

CN1_Cursul 11.1 Arhitectura unui calculator

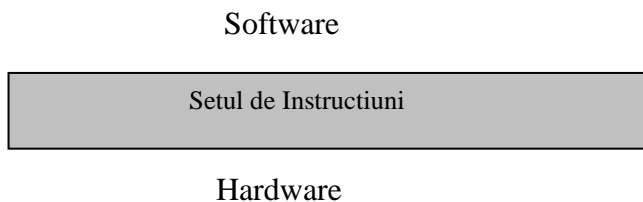
**Arhitectura unui calculator = Arhitectura Setului de Instructiuni +
Organizarea calculatorului/masinii**

Arhitectura Setului de Instructiuni – ASI- (ISA- Instruction Set Architecture)
conform lui: Amdahl, Blaaw si Brooks (1964)

.....atributele unui sistem (de calcul) vazute de catre programator: structura conceptuala si comportarea functionala, spre deosebire de organizarea fluxurilor de date si de control, de proiectarea logica si de implementarea fizica:

- Organizarea memoriei pentru stocarea programelor,
- Tipurile de date si structurile de date: codificare si reprezentari,
- Setul de Instructiuni
- Formatele Instructiunilor
- Modurile de adresare si accesare ale obiectelor reprezentand date si instructiuni
- Conditiiile de exceptie.

Setul de instructiuni realizeaza interfata intre software si hardware:



Exemple de Arhitecturi de Seturi de Instructiuni:

- Digital Alpha (v1,v3) 1992-1997
- HP PA (Precision Architecture) (v1.1,v2.0) 1986- 1996
- Sun Sparc (v8,v9) 1987-1995
- SGI (MIPS I, II, III, IV, V) 1986-1996
- Intel (8086,80286,80386,80486,Pentium,MMX ,.....) 1978-1996

Arhitectura Setului de Instructiuni pentru MIPS R3000 (rezumat)

Categorii de instructiuni:

- Incarca/Stocheaza (Load/Store)
- Aritmetice-Logice (Instructiuni de Calcul)
- Salt si Ramificare
- Virgula Mobila
- coprocesor
- Gestiune/Management Memorie
- Speciale

Registre:

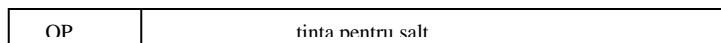
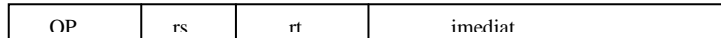
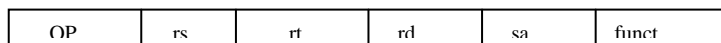
R0 – R31

PC

HI

I.O

Trei Formate de Instructiuni cu lungimea de 32 de biti.



Organizarea calculatorului se refera la:

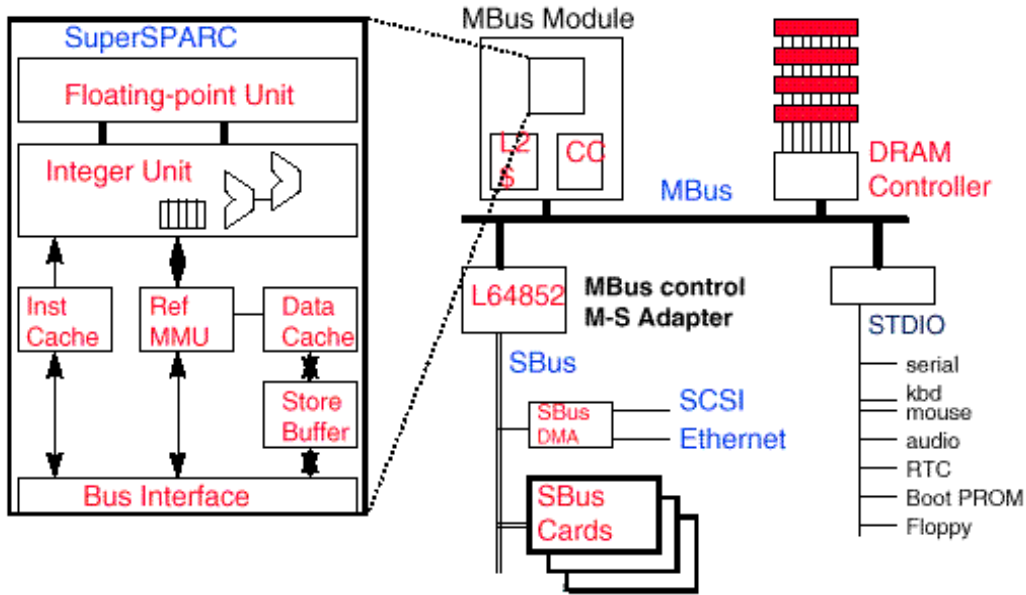
- Capabilitatile, Performantele si Caracteristicile principalelor Unitati Functionale (ex.: Registe, UAL, Unitati Logice, Circuite de Deplasare,...)
- Modurile in care aceste componente sunt interconectate;
- Fluxul informatiei intre componente;
- Logica si mijloacele folosite pentru controlul fluxului informatiei;
- Sincronizarea operarii Unitatilor Functionale pentru a realiza ASI;
- Descrierea operarii sistemului numeric la Nivelul Transferurilor intre Registre, NTR, (RTL – Register Transfer Level).

Punctul de vedere al Proiectantului la Nivel Logic:

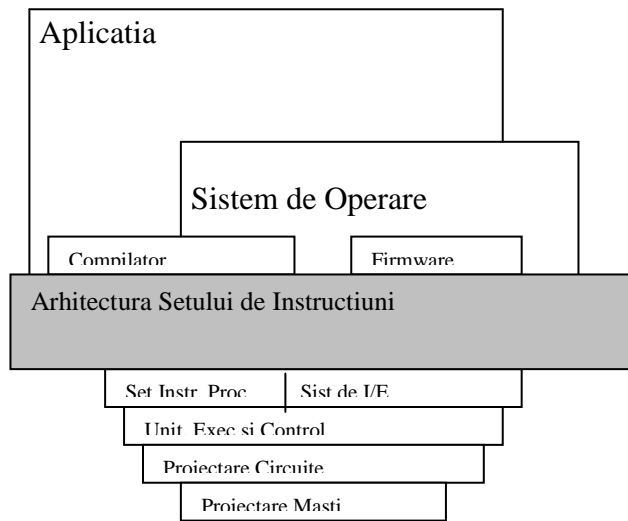
Nivel ASI \longleftrightarrow Unitati Functionale si Interconexiuni

Exemplu de Organizare:

TI SuperSPARC TMS390Z50 din statia Sun SPARCstation20:



Ce este "Arhitectura Calculatorului?"



- Coordonarea mai multor niveluri de abstractizare
- Existenta mai multor forte care se modifica rapid:
 - Tehnologia;
 - Aplicatiile
 - Limbajele de programare;
 - Sistemele de operare;
 - Istoria/Traditia;
 - Ingeniozitatea proiectantilor
- Proiectare, Masurare si Evaluare

Tehnologia

Tehnologia se perfectioneaza continuu:

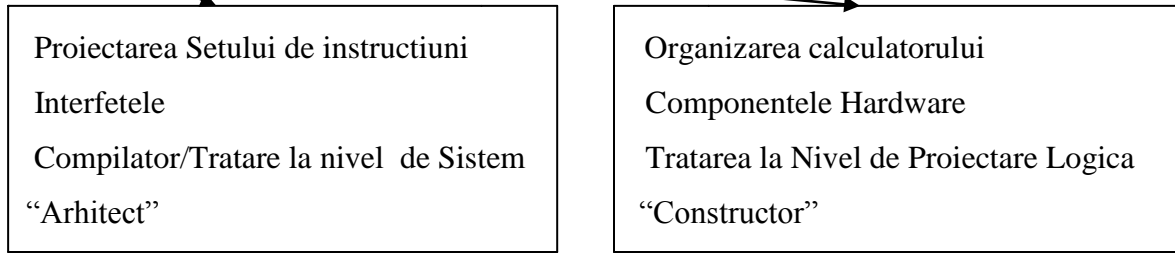
- Procesoarele:
 - Capacitatea logica creste cu circa 30%/an;
 - Frecventa ceasului creste cu circa 20%/an.
- Memoria:
 - Capacitatea memoriei Dinamice DRAM creste cu circa 60%/an
 - Viteza memoriei creste cu circa 10%/an
 - Costul pe bit scade cu circa 25%/an.
- Disc:
 - Capacitatea creste cu circa 60%/an

Capacitatea circuitelor de memorie DRAM:

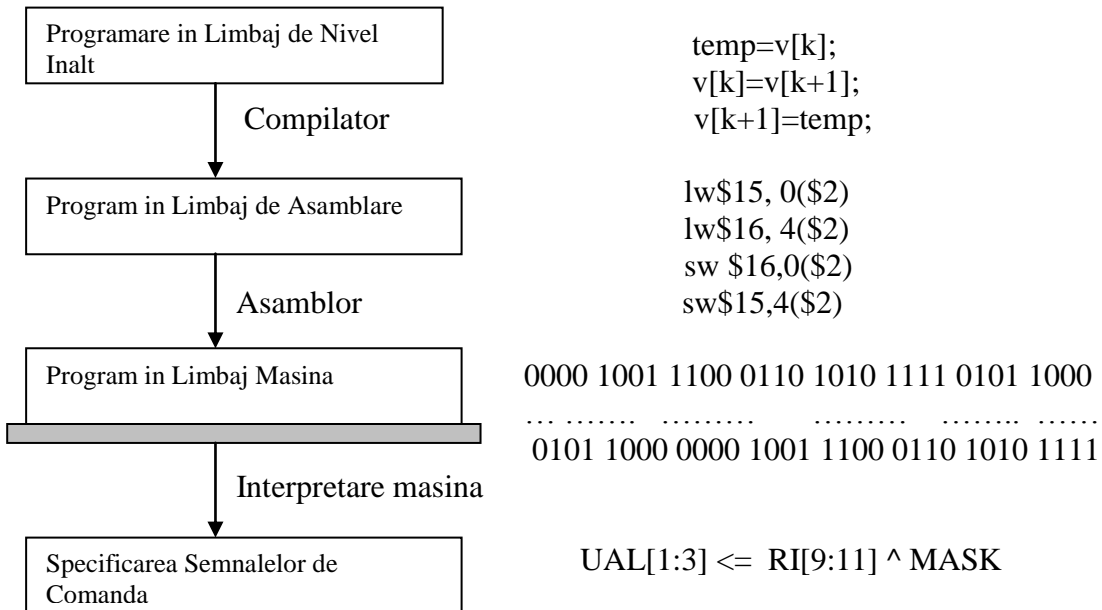
Anul :	1980	1983	1986	1989	1992	1996	1999	2001
Capacitatea:	64Kb	256Kb	1Mb	4Mb	16Mb	64Mb	256Mb	1Gb

In 1985 au aparut procesorul pe o singura pastila si calculatorul pe o singura placheta. Aceste realizari au propulsat puternic: statiile de lucru, calculatoarele personale, sistemele multiprocesor. Dupa 2002, acestea din urma pot aparea in postura de sisteme “mainframes” in comparatie cu calculatoarele pe una sau doua pastile

Arhitectura si Ingineria Calculatoarelor

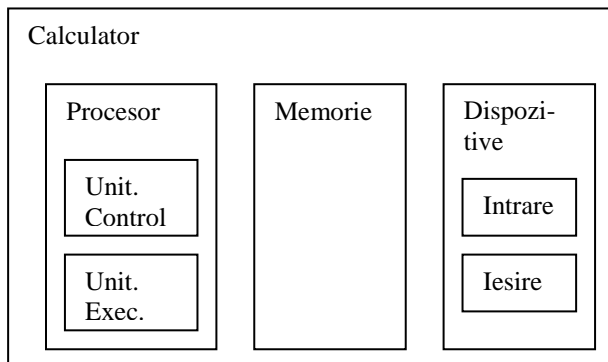


Niveluri de Reprezentare



Niveluri de Organizare

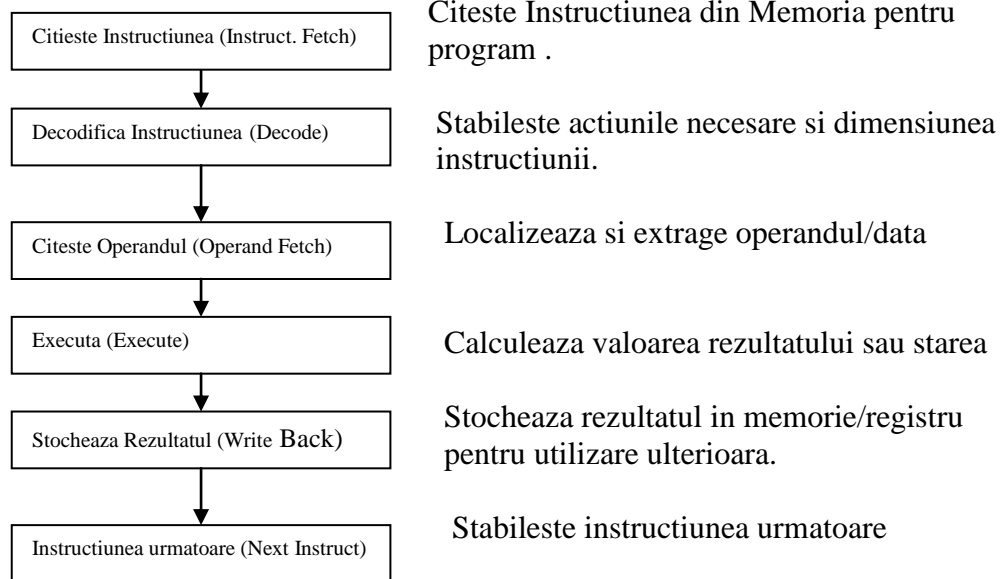
Pe exemplul SPARCstation20



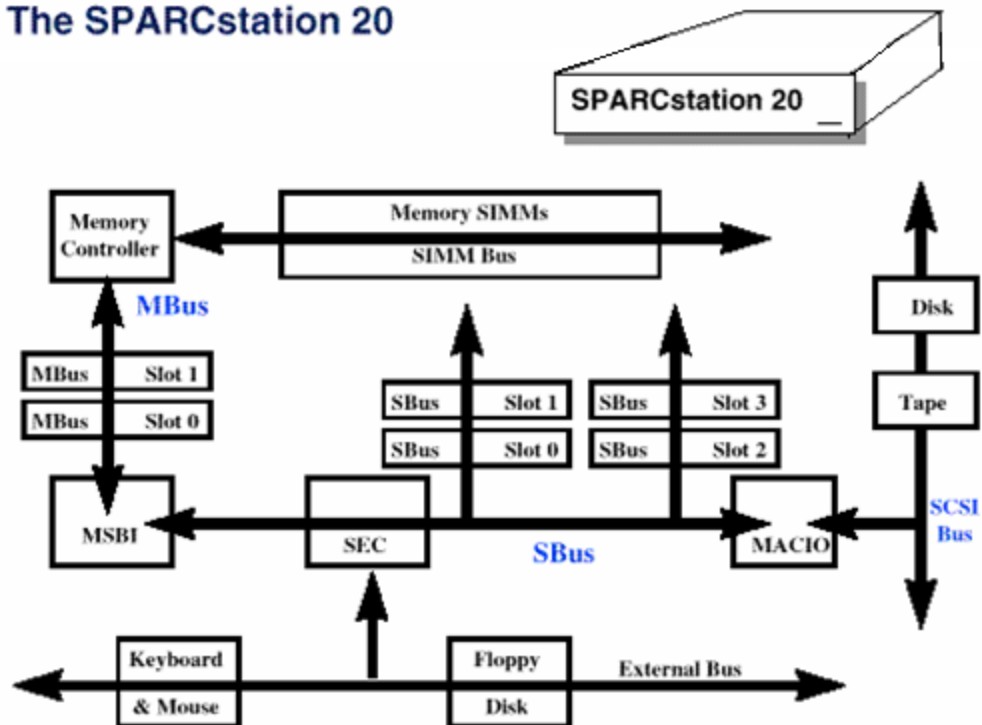
Tinta Proiectarii Statiilor de Lucru:

- Cost Procesor ~25%
- Cost Memorie cap. minima ~ 25%
- Cost Dispozitive de I/E, Surse alimentare, cabinet etc. ~ 50%

Ciclul de Operare



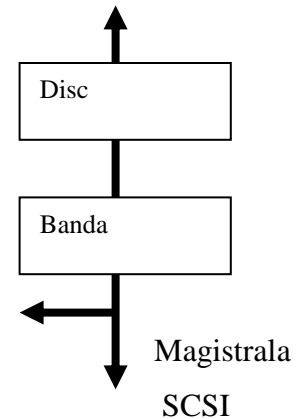
The SPARCstation 20



Dispozitive Standard de I/E

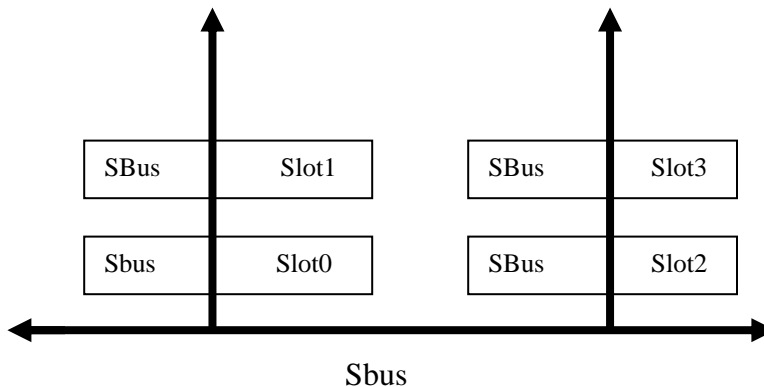
(SPARCstation 20 –SS20)

- SCSI = Small Computer Systems Interface
- Interfata Standard (IBM, Apple, HP, Sun etc)
- Calculatoarele si Dispozitivele de I/E comunica unul cu altul.
- Discul dur este unul dintre dispozitivele de I/E , care se conecteaza la Magistrala SCSI



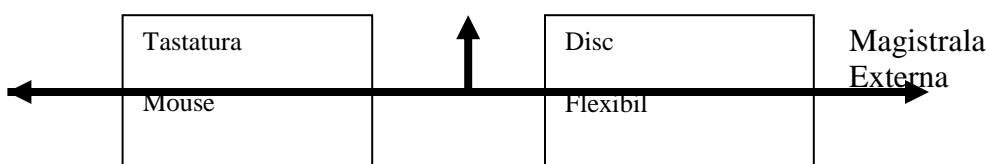
Dispozitive de I/E rapide

- SBus reprezinta magistrala proprietara SUN, pentru dispozitive rapide de I/E
- SS20 dispune de patru conectori SBus, pentru dispozitive de I/E
- Exemple: accelerator grafic, adaptor video etc..
- Termenii de viteza ridicata si viteza coborata sunt relativi



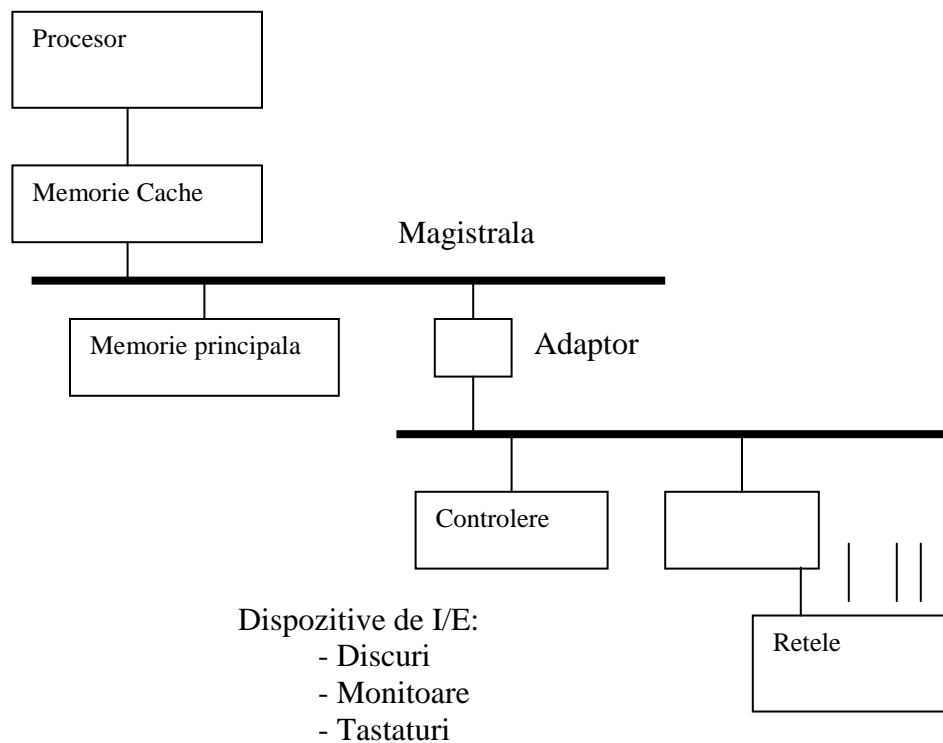
Dispozitive de I/E lente

- SS20 poseda numai patru conectori Sbus, deoarece spatiul pe placa este limitat
- Viteza unor dispozitive de I/E este limitata de timpul de reactie a operatorului, care este extrem de mare, din punctul de vedere al calculatorului
- Exemple: tastatura si mouse-ul
- Nu sunt motive pentru utilizarea unui conector SBus costisitor.



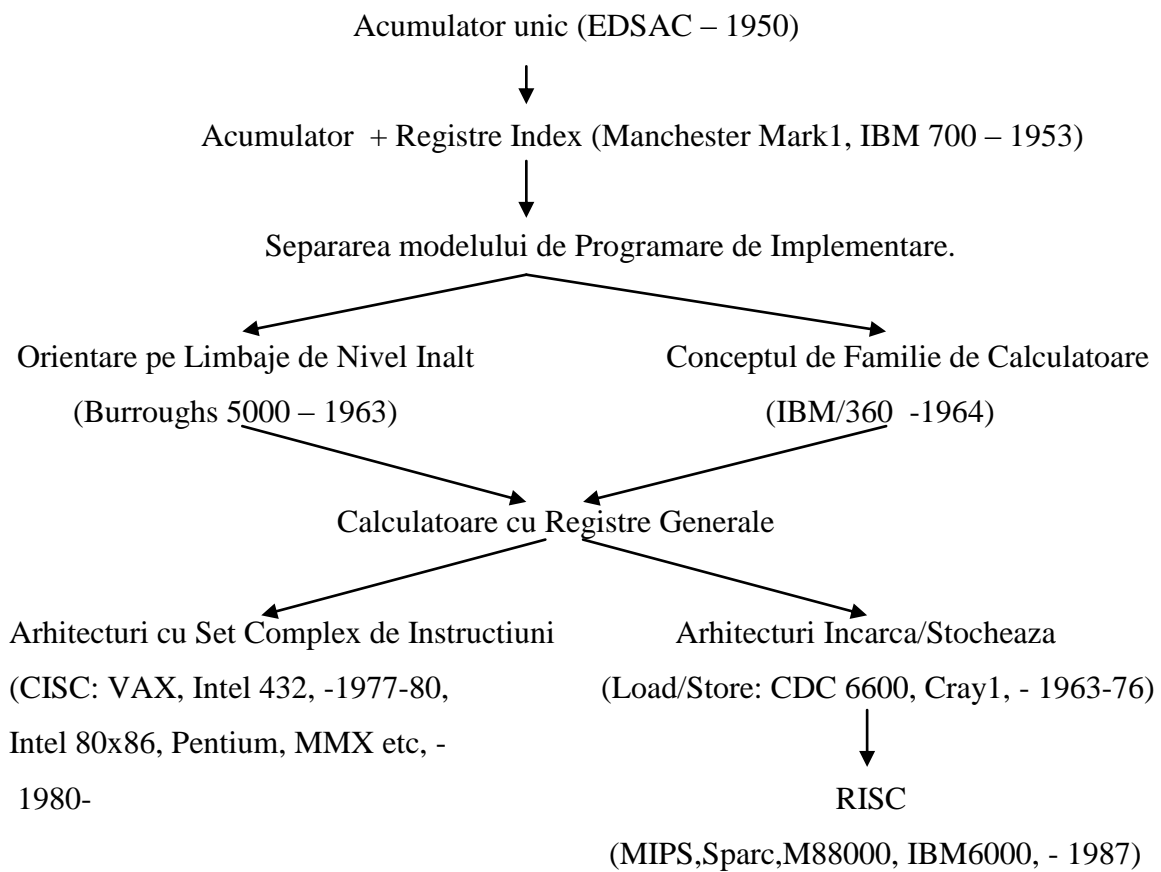
Rezumat

- Toate calculatoarele poseda cinci componente:
 - (1) Unitatea de Executie } Procesor
 - (2) Unitatea de Comanda }
 - (3) Memoria
 - (4) Dispozitivele de intrare
 - (5) Dispozitivele de iesire
- Memoria nu este omogena ca tehnologie, amplasare, cost, performanta etc
 - Memoria Cache (intermediara) este costisitoare, rapida si plasata in apropierea procesorului.
 - Memoria principala este mai putin costisitoare si este solicitata la capacitati din ce in ce mai mari
- Interfetele intre unitatile functionale si intre calculator si mediul inconjurator ridica probleme
- Proiectarea intregului sistem se realizeaza in conditiile unor restrictii de performanta, putere consumata, arie ocupata si cost



ASI - Clase fundamentale (cele mai multe masini reale constituie hibrizi ai acestor clase).

Evolutia Arhitecturii Setului de Instructiuni.



Acumulator (un registru):

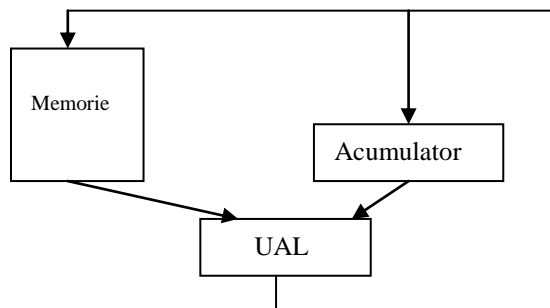
1 adresa add A

acc ← acc + mem[A]

1 adresa + x *) addx A

acc ← acc + mem[A+x]

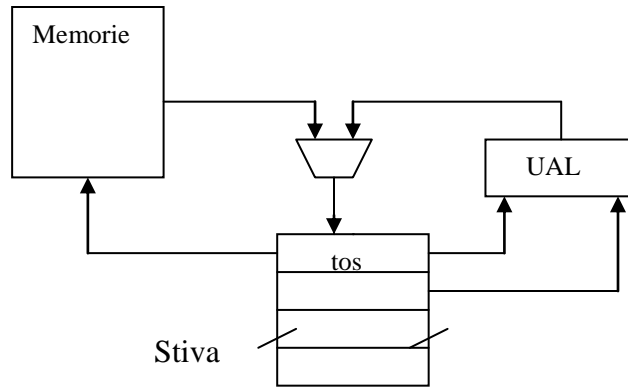
*) x este registru index



Fluxul datelor

Stiva:

0 (zero) adrese add $tos \leftarrow tos + urmator$



Fluxul datelor

Registre Generale (poate fi memorie/memorie)

2 adrese add A B $EA[A] \leftarrow EA[A] + EA[B]$

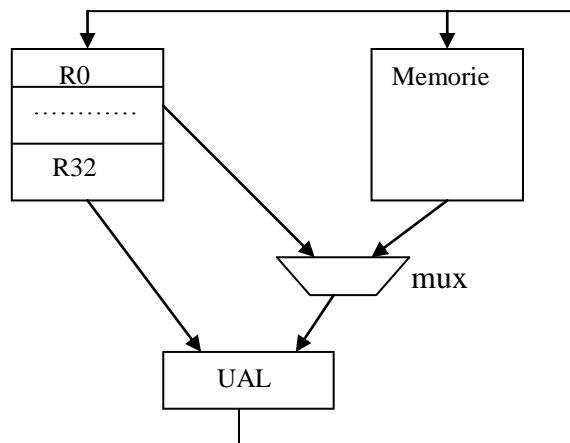
3 adrese add A B C $EA[A] \leftarrow EA[B] + EA[C]$

Incarca/Stocheaza: (Load/Store)

3 adrese add Ra Rb Rc $Ra \leftarrow Rb + Rc$

2 adrese load Ra Rb $Ra \leftarrow mem[Rb]$

store Ra Rb $mem[Rb] \leftarrow Ra$



Fluxul datelor

Comparatii:

- Octeti pe instructiune?
- Numar de instructiuni?
- Cicluri pe instructiune?

Compararea numarului de instructiuni.

Fie secventele de coduri pentru expresia $C = A + B$

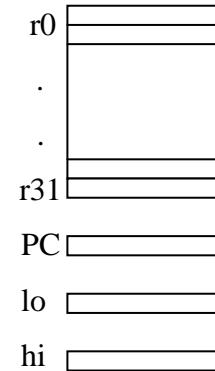
Stiva	Acumulator	Registre (reg-mem)	Registre (citeste/stocheaza)
Push A	Load A	Load R1, A	Load R1, A
Push B	Add B	Add R1, B	Load R2, B
Add	Store C	Store C, R1	Add R3, R1, R2
Pop C			Store C, R3

Organizarea bazata pe registre generale este dominanta

- Toate calculatoarele realizate intre 1975 si 2000 utilizeaza registre generale
- Avantajele folosirii registrelor generale:
 - Registrele sunt mai rapide decat memoria
 - Registrele sunt mai usor de utilizat de catre compilator, de ex.:
in expresia $(A*B) - (C*D) - (E*F)$ inmultirile se pot efectua indiferent de ordine, in comparatie cu stiva.
 - Registrele pot stoca variabile:
 - traficul cu memoria este redus, programul se poate executa mai repede (registrele sunt mai rapide decat memoria).
 - densitatea codului creste deoarece numele registrelor pot fi codificate cu mai putini biti decat locatiile de memorie

Registreele procesorului MIPS I

- Memoria programabila:
 - 2^{32} octeti de memorie
 - 31 x 32- biti RG (Registre Generale, R0=0)
 - 32 x 32 biti registre FP (DP-perechi)
 - HI, LO, PC

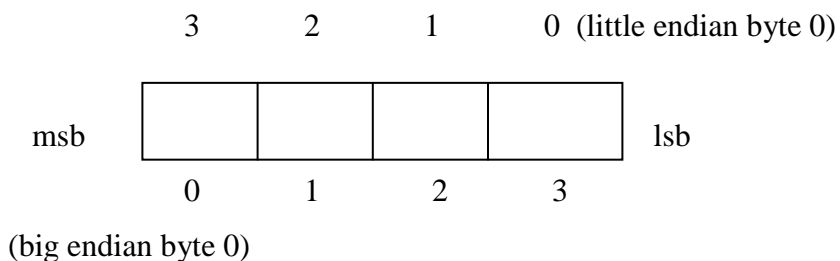


Adresarea Memoriei

- Incepand cu 1980 cele mai multe calculatoare utilizeaza adrese la nivel de octet – byte
- Pentru proiectarea ASI apar doua probleme:
 - Un cuvânt de 32 de biti poate fi extras prin patru citiri de octeti succesivi, de la adrese secventiale de octeti, sau poate fi extras ca un singur cuvânt de la o adresa de octet. Cum se mapeaza adresele de octeti in cuvinte?
 - Un cuvânt poate fi plasat la oricare pozitie de octet?

Adresarea Obiectelor: “Endiani” si Aliniere.

- Big Endian: adresa celui mai semnificativ octet = adresa cuvântului.
(xx00 = “Big End” al cuvântului)
IBM 360/370, Motorola 68k, MIPS, Sparc, HP PA
- Little Endian: adresa celui mai puțin semnificativ octet = adresa cuvântului
(xx00= “Little End” al cuvântului)
Intel 80x86, DEC Vax, DEC Alpha (Windows NT)



Aliniere: toate obiectele se plaseaza la adrese care sunt multipli ai dimensiunilor lor.

Moduri de Adresare (VAX 11/780)

Moduri de Adresare	Exemplu	Semnificatie
Registru	Add R4, R3	$R4 \leftarrow R4+R3$
Imediat	Add R4,#3	$R4 \leftarrow R4+3$
Deplasata	Add R4,100(R1)	$R4 \leftarrow R4+Mem[100+R1]$
Registru Indirect	Add R4,(R1)	$R4 \leftarrow R4+Mem[R1]$
Indexat/Bazat	Add R3,(R1+R2)	$R3 \leftarrow R3+Mem[R1+R2]$
Direct sau Absolut	Add R1,(1001)	$R1 \leftarrow R1 + Mem[1001]$
Memorie Indirect	Add R1,@(R3)	$R1 \leftarrow R1 + Mem[Mem[R3]]$
Post-increment	Add R1,(R2)+	$R1 \leftarrow R1 + Mem[R2]; R2 \leftarrow R2+d$
Pre-decrement	Add R1,-(R2)	$R2 \leftarrow R2-d; R1 \leftarrow R1 + Mem[R2]$
Scalata	Add R1,100(R2)[R3]	$R1 \leftarrow R1 + Mem[100+R2+R3*d]$

Utilitatea adresarilor Post-increment, Pre-decrement, Scalata?

Utilizarea Modurilor de Adresare (se ignora modul registru)

Trei programe masurate pe un calculator cu toate modurile de adresare (VAX)

- Deplasare:	42% medie, 32%-55%,	↑ 75%	↑ 85%
- Imediat:	33% medie, 17% - 43%	↓	
- Registru indirect:	13% medie, 3% - 24%	↓	
- Scalate:	7% medie, 1% -16%		
- Memorie indirect:	3% medie, 1% - 6%		
- Diverse:	2% medie, 0% - 3%		

75% cu deplasare si imediate

88% cu deplasare, imediate si registru indirect.

Dimensiunea Campului Deplasare?

Media asuratorilor efectuate pe programele 5SPECint92 si 5SPECfp92:

- 1% din adrese >16 biti

- 12-16 biti sunt necesari pentru deplasare.

Dimensiunea pentru campul Imediat?

- 50% - 60% se incadreaza in 8 biti
- 75% - 80% se incadreaza in 16 biti

Adresare (Rezumat).

- Modurile de adresare a datelor care sunt importante:
Deplasare, Imediat, Registru indirect.
- Dimensiunea campului Deplasare trebuie sa fie de 12 – 16 biti.
- Dimensiunea campului Imediat trebuie sa fie de 8 – 16 biti

Intrebari.

1. Cum se poate defini arhitectura unui calculator?
2. Care sunt componentele arhitecturii unui calculator?
3. Ce reprezinta Arhitectura Setului de Instructiuni? Exemple.
4. Ce reprezinta organizarea unui calculator? Exemplu.
5. Nivelurile de reprezentare a unui calculator.
6. Etapele/ciclurile derularii unei instructiuni.
7. Organizarea unui calculator la nivel de magistrale.
8. Clase de arhitecturi de seturi de instructiuni.
9. Arhitecturi bazate pe registre generale: CISC, RISC.
10. Adresarea memoriei. Big Endian, Little Endian.
11. Exemple de moduri de adresare: Vax 11/780, MIPS3000