ARHITECTURA CALCULATOARELOR 2003/2004
CURSUL 1

CAPITOLUL 1: Stocarea datelor

1.1 Stocarea biților

Calculatoarele utilizate în prezent reprezintă informațiile ca şiruri de biți. Un bit (binary unit - cifră binară) reprezintă una din cifrele 0 sau 1, pe care în acest context le vom considera mai degrabă simboluri decât numere. Într-adevăr, vom vedea că semnificația unui bit diferă de la o aplicație la alta. Stocarea unui bit într-un calculator necesită un dispozitiv care să poată să se afle într-una din cele două stări, ca de exemplu un întrerupător (pornit sau oprit), un releu (deschis sau închis) sau un steag de start (ridicat sau coborât). Una dintre cele două stări este utilizată pentru reprezentarea simbolului 0, iar cealaltă pentru reprezentarea simbolului 1. Scopul este să studiem modalitățile prin care sunt stocați biții într-un sistem de calcul.

1.1.1 Porți logice şi circuite basculante bistabile

Vom începe prin a reintroduce operațiile AND (ŞI), OR (SAU) şi XOR (SAU EXCLUSIV), prezentate pe scurt în figura 1.1. Aceste operații sunt similare cu operațiile aritmetice înmulțire şi adunare, care combină o pereche de valori, intrările operației, pentru a produce o a treia valoare, ieşirea (rezultatul) operației. Rețineți totuşi că singurele cifre cu care lucrează operațiile AND, OR şi XOR sunt 0 şi 1; aceste operații sunt considerate ca manipulând valorile adevărat (TRUE) şi fals (FALSE) - 1 pentru adevărat, 0 pentru fals. Operațiile care lucrează cu valorile adevărat/fals sunt denumite operații booleene (Boolean operations), după numele matematicianului George Boole.

Operația AND este proiectată să reflecte adevărul sau falsitatea unei expresii formate prin combinarea a două expresii mai mici cu ajutorul conjuncției ŞI. Aceste expresii au următoarea formă generală :

P AND *Q*

unde *P* reprezintă una dintre expresii, iar *Q* o reprezintă pe cealaltă, ca de exemplu :

Kermit este o broască AND Miss Piggy este o actriță.

Intrările operației AND sunt reprezentate de adevărul sau falsitatea componentelor expresiei; ieşirea reprezintă adevărul sau falsitatea expresiei compuse. De vreme ce o expresie de forma P AND Q este adevărată numai atunci când ambele sale componente sunt adevărate, vom trage concluzia că 1 AND 1 trebuie să dea ca rezultat 1, în timp ce toate celălalte combinații trebuie să aibă ca rezultat 0, conform celor prezentate în figura 1.1.

Similar, operația OR se bazează pe o expresie compusă de forma:

P OR *Q*

în care, din nou, *P* reprezintă o expresie iar *Q* reprezintă altă expresie. Asemenea expresii compuse sunt adevărate atunci când cel puțin una dintre componentele lor este adevărată, ceea ce coincide cu reprezentarea operației OR din figura 1.1.

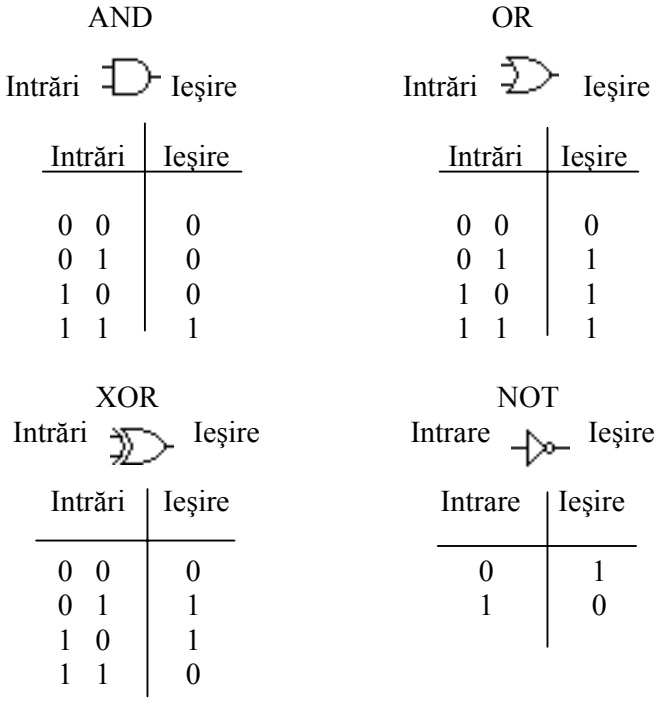
Figura 1.1 Operațiile AND, OR şi XOR

```
   0        0        1        1
AND 0    AND 1    AND 0    AND 1
   0        0        0        1

   0        0        1        1
 OR 0     OR 1     OR 0     OR 1
   0        0        0        0

   0        0        1        1
XOR 0    XOR 1    XOR 0    XOR 1
   0        1        1        0
```

Figura 1.2 Reprezentarea simbolică a porților logice AND, OR, XOR şi NOT precum şi a valorilor intrărilor şi ieşirilor acestora:



În limbajul curent, nu există o expresie care să corespunde semnificației operației XOR. XOR produce ca rezultat o ieşire cu valoarea 1 atunci când una dintre intrările sale este 1 şi cealaltă este 0. De exemplu, o propoziție de forma *P* XOR *Q* înseamnă ″sau *P*, sau *Q*, dar niciodată ambele″.

Operația NOT este altă operație booleană. Ea diferă de AND, OR şi XOR prin faptul că are o singură intrare. Ieşirea ei reprezintă opusul intrării; dacă intrarea operației NOT este adevărată, ieşirea este falsă şi viceversa. Astfel, dacă intrarea operației NOT este reprezentată de adevărul sau falsitatea propoziției:

Fozzie este un urs.

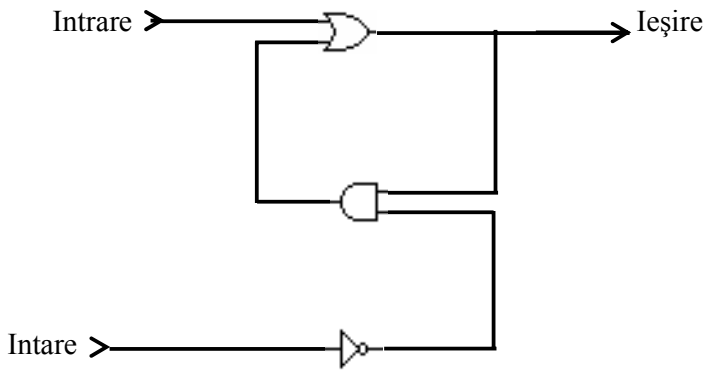
Ieşirea va reprezenta adevărul sau falsitatea propoziției:

Fozzie nu este un urs.

Un dispozitiv care produce rezultatul unei operații booleene atunci când i se aplică intrările operației este denumit poartă logică (gate). Porțile logice pot fi realizate utilizându-se o largă varietate de tehnologii, cum ar fi dispozitive mecanice, relee sau dispozitive optice. Calculatoarele de astăzi implementează de obicei porțile logice prin intermediul unor circuite electronice de mici dimensiuni în care valorile 0 şi 1 sunt reprezentate prin niveluri diferite de tensiune electrică. Dar nu ne vom preocupa acum de asemenea detalii. Pentru ceea ce ne-am propus aici, este suficient să reprezentăm porțile logice prin intermediul simbolurilor lor, prezentate în figura 1.2. Rețineți că porțile logice AND, OR, XOR şi NOT au asociate diagrame specifice, cu valorile intrărilor scrise de o parte şi cu valorile ieşirii de partea cealaltă.

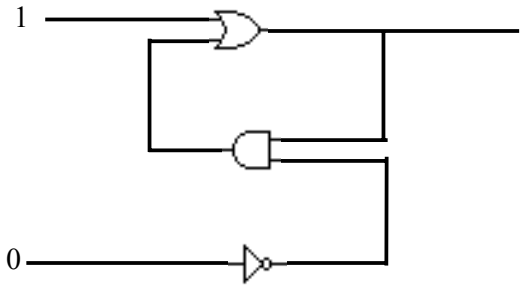
Astfel de porți logice alcătuiesc elementele constructive din care sunt realizate calculatoarele. Un pas important în această direcție este reprezentat de circuitul din figura 1.3. Acesta este un exemplu particular dintr-o clasă de circuite cunoscute sub numele de circuite basculante bistabile. Un **circuit basculant bistabil** (**flip-flop**) este un circuit care are la ieşire una din două valori posibile; ieşirea lui rămâne stabilă până când un impuls temporar de la alt circuit are ca efect comutarea lui la cealaltă valoare. Cu alte cuvinte, ieşirea trece de la o stare la cealaltă sub controlul unor stimuli externi. Atâta timp cât ambele intrări ale circuitului prezentat în figura 1.3 rămân la valoarea 0, ieşirea (fie 0 sau 1) nu se va modifica. În schimb plasarea temporară a unui 1 pe intrarea de sus va forța ieşirea să treacă în 1, în timp ce plasarea unui 1 pe intrarea de jos va forța ieşirea în 0.

Figura 1.3 Circuit basculant bistabil simplu

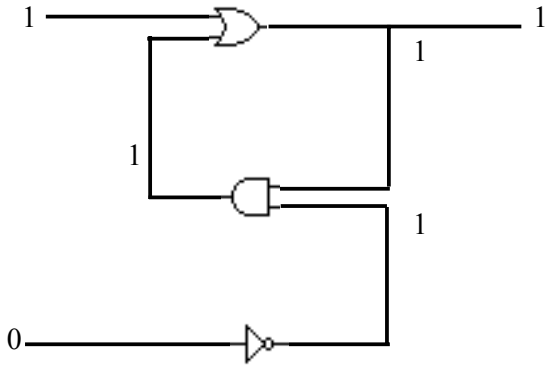


Să studiem mai îndeaproape funcționarea circuitului. Fără a cunoaşte valoarea curentă a ieşirii circuitului din figura 1.3, să presupunem că intrarea de sus este modificată la 1, în timp ce intrarea de jos rămâne în 0 (figura 1.4a). Acest fapt va avea ca efect trecerea ieşirii circuitului

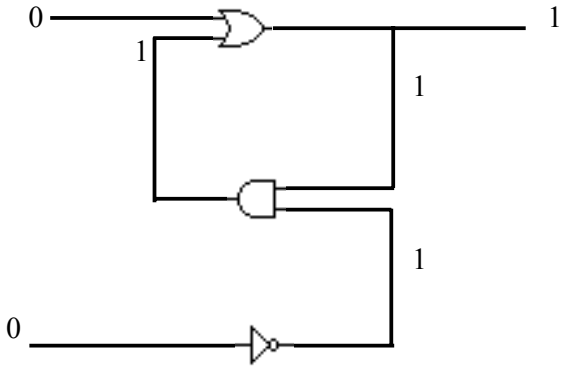
Figura 1.4 Setarea ieşirii circuitului basculant bistabil la valoarea 1:



(a) Intrarea de sus este trecută în 1



(b) Aceasta provoacă trecerea ieşirii porţii OR în 1 şi astfel ieşirea porţii AND devine 1



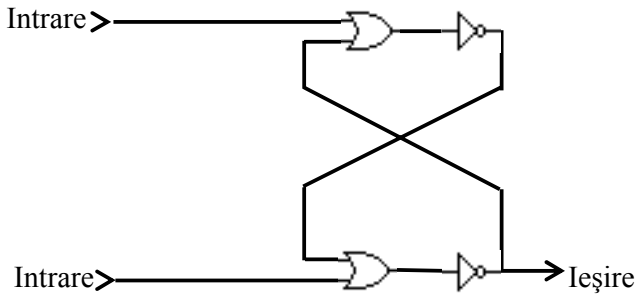
(c) Valoarea 1 de la ieşirea porţii AND menţine poarta OR în starea precedentă, după ce intrarea de sus revine la 0

OR în 1, indiferent de starea celeilalte intrări a acestei porţi logice. Ca efect, ambele intrări ale porţii AND vor fi acum 1, deoarece cealaltă intrare a acestei porţi este deja 1 (valoare obţinută prin trecerea intrării de jos a circuitului basculant bistabil prin poarta NOT). Ieşirea porţii AND va deveni acum 1, ceea ce înseamnă că a doua intrare a porţii OR va fi 1 (figura 1.4c). Pe scurt, ieşirea circuitului basculant bistabil devine 1 şi această valoare va fi menţinută şi după ce intrarea de sus revune la 0. Similar, plasarea temporară a valorii 1 pe intrarea de jos va forţa ieşirea circuitului basculant bistabil la valoarea 0, ieşirea rămânând în această stare şi după ce intrarea revine la valoarea 0.

Din punctul nostru de vedere, importanța unui circuit basculant bistabil constă în faptul că acesta este ideal pentru stocarea unui bit în calculator. Valoarea stocată într-un circuit basculant bistabil poate fi **citită şi modificată** cu uşurință prin intermediul altor circuite electronice. În plus, circuitele basculante bistabile pot fi realizate la dimensiuni atât de mici, încât milioane de asemenea circuite pot fi plasate pe un **cip** de mărimea unei unghii.

Porțile logice şi circuite precum circuitele basculante bistabile reprezintă primul nostru exemplu de utilizare a unor instrumente abstracte. Atunci când proiectăm circuitele de calculator, putem folosi la construirea blocurilor componente porți logice, neglijând detaliile tehnice referitoare la construcția internă a fiecărei porți. Mai mult chiar, o dată ce am realizat circuite basculante bistabile sau alte circuite folosind porți logice, putem utiliza aceste componente ca blocuri componente pentru sisteme mai complexe. Pentru a demonstra acest lucru, figura 1.5 prezintă alt tip de circuit basculant bistabil, se iau în calcul argumentele pro şi contra referitoare la cele două opțiuni. Însă, la proiectarea unui cicuit mai complex în care sunt utilizate circuite basculante bistabile, se lasă deoparte detaliile interne ale circuitului, atenția fiind îndreptată către modul încare acesta, ca unitate completă, interacționează cu celelalte componente din sistem.

Figura 1.5 Altă metodă de realizare a unui circuit basculant bistabil



1.1.2 Alte tehnici de stocare

În anii ′60, calculatoarele conțineau de obicei multe inele de material magnetic de mici dimensiuni, denumite **miezuri** (cores), pe care erau înfăşurate fire electrice. La trecerea curentului electric prin fire, fiecare miez putea fi magnetizat sau remagnetizat într-una din cele două direcții. Ulterior, direcția câmpului magnetic putea fi detectată observându-se efectul acestuia asupra unui curent electric care trecea prin centrul miezului. Astfel, un miez reprezenta o posibilitate de memorare a unui bit. Astfel de sisteme sunt depăşite astăzi, datorită dimensiunilor mari şi a necesarului ridicat de putere.

Cu toate acestea, tehnologia magnetică mai este utilizată pentru stocarea datelor în calculatoare. Şi în prezent calculatoarele înregistrează datele pe un suport magnetic într-un mod foarte apropriat de acela în care este înregistrată muzica pe bandă magnetică. Deasemenea este utilizată tehnologia laserelor pentru înregistrarea biților. Diferențele dintre proprietățile circuitelor basculante bistabile electronice şi dispozitivele de stocare magnetice (sau laser) reprezintă argumente pro şi contra în ceea ce priveşte aplicațiile lor. Faptul că sunt complet electronice şi pot fi acționate mai rapid decât dispozitivele magnetice sau laser (care după cum vom vedea, necesită pentru a funcționa deplasări mecanice) reprezintă un avantaj

în favoarea circuitelor basculante bistabile. Din acest motiv, ele sunt utilizate pentru stocarea datelor în circuitele interne ale calculatorului. Însă un circuit basculant bistabil electronic pierde informația stocată în el atunci când sursa de alimentare este oprită. În schimb, dispozitivele de stocare magnetice sau cu laser păstrează datele, ceea ce le recomandă pentru realizarea de sisteme de stocare în care longevitatea este un factor important.

Limitările tehnologice, considerentele economice, precum şi necesitatea stocării de copii de siguranță ale datelor vitale, au făcut ca arareori memoria principală a unui calculator să satisfacă cerințele impuse de diverse aplicații. De aceea, multe calculatoare sunt echipate, pe lângă memoria principală, cu sisteme de stocare de masă (**mass storage systems**, denumite şi **memorie secundară**). De obicei, stocarea datelor pe aceste sisteme se face în unități de mari dimensiuni denumite **fişiere (files).** Unul dintre principalele dezavantaje ale sistemelor de stocare de masă este acela că în general ele necesită mişcare mecanică, astfel că sunt mai lente la stocarea şi recuperarea datelor în comparație cu memoria principală a calculatorului, care realizează electronic toate aceste operații. Principalul avantaj al dispozitivelor de stocare în masă este acela că, în multe situații, sunt mai ieftine decât memoria principală, iar suportul pe care înregistrează datele poate fi extras din calculator şi depozitat într-un loc sigur în scopul recuperării ulterioare a datelor.

Cu referire la dispozitivele care pot fi cuplate şi decuplate de la calculator se folosesc termenii *on-line* şi *off-line*. **On-line** înseamnă că dispozitivul sau informațiile sunt conectate şi pot fi folosite de calculator, fără a fi necesară intervenția omului. Dimpotrivă, **off-line** înseamnă că este necesară intervenția umană înainte ca dispozitivul sau informațiile să poată fi utilizate de calculator, poate din cauză că dispozitivul trebuie pornit sau mediul care conține informațiile trebuie introdus într-un anumit mecanism.

Cele mai cunoscute şi utilizate dispozitive de stocare de masă sunt discuri magnetice flexibile (pe cale de dispariție) şi rigide (hard disk). Pentru evaluarea performanței discurilor se folosesc mai multe criterii. Unul dintre ele utilizează **timpul de căutare (seek time)** - timpul necesar deplasării capetelor de citire/scriere de la o pistă la alta. Alți parametri măsurabili sunt **timpul de întârziere (rotation delay** sau **latency time –** jumătate din timpulnecesar ca discul să efectueze o rotație completă, adică timpul mediu în care datele respective ajung în poziția capului de citire/scriere, după ce acesta a fost adus la pista dorită), **timpul de acces (acces time** - suma dintre timpul de căutare şi timpul de întârziere**)** şi **rata de transfer (trnsfer rate)** a datelor către sau de la disc.

Deoarece funcționalitatea discurilor presupune efectuarea unei mişcări mecanice, atât discurile fixe cât şi cele flexibile au perfomanțe mai scăzute decât circuitele electronice. Dacă timpul de întârziere se măsoară în cazul circuitelor electronice în nanosecunde (miliardimi de secundă) sau şi mai puțin, timpul de căutare, timpul de întârziere şi timpul de acces în cazul discurilor se măsoară în milisecunde (miimi de secundă). Nu e deci de mirare că regăsirea unei informații de pe disc pare să dureze la nesfârşit din punct de vedere al unui circuit electronic care aşteaptă informația respectivă.

Dispozitive de stocare de masă mai vechi utilizează banda magnetică. În acest caz, informațiile sunt înregistrate pe o peliculă magnetică depusă pe o bandă de material plastic care la rândul ei este stocată pe nişte role.

Diferența dintre accesul direct (la discuri) şi accesul secvențial (la benzi)!!

Tehnologia dispozitivelor de stocare de masă este într-o permanentă evoluție.În prezent, există deja pe piață sisteme optice care concurează dispozitivele magnetice de stocare. Cel mai promițător este discul compact (CD), care este compatibil cu cele utilizate în domeniul muzicii, cu diferența că, pentru a se obține rate ridicate de transfer a datelor, cititoarele de CD-uri din calculatoare rotesc în general mult mai rapid discul.

Înregistrări logice şi fizice

În timp ce datele din memoria principală a unui calculator pot fi apelate la nivelul celulelor de memorie de dimensiunea unui octet, proprietățile fizice ale dispozitivelor de stocare de masă impun ca manipularea datelor stocate să se facă utilizând unități ce au dimensiuni mai mari de un octet. De exemplu, fiecare sector de pe un disc magnetic trebuie tratat ca un şir lung de biți. Un bloc de date corespunzător caracteristicilor fizice ale unui dispozitiv de stocare este denumit **înregistrare fizică (phisical record).**

Spre deosebire de împărțirea datelor în înregistrări fizice ale căror dimensiuni sunt determinate de caracteristicile dispozitivului de stocare, fişierele care sunt stocate posedă în general o diviziune naturală. De exemplu, un fişier care conține informații referitoare la angajații unei companii este alcătuit de obicei din blocuri de informații referitoarela fiecare angajat. Aceste blocuri de date care apar în mod natural sunt denumite **înregistrări logice (logical records).**

Dimensiunile înregistrărilor logice se potrivesc foarte rar cu dimensiunea înregistrării fizice inpusă de un dispozitiv de stocare. În consecință, este posibil să existe mai multe înregistrări logice stocată pe două sau mai multe înregistrări fizice. Rezultatul este acela că pentru recuperarea datelor de pe sistemele de stocare de masă este necesară o anumită activitate de decodificare.

1.1.3 Sistemul de notație hexazecimal

Atunci când ne referim la activitățile din interiorul unui calculator, trebuie să lucrăm cu şiruri de biți, care pot fi uneori foarte lungi. Din nefericire mintea umană întâmpină dificultăți la manevrarea unor asemenea lucruri. Chiar şi numai transcrierea şirului 101100101100011 este fastidioasă şi propice erorilor. De aceea, pentru a simplifica reprezentarea şirurilor de biți, vom utiliza o notație prescurtată denumită **notație hexazecimală (hexadecimal notation).** Această notație profită de avantajul faptului că şirurile de biți dintr-un calculator au în general lungimi care sunt multipli de patru.

Notația hexazecimală utilizează un singur simbol pentru a reprezenta patru biți, ceea ce înseamnă că un şir de doisprezece biți poate fi scris utilizând numai trei simboluri.

Figura 1.6 prezintă sistemul de codificare hexazecimal. Coloana din stânga afişază toate combinațiile posibile de patru biți; coloana din dreapta arată simbolul utilizat în notația hexazecimală pentru reprezentarea combinației din stânga. Utilizând acest sistem, şirul de biți 10110101 este reprezentat ca B5. Această notație se obține prin împărțirea şirului de biți în subşiruri de lungime patru şi apoi prin înlocuirea fiecărui subşir cu echivalentul său în hexazecimal - 1011 este reprezentat de B, iar 0101 este reprezentat de 5. Procedând astfel, şirul de 16 biți 1010010011001000 poate fi redus la mult mai acceptabila formă A4C8.

Figura 1.6 Sistemul de notaţie hexazecimal

Cuvânt Binar	*Reprezentare hexazecimală*
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

1.2 Memoria principală

1.2.1 Biţi

În scopul stocării datelor, un calculator conţine un mare număr de circuite, fiecare dintre ele fiind capabil să stocheze un bit. Acest rezervor de biţi este cunoscut sub numele de **memorie principală** a calculatorului (**main memory**).

Memoria este acea parte a calculatorului unde sunt stocate programele şi datele. Unii informaticieni (în special cei britanici) folosesc termenii "stocare" sau "storage" decât memorie, cu toate că din ce în ce mai mult termenul "storage" este folosit pentru a se face referirea la disk storage. Fără o memorie din care procesorul poate citi informaţia sau pe care poate scrie, nu ar mai fi posibil de implementat conceptul de calculator numeric cu program memorat.

Unitatea de baza a unei memorii este cifra binară numit bit. Un bit poate conţine un 0 sau un 1. Este cea mai simplă unitate (un dispozitiv capabil sa stocheze numai zerouri ar putea greu forma baza unei memori; cel putin doua valori sunt necesare). Oanenii obişnuiesc să spună că în calculatoare se foloseşte aritemetica binarăpentru căeste "eficient". Ceea ce vor să pună (cu toate că rareori işi dau seama) este că informaţia numerică poate fi stocată prin distincţia care se poate face între diferite valori ale unei mărimi fizice continue cum ar fi tensiunea sau curentul. Cu cât sunt mai multe valorile care trebuie deosebinte, cu atât este mai mică separarea dintre valorile alăturate şi memoria mai puţin performantă. Sistemul numerelor binare necesită numai două valori pentru a fi distinse. În consecinţă, aceasta este cea mai sigură metodă de a codifica informaţia digitală. Unele calculatoare, cum ar fi IBM main frames, sunt cunoscute ca având implementată pe langă matematica binară şi matematica zecimală. Trucul constă în folosirea a 4 biţi pentru a stoca o cifră zecimală "decimal digit" folosind un cod numit BCD (Binary Code Deccimal). Patru biţi oferă 16 combinaţii folosite pentru 10 digiţi de la 0 la 9, şase combinaţii rămânând nefolosite. Numarul 1944 este prezentat mai jos codificat zecimal şi binar, folosind câte 16 biţi în fiecare exemplu.

zecimal: 0001 1001 0100 0100 binar: 0000011110011000

16 biţi în formatul zecimal pot stoca numere între 0 şi 9999, oferind numai 10000 de combinaţii, în timp ce 16 biţi în formatul binar pot stoca 65536 de combinaţii diferite. Pentru acest motiv oameni spun că formatul binar este mai eficient.

Cu toate acestea, gindiţi-vă ce s-ar fi întâmplat dacă un tînăr genial inginer electrician ar fi invantat un dispozitiv electronic de înaltă fiabilitate care ar putea stoca direct cifrele de la 0 la 9 divizând intervalul 0 - 10 volţi în 10 intervale. Patru dispozitive de acest fel ar putea stoca numerele zecimale de la 0 la 9999. Patru dispozitive oferă 10000 de combinaţii. Ele ar putea fi de asemenea folosite pentru a stoca numere binare, doar 0 şi 1 în fiecare caz, patru dintre ele ar putea stoca numai 16 combinaţii. Cu asemenea dispozitive, sistemul ar putea fi mult mai eficient.

1.2.2 Organizarea memoriei principale

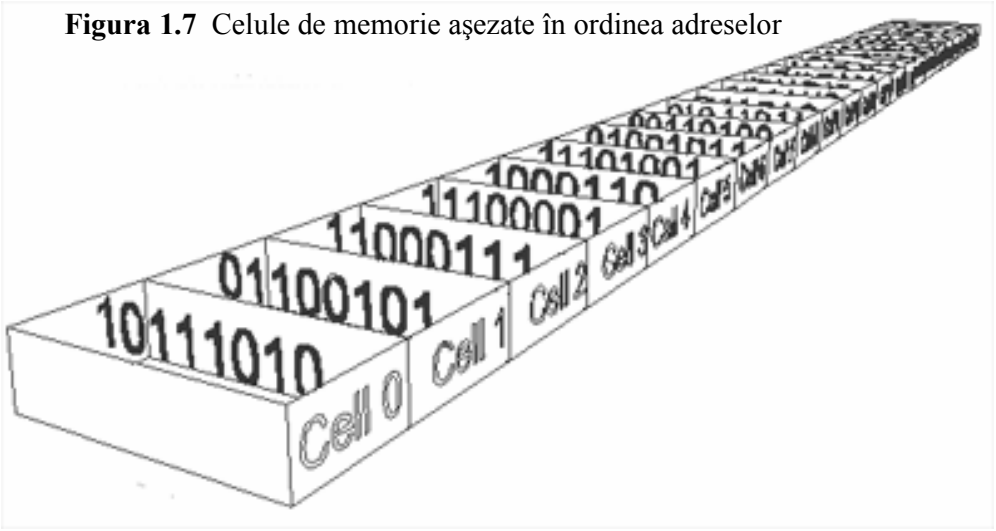
Circuitele de stocare din memoria principală a calculatorului sunt organizate în unităţi denumite **celule** (sau cuvinte), dimensiunea uzuală a unei celule fiind de opt biţi. De fapt, şirul de biţi de lungime opt a devenit atât de popular încât pe de o parte a dus la apariţia termenului **octet**, iar pe de altă a impus pentru cuvântul **byte** sensul de şir de biţi de această lingime.

Calculatoarele simple utilizate în cadrul aparatelor casnice, cum este cuptorul cu microunde, au de obicei memorii ale căror dimensiuni sunt de sute de celule sau chiar mai puţin, în timp ce calculatoarele complexe, utilizate pentru stocarea şi prelucrarea ansamblurilor de date de dimensiuni mari pot avea memoria principală alcătuită din miliarde de celule. Dimensiunea memoriei este măsurată în multipli de 1.048.576 celule. (Numărul 1.048.576 este o putere a lui 2, mai precis 2^{20} şi de aceea este mult mai natural să fie utilizat ca unitate de măsură în cadrul calculatoarelor decât 1.000.000, care este o putere a lui 10). Pentru a desemna această unitate de măsură se foloseşte termenul *mega*. Adesea, pentru termenul *megabyte* se foloseşte abrevierea MB. Astfel o memorie de 4 MB conţine 4.192.304 celule, fiecare dintre ele având un octet. Alte unităţi de măsură a dimensiunii memoriei sunt kilooctetul (kilobyte - prescurtat KB), care este egal cu 1024 octeţi (2^{10} octeţi) şi gigaoctetul (gigabyte - prescurtat GB) care este egal cu 1024 MB, respectiv 2^{30} octeţi.

Pentru identificarea celulelor individuale din memoria principală a unui calculator, fiecare are atribuit un nume unic, denumit **adresă**. Sistemul este analog cu tehnica de identificare - după adresă - a caselor unui oraş şi utilizează aceiaşi terminologie. Însă, în cazul celulelor de memorie, adresele utilizate sunt în întregime numerice. Pentru a fi mai precişi, celulele se consideră plasate toate pe un singur rând şi numerotate în ordine pornind de la valoarea 0. Celulele dintr-un calculator cu 4MB de memorie vor avea astfel de adrese ca 0, 1, 2, ... , 4192303. De observat că un astfel de sistem de adrese nu numai că ne oferă o cale de a identifica unic fiecare celulă, ci asociază în plus şi o relaţie de ordonare între celule (figura 1.7), permiţându-ne să facem referiri de tipul ″celula precedentă″ sau ″următoarea celulă″.

Pentru a completa structura memoriei principale a unui calculator, circuitele care stochează biţii sunt combinate cu circuitele necesare pentru a permite altor circuite să stocheze şi să recupereze datele din celulele de memorie. Astfel, alte circuite pot prelua date din memorie solicitând informaţii despre conţinutul unei anumite adrese (operaţie ce se numeşte citire) sau

Figura 1.7 Celule de memorie aşezate în ordinea adreselor



pot înregistra informaţii în memorie solicitând ca un anumit şir de biţi să fie plasat în celula aflată la o anumită adresă (operaţie ce poartă nimele de scriere).

O consecinţă importantă a modului de organizare a memoriei principale a calculatorului în celule de dimensiuni mici cu adresă este aceea că fiecare celulă poate fi apelată, cercetată şi modificată individual. O celulă de memorie cu o adresă mică este la fel de accesabilă ca una cu o adresă mare. În consecinţă, datele stocate în memoria principală aunui calculator pot fi prelucrate în orice ordine. De aceea memoria principală a unui calculator este adesea denumită **memorie cu acces aleator** (**random acces memory - RAM**). Acest acces aleator la mici unităţi de date se deosebeşte radical de sistemele de stocare de masă în cazul cărora şiruri lungi de biţi trebuie manipulate ca blocuri.