CN1- Cursul 8.
Arhitectura unui calculator

Termeni de baza:

Arhitectura: "Arta sau stiinta de a construi structuri locuibile".

Structurile in domeniul IT: Sistemele de Calcul.

Locuitorii: Programele calculatoarelor.

Arhitectura Calculatoarelor: arta/stiinta de a descrie si a construi sisteme de calcul, care pot executa programe de calculator sau, cu alte cuvinte, structura si comportamentul calculatorului vazute de catre programatorul la nivelul limbajului de asamblare.

Organizarea Calculatorului: descrierea componentelor principale logice si fizice utilizate pentru a construi un calculator, cat si modurile de interactiune ale acestora.

Implementarea Calculatorului: descrierea componentelor electronice utilizate pentru a construi un calculator, cat si interconectarea acestora in vederea executiei programelor de catre calculator.

Modelul de executie/operare al unui calculator: descrierea executiei unui program in cadrul unei arhitecturi de calculator.

Arhitectura setului de instructiuni: descrierea instructiunilor unui calculator la nivel sintactic si semantic.

Caracteristica dimensionala (μm): latimea minima, posibila, a portii unui tranzistor CMOS, intr-o anumita tehnologie.

Inca de la aparitia primelor calculatoare operationale s-au statuat

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*
- *conditiile de exceptie.*

Setul de instructiuni realizeaza interfata intre software si hardware:

Software

Setul de Instructiuni

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:

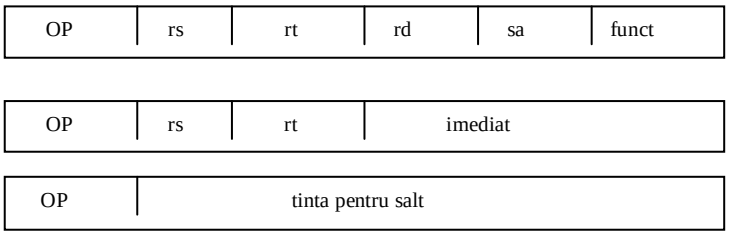
R0 – R31

PC

HI

LO

Trei Formate de Instructiuni cu lungimea de 32 de biti.



Organizarea calculatorului se refera la:

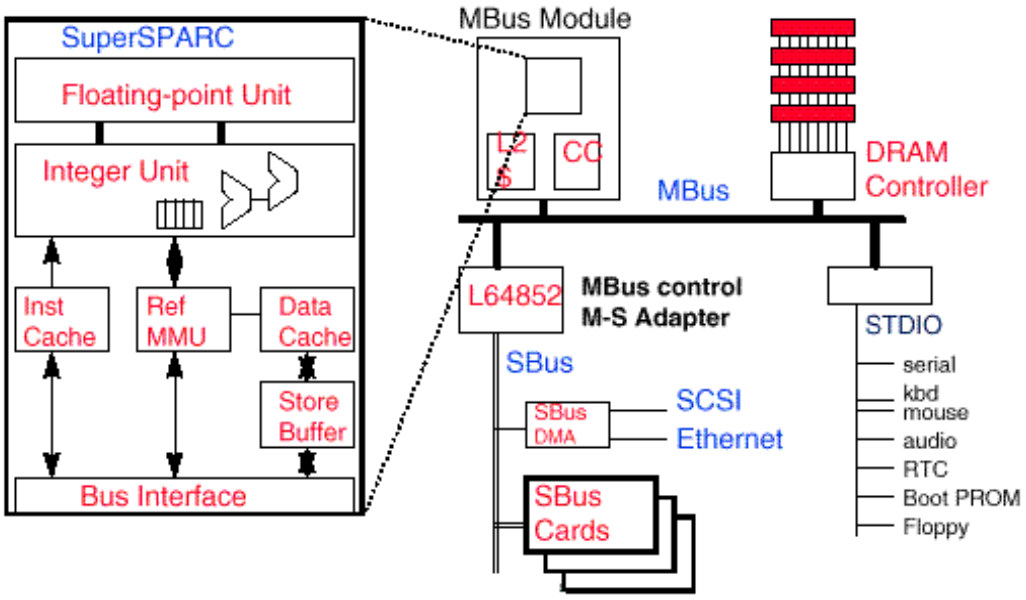
- Capabilitatile, Performantele si Caracteristicile principalelor Unitati Functionale (ex.: Registe, UAL, Unitati Logice, Circuite de Deplasare,…)
- Modurile in care aceste componente sunt interconectate;
- Fluxul informatiei intre componente;
- Logica si mijloacele folosite pentru controlul fluxului 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:

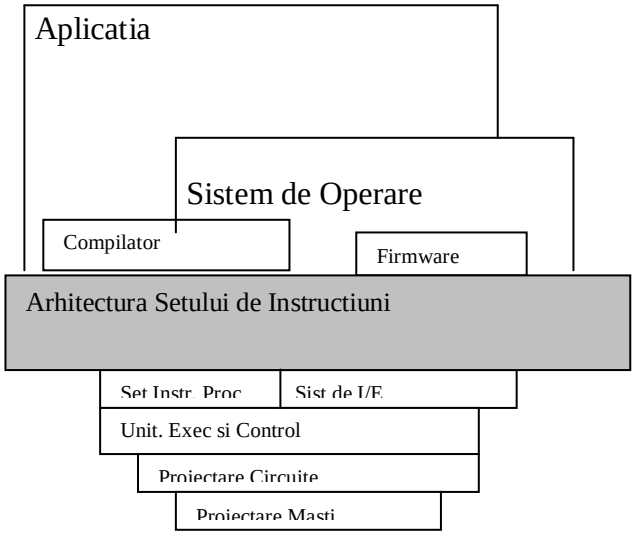
Nivel ASI ⟷ Unitati Functionale si Interconexiuni

Exemplu de Organizare:

TI SuperSPARC TMS390Z50 din statia Sun SPARCstation20:



“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:

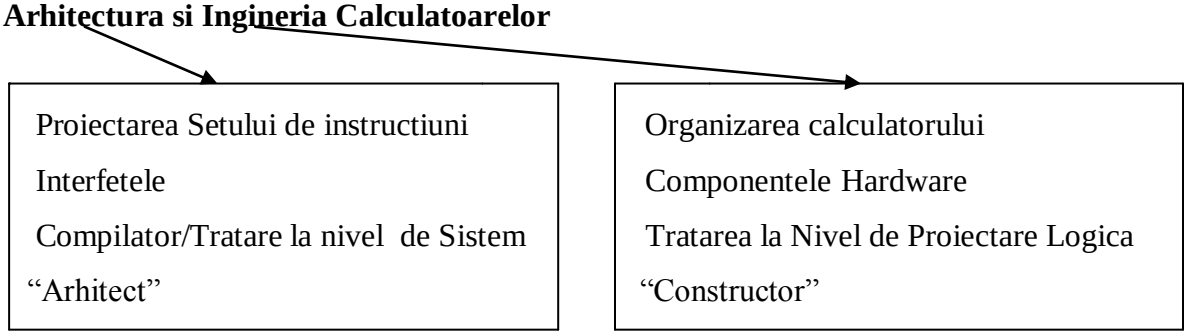
- Procesoarele:
  - Capacitatea logica creste cu circa 30%/an;
  - Frecventa ceasului creste cu circa 20%/an.
- Memoria:
  - Capacitatea memoriei Dinamice DRAM creste cu circa 60%/an
  - Viteza memoriei creste cu circa 10%/an
  - Costul pe bit scade cu circa 25%/an.
- Disc:
  - Capacitatea creste cu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

Capacitatea circuitelor de memorie DRAM:

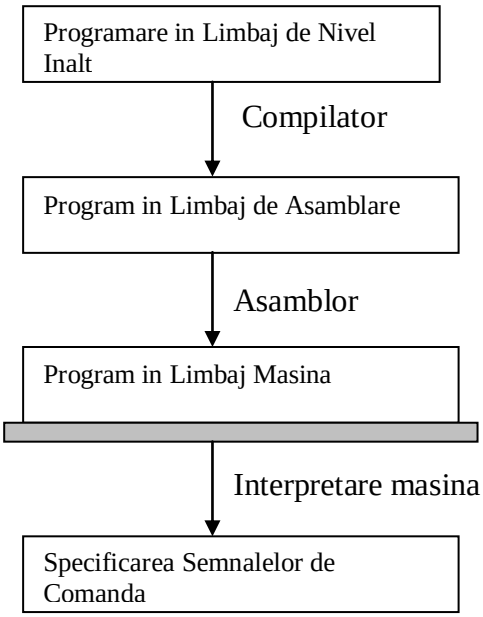
Anul :	1980	1983	1986	1989	1992	1996	1999	2001
Capacitatea:	64Kb	256Kb	1Mb	4Mb	16Mb	64Mb	256Mb	1Gb

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



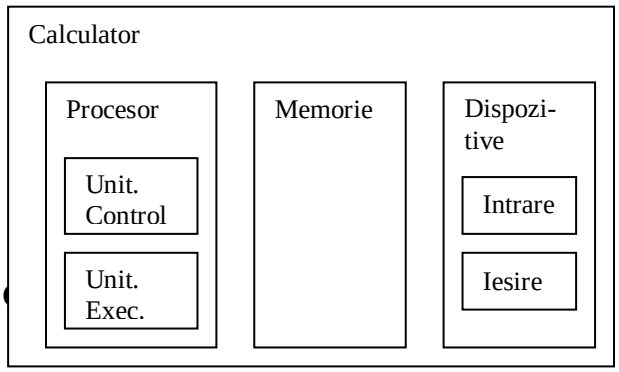
Niveluri de reprezentare/abordare




```
temp=v[k];
v[k]=v[k+1];
v[k+1]=temp;

lw$15, 0($2)
lw$16, 4($2)
sw $16,0($2)
sw$15,4($2)

0000 1001 1100 0110 1010 1111 0101 1000
… ……. ……. ……. …….. ……
0101 1000 0000 1001 1100 0110 1010 1111

UAL[1:3] <= RI[9:11] ^ MASK
```

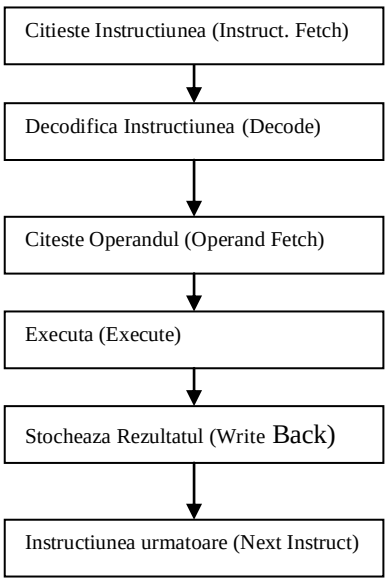
Niveluri de Organizare

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%



Citeste Instructiunea din Memoria pentru program .

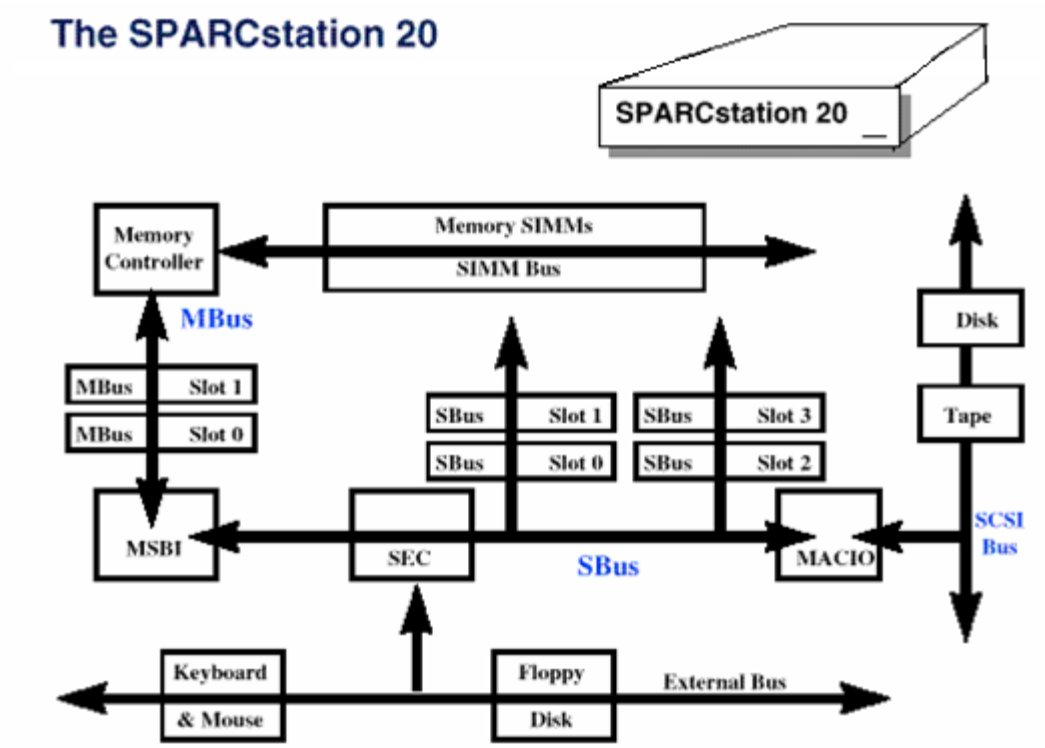
Stabileste actiunile necesare si dimensiunea instructiunii.

Localizeaza si extrage operandul/data

Calculeaza valoarea rezultatului sau starea

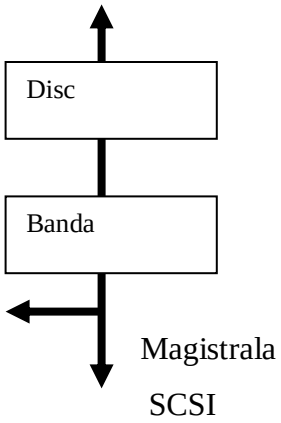
Stocheaza rezultatul in memorie/registru pentru utilizare ulterioara.

Stabileste instructiunea urmatoare



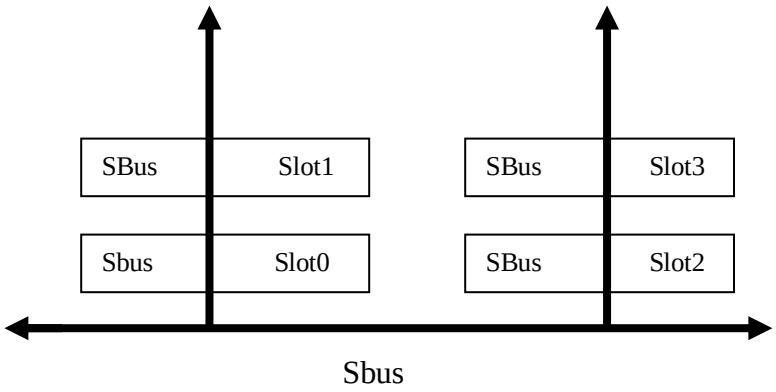
Dispozitive Standard de I/E (SPARCstation 20 –SS20)

- 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



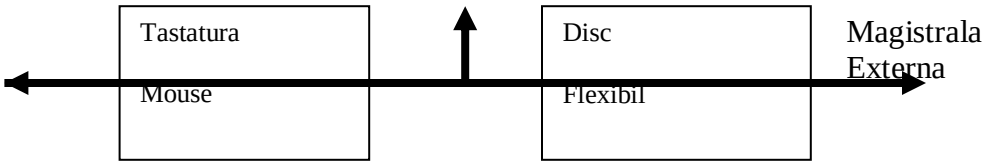
Dispozitive de I/E rapide

- SBus reprezinta magistrala proprietara SUN, pentru dispozitive rapide de I/E
- SS20dispune de patru conectori SBus, pentru dispozitivele de I/E
- Exemple: accelerator grafic, adaptor video etc..
- Termenii de viteza ridicata si viteza coborata sunt relativi



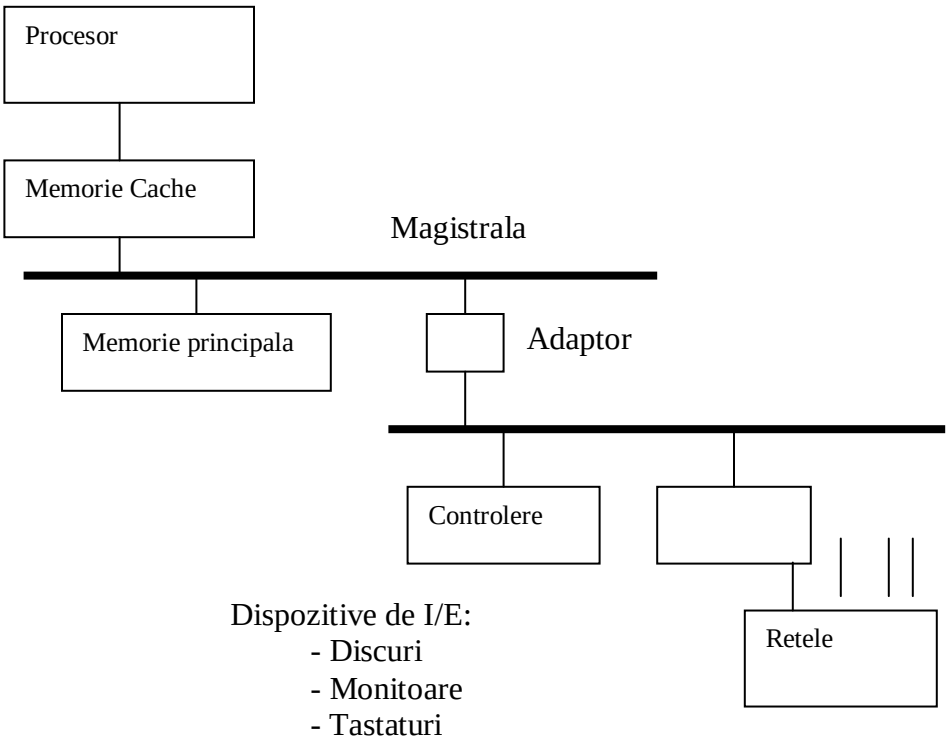
Dispozitive de I/E lente

- 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.

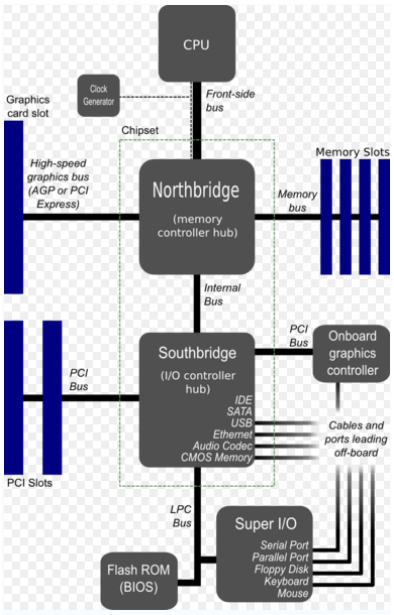


Rezumat

- 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



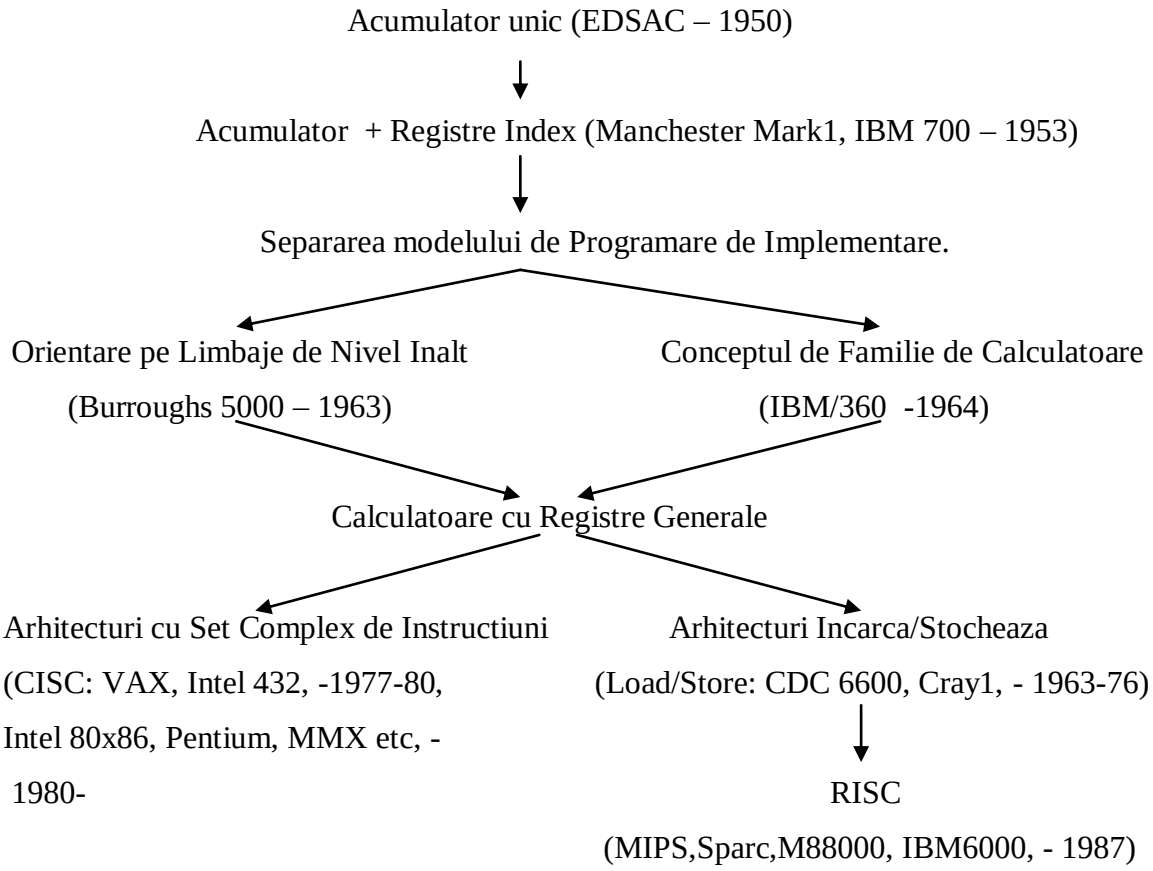
În figura de mai jos se prezintă o soluție modernă de cuplare a unității centrale de prelucrare (UCP) cu memoria internă, cu acceleratorul grafic şi cu echipamentele periferice. Setul de circuite, care asigură aceste funcții este constituit din secțiunea "Northbridge" (memory controller hub) şi "Southbridge" (I/O controller hub). "Northbridge" realizează legătura cu magistrala memoriei interne, magistrala de mare viteză pentru acceleratoarele grafice şi magistrala internă, care face legătura cu secțiunea "Southbridge". "Southbridge" implementează conexiunile cu magistralele PCI (Peripheral Computer Interface) şi LPC (Low Pin Count), cât şi cu magistralele IDE (Integrated Drive Electronics), SATA (Serial- Advanced Technology Attachment), USB (Universal Serial Bus), Ethernet s.a.



Soluție modernă de cuplare a UCP cu memoria internă, cu acceleratorul grafic şi cu echipamentele periferice.

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

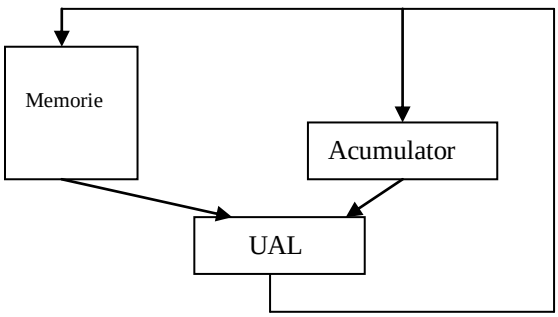
Evolutia Arhitecturii Setului de Instructiuni.



Acumulator (un registru):

1 adresa	add A	acc ← acc + mem[A]
1 adresa + x *)	addx A	acc ← acc + mem[A+x]

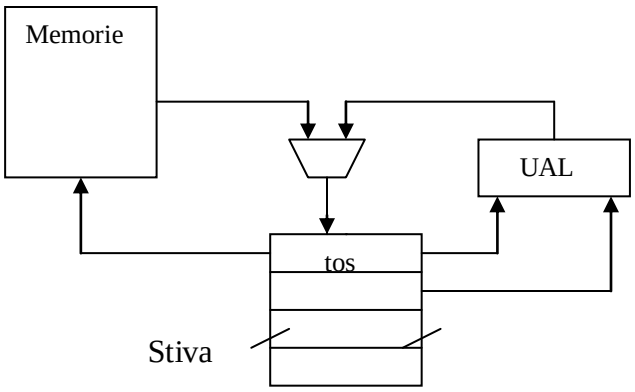
*) x este registru undex



Fluxul datelor

Stiva:

0 (zero) adrese add tos ←tos + urmator



Fluxul datelort

Registre Generale (poate fi memorie/memorie)

2 adrese add A B EA[A] ← EA[A] + EA[B]

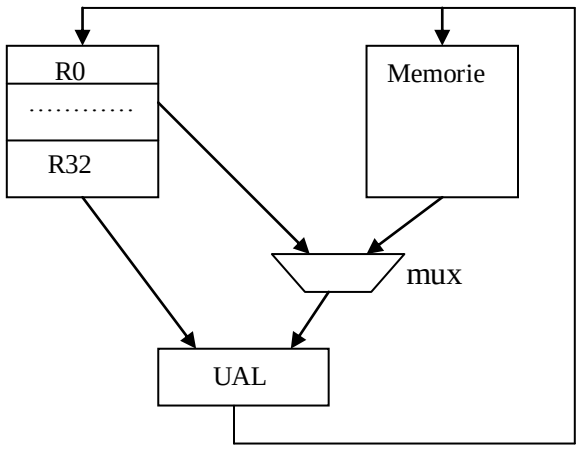
3 adrese add A B C EA[A] ← EA[B] + EA[C]

Incarca/Stocheaza: (Load/Store)

3 adrese add Ra Rb Rc Ra ← Rb + Rc

2 adrese load Ra Rb Ra ← mem[Rb]

store Ra Rb mem[Rb} ← 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

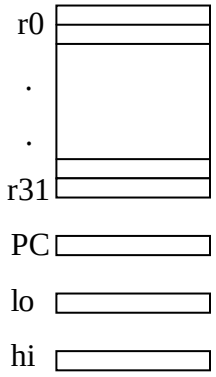
Stiva	Acumulator	Registre (reg-mem)	Registre (citeste/stocheaza)
Push A	Load A	Load R1, A	Load R1, A
Push B	Add B	Add R1, B	Load R2, B
Add	Store C	Store C, R1	Add R3, R1, R2
Pop C			Store C, R3

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

Registrele 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

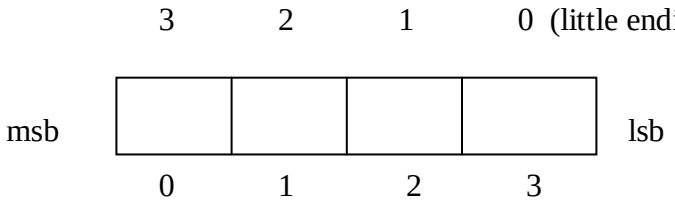


Adresarea Memoriei

- Incepand cu 1980 cele mai mule calculatoare utilizeaza adrese la nivel de octet – byte
- Pentru proiectarea ASI apar doua probleme:
  - Un cuvant 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 cuvant de la o adresa de octet. Cum se mapeaza adresele de octeti in cuvinte?
  - Un cuvant poate fi plasat la oricare pozitie de octet?

Adresarea Obiectelor: "Endiani" si Aliniere.

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



(big endian byte 0)

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 ←R4+R3
Imediat	Add R4,#3	R4 ← R4+3
Deplasata	Add R4,100(R1)	R4 ← R4+Mem[100+R1]
Registru Indirect	Add R4,(R1)	R4 ← R4+Mem[R1]
Indexat/Bazat	Add R3,(R1+R2)	R3 ←R3+Mem[R1+R2]
Direct sau Absolut	Add R1,(1001)	R1 ←R1 + Mem[1001]
Memorie Indirect	Add R1,@(R3)	R1 ← R1 + Mem[Mem[R3]]
Post-increment	Add R1,(R2)+	R1 ← R1 + Mem[R2]; R2 <- R2+d
Pre-decrement	Add R1,-(R2)	R2 ← R2-d; R1 <- R1 + Mem[R2]
Scalata	Add R1,100(R2)[R3]	R1 ←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%
- Imediat: 33% medie, 17% - 43% 85%
- 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 immediate

88% cu deplasare, immediate 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. Reprezentara unui calculator- Niveluri.
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

Performanta unui Calculator.

Utilizatorul unui calculator este interesat in reducerea cat mai accentuata a timpului de executie a unui program. Intervalul de timp intre momentul lansarii in executie a unui program si momentul terminarii executiei acestuia mai poarta numele de: timp de raspuns, timp de executie, latenta.

Managerul unui centru de calcul este interesat de productivitatea sistemului: numarul de job-uri executate in unitatea de timp, indicator care mai poarta numele de productivitate sau largime de banda.

Termenii timp de executie, timp de raspuns si productivitate sunt utilizati cand se examineaza intregul job, in timp ce termenii latenta si largime de banda au relevanta in raport cu memoria.

Exemplul 1. Urmatoarele masuri de modernizare a unui sistem pot contribui la cresterea productivitatii, scaderea timpului de raspuns sau a ambelor?

1. Cresterea frecventei ceasului.
2. Procesoare multiple pentru task-uri diferite (ex: gestiunea rezervarilor de bilete de avion, pentru o tara intreaga).
3. Prelucrare paralela pentru probleme cu caracter stiintific.

Raspuns: 1 si 3 imbunatatesc timpul de raspuns si productivitatea. 2 creste productivitatea.

Uneori aceste masuratori sunt mai bine descrise cu ajutorul distributiilor de probabilitate decat cu valori constante. De exemplu, terminarea unei operatii de I/E la un disc, in termeni de timp de raspuns depinde de o serie de factori nondeterministi, cum ar fi starea discului in momentul lansarii cererii, numarul de cereri de I/E aflate in coada de asteptare etc. Intrucat aceste valori sunt variabile are sens sa se vorbeasca de timpul mediu de raspuns al unitatii de disc.

La compararea unor proiecte alternative se afirma, de exemplu, ca “sistemul X este mai rapid decat sistemul Y”, ceea ce inseamna ca timpul de executie pe sistemul X este mai mic decat cel de executie pe sistemul Y sau ca “X este cu *n*% mai rapid decat Y”, adica:

$$\text{Timp de executie Y} / \text{Timp de executie X} = 1 + n/100$$

Intrucat timpul de executie este inversul performantei se pot scrie urmatoarele relatii:

1 + n/100 = Timp de executie Y/ Timp de executie X = PerformantaX/ PerformantaY

Cresterea performantei cu *n* mai poate fi descrisa si astfel:

n = 100(PerformantaX- PerformantaY)/ (PerformantaY)

Exemplul 2. Daca sistemul X executa un program in 10 s iar sistemul Y il executa in 15 s, cu cat este mai rapid sistemul X decat sistemul Y?

Raspuns: Deoarece.

n = 100(PerformantaX- PerformantaY)/ (PerformantaY)

rezulta ca: n = 100(1/10 – 1/15)/ (1/15) = 100(15/10 -1) = 50

Sistemul X este mai rapid cu 50% decat sistemul Y.

Principii Cantitative in Proiectarea Calculatoarelor. Legea lui Amdahl.

Unul din principiile cele mai importante ale proiectarii calculatoarelor este acela de a face cat mai rapid cazul cel mai frecvent intalnit. In procesul de proiectare trebuie sa se aprecieze care caz este cel mai frecvent intalnit si cu cat va creste peformanta in situatia in care acest caz va fi accelerat.

Pentru cuantificarea acestui principiu se poate folosi **Legea lui Amdahl.** Cresterea de performanata obtinuta prin modernizarea unei parti a unui calculator se poate calcula folosind Legea lui Amdahl, care stabileste ca marirea performantei, utilizand un mod de executie mai accelerat, este limitata de fractiunea de timp in care acest mod rapid de operare este folosit.

Legea lui Amdahl defineste cresterea de viteza/accelerarea (speedup) care se poate obtine folosind o solutie mai moderna, pentru o parte a masinii.

Cresterea de viteza = (Performanta pentru intregul task folosind solutia moderna, cand este posibil)/(Performanta pentru intregul task fara solutia moderna)

sau

Cresterea de viteza = (Timpul de executie pentru intregul task fara solutia moderna)/ (Timpul de executie pentru intregul task folosind solutia moderna, cand este posibil)

Exemplul 3. Se considera ca trebuie parcurs un traseu constituit dintr-un segment de 200 Km, care poate fi strabatut pe jos, in 50 de ore, sau cu ajutorul a 4 mijloace de transport A,B,C,D si un alt segment care va fi parcurs numai pe jos in 20 de ore. Mijloacele de transport asigura urmatoarele viteze:

- A: 10 Km/h
- B: 50 Km/h
- C: 100 Km/h
- D: 500 Km/h

In cat timp va fi parcurs intregul traseu folosind toate mijloacele si care va fi cresterea de viteza la parcurgerea intregului traseu cu cele 4 mijloace de transport in raport cu parcurgerea pe jos a traseului?

Raspuns: Se determina cat va dura parcurgerea segmentului de 200Km in cele 5 situatii, dupa care se adauga cele 20 ore pentru portiunea care trebuie parcursa pe jos. Rezultatele sunt date in tabelul de mai jos:

Mijloace pentru segmentul de 200Km	**Ore pentru segmentul de 200Km**	**Accelerarea pentru segmentul de 200Km**	**Ore pentru intregul traseu**	**Accelerarea pentru intregul traseu**
Pe jos	50	1	70	1
A	20	2,5	40	1,75
B	4	12,5	24	2,91
C	2	25	22	3,18
D	0,4	125	20,4	3,43

Legea lui Amdahl pleaca de la ideea ca timpul de executie al unui task, utilizand sistemul original modernizat, va fi timpul de executie folosind sectiunea din masina nemodernizata plus timpul de executie utilizand masina modernizata.

$$\text{Timpul de executie}_{nou} = \text{Timpul de executie}_{vechi} * ((1 - \text{Fractiunea}_{modernizata}) + \frac{\text{Fractiunea}_{modernizata}}{\text{Accelerarea}_{modernizata}})$$

$$\text{Accelerarea}_{globala} = \frac{\text{Timpul de executie}_{vechi}}{\text{Timpul de executie}_{nou}} = \frac{1}{(1 - \text{Fractiunea}_{modernizata}) + \frac{\text{Fractiunea}_{modernizata}}{\text{Accelerarea}_{modernizata}}}$$

Exemplul 4: Fractiunea de 40% modernizata din sistemul original opereaza de 10 ori mai repede decat sistemul nemodernizat original, intr-un interval de timp egal cu 40% din timpul total de executie. Care va fi accelerarea globala obtinuta?

Raspuns:

$\text{Fractiunea}_{modernizata} = 0,4$

$\text{Accelerarea}_{modernizata} = 10$

$\text{Accelerarea}_{globala} = 1 / (0,6 + 0,4/10) = 1/0,64 = 1,56$

Se constata ca daca se utilizeaza sistemul modernizat numai pentru o fractiune din task accelerarea nu poate fi mai mare decat inversul lui:

" $1 - \text{fractiune}_{modernizata}/\text{Accelerarea}_{modernizata}$ "

Exemplul 5. Se presupune ca viteza UCP dintr-un sistem se poate mari de 5 ori, fara a afecta performanta subsistemului de I/E, la un cost de 5 ori mai mare fata de cel al UCP-ului nemodernizat. UCP este utilizat efectiv 50% din timp, restul de 50% din timp fiind in starea de asteptare pentru efectuarea operatiilor de I/E. Daca costul UCP-ului este 1/3 din costul toatal al sistemului cresterea vitezei UCP-ului de 5 ori reprezinta o buna investitie din punctul de vedere al indicatorului cost/performanta?

Raspuns:

$\text{Accelerarea} = 1 / (0,5 + 0,5/5) = 1/0,6 = 1,67$

$\text{Noul cost} = 2/3 * 1 + 1/3 * 5 = 2,33 * \text{Vechiul cost}$

Astfel, cresterea costului este mai mare decat cresterea performantei, indicatorul cost/performanta nefiind ameliorat.

SPEC

(SPEC) (Standard Performance Evaluation Corporation, Warrenton, VA, www.specbench.org).

O organizatie fondata in 1988 pentru stabilirea unor teste de performanta standard pentru calculatoare. Primul test utilizat s-areferit la evaluarea unui singur procesor ca "SPECmark", in care un SPECmark a fost echivalent ca performanta cu un sistem VAX 11/780. Desi testele de performanta SPEC se folosesc in continuare pentru evaluarea procesoarelor, SPEC are o varietate de teste pentru masurarea subsistemelor grafice, Java,, serverele de web, serverele de mail, serverele de aplicatii si serverele de fisiere. Mai jos se dau numele testelor de performanta curente si ale celor pana de curand utilizate.

	SPEC CPU Benchmark for	
Current	**Integer**	**Floating Point**
SPEC CPU2006	CINT2006	CFP2006
Retired		
SPEC CPU2000	CINT2000	CFP2000
SPEC CPU95	SPECint95	SPECfp95

Benchmarks

Current

- SPECapc for 3ds Max™ 2011, performance evaluation software for systems running Autodesk 3ds Max 2011.
- SPECapcSM for Lightwave 3D 9.6, performance evaluation software for systems running NewTek LightWave 3D v9.6 software.
- SPEC CPU2006, combined performance of CPU, memory and compiler.
    - CINT2006 ("SPECint"), testing integer arithmetic, with programs such as compilers, interpreters, word processors, chess programs etc.
    - CFP2006 ("SPECfp"), testing floating point performance, with physical simulations, 3D graphics, image processing, computational chemistry etc.

- SPECjbb2005, evaluates the performance of server side Java by emulating a three-tier client/server system (with emphasis on the middle tier).
- SPECjEnterprise2010, a multi-tier benchmark for measuring the performance of Java 2 Enterprise Edition (J2EE) technology-based application servers.
- SPECjms2007, Java Message Service performance
- SPECjvm2008, measuring basic Java performance of a Java Runtime Environment on a wide variety of both client and server systems.
- SPECapc, performance of several 3D-intensive popular applications on a given system
- SPEC MPI2007, for evaluating performance of parallel systems using MPI (Message Passing Interface) applications.
- SPEC OMP2001 V3.2, for evaluating performance of parallel systems using OpenMP (http://www.openmp.org) applications.
- SPECpower_ssj2008, evaluates the energy efficiency of server systems.
- SPECsfs2008, File server throughput and response time supporting both NFS and CIFS protocol access
- SPECsip_Infrastructure2011, SIP server performance
- SPECviewperf 11, performance of an OpenGL 3D graphics system, tested with various rendering tasks from real applications
- SPECvirt_sc2010 ("SPECvirt"), evaluates the performance of datacenter servers used in virtualized server consolidation environments

Future

- SOA: according to SPEC's web site in late 2010, a subcommittee is investigating benchmarks for Service Oriented Architecture (SOA).

Culture

SPEC attempts to create an environment where arguments are settled by appeal to notions of technical credibility, representativeness, or the "level playing field". SPEC representatives are typically engineers with expertise in the areas being benchmarked. Benchmarks include "run rules", which describe the conditions of measurement and documentation requirements. Results that are published on SPEC's website undergo a peer review by members' performance engineers.