

Arhitectura calculatoarelor

Arhitectura calculatoarelor



EDITURA UNIVERSITARĂ
Bucuresti, 2014

Redactor: Gheorghe Iovan
Tehnoredactor: Carmen Butnic
Coperta: Monica Balaban
Figurile: Mihai Petcu

Editură recunoscută de Consiliul National al Cercetării Stiințifice (C.N.C.S.) și inclusă de Consiliul National de Atestare a Titlurilor, Diplomelor și Certificatelor Universitare (C.N.A.T.D.C.U.) în categoria editurilor de prestigiu recunoscut.

Descrierea CIP a Bibliotecii Nationale a României

BAICU, FLOAREA

Arhitectura calculatoarelor / Floarea Baicu, Liviu
Gabriel Baicu. - București : Editura Universitară, 2014 ISBN 978606-591-987-7
I. Baicu, Liviu Gabriel 004

DOI: (Digital Object Identifier): 10.5682/9786065919877

© Copyright 2014

Toate drepturile prezentei ediții sunt rezervate autorilor. Nici o parte din lucrare nu poate fi reprodusă, stocată sau transmisă, indiferent de formă, fără acordul prealabil scris al autorilor.

Copyright © 2014 Editura Universitară
Editor: Vasile Muscalu
B-dul. N. Bălcescu nr. 27-33, Sector 1, București Tel.: 021 –
315.32.47 / 319.67.27 www.editurauniversitara.ro
email: redactia@editurauniversitara.ro

Distribuție: tel.: 021-315.32.47 / 319.67.27 / 0744 EDITOR / 07217 CARTE comenzi@editurauniversitara.ro O.P.
15, C.P. 35, București www.editurauniversitara.ro

Prefață

A scrie o carte într-un domeniu în care în fiecare săptămână apare ceva nou, în fiecare lună este o întâlnire, o conferință, o lansare de produs nou, este o provocare, fără prea mulți sorți de izbândă. Și totuși, această carte apare acum și conține noutăți de ultimă oră, alături de principii fundamentale și adevăruri stabilite cu mult timp înainte.

*Bazele arhitecturii calculatoarelor moderne au fost puse acum aproape 70 ani de genii creatoare precum Von Neumann, Alan Turing și colaboratorii lor. Tot ce se face acum în domeniul arhitecturii calculatoarelor se bazează pe **principiul programului memorat** și se îmbunătățește permanent, se pun la punct dispozitive noi de prelucrare și stocare a informațiilor, pentru rezolvarea facilă a oricărei probleme matematice, tehnice sau de altă natură. Toate noile apariții în domeniul tehnicii de calcul, a limbajelor de programare sau de asamblare sunt rezolvări optimizate ale unor probleme de evoluție firească în acest domeniu.*

Dezvoltarea limbajelor noi, a microelectronicii și tehnologiilor de realizarea circuitelor digitale integrate la scară foarte largă și mult miniaturizate, oferă soluții tehnice de neimaginat acum 70 ani pentru a răspunde unor necesități din ce în ce mai pretențioase.

Există principii de bază care au rămas valabile, precum reprezentarea numerelor în binar, aritmetica acestora și algebra booleană, modul de calcul a erorilor și nevoia de a diminua acestor erori, de a îmbunătăți precizia, viteza de procesare și securitatea informațiilor prelucrate.

Modul de implementare al operațiilor cu ajutorul dispozitivelor semiconductoare VLSI este diferit, mult mai facil, mai rapid și cu consumuri mult reduse față de acum 70 ani.

Au apărut numeroase tipuri de microprocesoare, unele de uz general, altele cu funcții ultraspecializate, cu nenumărate instrucțiuni pe care se bazează realizarea funcțiilor lor (peste 15000 la Intel). Totuși cele mai performante procesoare lucrează cu aceleași tipuri de instrucțiuni (aritmetice, logice, de transfer, deplasare și rotire a datelor, de ramificare și control) ca și acum 40 ani. Ce s-a schimbat este viteza de lucru și capacitatea de procesare, ceea ce presupune un management mai bun al capacităților de procesare și stocare.

Modul de implementare tehnică a acestor instrucțiuni este în continuare o provocare. Miniaturizarea, virtualizarea, creșterea capacității de prelucrare și a vitezei de răspuns sunt preocupări majore la nivel mondial, preocupări cu investiții majore de timp, bani, inteligență, dar și cu realizări pe măsura efortului.

Referitor la memorii, apar mereu noi tipuri de memorii semiconductoare, cu funcții speciale, toate sunt însă îmbunătățiri pornind de la clasicele ROM și RAM, de la celula elementară de memorie cu porți logice.

Dispozitivele de intrare/ieșire se îmbunătățesc permanent adaptate la cerințele de piață și exigențele tinerilor utilizatori, aceasta impune optimizarea interfețelor și modului de acces.

A scrie o carte despre un domeniu atât de dinamic este o mare provocare, și totuși am îndrăznit în idea că va ajunge în mâna cuiva care va ști ce să facă cu ea, care va folosi cunoștințele, entuziasmul și experiența autorilor puse în carte în scop creator și că va aduce o mică contribuție la dezvoltarea armonioasă a științei calculatoarelor.

Martie 2014, București

AUTORII

Introducere

Arhitectura calculatoarelor este știința și arta de a proiecta și construi calculatoare, știință care are două componente principale: *Arhitectura Setului de Instrucțiuni*, numită în engleză *Instruction Set Architecture* – ISA, care definește codurile prin care procesorul citește și acționează pentru prelucrarea datelor de la utilizator și cea de-a doua componentă care se ocupă cu studiul *organizării* elementelor de prelucrare și stocare a datelor, a modului în care sunt implementate seturile de instrucțiuni și căilor de comunicare între aceste elemente.

Este o știință relativ nouă și este considerată o ramură a tehnologiei informației, deși cel mai vechi mecanism cunoscut care funcționa ca o mașină de calcul este considerat mecanismul din Antikythera care datează din anul 87 î.e.n. De-a lungul timpului au fost preocupări susținute pentru dezvoltarea sistemelor de calcul, dar în ultimii 70 de ani a cunoscut o dezvoltare extrem de rapidă, astfel încât azi nu ne mai putem imagina viața fără ajutorul sistemelor de calcul și nici nu știm cum am putut trăi fără ele!

Arhitectura calculatoarelor, ca știință modernă, se consideră a fi inițiată de John von Neumann în 1945, la Universitatea din Princeton. Majoritatea calculatoarelor moderne funcționează pe baza *principiului programului memorat* care consideră că orice problemă care se dorește a fi rezolvată de către calculator trebuie prezentată sub forma unei succesiuni de operații, a unui algoritm de calcul. Conform lui Von Neumann, o instrucțiune și datele cu care operează sunt codificate în același mod, păstrate în aceeași memorie dar la adrese diferite și folosesc aceleași magistrale.

Arhitectura Harvard este o altă arhitectură pentru calculatoare, mai puțin utilizată, care consideră că instrucțiunile și datele sunt codificate diferit și **13** folosesc seturi diferite de magistrale, unul pentru scrierea și citirea datelor în/din memorie și un alt set de magistrale pentru preluarea instrucțiunilor.

Pornind de la principiul Von Neumann se consideră o structură a calculatorului în cinci unități:

1. Unitatea de Intrare – IU (*Input Unit*);
2. Unitatea de Memorie –MU (*Memory Unit*);
3. Unitatea Aritmetico-Logică –ALU (*Aritmetic and Logic Unit*);

4. Unitatea de control –UC (*Control Unit*) și

5. Unitatea de ieșire –OU (*Output Unit*)

În mod uzual Unitatea Aritmetico-Logică și Unitatea de Control sunt cuprinse în procesorul computerului, în Unitatea Centrală de Procesare – CPU (*Central Processing Unit*), iar unitățile de intrare și cele de ieșire se consideră împreună, așa că se discută de structura în 3 unități a calculatorului:

1. Unitatea Centrală de Procesare (CPU);
2. Unitatea de Memorie (MU);
3. Unitatea de Intrare/Ieșire (I/OU).

Pentru a putea înțelege arhitectura unui calculator electronic, în primul capitol am prezentat principiile de bază care stau la baza arhitecturii calculatorului electronic, principiile și structura Von Neumann a calculatorului, structura fizică și funcțională a unui calculator, prezentând pe scurt modul de funcționare al fiecărei unități componente și de interacțiune între ele, organizarea pe nivele logice a calculatorului, considerat ca o succesiune de mașini virtuale care funcționează pe baza unor limbaje specifice.

Capitolul 2 al cărți prezintă modul de *reprezentare a numerelor* în calculatorul electronic, conversia între bazele de numerație uzuale, operații aritmetice de bază efectuate în binar. Am prezentat modul de reprezentare a numerelor cu virgulă fixă și mobilă (*floating point*), interval și precizie în ambele reprezentări a numerelor și limitările impuse de precizia ridicată și intervalul de reprezentare corelate cu capacitatea de prelucrare a calculatorului, cunoștințe necesare oricărui specialist în știința calculatoarelor.

Am insistat asupra erorilor în reprezentarea numerelor în calculator, caracteristicile considerate la reprezentarea numerelor în virgulă mobilă, coduri detectoare și corectoare de erori, respectiv bitul de paritate, codul Hamming, CRC și am prezentat principalele coduri de caractere utilizate în calculator (ASCII, EBCDIC, UNICODE).

Un capitol amplu în această carte îl reprezintă capitolul 3, *Circuite logice*, circuite electronice digitale care stau la baza realizării tuturor componentelor hardware ale calculatorului electronic, chiar dacă discutăm despre memorie, registrele procesorului, Unitate Aritmetico-Logică sau Unitate Centrală etc.

Am considerat că o bună înțelegere a funcționării calculatorului presupune o înțelegere corectă a funcțiilor logice și aritmetice realizate de calculator și modul de implementare hardware a acestor funcții prin intermediul porților logice. Am prezentat principalele funcții logice și modul de realizare cu porți logice cu dispozitive semiconductoare, vizualizarea tabelor de adevăr pentru

aceste porți cu ajutorul unui program specializat, *Circuit Maker* oferit liber în varianta *Student* pe Internet.

Am considerat important studiul parametrilor porților logice al modului de funcționare al circuitelor combinaționale de bază, respectiv codificator, decodificator, multiplexor și demultiplexor, sumator și multiplicator combinațional.

Am prezentat pe larg circuitele basculante bistabile care stau la baza realizării celulelor de memorie, a numărătoarelor, divizoarelor de frecvență și registrelor procesorului cu prezentarea ecuațiilor care descriu funcționarea lor și tabelele de stare. Au fost prezentate circuite basculate de tip R-S, Delay, T, J-K și numărătoarele sincrone și asincrone realizate cu circuite basculante bistabile, registre de deplasare.

În capitolul 4 am prezentat arhitectura setului de instrucțiuni – ISA, caracteristicile setului de instrucțiuni, formatul instrucțiunii explicitând etapele ciclului instrucțiunilor. Am prezentat principalele tipuri de instrucțiuni, exemplificând pe procesoare Intel, respectiv instrucțiuni aritmetice, logice, pentru transferul informațiilor, pentru controlul procesorului, pentru deplasarea și rotirea datelor, de ramificare pentru controlul derulării secvenței de operații și instrucțiuni pentru operații cu șiruri de date.

Am insistat asupra modurilor de adresare a instrucțiunilor, pentru adresele operanzilor de prelucrat, de stocare a rezultatului sau a următoarei instrucțiuni, respectiv: adresarea imediată, directă, indirectă, relativă și indexată.

În capitolul 5 am prezentat sistemul de memorie al calculatorului electronic începând cu modul de organizare și principiul de funcționare al memoriei calculatorului, caracteristicile, modul de programare și ștergere, persistența conținutului, modul de acces. Am prezentat pe larg memoriile ROM și RAM cu variantele lor constructive, respectiv EPROM, EEPROM, SRAM, DRAM, CAM, la fiecare dintre ele urmărind descrierea principiilor de funcționare, a caracteristicilor temporare, modului de scriere, citire, adresare, selectare, activare/dezactivare/menținere a datelor. Am prezentat pe scurt memoria *cache* cu principalele ei caracteristici, memoria externă respectiv hard-discul, flash memory, SSD-ul. Capitolul 5 se încheie cu modul de gestionare și protecție a memoriei prin mecanisme de segmentare, paginare și virtualizare.

Capitolul 6 este dedicat *Unității Centrale de Procesare* (CPU) pornind de la cele 3 funcții esențiale pe care le realizează: efectuarea de operații aritmetice și logice, transferul de date dintr-o locație în alta și luarea de decizii referitoare la prelucrarea instrucțiunii. Unitatea Centrală de Procesare este cunoscută ca procesor de uz general cu un set de instrucțiuni memorat, unicore sau multicore. Un procesor cuprinde o *Unitate de Prelucrare a datelor* formată dintr-o *Unitate Aritmetico-Logică* și un *Set de Registre*, precum și o *Unitate*

de Control. În acest capitol am prezentat pe larg registrele procesorului și modul de organizare și funcționare al acestora, modul de comunicare în interiorul CPU și cu exteriorul prin intermediul magistralelor de date, de adresă și control.

Am detaliat categorii de semnale din magistralele interne, semnalele de control pentru transferul datelor, a cererilor de cedare a controlului, semnale de sincronizare și indicatoare de stare, alte semnale utilitare.

Un spațiu important l-am alocat studiului întreruperilor și acordare a priorităților pe baza tabelului vectorilor de întreruperi.

Capitolul 7 este dedicat unității de intrare-ieșire, respectiv funcțiilor și structurii modului de I/O, echipamentelor periferice, circuitelor de interfață, tipurilor de magistrale și metodelor de transfer a datelor prin program, întreruperi, acces direct la memorie (DMA) și canale de I/O.

