

## Arhitectura Calculatoarelor

### T.4 –Structura și caracteristicile unității centrale

Structura Unității Centrale, Unitatea Aritmetică și Logică (UAL), Unitatea de Comandă (UCd), Registrele generale (RG). Caracteristicile unității centrale. Lungimea cuvântului, Frecvența ceasului, Numărul de instrucțiuni executate în unitatea de timp, Gradul de paralelism, Paralelism la nivel de proceso

Scopul Lecției: De a face cunoștință cu noțiunea de Unitate Centrală structura ei și caracteristicile. Funcția UAL, UCd și RG

Studentul trebuie să cunoască:

- § Schema conceptuală a UC
- § Funcțiile Componentelor UC
- § Structura Componentelor UC
- § Noțiunea de lungimea cuvântului
- § Influența frecvenței ceasului asupra performanței
- § Paralelismul la nivel de instrucțiune (pipeline)

Conf. Univ. Dr. Crețu Vasilii

---

---

---

---

---

---

---

---

---

---

Unitatea centrală (UC), în engleză CPU=*Central Processing Unit*, este partea din calculator care are rolul de a interpreta și executa instrucțiunile unui program, de a citi sau salva în memorie rezultatele și de a comunica cu unitățile de schimb. Toate aceste activități sunt cadentate de un ceas la frecvență constantă care împarte timpul în fracțiuni de aceeași durată numite cicli.

- Unitatea Centrală are în componență:
- Unitatea Aritmetică și Logică (UAL);
  - Unitatea de comandă (UCd);
  - Registrele generale (RG).




---

---

---

---

---

---

---

---

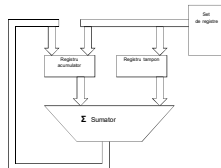
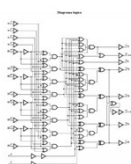
---

---

#### Unitatea Aritmetică și Logică (UAL)

UAL execută operații aritmetice (adunare, scădere, înmulțire împărțire, complement față de 1, complement față de 2 etc.), operații logice (negare, și, sau, suma modulo 2), decalaje și rotații. Are două intrări de date, pe n biți, o ieșire corespunzătoare operației efectuate, pe n biți, eventual o ieșire corespunzătoare flagurilor poziționate de operație și o intrare de comandă care selecționează operația de efectuat.

Toată Unitatea Aritmetică și Logică este grupată în jurul unui sumator paralel care poate aduna conținutul a două registre multiplexate la cele două intrări ale sumatorului. Operațiile de înmulțire și împărțire se realizează, cu ajutorul diferiților algoritmi, prin adunări și deplasări stânga/dreapta succesive. Scăderea se realizează ca o adunare cu complementul scăzătorului.




---

---

---

---

---

---

---

---

---

---

**Unitatea de Comandă (UCd)**

Unitatea de comandă este formată din:

- Generatorul de Faze (GF);
- Generatorul de Tact (GT);
- Blocul Circuitelor de Comandă (BCC).

Blocul circuitelor de comandă (BCC) dirijează toate operațiile executate în cadrul unei instrucțiuni. Există *microoperații*, care sunt operațiile elementare executate într-o instrucțiune, și *microcomenzi*, care sunt semnalele generate de BCC pentru execuția microoperațiilor. Microcomenzile sunt trimise elementelor de execuție din structura calculatorului: registre, UAL, memorie, porturi etc. O instrucțiune este, de fapt, o succesiune de microoperații.

Toate microoperațiile care se execută în același timp definesc o stare în execuția unei instrucțiuni, stare numită *fază*.

Generatorul de faze (GF) construiește succesiunea fazelor necesară pentru execuția instrucțiunii.

Generatorul de tact (GT) dă cadența schimbărilor de stare pentru toate circuitele secvențiale.

**Registrele generale (RG)**

Registrele generale (RG) sunt considerate o memorie foarte rapidă și de foarte mică capacitate. Structural fac parte din UC și, în marea majoritate a arhitecturilor, ele sunt adresabile pe magistrală.

Există două moduri de conexiune a registrelor generale:

- RG conectate direct între ele;
- RG conectate la magistrale.

---

---

---

---

---

---

---

---

---

---

**Caracteristicile Unității Centrale**

Performanțele în funcționarea unei Unități Centrale sunt redată prin următoarele caracteristici:

- Lungimea cuvântului;
- Frecvența ceasului;
- Numărul de instrucțiuni executate în unitatea de timp;
- Gradul de paralelism.

**Lungimea cuvântului**

Calculatoarele lucrează cu ajutorul cuvintelor de cod a căror cantitate de informație este măsurată în biți. Numărul de biți reprezintă lungimea unui cuvânt și este multiplu de doi. Un cuvânt poate reprezenta:

- o instrucțiune;
- un segment de date.

Într-un calculator, lungimea cuvântului se identifică cu numărul de biți ai instrucțiunii. Cele mai noi calculatoare au instrucțiuni pe 64 biți. Aceasta este o caracteristică principală a UC.

Nu este obligatoriu ca lungimea cuvântului să fie aceeași cu dimensiunea magistralei de memorie pe care se aduce instrucțiunile din UM în UC. Un calculator poate avea, de exemplu, lungimea instrucțiunii de 64 biți și lărgimea magistralei de 32 biți; pentru aducerea unei instrucțiuni din memorie sunt necesare, în acest caz, două apeluri la memorie.




---

---

---

---

---

---

---

---

---

---

**Frecvența ceasului**

Orice calculator are un generator de impulsuri, numit ceasul unității centrale. Acesta este realizat cu un cuarț care emite impulsuri cu frecvență fixă. Ceasul inițial suferă două tipuri de operații:

- operația de divizare a ceasului, ceea ce înseamnă că ceasul inițial, cu cuarț, suferă modificarea frecvenței sale;
- operația de amplificare a semnalului de ceas.

Pe acest ceas, care este inima calculatorului, au loc toate evenimentele hard din UC.

**Numărul de instrucțiuni executate în unitatea de timp**

Dacă facem următoarele notații:

$f$  = frecvența ceasului, în Hz

$N$  = numărul mediu de ceasuri în care se execută o instrucțiune

$n$  = numărul de instrucțiuni executate într-o secundă

atunci

$$n = \frac{f}{N}$$

De exemplu, pentru un calculator care are frecvența ceasului de 2 GHz și care execută două instrucțiuni pe ceas, numărul de instrucțiuni executate în unitatea de timp este:

$$n = \frac{f}{N} = \frac{2 \text{ GHz}}{2} = 4 \cdot 10^9 = 4 \text{ miliarde instrucțiuni pe secundă}$$

---

---

---

---

---

---

---

---

---

---

Pentru calculatoarele moderne se adoptă o unitate de măsură numită MIPS (milioane de instrucțiuni executate într-o secundă). În exemplul precedent, parametrul  $n$  va fi de 4000 MIPS.

Trebuie precizat că acest parametru ( $n$ ) este mai aproape de adevăr decât frecvența ceasului ( $f$ ). Există calculatoare cu frecvență mai mică dar care sunt inferioare calculatoarelor cu frecvența ceasului mai mare, tocmai datorită mărimii  $N$ . De exemplu Intel 586 cu frecvența 100 MHz și Pentium I cu frecvența 66 MHz. Ar fi mai bine să fie afișat parametrul  $n$  în locul parametrului  $f$ , dar partea dificilă este calcularea practică a lui  $N$ . Este dificilă, deoarece, în setul de instrucțiuni, fiecare are  $N$  diferit și chiar aceeași instrucțiune are  $N$  diferit în funcție de contextul rullării.

---

---

---

---

---

---

---

---

---

---

**Gradul de paralelism**

Există două feluri de paralelism:

- paralelism la nivel de instrucțiuni (pipeline)
- paralelism la nivel de procesor.

**a. Paralelism la nivel de instrucțiuni (pipeline)**

a) *Principiul pipeline-ului* este acela al liniei de montaj (de asamblare):

- împărțirea unei sarcini în mai multe subsarcini de durate egale, numite etaje;
- execuțarea simultană a diferitelor subsarcini din mai multe sarcini.

În felul acesta se mărește debitul de sarcini al sistemului. Sistemul va fi caracterizat prin doi parametri: durata individuală a unui etaj ( $T$ ) și numărul de etaje ( $L$ ) din pipeline. Latența  $L$  este durata totală de execuție a unei sarcini:  $L = LT$

Debitul,  $d$ , al pipeline-ului depinde de numărul  $n$  de sarcini de executat. Timpul necesar pentru execuția în pipeline a  $n$  sarcini este timpul de execuție a primei sarcini,  $L = LT$ , plus timpul necesar pentru a le termina pe celelalte  $n-1$  următoare:

$$d = \frac{n}{L + (n-1)T}$$

Apare rezultatul esențial: pentru un mare număr de sarcini, debitul nu depinde de latență ci de durata individuală a fiecăruia etaj. Deci, debitul optim al va fi atins fracționând cât mai fin posibil sarcina în subsarcini. Evident, fracționarea are limite tehnice.

---

---

---

---

---

---

---

---

---

---

*b) Clasicul pipeline pentru execuția instrucțiilor*

În calculatoare, pipeline-ul constă în fracționarea execuției unei instrucțiuni în mai multe module, fiecare modul executând o parte de instrucțiune. Conceptul de pipeline înseamnă de fapt o bandă de asamblare cu segmente, fiecare segment executând o parte de instrucțiune.

De exemplu, o bandă de asamblare cu 5 segmente. Aceste 5 segmente sunt:

- S1 – unitate de extragere a instrucțiunii
- S2 – unitate de decodificare a instrucțiunii
- S3 – unitate de calcul și extragere a operandilor
- S4 – unitate de execuție propriu zisă a operației instrucțiunii
- S5 – unitate de scriere a rezultatelor

S1	1	2	3	4	5	6	7	8	9	10	11
S2		1	2	3	4	5	6	7	8	9	10
S3			1	2	3	4	5	6	7	8	9
S4				1	2	3	4	5	6	7	8
S5					1	2	3	4	5	6	7
Timp de execuție a operandilor	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$

**Schema unei benzi de asamblare cu 5 segmente: (a) schema de funcționare; (b)diagrama temporară a execuției segmentelor.**

---

---

---

---

---

---

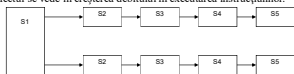
---

---

---

---

la un moment de timp fiecare segment execută unul din cele 5 segmente de instrucțiune diferite. Dacă timpul de execuție a unui segment este de 1 secundă, ( $t_1=t_2=...=t_5=1s$ ) atunci timpul de execuție a 7 instrucțiuni este, așa cum se vede în figură, de 11 secunde. Pe o mașină normală, fără pipeline, timpul de execuție a 7 instrucțiuni, fiecare necesitănd 5 secunde, ar fi de 35 sec. Efectul se vede în creșterea debitului în executarea instrucțiunilor.



Pipeline cu două benzi de asamblare.

Un astfel de sistem are o singură unitate de extragere a instrucțiunii (S1) care extrage perechi de instrucțiuni și le plasează pe cele două benzi. Condiția de a lucra în paralel este ca cele două instrucțiuni să nu își dispute aceeași resursă și să nu depindă una de rezultatul celeilalte. Această condiție este garantată fie de compilator, fie de un hard suplimentar dotat cu un sistem de predicție.

Calculatoarele Pentium I erau dotate cu două benzi de asamblare:

- bandă de asamblare U (U pipeline), bandă principală, care putea executa orice instrucțiune;
- bandă de asamblare V (V pipeline), care putea executa doar instrucțiunile simple în numere întregi și o singură instrucțiune simplă în virgulă mobilă.

Existau reguli destul de complicate pentru împerecherea instrucțiunilor. Erau extrase câte două instrucțiuni și dacă erau compatibile erau executate, dacă nu, se executa doar prima pe banda U, a doua fiind plăstrată și împerecheată cu cea care urma. În acest mod, s-a constatat că Pentium I era de două ori mai rapid decât un 586 la aceeași frecvență.

---

---

---

---

---

---

---

---

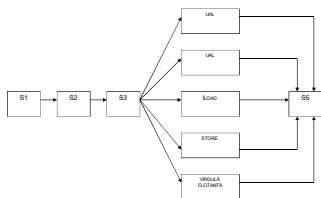
---

---

---

c) Arhitecturi superscalare

Pornind de la ideea benzilor de asamblare, s-a ajuns la concluzia că este mai benefic de a avea o singură bandă de asamblare dar cu mai multe unități funcționale. O astfel de arhitectură se numește *superscalară*, termen introdus de Agerwals și Cocke în 1987. În figura 2.4, este dată o astfel de arhitectură.



Procesor superscalat cu 5 unități funcționale.

Ideea arhitecturii superscalare este că segmentul S3 poate lansa instrucțiuni mult mai rapid decât le poate executa S4, deci, la o bandă simplă există o gâtuire de timp între S3 și S4. La arhitectura superscalară, se împarte unitatea de execuție S4 în unități funcționale pe tipuri de instrucțiuni (de încărcare, LOAD, de memorie, STORE, de virgulă mobilă)

---

---

---

---

---

---

---

---

---

---

---

a. Paralelism la nivel de procesor

Banda de asamblare sau arhitectura superscalară nu cresc performanțele în mod simțitor. Mult mai eficient este mărirea numărului de procesoare. Acest lucru se întâmplă în :

- mașinile SIMD, calculatoare vectoriale;
- mașinile MIMD, multiprocesoare;
- mașinile MIMD, multicalculatoare.

---

---

---

---

---

---

---

---

---

---

---

**MODURI DE ADRESARE**

După posibilitatea găsirii operandilor, există mai multe moduri de adresare:

- adresare imediată;
- adresare directă;
- adresare indirectă;
- adresare indexată.

**a) Adresare imediată**

Operandul se află chiar în câmpul instrucțiunii.



MOV R 20000

Se transferă valoarea 20000 în registrul R.

---

---

---

---

---

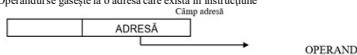
---

---

---

**Adresare directă**

Operandul se găsește la o adresă care există în instrucțiune

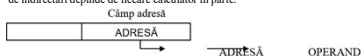


MOV R 100  
100 20000

În câmpul de adresă al instrucțiunii se află o adresă (100) la care există operandul.

**Adresare indirectă**

În câmpul de adresă al instrucțiunii se află o adresă. La acea adresă se află operandul. Numărul de indirecționări depinde de fiecare calculator în parte.



MOV R 100  
100 200  
200 2000

La adresa 100 din câmpul instrucțiunii se află altă adresă, 200, iar la această adresă se află operandul

---

---

---

---

---

---

---

---

**Adresare indexată**

La adresarea indexată participă un registru numit registru INDEX. Adresa operandului este:

Adresa operand = valoare<sub>CÂMP ADRESĂ</sub> + valoare<sub>REGISTRU INDEX</sub>

MOV R 100

RINDEX 600

100 700

Adresa calculată: 600 + 700 = D00

D00 20000

---

---

---

---

---

---

---

---