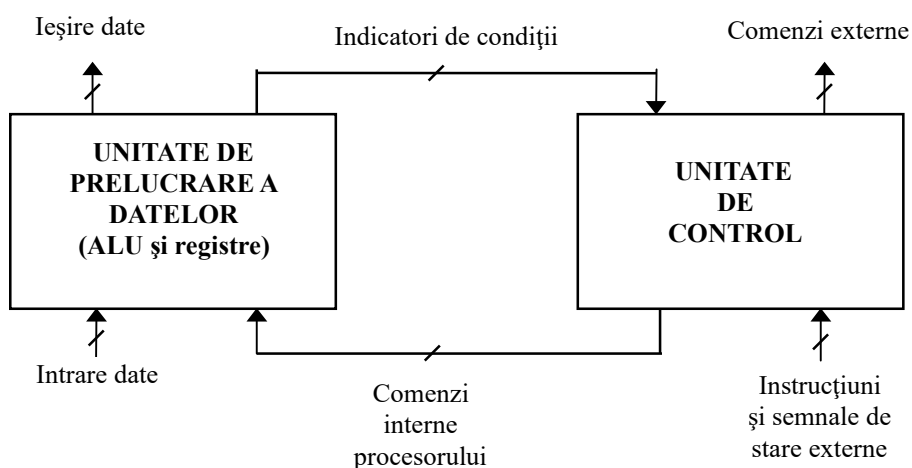


Figura 3.1. Structura de procesor, ca automat de ordinul 3. (liniile oblice pe căile de intrare și ieșire a semnalelor indică existența mai multor linii de semnal, simbolizate prin linie simplă în figură)

Sucesiunea de comenzi primită de ALU este generată de către un automat cu algoritm de stare încorporat, numit *Unitate de Control* (UC). Algoritmul de stare definește succesiunea stărilor automatului, respectiv succesiunea și natura comenzilor. Algoritmul de stare poate fi implementat fie în structura fizică (hardware) a UC, fie prin înscrierea comenzilor binare secvențiale într-o memorie de control de mare viteză. În primul caz vorbim de unitate de *control cablată*, iar în al doilea caz de unitate de control *microprogramată*.

*Sistemul digital format dintr-un automat de prelucrare aritmetică și logică și o unitate de control este numit procesor.*

Datorită reacției pe care UC o primește de la Unitatea de prelucrare, UC examinează caracteristicile rezultatelor (prin testarea valorii indicatorilor de condiții) și astfel poate modifica secvența de comenzi în mod corespunzător. UC primește din exterior "instrucțiuni", fiecare din acestea producând generarea unei secvențe de comenzi elementare către unitatea de prelucrare și către exteriorul procesorului. Aceste comenzi controlează succesiunea micro-operațiilor pe care le efectuează procesorul pentru a *executa* operația ce corespunde instrucțiunii primite. Instrucțiunile se păstrează într-o memorie externă procesorului. În cazul în care procesorul este "creierul" unui calculator numeric, el trebuie să decidă și succesiunea în care instrucțiunile vor fi aduse din memoria operativă (funcția de *secvențiere*). Astfel că UC trebuie să genereze în exterior și comenzi (informații) de adresare a instrucțiunilor în memorie. Pentru ca transferul instrucțiunilor binare să se facă corect, alte comenzi de la UC vor activa și circuitele ce comandă liniile de legătură între intrările UCP și ieșirile circuitelor de memorie. Ordinea și momentele de timp la care se generează comenzile externe poate fi influențată de semnalele de stare externe pe care le primește UC (de exemplu, semnale care indică unității de control că un circuit comandat nu poate răspunde în timp la aceste comenzi, deci solicită o întârziere). De aceea, se poate spune că unitatea de control poate fi împărțită funcțional, în două sub-unități de control: prima sub-unitate face controlul la nivel de micro-operații, iar cea de-a doua face control la nivel de secvențiere a instrucțiunilor din memoria externă (automat de adresare).

Interacțiunea unității de control cu dispozitive externe necesită "sincronizarea" acțiunilor. **Sincronizarea** operațiilor UC (și implicit a UCP) cu exteriorul poate fi clasificată în *trei niveluri de sincronizare*:

1. nivelul elementar al semnalului de *ceas* aplicat din exteriorul UCP;
2. nivelul semnalelor de sincronizare și control pentru *transferurile de date* cu memoria principală și porturile de I/O;
3. nivelul semnalelor de *întrerupere* pentru sincronizarea UCP cu evenimente externe.

Funcționarea procesorului este ciclică, conform schemei bloc din figura 3.2, până la terminarea execuției tuturor instrucțiunilor programului.

În prima fază unitatea de control adresează și controlează aducerea unei instrucțiuni din memoria principală. Apoi UC decodifică codul instrucțiunii și generează secvența de comenzi specifică instrucțiunii către unitatea de prelucrare și către exterior (de exemplu pentru citirea operandilor). În faza de execuție ALU execută comenzile (realizând operații aritmetice sau logice asupra operandilor) și setează indicatori de condiții, care sunt citați de UC. Stocarea rezultatului prelucrării se poate face în registre interne ale ALU, sau în registre externe UCP (de exemplu memorie, sau port de intrare / ieșire). Dacă mai sunt instrucțiuni în program unitatea de control revine la prima fază, de aducere a instrucțiunii următoare (ciclul de fetch).

Funcția principală a unui procesor de uz general, ca Unitate Centrală de Procesare (UCP) a calculatorului, este să execute secvența de instrucțiuni stocată într-o memorie externă (numită **memoria principală (MP)** a calculatorului). Secvența de operații implicate în aducerea și execuția

unei instrucțiuni, descrisă în figura 3.2, se produce într-un interval de timp, multiplu al perioadei impulsurilor de ceas, interval numit **ciclu instrucțiune**.

Comportarea UCP în timpul ciclului instrucțiune e definită printr-o secvență de microoperații, fiecare din acestea implicând operații de transfer între registre. Timpul  $t_{UCP}$  cerut pentru cea mai scurtă și bine definită micro-operație a UCP este numit **timp de ciclu al UCP** (sau **stare** a UCP, corespunzătoare unei stări a automatului UC) și este unitatea de timp de bază pentru măsurarea tuturor acțiunilor UCP. Inversul acestui timp este frecvența de ceas (clock) la care funcționează UCP, măsurată în MHz sau GHz.

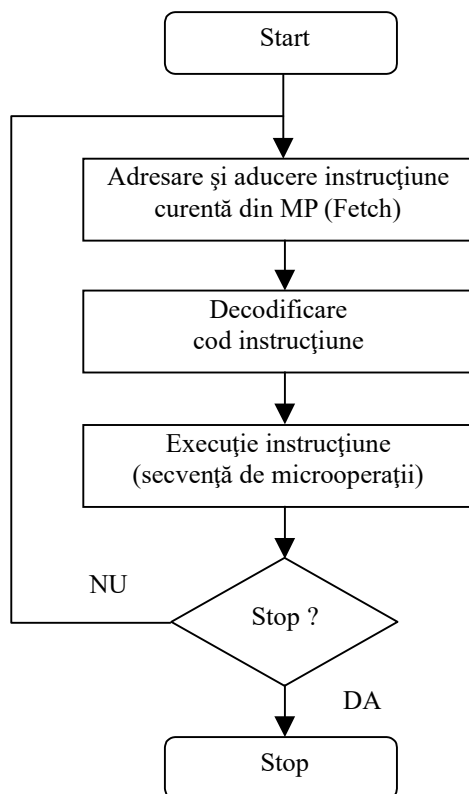


Figura 3.2. Schema bloc care indică funcționarea ciclică a procesorului. (MP = memoria principală)

Aceasta frecvență este dependentă direct de arhitectura și de tehnologia de fabricație a UCP. În cele mai multe dintre (micro-)procesoarele implementate frecvența la care funcționează UCP nu este identică cu frecvența oscilatorului de ceas cu cuarț, care fixează baza de timp a microprocesorului. Frecvența de lucru a UCP este un multiplu, sau submultiplu al frecvenței oscilatorului de ceas.

În plus, pentru a executa programele stocate în memoria principală, UCP controlează celelalte componente ale sistemului, prin linii de comandă și stare speciale. De exemplu, UCP controlează direct sau indirect operațiile de I/O, cum ar fi transferul datelor între dispozitivele de I/O și memoria principală (MP).

De multe ori operațiile cu porturile de I/O cer rar intervenția UCP și de aceea este mai eficient pentru UCP să ignore dispozitivele de I/O până când acestea cer servicii de la UCP. Asemenea cereri sunt numite **întreruperi**. Dacă apare o cerere de întrerupere, UCP poate suspenda execuția programului executat pentru a trece temporar controlul la o rutină de tratare a cererii de întrerupere. Un test al prezentei cererilor de întrerupere se face la sfârșitul fiecărui ciclu instrucțiune.

În cadrul unui calculator pot însă să existe și cereri simultane de acces la un anumit dispozitiv sau la o magistrală. Motivul principal îl constituie comunicarea, în general asincronă (în sensul că diferitele dispozitive nu pot fi sincronizate direct de un semnal comun de ceas), dintre procesor și alte dispozitive. Comunicarea asincronă este dictată de mai multe cauze:

- Există un mare grad de independență între componentele sistemului. De exemplu, într-un calculator, UCP și procesoarele de intrare-ieșire (IOP) pot executa programe diferite, ceea ce înseamnă că cele două componente interacționează relativ rar și la momente de timp nepredictibile.
- Viteza de funcționare a componentelor fizice ale calculatorului variază în limite largi. De exemplu UCP poate lucra la viteze de zeci de ori mai mari față de dispozitivele de memorie principală, în timp ce viteza MP poate fi cu câteva ordine de mărime mai mare ca viteza dispozitivelor de I/O.
- Distanța fizică între componente poate fi prea mare pentru a permite transmiterea sincronă a informației între ele.

Operațiile efectuate într-un ciclu instrucțiune pot fi împărțite în operații intermediare de tip: aducere cod instrucțiune (citire memorie), aducere operand (citire memorie sau registru I/O), scriere memorie sau registru I/O, etc. Ciclul ce corespunde desfășurării în timp a acestor operații intermediare va fi numit **ciclu mașină (CM)**. El cuprinde deci suma activităților ce rezolvă o operație intermediară bine definită, cu o finalitate clară. Primul ciclu mașină al oricărui ciclu instrucțiune este ciclul de **fetch** (extragere-aducere a codului operației).

Concluzionând putem spune că un ciclu instrucțiune (CI) cuprinde mai multe cicluri mașină, fiecare ciclu mașină desfășurându-se de-a lungul mai multor stări ale UCP.

Din punctul de vedere al liniilor de legătură cu exteriorul UCP (blocul procesor a fost redesenat în figura 3.3.), de obicei liniile de intrare date ( $D_{in}$ ) și ieșire date ( $D_{out}$ ) se unifică în linii comune de date de intrare-ieșire ce sunt legate la o magistrală (bus) externă bidirecțională (fig. 3.3b).

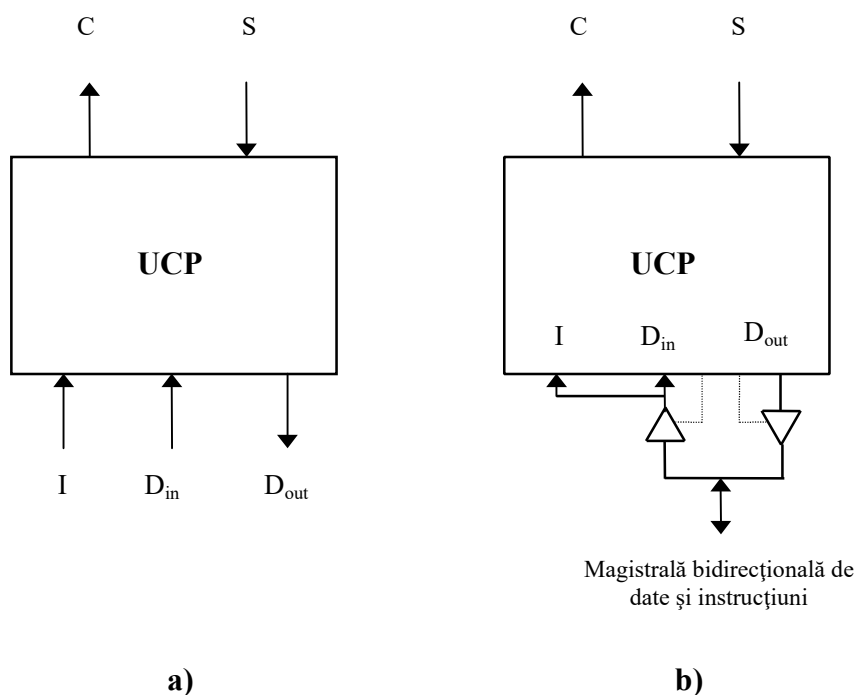


Figura 3.3. Blocul procesor, cu indicarea legăturilor externe. Din - date de intrare, Dout - date de ieșire, C - comenzi externe-inclusiv adrese de MP, I - instrucțiuni, S - semnale de stare din exterior

În acest caz UC trebuie să comande (liniile punctate) și sensul de transfer a datelor prin intermediul unor circuite tampon (buffer) de interfață pentru a elimina conflictele electrice pe liniile bidirecționale (circuitul are posibilitatea invalidării - blocării ieșirilor pe baza circuitelor de tip TSL - trei stări logice, starea blocat corespunzând unei înalte impedanțe (HiZ) la ieșire).

La majoritatea procesoarelor pe magistrala bidirecțională de date se aduc și instrucțiunile (I), reducând astfel numărul de conexiuni către exterior. Este clar, că din punctul de vedere al vitezei de funcționare, o organizare cu magistrale separate pentru date și instrucțiuni este mai rapidă. Acest ultim mod de organizare corespunde așa-numitei *arhitecturi Harvard* a procesorului. Arhitecturile Harvard construite în prezent au o magistrală externă unică de date și instrucțiuni, dar magistrale interne (și circuite de memorie tampon locale) separate pentru date și instrucțiuni.

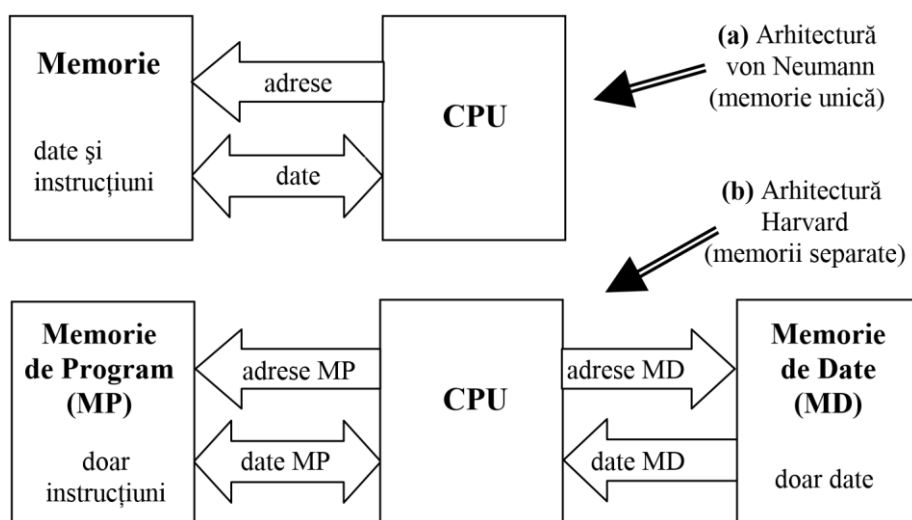


Figura 3.4. Diagrame bloc pentru exemplificarea deosebirilor dintre arhitecturile: (a) von Neumann și (b) Harvard.

Magistrala unică de date și instrucțiuni este numită magistrală de date bidirecțională și ea oferă anumite avantaje sistemului, din punctul de vedere al flexibilității modului de lucru cu exteriorul. Câteva din aceste avantaje ale unei magistrale unice de date și instrucțiuni sunt:

- se folosește o memorie unică pentru stocarea datelor și instrucțiunilor (memoria principală, externă procesorului). Rezultă simplificarea UCP din punctul de vedere al automatului de adresare a memoriei și al numărului de registre implicate.
- folosirea unei memorii unice, unde se stochează atât date cât și instrucțiuni crește flexibilitatea sistemului, pentru că nu există restricții privind adresele de stocare a datelor și instrucțiunilor în memoria principală.

Acest mod de organizare are și dezavantaje, din punctul de vedere al vitezei de lucru și din punctul de vedere al unei relative creșteri în complexitate a unității de control, care trebuie să interpreteze în mod corect cuvintele citite din memorie ca fiind date, sau instrucțiuni.

În cadrul liniilor notate ca și comenzi de ieșire (C) am inclus și adresele binare pe care procesorul le trimite către memorie sau circuitele de intrare - ieșire ale calculatorului. Acestea sunt grupate de obicei într-o magistrală separată, numită *magistrală de adrese*.

Deosebirile dintre arhitecturile von-Neumann și Harvard pentru organizarea memoriei principale și a magistralelor de acces cu această memorie sunt indicate în schemele bloc din figura 3.4.

Prin conectarea procesorului cu un circuit de memorie, în care se stochează programul de executat (ansamblul instrucțiunilor binare) împreună cu datele eferente programului, se realizează o *structură de calculator*. Aici reacția, care conduce la automatul de ordin superior (ordinul 4) se încheie prin programul stocat în memorie, program care controlează comportarea în timp a procesorului.

Schema bloc din figura 3.5. indică faptul că procesorul are circuite de interfață (tampoane) pe toate liniile de conectare cu exteriorul. Acestea permit "deconectarea" procesorului de la magistralele externe de date și adrese și de asemenea eliberarea unor linii de control externe, astfel încât alte dispozitive active (procesoare) din sistem să poată prelua temporar controlul magistralelor. Uneori tampoanele de la interfața procesorului cu exteriorul, au în afara capacității de trecere în stare de înaltă impedanță (HiZ) și capacitatea de memorare (latch).

Trebuie menționat că rolul principal al tampoanelor la interfața procesorului cu exteriorul nu constă în posibilitatea de HiZ, ci în capacitatea de a asigura factorii de încărcare de ieșire (fan-out<sup>1</sup>) și intrare (fan-in<sup>2</sup>) necesari circuitelor logice, pentru asigurarea nivelurilor corecte de tensiune. Circuitele tampon au de asemenea rol de izolare între circuitele interne și cele externe procesorului și rolul de a permite controlul unor valori de capacitate parazită limită, pe fiecare linie.

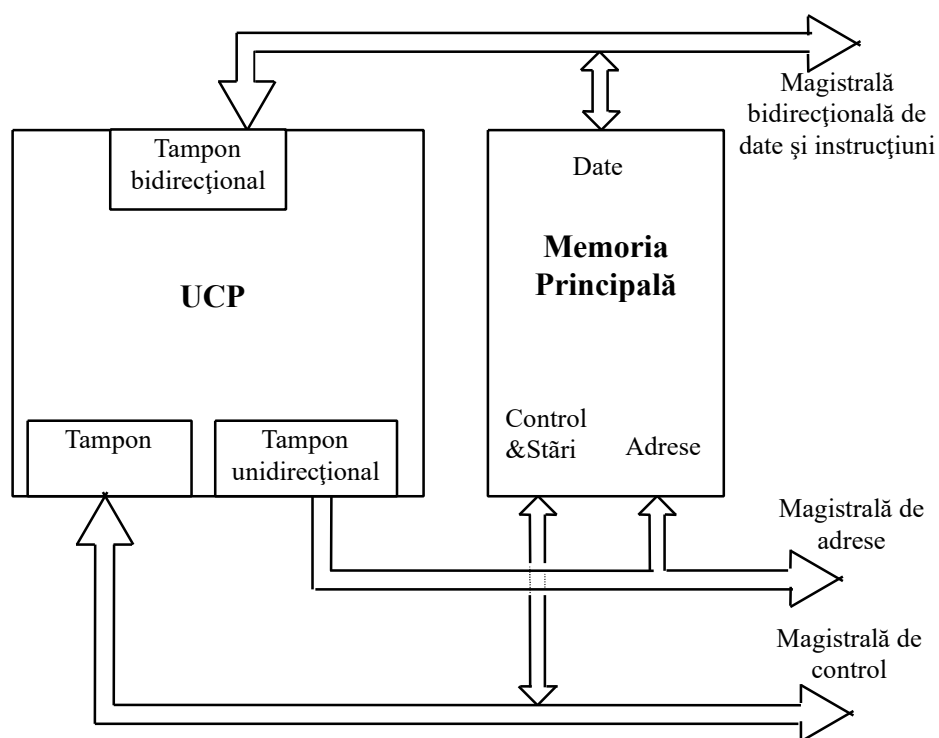


Figura 3.5. Structură de calculator, formată din UCP și memorie principală, cu indicarea circuitelor tampon la interfața procesorului cu exteriorul.

