Dicționar Complet de Abrevieri
Arhitectura Sistemelor de Calcul

Sursă: Cursuri & Seminar

Acest document reunește toate acronimele și termenii tehnici discutați, incluzând tipurile de memorie (desktop și mobile) și extensiile de procesor.

1 Arhitectură și Procesoare

Abreviere	Termen Complet (Engleză)	Descriere (Română)
CPU	Central Processing Unit	Unitatea Centrală de Prelucrare.
GPU	Graphics Processing Unit	Procesor grafic, specializat în operații paralele (throughput mare).
ALU	Arithmetic Logic Unit	Unitatea care execută calcule matematice și logice.
FPU	Floating Point Unit	Coprocesor pentru calcule cu virgulă mobilă.
CU	Control Unit	Unitatea care decodifică instrucțiunile și controlează fluxul datelor.
ISA	Instruction Set Architecture	Interfața dintre hardware și software (setul de instrucțiuni).
CISC	Complex Instruction Set Computer	Arhitectură cu instrucțiuni complexe (ex: x86).
RISC	Reduced Instruction Set Computer	Arhitectură cu instrucțiuni simple (ex: ARM, RISC-V).
PC	Program Counter	Registru care ține adresa următoarei instrucțiuni.
IR	Instruction Register	Registru care ține instrucțiunea curentă.
SIMD	Single Instruction Multiple Data	Procesare paralelă a datelor (vectorizare).
SSE	**Streaming SIMD Extensions**	Extensie a setului de instrucțiuni x86 pentru a spori performanța în procesarea multimedia/3D (folosind SIMD).
MIMD	Multiple Instruction Multiple Data	Paralelism complet (procesoare multiple independente).

Abreviere	Termen Complet (Engleză)	Descriere (Română)
ILP	Instruction Level Parallelism	Tehnica de a executa mai multe instrucțiuni simultan (Pipeline).

2 Memorie și Stocare

RAM	Random Access Memory	Memorie volatilă cu acces aleator.
SRAM	Static RAM	Memorie rapidă (cache), scumpă, nu necesită refresh.
DRAM	Dynamic RAM	Memorie principală, ieftină, necesită refresh.
SDRAM	Synchronous DRAM	DRAM sincronizat cu ceasul sistemului.
DDR	Double Data Rate (SDRAM)	Transferă date de două ori pe ciclu de ceas.
LPDDR	**Low Power Double Data Rate**	Tip de memorie DDR optimizat pentru consum redus de energie (telefoane mobile, tablete).
ROM	Read-Only Memory	Memorie care nu poate fi ștearsă (sau se șterge greu).
PROM	Programmable ROM	ROM scris o singură dată.
EPROM	Erasable Programmable ROM	ROM care se șterge cu ultraviolete.
EEPROM	Electrically Erasable PROM	ROM care se șterge electric (byte cu byte).
Flash	Flash Memory	Memorie EEPROM modernă, ștearsă pe blocuri (SSD/USB).
VRAM	Video RAM	Memorie dedicată plăcii video.
L1/L2/L3	Level 1 / 2 / 3 Cache	Ierarhia cache-ului (L1 în nucleu $\rightarrow$ L3 partajat).
SSD	Solid State Drive	Stocare rapidă bazată pe Flash.
HDD	Hard Disk Drive	Stocare magnetică mecanică.
MMU	Memory Management Unit	Gestionează memoria virtuală (traduce adresele).
TLB	Translation Lookaside Buffer	Cache pentru MMU (stochează traducerile recente).

3 Performanță și Măsurători

CPI	Cycles Per Instruction	Număr mediu de cicli per instrucțiune.
IPC	Instructions Per Cycle	Instrucțiuni finalizate per ciclu (inversul CPI).
IC	Instruction Count	Numărul total de instrucțiuni executate.

MIPS	Million Instructions Per Second	Unitate de măsură a vitezei (a nu se confunda cu arhitectura MIPS).
FLOPS	Floating Point Operations/Sec	Măsură pentru performanța calculelor reale/științifice.
Hz	Hertz	Unitate de frecvență ($1/s$).

4 Reprezentarea Datelor și Logică

MSB	Most Significant Bit	Bitul cel mai din stânga (de obicei semnul).
LSB	Least Significant Bit	Bitul cel mai din dreapta.
NaN	Not a Number	Valoare IEEE 754 pentru operații invalide (ex: 0/0).
ASCII	American Standard Code for Info. Interchange	Standard de codare a textului.
IEEE 754	Standard for Floating-Point	Standardul pentru numere reale în calculator.
HEX	Hexadecimal	Baza 16.

5 Istoric și Diverse

ENIAC	Electronic Numerical Integrator and Computer	Primul calculator electronic general.
EDVAC	Electronic Discrete Variable Automatic Computer	Succesor ENIAC, arhitectură Von Neumann.
BIOS	Basic Input/Output System	Softul de inițializare al plăcii de bază.
OS	Operating System	Sistemul de operare.
I/O	Input / Output	Operații de intrare/ieșire.

OPERAȚII PE BIȚI

C:

```
  0101110011110011
+ 0111000011110000
------------------
  1100110111100011
```

Adunare: 1 + 1 -> 0 și carry 1

Scădere: 0 - 1 -> 1 și carry -1

Înmulțire: Verific dacă se înmulțește cu o putere a lui 2 (2, 4, 8, 16).

Atunci shiftez la stânga (1, 2, 3, 4).

bit - **bi**nary digi**t**
baza x - are cifre de la 0 la x – 1

Numere pozitive:
Numărul Y din baza X = $\sum_0^{n-1} b_i \cdot X^i$, n este numărul de biți din reprezentare
b_0 e LSB, b_{n-1} e MSB

Conversii:
- $B_{old} = 10$ are 10 cifre (0 – 9)
- $B_{new} = 100$ are 100 de cifre (0 – 99)

B -> B^p – Grupăm din B în câte P cifre, $p = \log_b B^P$

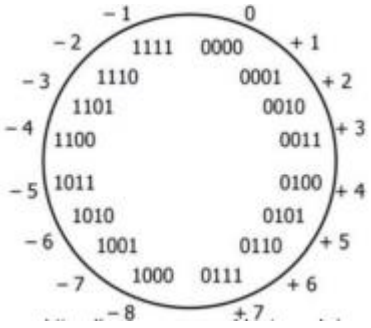
Numere negative (Pentru care, uneori, e nevoie de circuite speciale)
/S/_____ - S e MSB, bit rezervat pentru semn, deci sunt disponibili n – 1 biți din reprezentare!

Numărul -Y din baza X = $-b_i \cdot 2^{n-1} + \sum_0^{n-2} b_i \cdot 2^i$

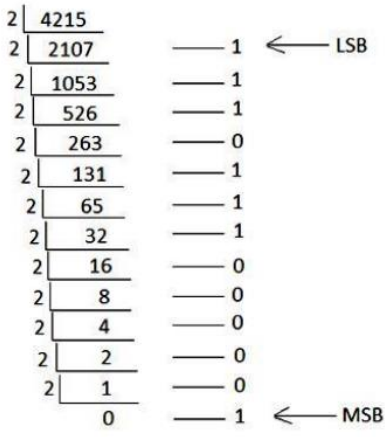
Complement față de doi:

$(-x)_{b10}$ -> b2	1. Scriem |x| în binar	2. Inversăm biții	3. Adăugăm 1
$(-x)_{b2}$ -> b10	1. Scriem numărul așa cum e	2. Inversăm biții	3. Adăugăm 1

Folosim sistemul binar: Același algoritm de adunare, pot fi folosite aceleași circuite pentru adunare naturală



- **un exemplu explicit: $(4215)_{10} = (1000001110111)_2$**



Cel mai mare număr care se poate reprezenta pe N biți: $2^N - 1$

Cel mai mare număr care se poate reprezenta (complement față de 2) pe N biți: $2^{N-1}-1$
Cel mai mic: -2^{N-1}
MSB pe poziția: N – 1 (maximal în reprezentare)

X – Număr natural. Biți necesari pentru reprezentare: ceil($\log_2 X$)
X – K cifre în HEX, de câți biți e nevoie pentru BIN: K * $\log_2 16 = 4$ HEX_{old} BIN_{new}
X – K cifre în BIN, de câți biți e nevoie pentru HEX: ceil(k / 4) BIN_{old} HEX_{new}
X – K cifre în DEC, de câți biți e nevoie pentru BIN: ceil(k * $\log_2 10$) DEC_{old} BIN_{new}

De ce funcționează complementul față de 2?

$$-\left(-2^N + \sum_{i=0}^{N-1} b_i 2^i\right) = 2^N - \sum_{i=0}^{N-1} b_i 2^i$$

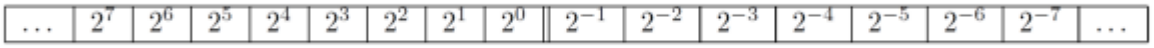
$$2^{N+1} = \sum_{i=0}^{N} 2^i + 1$$

$$= \sum_{i=0}^{N-1} 2^i + 1 - \sum_{i=0}^{N-1} b_i 2^i$$

$$= \sum_{i=0}^{N-1} (1 - b_i) 2^i + 1$$

$$= (\text{inversam bitii}) + 1$$

BINARY FIXED-POINT



2^-1	0.5
2^-2	0.25
2^-3	0.125
2^-4	0.0625
2^-5	0.03125
2^-6	0.015625
2^-7	0.0078125
2^-8	0.00390625
2^-9	0.001953125
2^-10	0.0009765625
2^-11	0.00048828125
2^-12	0.000244140625
2^-13	0.0001220703125

Ex: 101.<u>101</u>

$$\begin{cases} 101 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 1 = 5 \\ 101 = 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 0.5 + 0.125 = 0.625 \end{cases}$$

⇨ $101.\underline{101} = 5.625$
⇨

Ex: 3.<u>75</u>

$$\begin{cases} 3 = 1 \cdot 2^1 + 1 \cdot 2^0 \\ 0.75 = 1 \cdot 2^{-1} + 1 \cdot 2^{-2} \end{cases}$$

⇨ $3.\underline{75} = 11.11$

LOGARITM ÎNTREG

9. Demonstrați că $\lfloor \log_2 x \rfloor = i_{\max}$ unde x este un număr dat pe N biți iar $i_{\max} = \max \{i \mid b_i = 1,\ \forall\ i = 0, \ldots, N-1\}$ unde b_i reprezintă al i-lea bit din reprezentarea binară a numărului x. De exemplu: dacă $x = 00101110$ (46 zecimal) atunci $\lfloor \log_2 x \rfloor = 5$.

- **arătați că** $\lfloor \log_2 x \rfloor = i_{\max}$
- **pornim de la reprezentarea binară și aplicăm logaritmul**

$$x = \sum_{i=0}^{N-1} b_i 2^i$$

$$\begin{aligned}
\log_2 x &= \log_2 \left(\sum_{i=0}^{N-1} b_i 2^i \right) \\
&= \log_2 \left(2^{i_{\max}} \left(\sum_{i=0}^{N-1} b_i \frac{2^i}{2^{i_{\max}}} \right) \right) \\
&= \log_2 2^{i_{\max}} + \log_2 \left(\left(\sum_{i=0}^{N-1} b_i \frac{2^i}{2^{i_{\max}}} \right) \right) \\
&= i_{\max} + C,\ C < 1
\end{aligned}$$

OVERFLOW-SAFE BINARY-SEARCH

```
int binarySearch1(int arr[], int start, int end, int x)
{
        if (end >= start)
        {
                int mid = start + (end - start) / 2;

                if (arr[mid] == x)
                        return mid;

                if (arr[mid] > x)
                        return binarySearch1(arr, start, mid - 1, x);

                return binarySearch1(arr, mid + 1, end, x);
        }

        return -1;
}
```

GENERAL / ISTORIC

ENIAC – 1945 – 1955 / USA – 1000 op/s (aka Electronic Numerical Integrator and Computer)
HPE CRAY – 2021 / USA – 1714 petaoperații/s
”compute” – Infrastructura hardware pe care rulează algoritmii
Putem scădea costul de calcul pentru ML prin: algoritmi eficienți, hardware dedicat

Varianta A – Cea mai rapidă
(Calculul a două matrice)

```
# varianta A
for (int i = 0; i < n; ++i)
  for (int j = 0; j < n; ++j)
     for (int k = 0; k < n; ++k)
        C[i][j] += A[i][k] * B[k][j];
```

Complexitate: $O(n^3)$
Operații elementare: Adunare, înmulțire
Numărul operațiilor elementare: 2 pentru fiecare for (pe lângă asta, în fiecare for sunt câte n operații elementare de incrementare)
Număr de operații: $2 * n^3$

PERSONALITĂȚI

Richard Hamming - “The purpose of computing is insight, not numbers.”. Codul Hamming pentru integritatea mesajelor transmise la distanțe mari.

Blaise Pascal – Creează Pascaline (1642) – Calculator mecanic, capabil de +-, jucăria arisotcrației | Limbajul Pascal e numit în onoarea lui

Gottfried Wilhelm Von Leibniz – Studiază sistemul binar, extinde mașina lui Pascal (adaugă * /)

George Boole – Scrie ”The Laws of Thought” | Introduce logica booleană și analizează operațille logice de bază: Negație / Conjucție / Disjuncție / Disjuncție exclusivă | Ele stau la baza teoriei informației

Charles Babbage – Proiectează Mașina Diferențială Nr. 2 – prima mașină de calcul mecanică programabilă – prototipuri de 13 tone – Tatăl calculatoarelor moderne

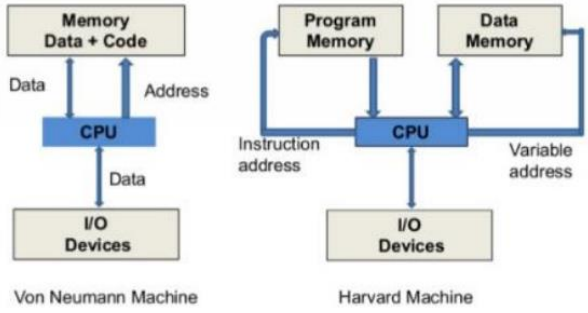
Ada Lovelace – Colaborează cu Babbage | Scrie un program care calculează numerele numere Bernoulli – Nu existau limbaje de programare, dar scrie pași pentru execuție – Primul programator

Konrad Zuse – Introduce o serie de calculatoare: Z1, Z2, Z3, Z4 | Folosește sistemul binar | Instrucțiuni stocate pe o bandă | Introduce reprezentarea și calculul în virgulă mobilă / floating point | Aproape totul în izolare, pe perioada celui de-Al II-lea Război Mondial.

Alan Turing – Decriptează rapid mesaje Enigma folosind mașina The Bomb - Brute-Force search | Introduce Mașina Turing pentru algoritmi Turing Complete – Sistem de analiză și recunoaștere a unui număr mare de date | Introduce Testul Turing – Pot mașinile să gândească?

John Von Neumann – Contribuie la crearea primului calculator electronic: ENIAC | Creează EDVAC – Sistem binar cu programe stocate. Introduce arhitectura von Neumann

Arhitectura von Neumann:



Claude Shannon: Părintele teoriei informaţiei (o inventează literally) | Demonstrează că problemele de logică booleană se pot rezolva cu circuite electronice | Teorema de eşantionare Shannon – Nyquist: Analog <-> Digital fără a pierde date

Grace Hopper – Contribuitoare în dezvoltarea de limbaje high-level şi scrie compilatorul COBOL

Margaret Hamilton - Scrie cod pentru software-ul de la bord pentru misiunea Apollo

Barbara Liskov – Design pattern: Principiul substituţiei Liskov + altele în distributed computing

Rivest Shamir Adleman – Creatorii primului sistem de criptare cu chei, public: RSA. Foloeşte numere prime. Foarte util în tranzacţiii bancare.

Diffie Hellman – Schimbul de cheie Diffie-Hellman – Soluţia în a trimite mesaje secrete într-un canal de comunicaţie nesecurizat.

Ritchie and Thompson – Dezvoltatorii limbajului C | Pun bazele sistemelor open-source, creează Unix OS.

Richard Stallman – Contributori la GNU Project.

Linus Tolvads – Creatorul Linux şi GIT (pentru controlul versiunilor)

Steve Jobs – Cofondează Apple, Pixar

Bill Gates – Fondator Microsoft <3 | Acum CEO este Satya Nadella

Jeff Bezos – Fondator & fost CEO Amazon

Mark Zuckerberg – CEO Meta

Larry Page and Sergey Brin – Fondatorii Google

Idei mari:

- De la mecanic la electric
- De la o maşină care face un singur lucru la o maşină programabilă
- Design modular (pe module)
- Teorie despre ce este posibil pe aceste maşini
- Dorinţa de a face lucrurile optim, la limită şi fără risipă

După Al Doilea Război Mondial, dorinţa de cercetare în domeniul calculatoarelor creşte în ritm exponenţial

Actori importanţi: Grupuri profesionale (IEEE, ACM, Bell Labs) şi state (Statele Unite: DARPA)

TEORIA PROBABILITĂȚILOR

$P = \frac{\text{Numărul cazurilor favorabile}}{\text{Numărul cazurilor posibile}}$ (atunci când evenimentul nu este influențat de mediul exterior)

$P1 \cdot P2$ (probabilitatea ca două evenimente **INDEPENDENTE** să se întâmple, fără să fie influențate de mediul exterior)Este intersecția dintre evenimentul care se putea întâmpla primul și cele totale. Trebuie să ținem cont de cazurile care sunt favorabile pentru ambele evenimente.

TEORIA INFORMAȚIEI

Informația: Date care afectează (aproape mereu reduce) incertitudinea despre un fenomen.
Se poate acumula, este constantă (nu este creată/distrusă).

$I(x_i) = \log_2\left(\frac{1}{p_i}\right)$ (Câtă informație ne oferă o valoare în funcție de probabilitatea ca ea să apară)

Calculăm cantitatea de informație folosind X (variabila aleatoare):
N – Numărul de valori distincte: $x_1, x_2 \dots x_n$
P - Fiecare valoare apare cu probabilitatea $p_1, p_2 \dots p_n$
p_i mai mic -> Obținem cantitate mai mare de informație

Putem privi formula astfel: $I(x_i) = \log_2\left(\frac{1}{M^{\frac{1}{N}}}\right) = \log_2\left(\frac{N}{M}\right) =$ rezultat în biți!

N – Numărul total de evenimente
M – Numărul de evenimente favorabile

- **entropia**
  - valoarea medie de informației primită despre o variabilă X

$$H(X) = E(I(X)) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} = \sum_{i=1}^{N} -p_i \log_2 p_i$$

  - *H(X)* se numește entropia lui X
  - *I(X)* este informația despre X
  - *E* este "expected value", operația care calculează valoarea medie

  - exemplu: X = {A, B, C, D} cu probabilități {1/3, ½, 1/12, 1/12}
    - $H(X) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{1}{2}\log_2\frac{1}{2} - \frac{1}{12}\log_2\frac{1}{12} - \frac{1}{12}\log_2\frac{1}{12} = 1.626$ biti

I) 12 piese: 3 cuburi, 3 conuri, 3 sfere, 3 piramide | Fiecare formă are una dintre culorile: roşu, galben, albastru

A extrage o piesă roşie. Câtă informaţie primeşte copilul B, care vrea să afle tipul piesei?

Iniţial: 12 variante, dar 4 piese sunt de culoare roşie, deci $p_i = 4 \cdot \frac{1}{12}$

$$I(x_i) = \log_2\left(\frac{1}{4 \cdot \frac{1}{12}}\right) = \log_2(3)$$ (x_i este o piesă roşie)

II) În urnă: 5 bile roşii, 3 bile albastre

Se extrage o bilă albastră.
Primim I(bilă albastră) cantitate de informaţie

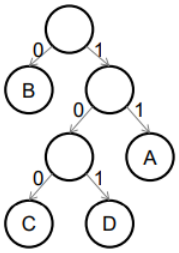
$$I(\text{bila albastra}) = \log_2\left(\frac{1}{\frac{3}{8}}\right) = \log_2\left(\frac{8}{3}\right) = 1.42 \text{ biti}$$

$$H(\text{urna}) = \frac{5}{8}\log_2\left(\frac{8}{5}\right) + \frac{3}{8}\log_2\left(\frac{8}{3}\right) = 0.95 \text{ biti}$$

$$H(\text{urna dupa extragere}) = \frac{5}{7}\log_2\left(\frac{7}{5}\right) + \frac{2}{7}\log_2\left(\frac{7}{2}\right) = 0.86 \text{ biti}$$

Entropia spune că: Nu se poate face o reprezentare pe biţi mai optimă decât valoarea ei fără a pierde informaţie. (Ceva compresat perfect se numeşte zgomot)
Este limita de compresie posibilă!

Codarea eficientă şi unică se face cu arbori binari:
Frunzele – Codurile
Stânga/Dreapta – Decis de 0/1



Implementare: Algoritmul lui Huffman
(Optim când considerăm un singur caracter pe rând)
Input: Probabilitatea fiecărui eveniment, ex: {1/3, 1/2, 1/12, 1/12}
Output: Codurile de pe un arbore binar
Cheia: O codare mai scurtă pentru event. cu probabilităţi mari, una mai scurtă -- //-- mici.
Dacă toate evenimentele sunt echiprobabile, nu putem face nimic.

```
00100111010011   0 0 100 11 101 0 0 11
                 B B C   A  D   B B A (Parcurgem arborele)
```

Detectarea erorilor: Ineficientă ca stocare.

Metoda I) Adăugăm biţi de paritate simbolurilor, pentru a recupera informaţia.

Corecţia erorilor: Distanţă Hamming – O distanţă mare poate duce şi la corectarea erorilor
O distanţă Hamming de 2N+1 poate corecta E erori. Distanţa hamming minimă pentru 4 biţi schimbaţi este 4.

Orice comunicaţie / Stocare este redundantă!

Aproximarea lui Stirling: $\log_2(a!) \approx a\log_2(a) - a\log_2 e$

Probabilităţile pentru fiecare caz:

(În principiu : Cât ştim / Cât vrem să ştim)

a) x are exact două valori "1" în reprezentarea sa binară: $p = \frac{C_N^2}{2^N}$, pp. n par.

b) x are exact N/2 valori "1" în reprezentarea sa binară (N par): $p = \frac{C_N^{N/2}}{2^N}$, pp. n par.

c) x are o secvenţă continuă de N/4 biţi de "1" (restul sunt "0", N divizibil la 4) :

$p = \frac{(3N/4+1)}{2^N}$, pp. 4 | n

d) x are MSB (Most Significant Bit) setat la "1": $p = \frac{2^{N-1}}{2^N} = \frac{1}{2}$

e) x este impar: $p = \frac{2^{N-1}}{2^N} = \frac{1}{2}$

f) x este o putere a lui 2: $p = \frac{N}{2^N}$

g) x are primii N/2 biţi din reprezentarea sa binară setaţi la "0" (N par): $p = \frac{2^{N/2}}{2^N}$

h) x este un număr prim (aproximativ estimat):

$$p \approx [2N/l\,n(2N)]2N = 1l\,n(2N)\,p \approx \frac{[2^N/\ln(2^N)]}{2^N} = \frac{1}{\ln(2^N)}$$

i) x are un număr par de biţi setaţi la "1" (N par): $p = \frac{1+\sum_{i=1}^{N/2} C_N^{2i}}{2^N} = \frac{1}{2}$

j) x = 42: $p = \frac{1}{2^N}$

ASSEMBLY x86

Utilizări: Securitate informatică: Hacking, Reverse Engineering | Optimizare: Dezvoltare jocuri și ML / AI | Debugging | Dezvoltare software low-level: Sisteme embedded / Sisteme de operare

În Assembly x86, rezultatul înmulțirii poate fi stocat pe 64 de biți (%edx * 2^{32} + %eax)

Shift: |S|XXX.XXXX >> 2 = 00|S|X.XXXX – Se pierde bitul de semn!
SAR/SAL – Shift artimetic, ține cont de semn.

jmp *%eax – Sare la o adresă

aritate 1 – Operația se aplică unui singur număr

.asciz -> Lungimea șirului + 1 pentru \n

S1: ”ASC”, S2: ”FMI” -> syscall de print pentru $S1 cu lungimea 5 -> ASCFM

¬ NOT (are aritate 1) ∧ AND ∨ OR Tsunotshi DX – eu irl | ORNOTAND

CMP – Compară două valori și setează flag-uri:
- Zero Flag (ZF) op1 = op2
- Sign Flag (SF) Rezultatul comparației este negativ
- Carry Flag (CF) op1 < op2

$a(b, c, d) = a + b + c \cdot d$

operand	Descriere	Semn (Da/Nu)	Flaguri setate	Condiție Flag
jc	jump dacă este carry setat	Nu	CF (Carry Flag)	CF = 1
jnc	jump dacă nu este carry setat	Nu	CF	CF = 0
jo	jump dacă este overflow setat	Nu	OF (Overflow Flag)	OF = 1
jno	jump dacă nu este overflow setat	Nu	OF	OF = 0
jz	jump dacă este zero setat	Nu	ZF (Zero Flag)	ZF = 1
jnz	jump dacă nu este zero setat	Nu	ZF	ZF = 0
js	jump dacă este sign setat	Da	SF (Sign Flag)	SF = 1
jns	jump dacă nu este sign setat	Da	SF	SF = 0
jb	jump if below (unsigned $op2 < op1$)	Nu	CF	CF = 1
jbe	jump if below or equal (unsigned $op2 \leq op1$)	Nu	CF, ZF	CF = 1 or ZF = 1
ja	jump if above (unsigned $op2 > op1$)	Nu	CF, ZF	CF = 0 and ZF = 0
jae	jump if above or equal (unsigned $op2 \geq op1$)	Nu	CF	CF = 0
jl	jump if less than (signed $op2 < op1$)	Da	SF, OF	SF ≠ OF
jle	jump if less than or equal (signed $op2 \leq op1$)	Da	SF, OF, ZF	SF ≠ OF or ZF = 1
jg	jump if greater than (signed $op2 > op1$)	Da	SF, OF, ZF	SF = OF and ZF = 0
jge	jump if greater than or equal (signed $op2 \geq op1$)	Da	SF, OF	SF = OF
je	jump if equal ($op1 = op2$)	Nu	ZF	ZF = 1
jne	jump if not equal ($op1 \neq op2$)	Nu	ZF	ZF = 0

TABELA 1. Operandi de salt condiționat, flagurile setate și condițiile lor

CIRCUITE

CIRCUIT DIGITAL COMBINAŢIONAL

La ieşire, este o combinaţie (funcţie logică care combină) toate sau o parte a intrărilor

Eficiente în general. Problema majoră:
sunt one-shot
- Nu putem itera
- Nu permit niciun fel de logică internă / memorie internă (nu are stări interne)
- Sunt prea simple: Pui un semnal digital constant la intrare şi ai un semnal digital constant la ieşire
- Logica combinaţională este insuficientă pentru anumite implementări

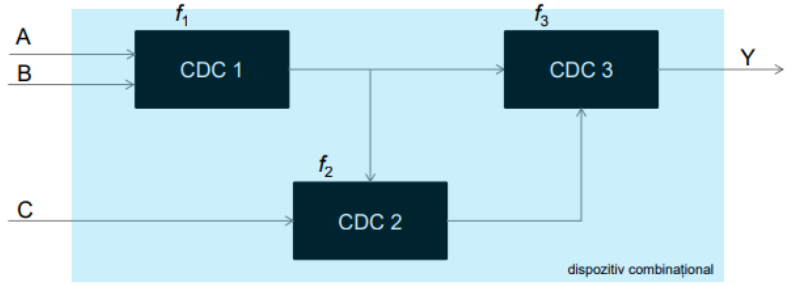
Timp de propagare t_p: Timp maxim necesar pentru a produce la ieşire semnale digitale corecte şi valide

Pentru fiecare intrare, trebuie să ştim care e ieşirea

A	B	C	X	Y
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Dispozitiv combinaţional

Elementele sale: Circuite combinaţionale
O intrare este conectată la exact o ieşire / la o constantă
Fără cicluri în graful direcţional al dispozitivului



Funcţia dispozitivului: Y = f3(f1(A, B), f2(f1(A, B), C)) – Ne uităm ce intră în Y şamd.

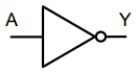
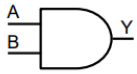
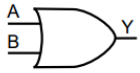
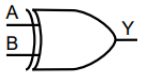
Timpul total de propagare: $t_{p,\,total} = t_{p,\,1} + t_{p,\,2} + t_{p,\,3}$ (cea mai lungă cale)
Timpul maxim după care avem o ieşire validă dacă avem intrări valide

Un computer care funcţionează la 1GHz trimite comenzi o dată la 1ns

De ce folosim semnale digitale în loc de analogice:
- Într-un sistem analogic zgomotul se acumulează
- Într-un sistem digital, avem corecţiile de zgomot (avem margini)

De ce folosim sistemul binar?
- Număr redus de stări (2) (pentru HEX ar trebui 16, deci ar trebui distinse 16 nivele de voltaj)
- Pentru baza 4, am avea 4 nivele, deci am fi de 2 ori mai eficienţi

Tot ce facem pe sistemul de calcul trebuie redus la circuite care sunt porţi logice

Operația	Valori	Notație
NOT	 <table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	$\bar{A}$ (A bar, A complement) sau !A
AND	 <table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	AB
OR	 <table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A + B
XOR	 <table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table> $A \oplus B = \bar{A}B + A\bar{B}$	$A \oplus B$

A	B	C	X	Y
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

o expresie booleană care conține regulile din tabel?

- $X = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

poartă	întârziere (ps)	suprafață (μm²)
AND-2	50	25
NAND-2	30	15
OR-2	55	26
NOR-2	35	16
AND-4	90	40
NAND-4	70	30
OR-4	100	42
NOR-4	80	32

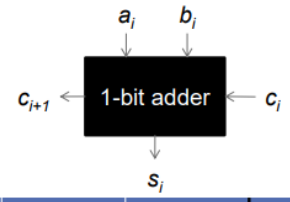
N* - Mai rapide, mai mici ca suprafață față de *
*-2 – Mai rapide, mai mici ca suprafață față de *-4

Tranzitoare CMOS: Circuitele analogice sunt mai eficiente pe logică negată (mai puține componente electrice)

CIRCUIT	INPUT	OUTPUT	NR INTRĂRI	NR IEȘIRI
ADUNARE	A (N biți) B (N biți)	S (N + 1 biți)	2N	N + 1

Cât de mare va fi circuitul? $(N + 1)2^{2N-1}$

Putem defini un circuit bloc fundamental pe care bazăm totul:



$s_i = a_i \oplus b_i \oplus c_{in}$

$c_{out} = a_i b_i + (a_i + b_i)c_{in}$

a_i	b_i	c_{in}/c_i	s_i	c_{out}/c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

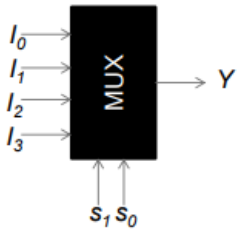
Folosim circuitul în cascadă
$t_p = N \cdot t_{p,bit\text{-}adder}$

Trebuie așteptată calcularea biților de carry

Putem generaliza

- notăm $g_i = a_i b_i$ și $p_i = a_i + b_i$
  - *dacă $g_i = 1$ atunci $c_{out} = 1$*
  - *dacă $g_i = 0$ și $p_i = 0$ atunci $c_{out} = 0$*
  - *dacă $g_i = 0$ și $p_i = 1$ atunci $c_{out} = c_{in}$*

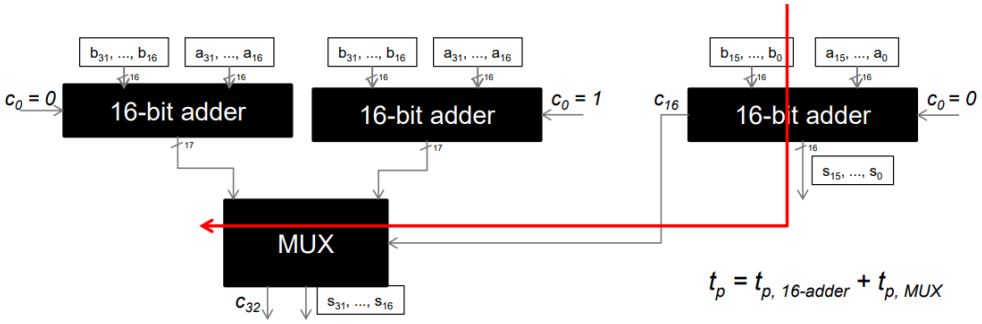
Multiplexor (MUX 😵)
Circuitul selectează: un semnal digital de la intrare pe baza unui semnal de activare S
Util pentru implementare hardware pentru ”if”, ”case”, operații shift



s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Inputuri: 2^n -> n linii de selecție

Îmbunătățire: Adunare pe 32 de biți



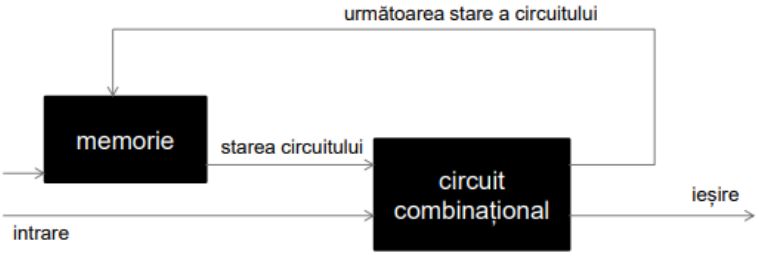
Putem aplica aceeași idee pentru circuitele de 16 biți de mai sus: $t_p = O(\log_2 N)$

CIRCUITE SECVENȚIALE

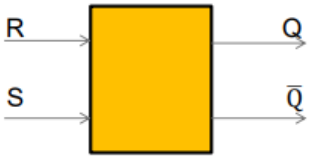
Permit elemente de tip ”memorie” > putem adăuga o stare circuitului (există stare internă)

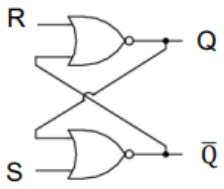
Au variabila de timp: Intrările / Ieșirile nu sunt fixe | Număr variabil de pași în rezolvare

Biții pe care îi reprezentăm - voltaj.
Energia electrică - dificil de stocat (provoacă fenomenul de scurgere / leakage)
Pentru a memora ceva > Refresh din când în când pentru actualizarea nivelului de energie electrică



SR Latch (Set-Reset Latch) | Memorează un bit de informație | Bun, dar are două intrări

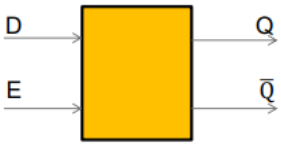


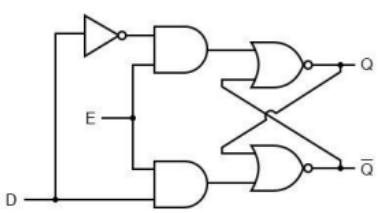


S	R	Q	$\overline{Q}$	
0	0	latch	latch	nu se schimbă nimic
0	1	0	1	aici punem “0” în memorie
1	0	1	0	aici punem “1” în memorie
1	1	0	0	stare invalidă

D Latch | Bun, are o intrare dar vrem să sincronizăm mai multe dispozitive

Se activează când E este activ. Vrem să se activeze când E creşte.



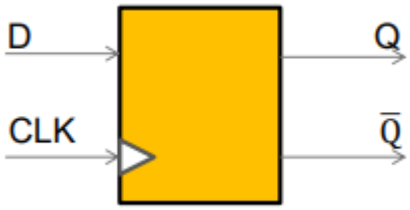


E	D	Q	$\overline{Q}$	
0	0	latch	latch	nu se schimbă nimic
0	1	latch	latch	nu se schimbă nimic
1	0	0	1	aici punem "0" în memorie
1	1	1	0	aici punem "1" în memorie

E de la Enable, adică activare
dacă E = 0 nu se întâmplă nimic

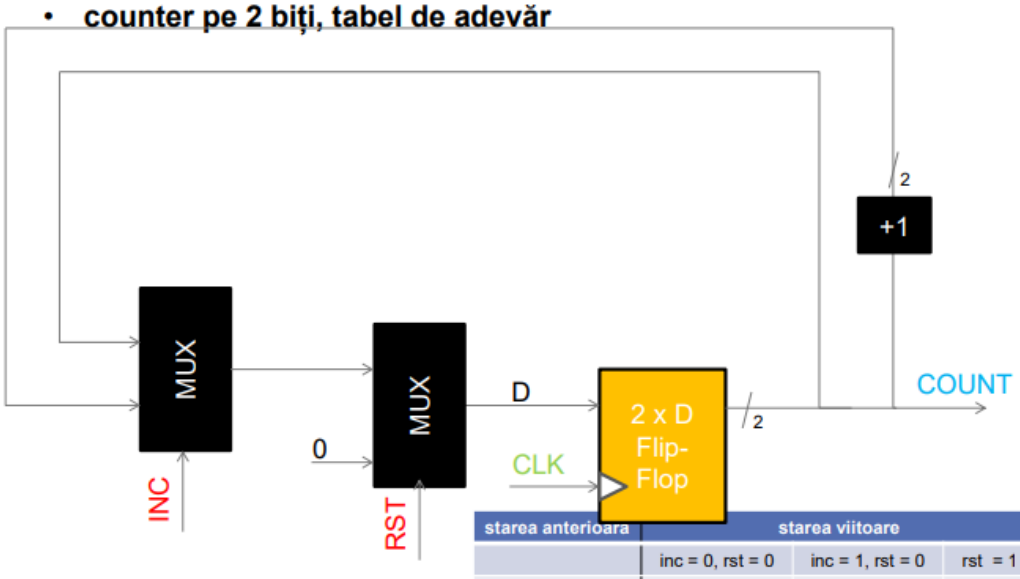
D Flip-Flop

E devine clock (ceasul sistemului)
CLOCK: La un interval fix de timp (la un ciclu), sistemul face ceva



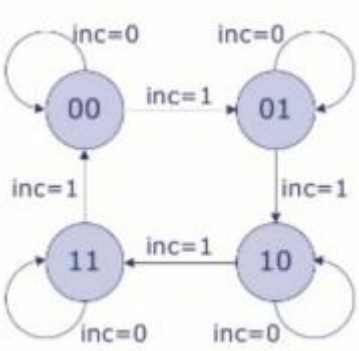
Registru = Un set de câteva D Flip Flops care au acelaşi CLK

- counter pe 2 biți, tabel de adevăr



starea anterioara	starea viitoare		
	inc = 0, rst = 0	inc = 1, rst = 0	rst = 1
00	00	01	00
01	01	10	00
10	10	11	00
11	11	00	00

Stările



LEGILE DE MORGAN

- !(!A+!B) = AB
- !(!A!B) = A+B
- !(A+B+C) = !A!B!C
- !(ABC) = !A+!B+!C
- !(A+B)!A!B = !A!B
- !(AB)(!A+!B) = !A+!B
- !(A+B)(!A+!B) = !A!B
- !A!B!(AB) = !A!B
- C+!(CB) = 1
- !(AB)(!A+B)(!B+!B) = !A!B

SIMPLIFICĂRI

$(A+C)(AD+A\bar{D})+AC+C$

$(A+C)A(D+\bar{D})+AC+C$ //distribuim, invers

$(A+C)A+AC+C$ //suma variabila si complement

$A((A+C)+C)+C$ //distribuim, invers

$A(A+C)+C$ //asociem, idempotent

$AA+AC+C$ //distribuim

$A+(A+1)C$ //idempotent, identitate, factor

$A+C$ //identitate de doua ori

A+0 = A
!Ax0 = 0
A+!A = 1
A+A = A
A+AB = A
A+!AB = A+B
A(!A+B) = AB
AB+!AB = B
(!A!B+!AB) = !A
A(A+B+C+...) = A

AB+A!B = A
!A+B!A = !A
(D+!A+B+!C)B = B
(A+!B)(A+B) = A
C(C+CD) = C
A(A+AB) = A
!(!A+!A) = A
!(A+!A) = 0
D+(D!CBA) = D
!D!(DBCA) = !D
AC+!AB+BC = AC+!AB
(A+C)(!A+B)(B+C) = AB+!AC
!A+!B+AB!C = !A+!B+!C
$(A+B)^2+(A+B)^3+A+3!A+A^3 = 1$

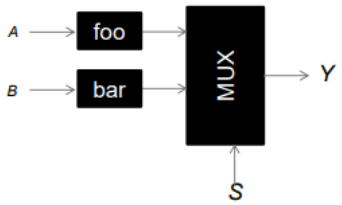
A+A!A = A

- implementați o poartă NOT cu un MUX: Y = NOT A
- $Y = I_0\bar{s}_0 + I_1 s_0 = 1\bar{A} + 0A = \bar{A}$

- implementați o poartă AND cu un MUX: Y = A AND B
- $Y = I_0\bar{s}_0 + I_1 s_0 = 0\bar{A} + BA = AB$

- implementați o poartă OR cu un MUX: Y = A OR B
- $Y = I_0\bar{s}_0 + I_1 s_0 = B\bar{A} + 1A = A + B\bar{A} = A + B$ [vezi ex. 4 f)]

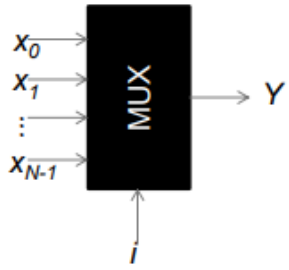
- **Y = S ? foo(A) : bar(B)**



I_0 foo(X)	I_1 bar(Y)	S	Y
*	*	0	I_1
*	*	1	I_0

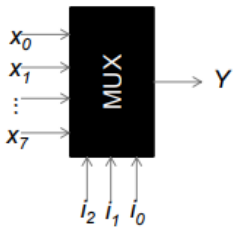
- **care e diferența cu un limbaj de programare?**
  - indiferent de valoarea lui S, se execută foo(A) și bar(B)
  - doar că la ieșire vedem doar una dintre funcții (cea selectată de S)

Vrem să accesăm x[i]



Intrări: Vectorul x
Semnal: Indexul i
Dimensiunea intrării: $N - 1 - 0 + 1 = N$
Dimensiunea lui S: ceil($\log_2 N$)

s_0 (i)	Y
000	I_0 (x_0)
001	I_1 (x_1)
010	I_2 (x_2)
011	I_3 (x_3)
100	I_4 (x_4)
101	I_5 (x_5)
110	I_6 (x_6)
111	I_7 (x_7)



Câte MUX cu 2 intrări pentru a simula MUX cu N intrări?
N – 1

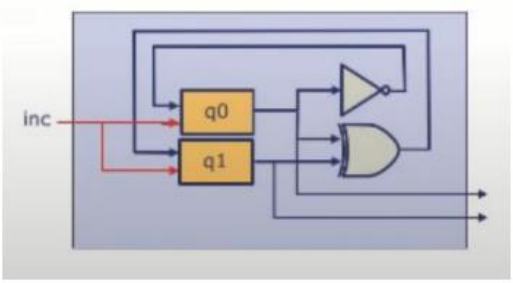
Circuit secvenţial COUNTER pe 2 biţi

Relaţia dintre starea anterioară şi starea viitoare

- $q_0^{(t+1)} = !\text{INC} \times q_0^{(t)} + \text{INC} \times !q_0^{(t)}$, $q_1^{(t+1)} = !\text{INC} \times q_1^{(t)} + \text{INC} \times (q_1^{(t)} \otimes q_0^{(t)})$

$q_1^{(t)}$	$q_0^{(t)}$	$q_1^{(t+1)}$	$q_0^{(t+1)}$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Desenul circuitului secvenţial (stările sunt q0 şi q1)

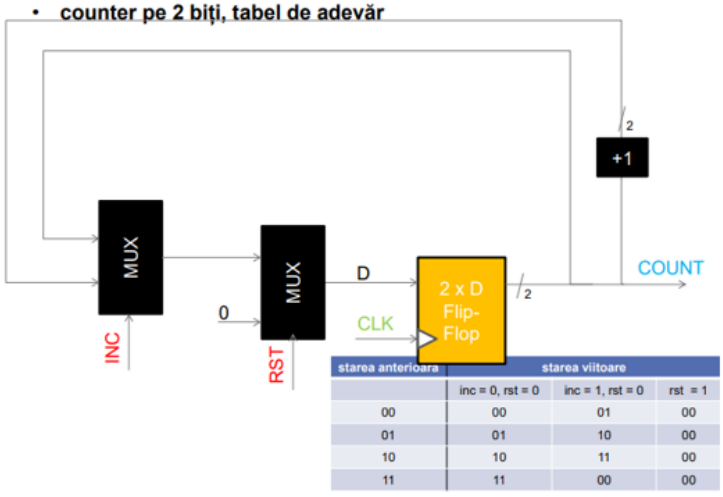


- **aici INC e pe post de semnal Enable**

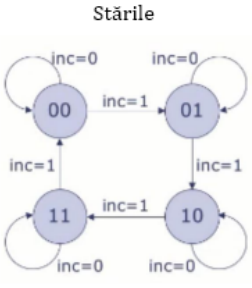
c) Codul Gray – cod binar, proprietate: diferenţa de la un simbol la altul este un singur bit care se schimbă (Pentru 3 biţi, codul Gray: 000, 001, 011, 010, 110, 111, 101, 100).

$q_2^{(t)}$	$q_1^{(t)}$	$q_0^{(t)}$	$q_2^{(t+1)}$	$q_1^{(t+1)}$	$q_0^{(t+1)}$
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	1

- $q_0^{(t+1)} = !q_2^{(t)}!q_1^{(t)}!q_0^{(t)} + !q_2^{(t)}!q_1^{(t)}q_0^{(t)} + q_2^{(t)}q_1^{(t)}!q_0^{(t)} + q_2^{(t)}q_1^{(t)}q_0^{(t)}$
  $= q_2^{(t)}q_1^{(t)} + !q_2^{(t)}!q_1^{(t)}$
- $q_1^{(t+1)} = \ldots$
- $q_2^{(t+1)} = \ldots$
- **counter pe 2 biţi, tabel de adevăr**

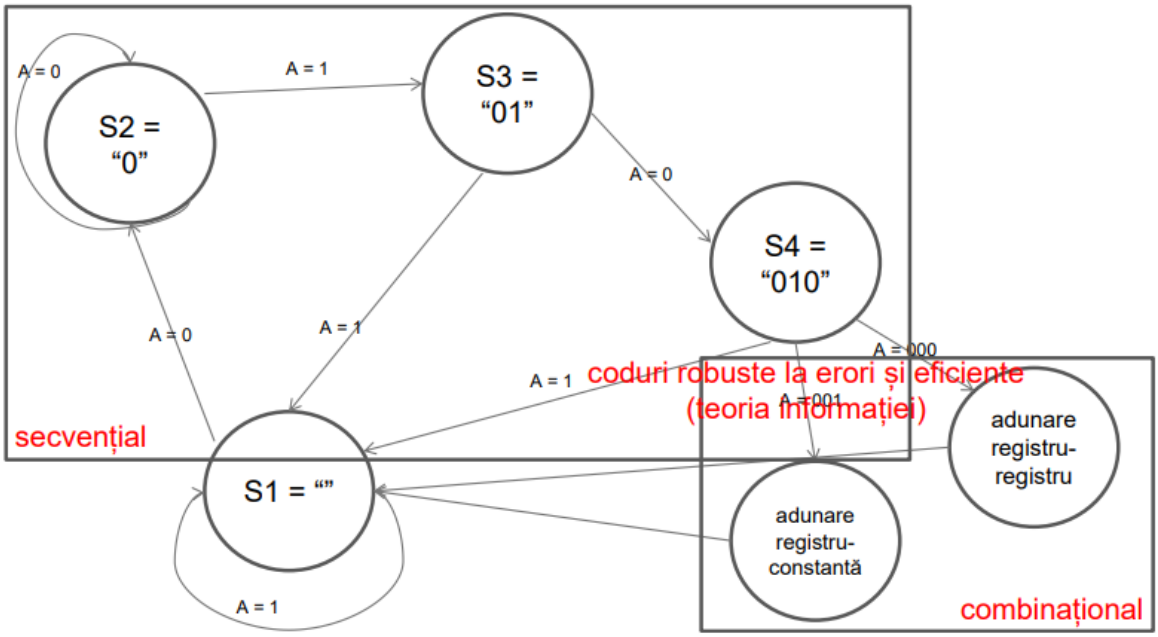


starea anterioara	starea viitoare		
	inc = 0, rst = 0	inc = 1, rst = 0	rst = 1
00	00	01	00
01	01	10	00
10	10	11	00
11	11	00	00



Un semnal digital A poate lua valori $\{0, 1\}$ în timp iar noi vrem să detectăm dacă semnalul are valoarea 010 la un moment dat. Dacă acestă secvență de biți este detectată în A atunci o variabilă Y este setată la 1, altfel această variabilă este 0. Cerințe:

(a) desenați diagrama de stări și tranzițiile între aceste stări;
(b) cum arată circuitul secvențial asociat diagramei realizate anterior?
(c) calculați tabelul de stări pentru circuitul secvențial;
(d) scrieți expresiile pentru logica combinațională.



cod mașină ...0100001011010001010000100011001011 ...

Circuit secvenţial pentru CMMDC

def cmmdc(a, b): if a == b: return b elif a > b: return cmmdc(a-b, b) else: return cmmdc(b, a)

Avem variabilele: $a^{(t)}$, $b^{(t)}$

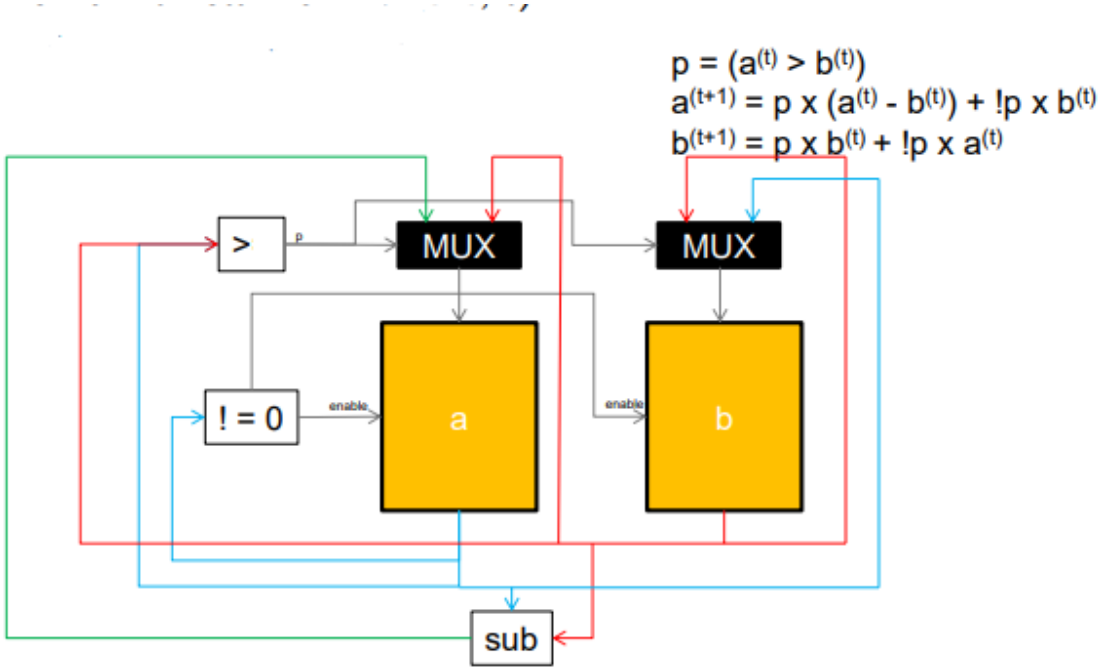
Ecuaţiile de evoluţie:

facem ceva doar dacă $a^{(t)} != 0$

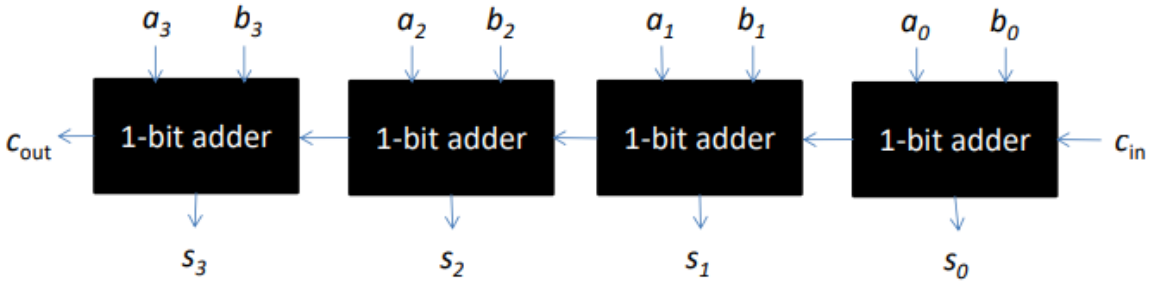
$p = (a^{(t)} > b^{(t)})$

$a^{(t+1)} = p \times (a^{(t)} - b^{(t)}) + !p \times b^{(t)}$

$b^{(t+1)} = p \times b^{(t)} + !p \times a^{(t)}$



Circuit de adunare pe 4 biţi



Circuit de adunare pe 4 biți.

ÎMPĂRŢIREA NUMERELOR ÎNTREGI

Tratăm împărţitorul ca număr natural în baza 10, fie el **n** în exemplu:

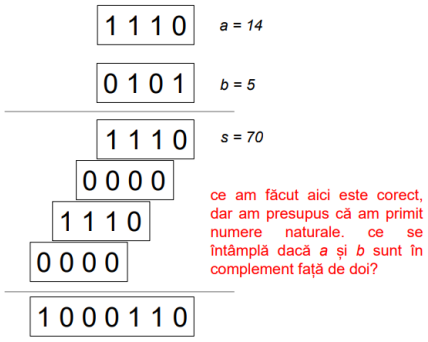
```
int Divide(NrMare x, int n)
//x = x /n, returneaza x%n
{
  int i,r=0;
  for(i=x[0];i>0;i--)
  {
    r=2*r+x[i];
    x[i]=r/n;
    r%=n;
  }
  for(;x[x[0]]==0 && x[0]>1;)
    x[0]--;
  return r;
}
```

- **s = a ÷ b**
  - ce se întâmplă dacă *a* sau *b* sunt variabile negative?
  - rezultatul este negativ dacă *a* și *b* au semne diferite (XOR logic)
  - în general
    - a = s x b + r
    - semnul lui r este semnul lui a

ÎNMULŢIREA NUMERELOR ÎNTREGI

De la dreapta la stânga
Dacă suntem în B pe bit 1, copiem A
Dacă suntem în B pe bit 0, punem 0 peste tot

A şi B naturale



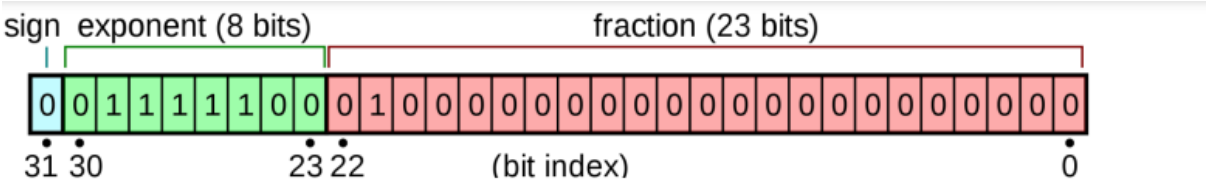
A si B întregi

Extindem numerele:

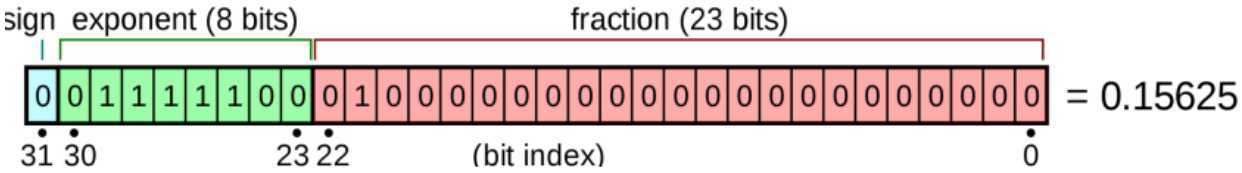
A = -2 A = 1110 A = 1111.1110
B = 5 B = 0101 B = 0000.0101

Se calculează la fel ca la numere naturale, dar **rezultatul este în complement faţă de doi.**

FLOATING POINT



- -1313.3125
  - partea întreagă este: 1313
  - partea fracționară: 0.3125
    - 0.3125 × 2 = 0.625 => 0
    - 0.625 × 2 = 1.25 => 1
    - 0.25 × 2 = 0.5 => 0
    - 0.5× 2 = 1.0 => 1
  - deci, $1313.3125_{10} = 10100100001.0101_2$
  - normalizare: $10100100001.0101_2 = 1.01001000010101_2 \times 2^{10}$
  - mantisa este 01001000010101000000000
  - exponentul este $10 + 127 = 137 = 10001001_2$
  - semnul este 1



- **$x = (-1)^s \ 1.mmmmmmmmmmmmmmmmmmmmmmm \ 2^{(eeeeeeee)_2 - 127}$**

Schimbați semnul lui a: a = a ^ (1 << 31)
Exponentul pe: 0x7F80.0000
Extragem exponentul: (a & 0x7F80.0000) >> 23
Pentru a împărți la 4
Exponent > 1 => Exponent = Exponent – 2 | Altfel: a = 0
a = (a & ~MASK) | (exponent << 23)

SEMN	a >> 31
EXPONENT	(a >> 23) & 0x000000FF
MANTISĂ	a & 0x007FFFFF
Nr. în baza 10 cu mantisă de M biți	$\lfloor \log_{10} 2 \times M \rfloor \approx M/3$

abs(a)	a = a & ~(1 << 31)
a x 2	a << 1 sau a + a
a x 2	(a < 0) ? -((-a) << 1)): a << 1
a x 16	a << 4
a x 3	a << 1 + a
a x 7	a << 3 - a
a / 8	a >> 3
a % 16	a & 0x000F
a % m	a&(m − 1)
a / 16	(a & FFF0) >> 4
a x 72	a << 6 + a << 3
a / 16 + a % 16	a & FFF0 + a & 000F

d) 0xDEADBEEF = 0b11011110101011011011111011101111

- S = 1
- E = 10111101
- M = 01011011011111011101111
- (-1)^S 1.M * 2^{E-127} = (-1) 1.01011011011111011101111 $2^{189-127}$
  = - 1.01011011011111011101111 2^{62}
  = -6259853398707798000

- **setați s = 0, e = 0, f = 0**
- **a = $(-1)^0 \times 1.00\ldots00 \times 2^{-127} = 2^{-127} \neq 0$**

- **0.2 + 0.3**

- **primul pas, trecem fiecare număr în format**
  - 0.2 = + 1.10011001100110011001100 x 2^{-3} = 0.19999998807907104
  - 0.3 = + 1.00110011001100110011001 x 2^{-2} = 0.29999998211860657

- **al doilea pas, alinierea**
  - 0.2 = + 0.11001100110011001100110|000 x 2^{-2}
  - 0.3 = + 1.00110011001100110011001|000 x 2^{-2}

- **al treilea pas, adunăm**
  - 0.2 + 0.3 = 1.11111111111111111111111|000 x 2^{-2}

- **a / 19**

$$a \times \frac{1}{19} \approx \frac{a \times \frac{2938661835}{2^{32}} + \frac{a - a \times \frac{2938661835}{2^{32}}}{2^1}}{2^4}$$
$$a \times \frac{1}{19} \approx (a \times 2938661835 \times 2^{-32} + (a - a \times 2938661835 \times 2^{-32}) \times 2^{-1}) \times 2^{-4}$$
$$a \times \frac{1}{19} \approx a \times \frac{7233629131}{137438953472}$$

- **soluția generală**

$$\frac{a}{D} \approx \frac{\frac{aC}{2^X} + \frac{a - \frac{aC}{2^X}}{2^Y}}{2^Z}$$
$$D \approx \frac{2^{X+Y+Z}}{C \times (2^Y - 1) + 2^X}$$

- **calculați aproximarea binară**
  - soluția: 0.099999904632568359375
- **care este diferența dintre valoarea calculată și 0.1**
  - soluția: 0.1 - 0.099999904632568359375
- **care este eroarea (de timp) după 100 de ore de operare**
  - soluția: 100x60x60x10x(0.1 - 0.099999904632568359375) ≈ 0.34
- **care este eroarea dacă reprezentăm 0.1 în formatul IEEE 754 FP?**
  - soluția: 100x60x60x10x(0.1 - 0.09999999403953552) ≈ 0.021
- **dacă rachetele SCUD pot atinge o viteza MACH 5, care este distanța pe care racheta o poate parcurge în timpul eroare calculat?**
  - soluția: 1715 m/s * 0.34 s ≈ 583 m, 1715 m/s * 0.021 s ≈ 36 m

FAST INVERSE SQUARE ROOT, Q III

```
552  float Q_rsqrt( float number )
553  {
554          long i;
555          float x2, y;
556          const float threehalfs = 1.5F;
557
558          x2 = number * 0.5F;
559          y  = number;
560          i  = * ( long * ) &y;                                   // evil floating point bit level hacking
561          i  = 0x5f3759df - ( i >> 1 );               // what the duck?
562          y  = * ( float * ) &i;
563          y  = y * ( threehalfs - ( x2 * y * y ) );   // 1st iteration
564  //      y  = y * ( threehalfs - ( x2 * y * y ) );   // 2nd iteration, this can be removed
```

- **prima instrucțiune interesantă e pe linia 560**

- **dacă avem în M mantisa și exponentul în E atunci**
  - numărul nostru în FP este 2^{23} x E + M
  - iar valoarea numărului este (1 + M / 2^{23}) x 2^{E-127}
  - observăm că:

$$\begin{aligned}\log_2\left(\left(1+\frac{M}{2^{23}}\right)\times 2^{E-127}\right) &= \log_2\left(1+\frac{M}{2^{23}}\right)+\log_2\left(2^{E-127}\right)\\ &= \log_2\left(1+\frac{M}{2^{23}}\right)+E-127\\ &\approx \frac{M}{2^{23}}+E-127+\mu \quad \text{(am folosit } \log_2(1+x)\approx x, \mu \text{ este o corectie)}\\ &= \frac{1}{2^{23}}(2^{23}\times E+M)+\mu-127\\ &= \frac{1}{2^{23}}(\text{reprezentarea pe biti})+\mu-127\end{aligned}$$

  - **de ce calculăm log(y)? defapt vrem 1/sqrt(y). dar:**

$$\log_2\left(\frac{1}{\sqrt{y}}\right)=\log_2\left(y^{-1/2}\right)=-\frac{1}{2}\log_2(y)=-(i\gg 1)$$

- $\pi \approx 3.14159265 = (-1)^0\ 1.10010010000111111011010\ 2^{b10000000\ -127}$
- $+0 = (-1)^0\ \ 1.00000000000000000000000\ 2^{00000000-127}$
- $-0 = (-1)^1\ \ 1.00000000000000000000000\ 2^{00000000-127}$
- signaling NaN: 0x7F800001 sau 0x7FBFFFFF sau între 0xFF800001 și 0xFFBFFFFF
- quiet NaN: 0x7FC00000 sau 0x7FFFFFFF sau între 0xFFC00000 și 0xFFFFFFFF

CONSECINȚE FLOATING POINT

- (0.1 + 0.2) == 0.3 versus (0.2 + 0.3) == 0.5 (rotunjiri)
- math.sqrt(3)*math.sqrt(3) == 3 versus math.sqrt(3*3) == 3
- (0.7 + 0.2) + 0.1 versus (0.7 + 0.1) + 0.2 (nu avem asociativitatea)
- diferența cu numere întregi
  - dacă folosim tip de date întreg: 16777216 + 1 = 16777217
  - dacă folosim tip de date FP: 16777216.0 + 1 = 16777216.0
  - float(123456789101112) + 1.0 = 123456789101113.0
  - float(1234567891011121) + 1.0 = 1234567891011122.0
  - float(12345678910111213) + 1.0 = 1.2345678910111212e+16

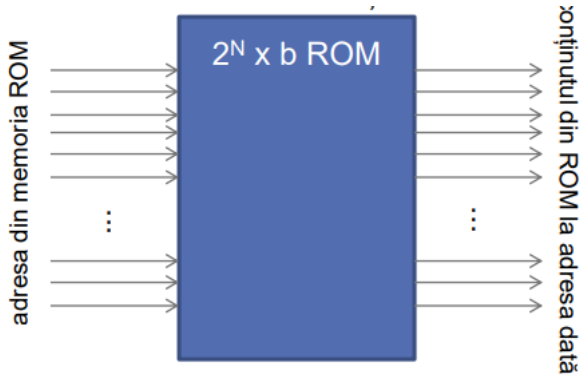
ARHITECTURA CALCULATOARELOR MODERNE

Pornirea sistemului
- Buton de power ON/OFF > Realizează alimentarea cu electricitate a componentelor
- CPU este activat > Caută pornește BIOS
    - Testează componentele HW (RAM, I/O, HDD, etc.)
    - Încarcă BIOS (scris pe placa de bază) din ROM (read only memory) în RAM pentru execuţie

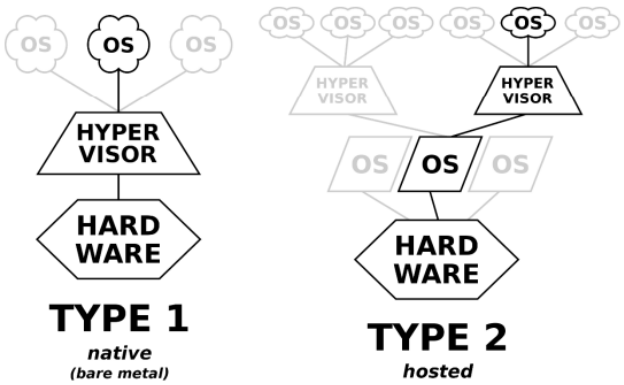
BIOS: Ştie cât e ceasul (CMOS Real-Time Clock) şi HW, se accesează cu F2 la pornirea sistemului

CPU/BIOS: Pornesc Boot Code (caută sistemul de operare)
    - În general, SO este pe HD (poate fi şi pe CD, stick). Se încarcă în RAM pentru execuţie

Componente
TOT BIOS: Scris în ROM, câteodată în Programmable ROM, Erasble Programmable ROM
„Să scriem în ROM": „burning" sau „flashing" the ROM
Este un circuit combinaţional

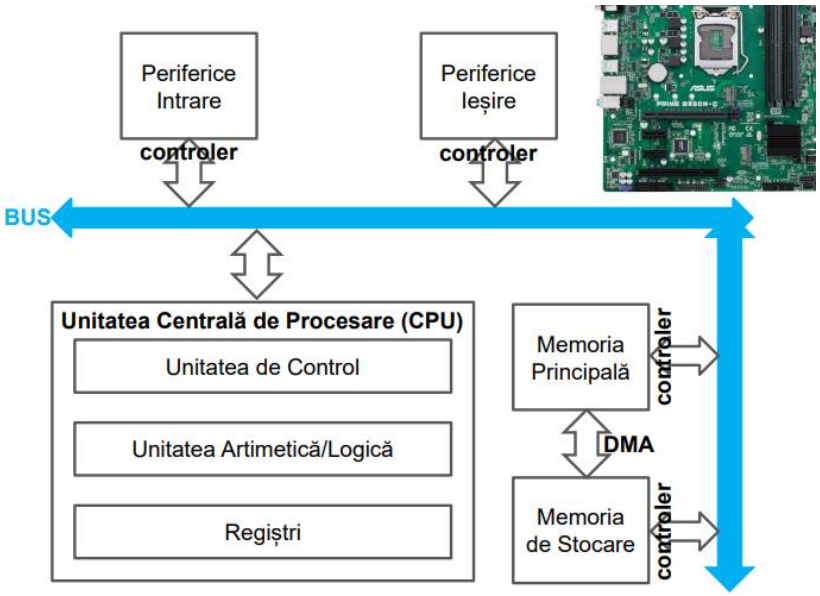


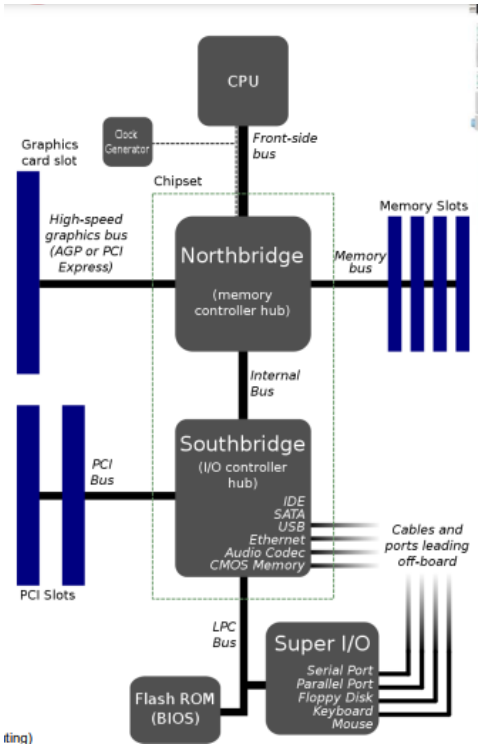
OS preia controlul de la BIOS
    - Doar OS are acces la periferice, prin drivere
    - Virtualizare/Emulare/Containere(Dockere)

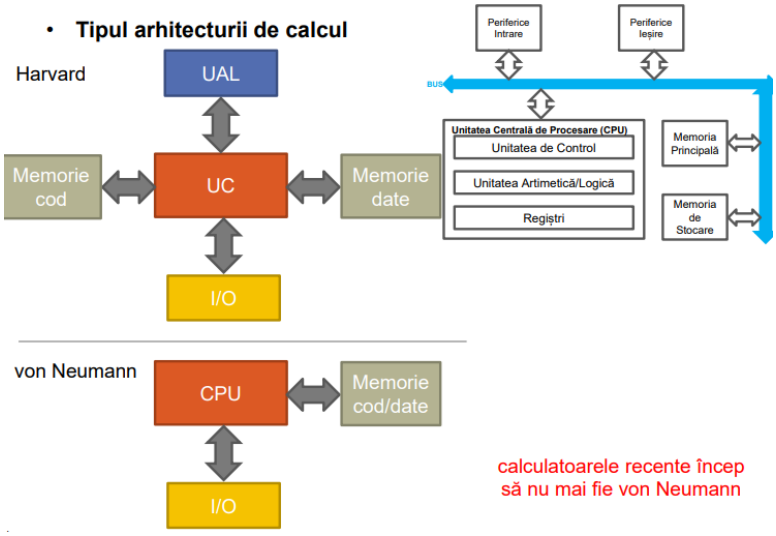


OS: Oferă imagine abstractizată a memoriei pentru fiecare proces pornit.
OS pornește > Sistemul de calcul intră în ciclul obișnuit de procesare
(Secvența boot se încheie)

Un sistem de calcul trebuie să
- Calculeze > Să execute instrucțiuni
- Să comunice > Să transfere biți între componente electronice
- Să stocheze > Date - folosite de instrucțiuni | Instrucțiuni pentru execuție







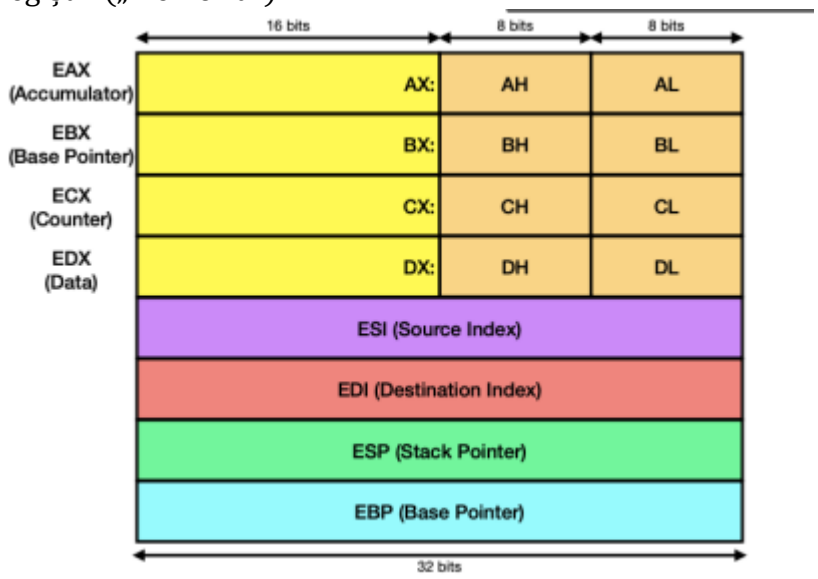
calculatoarele recente încep să nu mai fie von Neumann

CPU (5 componente) – Creierul unității de calcul / Execută instrucțiuni

1. Clock

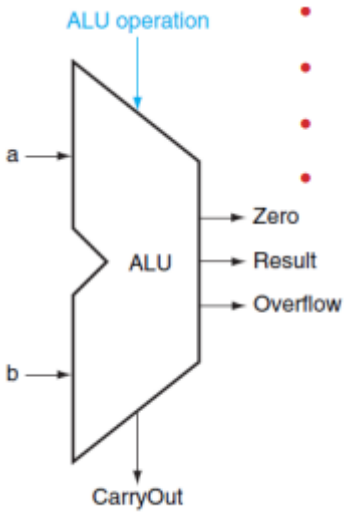
- Circuit special care generează „ceasul”
- Frecvența la care operează CPU (calcule + sincronizarea componentelor secvențiale)
- Frecvență mare -> Mai bine | Se măsoară în MHz sau GHz

2. Regiștri („memoria”)



3. UAL („operații”) – Unitatea aritmetică logică

- Operații logice
- Operații aritmetice cu int/float
- Operații speciale: sqrt, exp, trig



4. BUS

Sinteză Extinsă: Arhitectura Sistemelor de Calcul

Generat pe baza materialelor de curs

Cuprins

1 Personalități și Contribuții în Informatică

Tabelul de mai jos sintetizează figurile istorice menționate în cursuri, completat cu creatorii limbajelor de programare clasice.

Nume	**Realizare / Contribuție**
Al-Khwarizmi	Părintele algebrei, termenul de "algoritm".
Blaise Pascal	Calculatorul mecanic *Pascaline* (adunări/scăderi).
Gottfried Wilhelm von Leibniz	A studiat sistemul binar; a extins mașina lui Pascal pentru înmulțire/împărțire.
George Boole	Logica Booleană (baza teoriei informației și a circuitelor digitale).
Charles Babbage	"Tatăl calculatoarelor"; a proiectat *Difference Engine* și *Analytical Engine*.
Ada Lovelace	Primul programator din istorie; a scris primul algoritm pentru mașina lui Babbage.
Konrad Zuse	Seria de calculatoare Z1-Z4; primele calculatoare programabile funcționale (binare).
Alan Turing	Mașina Turing (model teoretic universal), spargerea codului Enigma, Testul Turing.
John von Neumann	Arhitectura Von Neumann (programe stocate în memorie alături de date), EDVAC.
Claude Shannon	"Părintele teoriei informației"; a demonstrat că logica booleană poate fi implementată cu circuite.
Grace Hopper	A creat primul compilator; a popularizat termenul "debugging"; a contribuit la COBOL.
Dennis Ritchie	**Creatorul limbajului C**; co-creator UNIX.
Ken Thompson	Co-creator UNIX; creatorul limbajului B (precursor C).
Bjarne Stroustrup	**Creatorul limbajului C++**.
Niklaus Wirth	**Creatorul limbajului Pascal**.
James Gosling	**Creatorul limbajului Java**.
Guido van Rossum	**Creatorul limbajului Python**.
Linus Torvalds	Creatorul kernel-ului Linux și al sistemului Git.
Richard Stallman	Fondatorul mișcării Free Software (GNU).
Larry Page & Sergey Brin	Fondatorii Google; algoritmul PageRank.
Rivest, Shamir, Adleman	Algoritmul de criptare RSA (cheie publică).
Diffie & Hellman	Pionieri în criptografia cu cheie publică (schimbul de chei).

2 Cap. 1: Introducere și Performanță

2.1 Cele 8 Mari Idei în Arhitectura Calculatoarelor

Cursul subliniază principii fundamentale pentru designul hardware:

1. **Design for Moore's Law:** Arhitecții trebuie să anticipeze creșterea puterii de calcul în timpul procesului de design.
2. **Use Abstraction to Simplify Design:** Ascunderea detaliilor hardware low-level (ex: ISA - Instruction Set Architecture este interfața dintre hardware și software).
3. **Make the Common Case Fast:** Optimizarea instrucțiunilor folosite cel mai des (ex: adunări, bucle) aduce cel mai mare câștig de performanță.
4. **Parallelism:** Executarea mai multor operații simultan.
5. **Pipelining:** Suprapunerea execuției instrucțiunilor (ca o linie de asamblare).
6. **Prediction:** Ghicirea rezultatului unei condiții (Branch Prediction) pentru a nu opri procesorul.
7. **Hierarchy of Memories:** Utilizarea cache-urilor pentru a simula o memorie care este și rapidă (ca SRAM) și mare (ca DRAM).
8. **Dependability via Redundancy:** Folosirea componentelor de rezervă pentru a tolera erorile (ex: RAID, ECC memory).

2.2 Măsurarea Performanței

Performanța este inversul timpului de execuție.

$$Performanța_X = \frac{1}{TimpExecutie_X} \tag{1}$$

2.2.1 Ecuația Performanței CPU

Timpul de execuție al unui program depinde de trei factori cheie:

$$CPU\_Time = \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle Time} \tag{2}$$

Unde:

- **Instruction Count (IC):** Numărul de instrucțiuni executate (depinde de compilator și arhitectură).
- **CPI (Cycles Per Instruction):** Numărul mediu de cicli necesari pentru o instrucțiune (depinde de organizarea procesorului).
- **Clock Cycle Time:** Durata unui ciclu de ceas (inversul frecvenței: $1/f$).

2.2.2 Legea lui Amdahl

Ne spune limita îmbunătățirii performanței atunci când optimizăm doar o parte a sistemului.

$$Timp_{nou} = Timp_{neafectat} + \frac{Timp_{afectat}}{Speedup} \tag{3}$$

Concluzie: Dacă o componentă este folosită doar 10% din timp, chiar dacă o facem infinit de rapidă, performanța totală nu va crește semnificativ.

3 Cap. 2: Reprezentarea Datelor

3.1 Numere Întregi (Complement față de 2)

Este metoda standard pentru numere cu semn.

- **Reprezentare:** Bitul cel mai semnificativ (MSB) are pondere negativă.
- **Formula:** Valoare $= -x_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} x_i \cdot 2^i$.
- **Negarea unui număr:** Inversarea tuturor biților (NOT) + 1.
- **Sign Extension:** Când mutăm un număr pe mai mulți biți (ex: 8 biți $\rightarrow$ 16 biți), copiem bitul de semn la stânga pentru a păstra valoarea.

3.2 Numere Reale (IEEE 754 Floating Point)

Standardul pentru numerele cu virgulă mobilă. Formatul pe 32 de biți (Single Precision):

Semn (S)	**Exponent (E)**	**Mantisă/Fracție (F)**
1 bit	8 biți	23 biți

Caracteristici cheie:

- **Bias (Deplasament):** Exponentul stocat nu este cel real. $E_{stocat} = E_{real} + 127$. Asta permite compararea ușoară a numerelor.
- **Normalizare:** Mantisa are un "1 implicit" în fața virgulei ($1.F...$).
- **Cazuri Speciale:**
  - $E = 0, F = 0 \Rightarrow$ **Zero**.
  - $E = 255, F = 0 \Rightarrow$ **Infinit** ($\pm\infty$).
  - $E = 255, F \neq 0 \Rightarrow$ **NaN** (Not a Number - ex: $0/0$).

4 Cap. 3: Arhitectura Setului de Instrucțiuni (ISA)

4.1 CISC vs. RISC

O comparație fundamentală în designul procesoarelor.

CISC (Complex Instruction Set Computer)	**RISC (Reduced Instruction Set Computer)**
Exemple: x86 (Intel, AMD)	**Exemple:** ARM, MIPS, RISC-V
Instrucțiuni complexe, pot face multe operații într-un singur pas (ex: încarcă din memorie, adună, salvează).	Instrucțiuni simple și atomice (Load, Add, Store sunt separate).
Lungime variabilă a instrucțiunilor (1-15 bytes).	Lungime fixă (de obicei 4 bytes / 32 biți).
Hardware complex (decodificator greu), compilator simplu.	Hardware simplu, compilator complex (trebuie să scrie mai mult cod).
Accent pe minimizarea numărului de instrucțiuni (Code density).	Accent pe viteza de execuție (CPI redus, frecvență mare).
Permite operanzi direct din Memorie.	Arhitectură "Load/Store" (operează doar pe regiștri).

5 Cap. 4: Procesorul (Datapath Control)

5.1 Etapele Execuției (Pipeline în 5 stadii - MIPS/RISC)

Procesorul modern execută instrucțiunile într-o "bandă de asamblare":

1. **IF (Instruction Fetch):** Citirea instrucțiunii de la adresa din PC (Program Counter).
2. **ID (Instruction Decode):** Decodificarea instrucțiunii și citirea valorilor din regiștri.
3. **EX (Execute):** ALU (Unitatea Aritmetică) calculează rezultatul sau adresa de memorie.
4. **MEM (Memory Access):** Citirea sau scrierea în memoria de date (doar pentru Load/Store).
5. **WB (Write Back):** Scrierea rezultatului final înapoi în regiștrii procesorului.

5.2 Hazards (Probleme în Pipeline)

Pipelining-ul crește viteza, dar apar conflicte:

- **Structural Hazards:** Două instrucțiuni vor să folosească aceeași resursă hardware simultan (ex: acces la memorie). *Soluție:* Memorii separate pentru Instrucțiuni și Date (L1 Cache split).
- **Data Hazards:** O instrucțiune are nevoie de rezultatul unei instrucțiuni anterioare care nu s-a terminat.

  - *Soluție (Software):* Introducerea de instrucțiuni NOP (No Operation).
  - *Soluție (Hardware):* **Forwarding** (trimiterea datelor direct din ALU în ALU, fără a aștepta scrierea în regiștri).

- **Control Hazards:** Procesorul nu știe ce instrucțiune urmează după un 'if' (branch) până nu evaluează condiția.
  - *Soluție:* **Branch Prediction** (Static - presupunem mereu "Not Taken"; Dinamic - învățăm din istoric).

6 Cap. 5: Ierarhia Memoriei

6.1 Tipuri de Memorie

- **SRAM (Static RAM):** Rapidă, nu necesită refresh, scumpă, densitate mică. Folosită pentru **Cache**.
- **DRAM (Dynamic RAM):** Mai lentă, necesită refresh (condensatori), ieftină, densitate mare. Folosită pentru **Memoria Principală**.
- **Flash/Disk:** Non-volatilă, cea mai lentă.

6.2 Memoria Cache

Rezolvă diferența de viteză dintre CPU și RAM.

- **Direct Mapped Cache:** Fiecare adresă de memorie se mapează într-un singur loc fix în cache. (Formula: $Address \mod CacheSize$). Simplu, dar multe coliziuni.
- **Set Associative Cache:** O adresă poate fi stocată în oricare din N locații ale unui set. Compromis ideal între complexitate și performanță.
- **Fully Associative:** O dată poate fi oriunde. Căutare lentă hardware, folosită rar (ex: TLB).

Politici de scriere:

- **Write-Through:** Scriem datele simultan în Cache și în RAM (sigur, dar lent).
- **Write-Back:** Scriem doar în Cache. Scriem în RAM doar când blocul e înlocuit (rapid, dar complex).

6.3 Memoria Virtuală

Permite rularea programelor mai mari decât memoria fizică.

- Împarte memoria în **Pagini** (ex: 4KB).
- **Page Table:** Dicționarul care traduce Adrese Virtuale în Adrese Fizice.
- **TLB (Translation Lookaside Buffer):** Un "Cache pentru Page Table" (esențial pentru viteză).
- **Page Fault:** Când datele nu sunt în RAM și trebuie aduse de pe HDD/SSD (proces foarte lent, gestionat de OS).

7 Cap. 6: Paralelism și GPU

- **GPU (Graphics Processing Unit):** Proiectat pentru throughput mare (multe date procesate la fel).
- Arhitectură **SIMD** (Single Instruction, Multiple Data): Aceeași instrucțiune se execută pe mii de pixeli/date simultan.

- Diferență față de CPU: CPU e optimizat pentru latență scăzută (să termine un task repede), GPU pentru lățime de bandă (să termine multe task-uri deodată).

Table 1: Tabel transpus și mărit

i	x=1	x=16	x=256	x=4096
0	0	0	0	0
1	1	16	256	4096
2	2	32	512	8192
3	3	48	768	12288
4	4	64	1024	16384
5	5	80	1280	20480
6	6	96	1536	24576
7	7	112	1792	28672
8	8	128	2048	32768
9	9	144	2304	36864
10	10	160	2560	40960
11	11	176	2816	45056
12	12	192	3072	49152
13	13	208	3328	53248
14	14	224	3584	57344
15	15	240	3840	61440

1 001	2 002	3 003	4 004	5 005	6 006	7 007
8 008	9 009	10 00A	11 00B	12 00C	13 00D	14 00E
15 00F	16 010	17 011	18 012	19 013	20 014	21 015
22 016	23 017	24 018	25 019	26 01A	27 01B	28 01C
29 01D	30 01E	31 01F	32 020	33 021	34 022	35 023
36 024	37 025	38 026	39 027	40 028	41 029	42 02A
43 02B	44 02C	45 02D	46 02E	47 02F	48 030	49 031
50 032	51 033	52 034	53 035	54 036	55 037	56 038
57 039	58 03A	59 03B	60 03C	61 03D	62 03E	63 03F
64 040	65 041	66 042	67 043	68 044	69 045	70 046
71 047	72 048	73 049	74 04A	75 04B	76 04C	77 04D
78 04E	79 04F	80 050	81 051	82 052	83 053	84 054
85 055	86 056	87 057	88 058	89 059	90 05A	91 05B
92 05C	93 05D	94 05E	95 05F	96 060	97 061	98 062
99 063	100 064	101 065	102 066	103 067	104 068	105 069
106 06A	107 06B	108 06C	109 06D	110 06E	111 06F	112 070
113 071	114 072	115 073	116 074	117 075	118 076	119 077
120 078	121 079	122 07A	123 07B	124 07C	125 07D	126 07E
127 07F	128 080	129 081	130 082	131 083	132 084	133 085
134 086	135 087	136 088	137 089	138 08A	139 08B	140 08C
141 08D	142 08E	143 08F	144 090	145 091	146 092	147 093
148 094	149 095	150 096	151 097	152 098	153 099	154 09A
155 09B	156 09C	157 09D	158 09E	159 09F	160 0A0	161 0A1
162 0A2	163 0A3	164 0A4	165 0A5	166 0A6	167 0A7	168 0A8
169 0A9	170 0AA	171 0AB	172 0AC	173 0AD	174 0AE	175 0AF
176 0B0	177 0B1	178 0B2	179 0B3	180 0B4	181 0B5	182 0B6
183 0B7	184 0B8	185 0B9	186 0BA	187 0BB	188 0BC	189 0BD
190 0BE	191 0BF	192 0C0	193 0C1	194 0C2	195 0C3	196 0C4
197 0C5	198 0C6	199 0C7	200 0C8	201 0C9	202 0CA	203 0CB
204 0CC	205 0CD	206 0CE	207 0CF	208 0D0	209 0D1	210 0D2
211 0D3	212 0D4	213 0D5	214 0D6	215 0D7	216 0D8	217 0D9
218 0DA	219 0DB	220 0DC	221 0DD	222 0DE	223 0DF	224 0E0
225 0E1	226 0E2	227 0E3	228 0E4	229 0E5	230 0E6	231 0E7
232 0E8	233 0E9	234 0EA	235 0EB	236 0EC	237 0ED	238 0EE
239 0EF	240 0F0	241 0F1	242 0F2	243 0F3	244 0F4	245 0F5
246 0F6	247 0F7	248 0F8	249 0F9	250 0FA	251 0FB	252 0FC
253 0FD	254 0FE	255 0FF	256 100	257 101	258 102	259 103
260 104	261 105	262 106	263 107	264 108	265 109	266 10A
267 10B	268 10C	269 10D	270 10E	271 10F	272 110	273 111
274 112	275 113	276 114	277 115	278 116	279 117	280 118
281 119	282 11A	283 11B	284 11C	285 11D	286 11E	287 11F
288 120	289 121	290 122	291 123	292 124	293 125	294 126
295 127	296 128	297 129	298 12A	299 12B	300 12C	301 12D
302 12E	303 12F	304 130	305 131	306 132	307 133	308 134
309 135	310 136	311 137	312 138	313 139	314 13A	315 13B
316 13C	317 13D	318 13E	319 13F	320 140	321 141	322 142
323 143	324 144	325 145	326 146	327 147	328 148	329 149
330 14A	331 14B	332 14C	333 14D	334 14E	335 14F	336 150
337 151	338 152	339 153	340 154	341 155	342 156	343 157
344 158	345 159	346 15A	347 15B	348 15C	349 15D	350 15E
351 15F	352 160	353 161	354 162	355 163	356 164	357 165
358 166	359 167	360 168	361 169	362 16A	363 16B	364 16C
365 16D	366 16E	367 16F	368 170	369 171	370 172	371 173
372 174	373 175	374 176	375 177	376 178	377 179	378 17A
379 17B	380 17C	381 17D	382 17E	383 17F	384 180	385 181
386 182	387 183	388 184	389 185	390 186	391 187	392 188
393 189	394 18A	395 18B	396 18C	397 18D	398 18E	399 18F
400 190	401 191	402 192	403 193	404 194	405 195	406 196
407 197	408 198	409 199	410 19A	411 19B	412 19C	413 19D
414 19E	415 19F	416 1A0	417 1A1	418 1A2	419 1A3	420 1A4
421 1A5	422 1A6	423 1A7	424 1A8	425 1A9	426 1AA	427 1AB
428 1AC	429 1AD	430 1AE	431 1AF	432 1B0	433 1B1	434 1B2
435 1B3	436 1B4	437 1B5	438 1B6	439 1B7	440 1B8	441 1B9

442 1BA	443 1BB	444 1BC	445 1BD	446 1BE	447 1BF	448 1C0
449 1C1	450 1C2	451 1C3	452 1C4	453 1C5	454 1C6	455 1C7
456 1C8	457 1C9	458 1CA	459 1CB	460 1CC	461 1CD	462 1CE
463 1CF	464 1D0	465 1D1	466 1D2	467 1D3	468 1D4	469 1D5
470 1D6	471 1D7	472 1D8	473 1D9	474 1DA	475 1DB	476 1DC
477 1DD	478 1DE	479 1DF	480 1E0	481 1E1	482 1E2	483 1E3
484 1E4	485 1E5	486 1E6	487 1E7	488 1E8	489 1E9	490 1EA
491 1EB	492 1EC	493 1ED	494 1EE	495 1EF	496 1F0	497 1F1
498 1F2	499 1F3	500 1F4	501 1F5	502 1F6	503 1F7	504 1F8
505 1F9	506 1FA	507 1FB	508 1FC	509 1FD	510 1FE	511 1FF
512 200	513 201	514 202	515 203	516 204	517 205	518 206
519 207	520 208	521 209	522 20A	523 20B	524 20C	525 20D
526 20E	527 20F	528 210	529 211	530 212	531 213	532 214
533 215	534 216	535 217	536 218	537 219	538 21A	539 21B
540 21C	541 21D	542 21E	543 21F	544 220	545 221	546 222
547 223	548 224	549 225	550 226	551 227	552 228	553 229
554 22A	555 22B	556 22C	557 22D	558 22E	559 22F	560 230
561 231	562 232	563 233	564 234	565 235	566 236	567 237
568 238	569 239	570 23A	571 23B	572 23C	573 23D	574 23E
575 23F	576 240	577 241	578 242	579 243	580 244	581 245
582 246	583 247	584 248	585 249	586 24A	587 24B	588 24C
589 24D	590 24E	591 24F	592 250	593 251	594 252	595 253
596 254	597 255	598 256	599 257	600 258	601 259	602 25A
603 25B	604 25C	605 25D	606 25E	607 25F	608 260	609 261
610 262	611 263	612 264	613 265	614 266	615 267	616 268
617 269	618 26A	619 26B	620 26C	621 26D	622 26E	623 26F
624 270	625 271	626 272	627 273	628 274	629 275	630 276
631 277	632 278	633 279	634 27A	635 27B	636 27C	637 27D
638 27E	639 27F	640 280	641 281	642 282	643 283	644 284
645 285	646 286	647 287	648 288	649 289	650 28A	651 28B
652 28C	653 28D	654 28E	655 28F	656 290	657 291	658 292
659 293	660 294	661 295	662 296	663 297	664 298	665 299
666 29A	667 29B	668 29C	669 29D	670 29E	671 29F	672 2A0
673 2A1	674 2A2	675 2A3	676 2A4	677 2A5	678 2A6	679 2A7
680 2A8	681 2A9	682 2AA	683 2AB	684 2AC	685 2AD	686 2AE
687 2AF	688 2B0	689 2B1	690 2B2	691 2B3	692 2B4	693 2B5
694 2B6	695 2B7	696 2B8	697 2B9	698 2BA	699 2BB	700 2BC
701 2BD	702 2BE	703 2BF	704 2C0	705 2C1	706 2C2	707 2C3
708 2C4	709 2C5	710 2C6	711 2C7	712 2C8	713 2C9	714 2CA
715 2CB	716 2CC	717 2CD	718 2CE	719 2CF	720 2D0	721 2D1
722 2D2	723 2D3	724 2D4	725 2D5	726 2D6	727 2D7	728 2D8
729 2D9	730 2DA	731 2DB	732 2DC	733 2DD	734 2DE	735 2DF
736 2E0	737 2E1	738 2E2	739 2E3	740 2E4	741 2E5	742 2E6
743 2E7	744 2E8	745 2E9	746 2EA	747 2EB	748 2EC	749 2ED
750 2EE	751 2EF	752 2F0	753 2F1	754 2F2	755 2F3	756 2F4
757 2F5	758 2F6	759 2F7	760 2F8	761 2F9	762 2FA	763 2FB
764 2FC	765 2FD	766 2FE	767 2FF	768 300	769 301	770 302
771 303	772 304	773 305	774 306	775 307	776 308	777 309
778 30A	779 30B	780 30C	781 30D	782 30E	783 30F	784 310
785 311	786 312	787 313	788 314	789 315	790 316	791 317
792 318	793 319	794 31A	795 31B	796 31C	797 31D	798 31E
799 31F	800 320	801 321	802 322	803 323	804 324	805 325
806 326	807 327	808 328	809 329	810 32A	811 32B	812 32C
813 32D	814 32E	815 32F	816 330	817 331	818 332	819 333
820 334	821 335	822 336	823 337	824 338	825 339	826 33A
827 33B	828 33C	829 33D	830 33E	831 33F	832 340	833 341
834 342	835 343	836 344	837 345	838 346	839 347	840 348
841 349	842 34A	843 34B	844 34C	845 34D	846 34E	847 34F
848 350	849 351	850 352	851 353	852 354	853 355	854 356
855 357	856 358	857 359	858 35A	859 35B	860 35C	861 35D
862 35E	863 35F	864 360	865 361	866 362	867 363	868 364
869 365	870 366	871 367	872 368	873 369	874 36A	875 36B
876 36C	877 36D	878 36E	879 36F	880 370	881 371	882 372

883 373	884 374	885 375	886 376	887 377	888 378	889 379
890 37A	891 37B	892 37C	893 37D	894 37E	895 37F	896 380
897 381	898 382	899 383	900 384	901 385	902 386	903 387
904 388	905 389	906 38A	907 38B	908 38C	909 38D	910 38E
911 38F	912 390	913 391	914 392	915 393	916 394	917 395
918 396	919 397	920 398	921 399	922 39A	923 39B	924 39C
925 39D	926 39E	927 39F	928 3A0	929 3A1	930 3A2	931 3A3
932 3A4	933 3A5	934 3A6	935 3A7	936 3A8	937 3A9	938 3AA
939 3AB	940 3AC	941 3AD	942 3AE	943 3AF	944 3B0	945 3B1
946 3B2	947 3B3	948 3B4	949 3B5	950 3B6	951 3B7	952 3B8
953 3B9	954 3BA	955 3BB	956 3BC	957 3BD	958 3BE	959 3BF
960 3C0	961 3C1	962 3C2	963 3C3	964 3C4	965 3C5	966 3C6
967 3C7	968 3C8	969 3C9	970 3CA	971 3CB	972 3CC	973 3CD
974 3CE	975 3CF	976 3D0	977 3D1	978 3D2	979 3D3	980 3D4
981 3D5	982 3D6	983 3D7	984 3D8	985 3D9	986 3DA	987 3DB
988 3DC	989 3DD	990 3DE	991 3DF	992 3E0	993 3E1	994 3E2
995 3E3	996 3E4	997 3E5	998 3E6	999 3E7	1000 3E8	1001 3E9
1002 3EA	1003 3EB	1004 3EC	1005 3ED	1006 3EE	1007 3EF	1008 3F0
1009 3F1	1010 3F2	1011 3F3	1012 3F4	1013 3F5	1014 3F6	1015 3F7
1016 3F8	1017 3F9	1018 3FA	1019 3FB	1020 3FC	1021 3FD	1022 3FE
1023 3FF	1024 400	1025 401	1026 402	1027 403	1028 404	1029 405
1030 406	1031 407	1032 408	1033 409	1034 40A	1035 40B	1036 40C
1037 40D	1038 40E	1039 40F	1040 410	1041 411	1042 412	1043 413
1044 414	1045 415	1046 416	1047 417	1048 418	1049 419	1050 41A
1051 41B	1052 41C	1053 41D	1054 41E	1055 41F	1056 420	1057 421
1058 422	1059 423	1060 424	1061 425	1062 426	1063 427	1064 428
1065 429	1066 42A	1067 42B	1068 42C	1069 42D	1070 42E	1071 42F
1072 430	1073 431	1074 432	1075 433	1076 434	1077 435	1078 436
1079 437	1080 438	1081 439	1082 43A	1083 43B	1084 43C	1085 43D
1086 43E	1087 43F	1088 440	1089 441	1090 442	1091 443	1092 444
1093 445	1094 446	1095 447	1096 448	1097 449	1098 44A	1099 44B
1100 44C	1101 44D	1102 44E	1103 44F	1104 450	1105 451	1106 452
1107 453	1108 454	1109 455	1110 456	1111 457	1112 458	1113 459
1114 45A	1115 45B	1116 45C	1117 45D	1118 45E	1119 45F	1120 460
1121 461	1122 462	1123 463	1124 464	1125 465	1126 466	1127 467
1128 468	1129 469	1130 46A	1131 46B	1132 46C	1133 46D	1134 46E
1135 46F	1136 470	1137 471	1138 472	1139 473	1140 474	1141 475
1142 476	1143 477	1144 478	1145 479	1146 47A	1147 47B	1148 47C
1149 47D	1150 47E	1151 47F	1152 480	1153 481	1154 482	1155 483
1156 484	1157 485	1158 486	1159 487	1160 488	1161 489	1162 48A
1163 48B	1164 48C	1165 48D	1166 48E	1167 48F	1168 490	1169 491
1170 492	1171 493	1172 494	1173 495	1174 496	1175 497	1176 498
1177 499	1178 49A	1179 49B	1180 49C	1181 49D	1182 49E	1183 49F
1184 4A0	1185 4A1	1186 4A2	1187 4A3	1188 4A4	1189 4A5	1190 4A6
1191 4A7	1192 4A8	1193 4A9	1194 4AA	1195 4AB	1196 4AC	1197 4AD
1198 4AE	1199 4AF	1200 4B0	1201 4B1	1202 4B2	1203 4B3	1204 4B4
1205 4B5	1206 4B6	1207 4B7	1208 4B8	1209 4B9	1210 4BA	1211 4BB
1212 4BC	1213 4BD	1214 4BE	1215 4BF	1216 4C0	1217 4C1	1218 4C2
1219 4C3	1220 4C4	1221 4C5	1222 4C6	1223 4C7	1224 4C8	1225 4C9
1226 4CA	1227 4CB	1228 4CC	1229 4CD	1230 4CE	1231 4CF	1232 4D0
1233 4D1	1234 4D2	1235 4D3	1236 4D4	1237 4D5	1238 4D6	1239 4D7
1240 4D8	1241 4D9	1242 4DA	1243 4DB	1244 4DC	1245 4DD	1246 4DE
1247 4DF	1248 4E0	1249 4E1	1250 4E2	1251 4E3	1252 4E4	1253 4E5
1254 4E6	1255 4E7	1256 4E8	1257 4E9	1258 4EA	1259 4EB	1260 4EC
1261 4ED	1262 4EE	1263 4EF	1264 4F0	1265 4F1	1266 4F2	1267 4F3
1268 4F4	1269 4F5	1270 4F6	1271 4F7	1272 4F8	1273 4F9	1274 4FA
1275 4FB	1276 4FC	1277 4FD	1278 4FE	1279 4FF	1280 500	1281 501
1282 502	1283 503	1284 504	1285 505	1286 506	1287 507	1288 508
1289 509	1290 50A	1291 50B	1292 50C	1293 50D	1294 50E	1295 50F
1296 510	1297 511	1298 512	1299 513	1300 514	1301 515	1302 516
1303 517	1304 518	1305 519	1306 51A	1307 51B	1308 51C	1309 51D
1310 51E	1311 51F	1312 520	1313 521	1314 522	1315 523	1316 524
1317 525	1318 526	1319 527	1320 528	1321 529	1322 52A	1323 52B

1324 52C	1325 52D	1326 52E	1327 52F	1328 530	1329 531	1330 532
1331 533	1332 534	1333 535	1334 536	1335 537	1336 538	1337 539
1338 53A	1339 53B	1340 53C	1341 53D	1342 53E	1343 53F	1344 540
1345 541	1346 542	1347 543	1348 544	1349 545	1350 546	1351 547
1352 548	1353 549	1354 54A	1355 54B	1356 54C	1357 54D	1358 54E
1359 54F	1360 550	1361 551	1362 552	1363 553	1364 554	1365 555
1366 556	1367 557	1368 558	1369 559	1370 55A	1371 55B	1372 55C
1373 55D	1374 55E	1375 55F	1376 560	1377 561	1378 562	1379 563
1380 564	1381 565	1382 566	1383 567	1384 568	1385 569	1386 56A
1387 56B	1388 56C	1389 56D	1390 56E	1391 56F	1392 570	1393 571
1394 572	1395 573	1396 574	1397 575	1398 576	1399 577	1400 578
1401 579	1402 57A	1403 57B	1404 57C	1405 57D	1406 57E	1407 57F
1408 580	1409 581	1410 582	1411 583	1412 584	1413 585	1414 586
1415 587	1416 588	1417 589	1418 58A	1419 58B	1420 58C	1421 58D
1422 58E	1423 58F	1424 590	1425 591	1426 592	1427 593	1428 594
1429 595	1430 596	1431 597	1432 598	1433 599	1434 59A	1435 59B
1436 59C	1437 59D	1438 59E	1439 59F	1440 5A0	1441 5A1	1442 5A2
1443 5A3	1444 5A4	1445 5A5	1446 5A6	1447 5A7	1448 5A8	1449 5A9
1450 5AA	1451 5AB	1452 5AC	1453 5AD	1454 5AE	1455 5AF	1456 5B0
1457 5B1	1458 5B2	1459 5B3	1460 5B4	1461 5B5	1462 5B6	1463 5B7
1464 5B8	1465 5B9	1466 5BA	1467 5BB	1468 5BC	1469 5BD	1470 5BE
1471 5BF	1472 5C0	1473 5C1	1474 5C2	1475 5C3	1476 5C4	1477 5C5
1478 5C6	1479 5C7	1480 5C8	1481 5C9	1482 5CA	1483 5CB	1484 5CC
1485 5CD	1486 5CE	1487 5CF	1488 5D0	1489 5D1	1490 5D2	1491 5D3
1492 5D4	1493 5D5	1494 5D6	1495 5D7	1496 5D8	1497 5D9	1498 5DA
1499 5DB	1500 5DC	1501 5DD	1502 5DE	1503 5DF	1504 5E0	1505 5E1
1506 5E2	1507 5E3	1508 5E4	1509 5E5	1510 5E6	1511 5E7	1512 5E8
1513 5E9	1514 5EA	1515 5EB	1516 5EC	1517 5ED	1518 5EE	1519 5EF
1520 5F0	1521 5F1	1522 5F2	1523 5F3	1524 5F4	1525 5F5	1526 5F6
1527 5F7	1528 5F8	1529 5F9	1530 5FA	1531 5FB	1532 5FC	1533 5FD
1534 5FE	1535 5FF	1536 600	1537 601	1538 602	1539 603	1540 604
1541 605	1542 606	1543 607	1544 608	1545 609	1546 60A	1547 60B
1548 60C	1549 60D	1550 60E	1551 60F	1552 610	1553 611	1554 612
1555 613	1556 614	1557 615	1558 616	1559 617	1560 618	1561 619
1562 61A	1563 61B	1564 61C	1565 61D	1566 61E	1567 61F	1568 620
1569 621	1570 622	1571 623	1572 624	1573 625	1574 626	1575 627
1576 628	1577 629	1578 62A	1579 62B	1580 62C	1581 62D	1582 62E
1583 62F	1584 630	1585 631	1586 632	1587 633	1588 634	1589 635
1590 636	1591 637	1592 638	1593 639	1594 63A	1595 63B	1596 63C
1597 63D	1598 63E	1599 63F	1600 640	1601 641	1602 642	1603 643
1604 644	1605 645	1606 646	1607 647	1608 648	1609 649	1610 64A
1611 64B	1612 64C	1613 64D	1614 64E	1615 64F	1616 650	1617 651
1618 652	1619 653	1620 654	1621 655	1622 656	1623 657	1624 658
1625 659	1626 65A	1627 65B	1628 65C	1629 65D	1630 65E	1631 65F
1632 660	1633 661	1634 662	1635 663	1636 664	1637 665	1638 666
1639 667	1640 668	1641 669	1642 66A	1643 66B	1644 66C	1645 66D
1646 66E	1647 66F	1648 670	1649 671	1650 672	1651 673	1652 674
1653 675	1654 676	1655 677	1656 678	1657 679	1658 67A	1659 67B
1660 67C	1661 67D	1662 67E	1663 67F	1664 680	1665 681	1666 682
1667 683	1668 684	1669 685	1670 686	1671 687	1672 688	1673 689
1674 68A	1675 68B	1676 68C	1677 68D	1678 68E	1679 68F	1680 690
1681 691	1682 692	1683 693	1684 694	1685 695	1686 696	1687 697
1688 698	1689 699	1690 69A	1691 69B	1692 69C	1693 69D	1694 69E
1695 69F	1696 6A0	1697 6A1	1698 6A2	1699 6A3	1700 6A4	1701 6A5
1702 6A6	1703 6A7	1704 6A8	1705 6A9	1706 6AA	1707 6AB	1708 6AC
1709 6AD	1710 6AE	1711 6AF	1712 6B0	1713 6B1	1714 6B2	1715 6B3
1716 6B4	1717 6B5	1718 6B6	1719 6B7	1720 6B8	1721 6B9	1722 6BA
1723 6BB	1724 6BC	1725 6BD	1726 6BE	1727 6BF	1728 6C0	1729 6C1
1730 6C2	1731 6C3	1732 6C4	1733 6C5	1734 6C6	1735 6C7	1736 6C8
1737 6C9	1738 6CA	1739 6CB	1740 6CC	1741 6CD	1742 6CE	1743 6CF
1744 6D0	1745 6D1	1746 6D2	1747 6D3	1748 6D4	1749 6D5	1750 6D6
1751 6D7	1752 6D8	1753 6D9	1754 6DA	1755 6DB	1756 6DC	1757 6DD
1758 6DE	1759 6DF	1760 6E0	1761 6E1	1762 6E2	1763 6E3	1764 6E4

1765 6E5	1766 6E6	1767 6E7	1768 6E8	1769 6E9	1770 6EA	1771 6EB
1772 6EC	1773 6ED	1774 6EE	1775 6EF	1776 6F0	1777 6F1	1778 6F2
1779 6F3	1780 6F4	1781 6F5	1782 6F6	1783 6F7	1784 6F8	1785 6F9
1786 6FA	1787 6FB	1788 6FC	1789 6FD	1790 6FE	1791 6FF	1792 700
1793 701	1794 702	1795 703	1796 704	1797 705	1798 706	1799 707
1800 708	1801 709	1802 70A	1803 70B	1804 70C	1805 70D	1806 70E
1807 70F	1808 710	1809 711	1810 712	1811 713	1812 714	1813 715
1814 716	1815 717	1816 718	1817 719	1818 71A	1819 71B	1820 71C
1821 71D	1822 71E	1823 71F	1824 720	1825 721	1826 722	1827 723
1828 724	1829 725	1830 726	1831 727	1832 728	1833 729	1834 72A
1835 72B	1836 72C	1837 72D	1838 72E	1839 72F	1840 730	1841 731
1842 732	1843 733	1844 734	1845 735	1846 736	1847 737	1848 738
1849 739	1850 73A	1851 73B	1852 73C	1853 73D	1854 73E	1855 73F
1856 740	1857 741	1858 742	1859 743	1860 744	1861 745	1862 746
1863 747	1864 748	1865 749	1866 74A	1867 74B	1868 74C	1869 74D
1870 74E	1871 74F	1872 750	1873 751	1874 752	1875 753	1876 754
1877 755	1878 756	1879 757	1880 758	1881 759	1882 75A	1883 75B
1884 75C	1885 75D	1886 75E	1887 75F	1888 760	1889 761	1890 762
1891 763	1892 764	1893 765	1894 766	1895 767	1896 768	1897 769
1898 76A	1899 76B	1900 76C	1901 76D	1902 76E	1903 76F	1904 770
1905 771	1906 772	1907 773	1908 774	1909 775	1910 776	1911 777
1912 778	1913 779	1914 77A	1915 77B	1916 77C	1917 77D	1918 77E
1919 77F	1920 780	1921 781	1922 782	1923 783	1924 784	1925 785
1926 786	1927 787	1928 788	1929 789	1930 78A	1931 78B	1932 78C
1933 78D	1934 78E	1935 78F	1936 790	1937 791	1938 792	1939 793
1940 794	1941 795	1942 796	1943 797	1944 798	1945 799	1946 79A
1947 79B	1948 79C	1949 79D	1950 79E	1951 79F	1952 7A0	1953 7A1
1954 7A2	1955 7A3	1956 7A4	1957 7A5	1958 7A6	1959 7A7	1960 7A8
1961 7A9	1962 7AA	1963 7AB	1964 7AC	1965 7AD	1966 7AE	1967 7AF
1968 7B0	1969 7B1	1970 7B2	1971 7B3	1972 7B4	1973 7B5	1974 7B6
1975 7B7	1976 7B8	1977 7B9	1978 7BA	1979 7BB	1980 7BC	1981 7BD
1982 7BE	1983 7BF	1984 7C0	1985 7C1	1986 7C2	1987 7C3	1988 7C4
1989 7C5	1990 7C6	1991 7C7	1992 7C8	1993 7C9	1994 7CA	1995 7CB
1996 7CC	1997 7CD	1998 7CE	1999 7CF	2000 7D0	2001 7D1	2002 7D2
2003 7D3	2004 7D4	2005 7D5	2006 7D6	2007 7D7	2008 7D8	2009 7D9
2010 7DA	2011 7DB	2012 7DC	2013 7DD	2014 7DE	2015 7DF	2016 7E0
2017 7E1	2018 7E2	2019 7E3	2020 7E4	2021 7E5	2022 7E6	2023 7E7
2024 7E8	2025 7E9	2026 7EA	2027 7EB	2028 7EC	2029 7ED	2030 7EE
2031 7EF	2032 7F0	2033 7F1	2034 7F2	2035 7F3	2036 7F4	2037 7F5
2038 7F6	2039 7F7	2040 7F8	2041 7F9	2042 7FA	2043 7FB	2044 7FC
2045 7FD	2046 7FE	2047 7FF	-2048 800	-2047 801	-2046 802	-2045 803
-2044 804	-2043 805	-2042 806	-2041 807	-2040 808	-2039 809	-2038 80A
-2037 80B	-2036 80C	-2035 80D	-2034 80E	-2033 80F	-2032 810	-2031 811
-2030 812	-2029 813	-2028 814	-2027 815	-2026 816	-2025 817	-2024 818
-2023 819	-2022 81A	-2021 81B	-2020 81C	-2019 81D	-2018 81E	-2017 81F
-2016 820	-2015 821	-2014 822	-2013 823	-2012 824	-2011 825	-2010 826
-2009 827	-2008 828	-2007 829	-2006 82A	-2005 82B	-2004 82C	-2003 82D
-2002 82E	-2001 82F	-2000 830	-1999 831	-1998 832	-1997 833	-1996 834
-1995 835	-1994 836	-1993 837	-1992 838	-1991 839	-1990 83A	-1989 83B
-1988 83C	-1987 83D	-1986 83E	-1985 83F	-1984 840	-1983 841	-1982 842
-1981 843	-1980 844	-1979 845	-1978 846	-1977 847	-1976 848	-1975 849
-1974 84A	-1973 84B	-1972 84C	-1971 84D	-1970 84E	-1969 84F	-1968 850
-1967 851	-1966 852	-1965 853	-1964 854	-1963 855	-1962 856	-1961 857
-1960 858	-1959 859	-1958 85A	-1957 85B	-1956 85C	-1955 85D	-1954 85E
-1953 85F	-1952 860	-1951 861	-1950 862	-1949 863	-1948 864	-1947 865
-1946 866	-1945 867	-1944 868	-1943 869	-1942 86A	-1941 86B	-1940 86C
-1939 86D	-1938 86E	-1937 86F	-1936 870	-1935 871	-1934 872	-1933 873
-1932 874	-1931 875	-1930 876	-1929 877	-1928 878	-1927 879	-1926 87A
-1925 87B	-1924 87C	-1923 87D	-1922 87E	-1921 87F	-1920 880	-1919 881
-1918 882	-1917 883	-1916 884	-1915 885	-1914 886	-1913 887	-1912 888
-1911 889	-1910 88A	-1909 88B	-1908 88C	-1907 88D	-1906 88E	-1905 88F
-1904 890	-1903 891	-1902 892	-1901 893	-1900 894	-1899 895	-1898 896
-1897 897	-1896 898	-1895 899	-1894 89A	-1893 89B	-1892 89C	-1891 89D

-1890 89E	-1889 89F	-1888 8A0	-1887 8A1	-1886 8A2	-1885 8A3	-1884 8A4
-1883 8A5	-1882 8A6	-1881 8A7	-1880 8A8	-1879 8A9	-1878 8AA	-1877 8AB
-1876 8AC	-1875 8AD	-1874 8AE	-1873 8AF	-1872 8B0	-1871 8B1	-1870 8B2
-1869 8B3	-1868 8B4	-1867 8B5	-1866 8B6	-1865 8B7	-1864 8B8	-1863 8B9
-1862 8BA	-1861 8BB	-1860 8BC	-1859 8BD	-1858 8BE	-1857 8BF	-1856 8C0
-1855 8C1	-1854 8C2	-1853 8C3	-1852 8C4	-1851 8C5	-1850 8C6	-1849 8C7
-1848 8C8	-1847 8C9	-1846 8CA	-1845 8CB	-1844 8CC	-1843 8CD	-1842 8CE
-1841 8CF	-1840 8D0	-1839 8D1	-1838 8D2	-1837 8D3	-1836 8D4	-1835 8D5
-1834 8D6	-1833 8D7	-1832 8D8	-1831 8D9	-1830 8DA	-1829 8DB	-1828 8DC
-1827 8DD	-1826 8DE	-1825 8DF	-1824 8E0	-1823 8E1	-1822 8E2	-1821 8E3
-1820 8E4	-1819 8E5	-1818 8E6	-1817 8E7	-1816 8E8	-1815 8E9	-1814 8EA
-1813 8EB	-1812 8EC	-1811 8ED	-1810 8EE	-1809 8EF	-1808 8F0	-1807 8F1
-1806 8F2	-1805 8F3	-1804 8F4	-1803 8F5	-1802 8F6	-1801 8F7	-1800 8F8
-1799 8F9	-1798 8FA	-1797 8FB	-1796 8FC	-1795 8FD	-1794 8FE	-1793 8FF
-1792 900	-1791 901	-1790 902	-1789 903	-1788 904	-1787 905	-1786 906
-1785 907	-1784 908	-1783 909	-1782 90A	-1781 90B	-1780 90C	-1779 90D
-1778 90E	-1777 90F	-1776 910	-1775 911	-1774 912	-1773 913	-1772 914
-1771 915	-1770 916	-1769 917	-1768 918	-1767 919	-1766 91A	-1765 91B
-1764 91C	-1763 91D	-1762 91E	-1761 91F	-1760 920	-1759 921	-1758 922
-1757 923	-1756 924	-1755 925	-1754 926	-1753 927	-1752 928	-1751 929
-1750 92A	-1749 92B	-1748 92C	-1747 92D	-1746 92E	-1745 92F	-1744 930
-1743 931	-1742 932	-1741 933	-1740 934	-1739 935	-1738 936	-1737 937
-1736 938	-1735 939	-1734 93A	-1733 93B	-1732 93C	-1731 93D	-1730 93E
-1729 93F	-1728 940	-1727 941	-1726 942	-1725 943	-1724 944	-1723 945
-1722 946	-1721 947	-1720 948	-1719 949	-1718 94A	-1717 94B	-1716 94C
-1715 94D	-1714 94E	-1713 94F	-1712 950	-1711 951	-1710 952	-1709 953
-1708 954	-1707 955	-1706 956	-1705 957	-1704 958	-1703 959	-1702 95A
-1701 95B	-1700 95C	-1699 95D	-1698 95E	-1697 95F	-1696 960	-1695 961
-1694 962	-1693 963	-1692 964	-1691 965	-1690 966	-1689 967	-1688 968
-1687 969	-1686 96A	-1685 96B	-1684 96C	-1683 96D	-1682 96E	-1681 96F
-1680 970	-1679 971	-1678 972	-1677 973	-1676 974	-1675 975	-1674 976
-1673 977	-1672 978	-1671 979	-1670 97A	-1669 97B	-1668 97C	-1667 97D
-1666 97E	-1665 97F	-1664 980	-1663 981	-1662 982	-1661 983	-1660 984
-1659 985	-1658 986	-1657 987	-1656 988	-1655 989	-1654 98A	-1653 98B
-1652 98C	-1651 98D	-1650 98E	-1649 98F	-1648 990	-1647 991	-1646 992
-1645 993	-1644 994	-1643 995	-1642 996	-1641 997	-1640 998	-1639 999
-1638 99A	-1637 99B	-1636 99C	-1635 99D	-1634 99E	-1633 99F	-1632 9A0
-1631 9A1	-1630 9A2	-1629 9A3	-1628 9A4	-1627 9A5	-1626 9A6	-1625 9A7
-1624 9A8	-1623 9A9	-1622 9AA	-1621 9AB	-1620 9AC	-1619 9AD	-1618 9AE
-1617 9AF	-1616 9B0	-1615 9B1	-1614 9B2	-1613 9B3	-1612 9B4	-1611 9B5
-1610 9B6	-1609 9B7	-1608 9B8	-1607 9B9	-1606 9BA	-1605 9BB	-1604 9BC
-1603 9BD	-1602 9BE	-1601 9BF	-1600 9C0	-1599 9C1	-1598 9C2	-1597 9C3
-1596 9C4	-1595 9C5	-1594 9C6	-1593 9C7	-1592 9C8	-1591 9C9	-1590 9CA
-1589 9CB	-1588 9CC	-1587 9CD	-1586 9CE	-1585 9CF	-1584 9D0	-1583 9D1
-1582 9D2	-1581 9D3	-1580 9D4	-1579 9D5	-1578 9D6	-1577 9D7	-1576 9D8
-1575 9D9	-1574 9DA	-1573 9DB	-1572 9DC	-1571 9DD	-1570 9DE	-1569 9DF
-1568 9E0	-1567 9E1	-1566 9E2	-1565 9E3	-1564 9E4	-1563 9E5	-1562 9E6
-1561 9E7	-1560 9E8	-1559 9E9	-1558 9EA	-1557 9EB	-1556 9EC	-1555 9ED
-1554 9EE	-1553 9EF	-1552 9F0	-1551 9F1	-1550 9F2	-1549 9F3	-1548 9F4
-1547 9F5	-1546 9F6	-1545 9F7	-1544 9F8	-1543 9F9	-1542 9FA	-1541 9FB
-1540 9FC	-1539 9FD	-1538 9FE	-1537 9FF	-1536 A00	-1535 A01	-1534 A02
-1533 A03	-1532 A04	-1531 A05	-1530 A06	-1529 A07	-1528 A08	-1527 A09
-1526 A0A	-1525 A0B	-1524 A0C	-1523 A0D	-1522 A0E	-1521 A0F	-1520 A10
-1519 A11	-1518 A12	-1517 A13	-1516 A14	-1515 A15	-1514 A16	-1513 A17
-1512 A18	-1511 A19	-1510 A1A	-1509 A1B	-1508 A1C	-1507 A1D	-1506 A1E
-1505 A1F	-1504 A20	-1503 A21	-1502 A22	-1501 A23	-1500 A24	-1499 A25
-1498 A26	-1497 A27	-1496 A28	-1495 A29	-1494 A2A	-1493 A2B	-1492 A2C
-1491 A2D	-1490 A2E	-1489 A2F	-1488 A30	-1487 A31	-1486 A32	-1485 A33
-1484 A34	-1483 A35	-1482 A36	-1481 A37	-1480 A38	-1479 A39	-1478 A3A
-1477 A3B	-1476 A3C	-1475 A3D	-1474 A3E	-1473 A3F	-1472 A40	-1471 A41
-1470 A42	-1469 A43	-1468 A44	-1467 A45	-1466 A46	-1465 A47	-1464 A48
-1463 A49	-1462 A4A	-1461 A4B	-1460 A4C	-1459 A4D	-1458 A4E	-1457 A4F
-1456 A50	-1455 A51	-1454 A52	-1453 A53	-1452 A54	-1451 A55	-1450 A56

-1449 A57	-1448 A58	-1447 A59	-1446 A5A	-1445 A5B	-1444 A5C	-1443 A5D
-1442 A5E	-1441 A5F	-1440 A60	-1439 A61	-1438 A62	-1437 A63	-1436 A64
-1435 A65	-1434 A66	-1433 A67	-1432 A68	-1431 A69	-1430 A6A	-1429 A6B
-1428 A6C	-1427 A6D	-1426 A6E	-1425 A6F	-1424 A70	-1423 A71	-1422 A72
-1421 A73	-1420 A74	-1419 A75	-1418 A76	-1417 A77	-1416 A78	-1415 A79
-1414 A7A	-1413 A7B	-1412 A7C	-1411 A7D	-1410 A7E	-1409 A7F	-1408 A80
-1407 A81	-1406 A82	-1405 A83	-1404 A84	-1403 A85	-1402 A86	-1401 A87
-1400 A88	-1399 A89	-1398 A8A	-1397 A8B	-1396 A8C	-1395 A8D	-1394 A8E
-1393 A8F	-1392 A90	-1391 A91	-1390 A92	-1389 A93	-1388 A94	-1387 A95
-1386 A96	-1385 A97	-1384 A98	-1383 A99	-1382 A9A	-1381 A9B	-1380 A9C
-1379 A9D	-1378 A9E	-1377 A9F	-1376 AA0	-1375 AA1	-1374 AA2	-1373 AA3
-1372 AA4	-1371 AA5	-1370 AA6	-1369 AA7	-1368 AA8	-1367 AA9	-1366 AAA
-1365 AAB	-1364 AAC	-1363 AAD	-1362 AAE	-1361 AAF	-1360 AB0	-1359 AB1
-1358 AB2	-1357 AB3	-1356 AB4	-1355 AB5	-1354 AB6	-1353 AB7	-1352 AB8
-1351 AB9	-1350 ABA	-1349 ABB	-1348 ABC	-1347 ABD	-1346 ABE	-1345 ABF
-1344 AC0	-1343 AC1	-1342 AC2	-1341 AC3	-1340 AC4	-1339 AC5	-1338 AC6
-1337 AC7	-1336 AC8	-1335 AC9	-1334 ACA	-1333 ACB	-1332 ACC	-1331 ACD
-1330 ACE	-1329 ACF	-1328 AD0	-1327 AD1	-1326 AD2	-1325 AD3	-1324 AD4
-1323 AD5	-1322 AD6	-1321 AD7	-1320 AD8	-1319 AD9	-1318 ADA	-1317 ADB
-1316 ADC	-1315 ADD	-1314 ADE	-1313 ADF	-1312 AE0	-1311 AE1	-1310 AE2
-1309 AE3	-1308 AE4	-1307 AE5	-1306 AE6	-1305 AE7	-1304 AE8	-1303 AE9
-1302 AEA	-1301 AEB	-1300 AEC	-1299 AED	-1298 AEE	-1297 AEF	-1296 AF0
-1295 AF1	-1294 AF2	-1293 AF3	-1292 AF4	-1291 AF5	-1290 AF6	-1289 AF7
-1288 AF8	-1287 AF9	-1286 AFA	-1285 AFB	-1284 AFC	-1283 AFD	-1282 AFE
-1281 AFF	-1280 B00	-1279 B01	-1278 B02	-1277 B03	-1276 B04	-1275 B05
-1274 B06	-1273 B07	-1272 B08	-1271 B09	-1270 B0A	-1269 B0B	-1268 B0C
-1267 B0D	-1266 B0E	-1265 B0F	-1264 B10	-1263 B11	-1262 B12	-1261 B13
-1260 B14	-1259 B15	-1258 B16	-1257 B17	-1256 B18	-1255 B19	-1254 B1A
-1253 B1B	-1252 B1C	-1251 B1D	-1250 B1E	-1249 B1F	-1248 B20	-1247 B21
-1246 B22	-1245 B23	-1244 B24	-1243 B25	-1242 B26	-1241 B27	-1240 B28
-1239 B29	-1238 B2A	-1237 B2B	-1236 B2C	-1235 B2D	-1234 B2E	-1233 B2F
-1232 B30	-1231 B31	-1230 B32	-1229 B33	-1228 B34	-1227 B35	-1226 B36
-1225 B37	-1224 B38	-1223 B39	-1222 B3A	-1221 B3B	-1220 B3C	-1219 B3D
-1218 B3E	-1217 B3F	-1216 B40	-1215 B41	-1214 B42	-1213 B43	-1212 B44
-1211 B45	-1210 B46	-1209 B47	-1208 B48	-1207 B49	-1206 B4A	-1205 B4B
-1204 B4C	-1203 B4D	-1202 B4E	-1201 B4F	-1200 B50	-1199 B51	-1198 B52
-1197 B53	-1196 B54	-1195 B55	-1194 B56	-1193 B57	-1192 B58	-1191 B59
-1190 B5A	-1189 B5B	-1188 B5C	-1187 B5D	-1186 B5E	-1185 B5F	-1184 B60
-1183 B61	-1182 B62	-1181 B63	-1180 B64	-1179 B65	-1178 B66	-1177 B67
-1176 B68	-1175 B69	-1174 B6A	-1173 B6B	-1172 B6C	-1171 B6D	-1170 B6E
-1169 B6F	-1168 B70	-1167 B71	-1166 B72	-1165 B73	-1164 B74	-1163 B75
-1162 B76	-1161 B77	-1160 B78	-1159 B79	-1158 B7A	-1157 B7B	-1156 B7C
-1155 B7D	-1154 B7E	-1153 B7F	-1152 B80	-1151 B81	-1150 B82	-1149 B83
-1148 B84	-1147 B85	-1146 B86	-1145 B87	-1144 B88	-1143 B89	-1142 B8A
-1141 B8B	-1140 B8C	-1139 B8D	-1138 B8E	-1137 B8F	-1136 B90	-1135 B91
-1134 B92	-1133 B93	-1132 B94	-1131 B95	-1130 B96	-1129 B97	-1128 B98
-1127 B99	-1126 B9A	-1125 B9B	-1124 B9C	-1123 B9D	-1122 B9E	-1121 B9F
-1120 BA0	-1119 BA1	-1118 BA2	-1117 BA3	-1116 BA4	-1115 BA5	-1114 BA6
-1113 BA7	-1112 BA8	-1111 BA9	-1110 BAA	-1109 BAB	-1108 BAC	-1107 BAD
-1106 BAE	-1105 BAF	-1104 BB0	-1103 BB1	-1102 BB2	-1101 BB3	-1100 BB4
-1099 BB5	-1098 BB6	-1097 BB7	-1096 BB8	-1095 BB9	-1094 BBA	-1093 BBB
-1092 BBC	-1091 BBD	-1090 BBE	-1089 BBF	-1088 BC0	-1087 BC1	-1086 BC2
-1085 BC3	-1084 BC4	-1083 BC5	-1082 BC6	-1081 BC7	-1080 BC8	-1079 BC9
-1078 BCA	-1077 BCB	-1076 BCC	-1075 BCD	-1074 BCE	-1073 BCF	-1072 BD0
-1071 BD1	-1070 BD2	-1069 BD3	-1068 BD4	-1067 BD5	-1066 BD6	-1065 BD7
-1064 BD8	-1063 BD9	-1062 BDA	-1061 BDB	-1060 BDC	-1059 BDD	-1058 BDE
-1057 BDF	-1056 BE0	-1055 BE1	-1054 BE2	-1053 BE3	-1052 BE4	-1051 BE5
-1050 BE6	-1049 BE7	-1048 BE8	-1047 BE9	-1046 BEA	-1045 BEB	-1044 BEC
-1043 BED	-1042 BEE	-1041 BEF	-1040 BF0	-1039 BF1	-1038 BF2	-1037 BF3
-1036 BF4	-1035 BF5	-1034 BF6	-1033 BF7	-1032 BF8	-1031 BF9	-1030 BFA
-1029 BFB	-1028 BFC	-1027 BFD	-1026 BFE	-1025 BFF	-1024 C00	-1023 C01
-1022 C02	-1021 C03	-1020 C04	-1019 C05	-1018 C06	-1017 C07	-1016 C08
-1015 C09	-1014 C0A	-1013 C0B	-1012 C0C	-1011 C0D	-1010 C0E	-1009 C0F

-1008 C10	-1007 C11	-1006 C12	-1005 C13	-1004 C14	-1003 C15	-1002 C16
-1001 C17	-1000 C18	-999 C19	-998 C1A	-997 C1B	-996 C1C	-995 C1D
-994 C1E	-993 C1F	-992 C20	-991 C21	-990 C22	-989 C23	-988 C24
-987 C25	-986 C26	-985 C27	-984 C28	-983 C29	-982 C2A	-981 C2B
-980 C2C	-979 C2D	-978 C2E	-977 C2F	-976 C30	-975 C31	-974 C32
-973 C33	-972 C34	-971 C35	-970 C36	-969 C37	-968 C38	-967 C39
-966 C3A	-965 C3B	-964 C3C	-963 C3D	-962 C3E	-961 C3F	-960 C40
-959 C41	-958 C42	-957 C43	-956 C44	-955 C45	-954 C46	-953 C47
-952 C48	-951 C49	-950 C4A	-949 C4B	-948 C4C	-947 C4D	-946 C4E
-945 C4F	-944 C50	-943 C51	-942 C52	-941 C53	-940 C54	-939 C55
-938 C56	-937 C57	-936 C58	-935 C59	-934 C5A	-933 C5B	-932 C5C
-931 C5D	-930 C5E	-929 C5F	-928 C60	-927 C61	-926 C62	-925 C63
-924 C64	-923 C65	-922 C66	-921 C67	-920 C68	-919 C69	-918 C6A
-917 C6B	-916 C6C	-915 C6D	-914 C6E	-913 C6F	-912 C70	-911 C71
-910 C72	-909 C73	-908 C74	-907 C75	-906 C76	-905 C77	-904 C78
-903 C79	-902 C7A	-901 C7B	-900 C7C	-899 C7D	-898 C7E	-897 C7F
-896 C80	-895 C81	-894 C82	-893 C83	-892 C84	-891 C85	-890 C86
-889 C87	-888 C88	-887 C89	-886 C8A	-885 C8B	-884 C8C	-883 C8D
-882 C8E	-881 C8F	-880 C90	-879 C91	-878 C92	-877 C93	-876 C94
-875 C95	-874 C96	-873 C97	-872 C98	-871 C99	-870 C9A	-869 C9B
-868 C9C	-867 C9D	-866 C9E	-865 C9F	-864 CA0	-863 CA1	-862 CA2
-861 CA3	-860 CA4	-859 CA5	-858 CA6	-857 CA7	-856 CA8	-855 CA9
-854 CAA	-853 CAB	-852 CAC	-851 CAD	-850 CAE	-849 CAF	-848 CB0
-847 CB1	-846 CB2	-845 CB3	-844 CB4	-843 CB5	-842 CB6	-841 CB7
-840 CB8	-839 CB9	-838 CBA	-837 CBB	-836 CBC	-835 CBD	-834 CBE
-833 CBF	-832 CC0	-831 CC1	-830 CC2	-829 CC3	-828 CC4	-827 CC5
-826 CC6	-825 CC7	-824 CC8	-823 CC9	-822 CCA	-821 CCB	-820 CCC
-819 CCD	-818 CCE	-817 CCF	-816 CD0	-815 CD1	-814 CD2	-813 CD3
-812 CD4	-811 CD5	-810 CD6	-809 CD7	-808 CD8	-807 CD9	-806 CDA
-805 CDB	-804 CDC	-803 CDD	-802 CDE	-801 CDF	-800 CE0	-799 CE1
-798 CE2	-797 CE3	-796 CE4	-795 CE5	-794 CE6	-793 CE7	-792 CE8
-791 CE9	-790 CEA	-789 CEB	-788 CEC	-787 CED	-786 CEE	-785 CEF
-784 CF0	-783 CF1	-782 CF2	-781 CF3	-780 CF4	-779 CF5	-778 CF6
-777 CF7	-776 CF8	-775 CF9	-774 CFA	-773 CFB	-772 CFC	-771 CFD
-770 CFE	-769 CFF	-768 D00	-767 D01	-766 D02	-765 D03	-764 D04
-763 D05	-762 D06	-761 D07	-760 D08	-759 D09	-758 D0A	-757 D0B
-756 D0C	-755 D0D	-754 D0E	-753 D0F	-752 D10	-751 D11	-750 D12
-749 D13	-748 D14	-747 D15	-746 D16	-745 D17	-744 D18	-743 D19
-742 D1A	-741 D1B	-740 D1C	-739 D1D	-738 D1E	-737 D1F	-736 D20
-735 D21	-734 D22	-733 D23	-732 D24	-731 D25	-730 D26	-729 D27
-728 D28	-727 D29	-726 D2A	-725 D2B	-724 D2C	-723 D2D	-722 D2E
-721 D2F	-720 D30	-719 D31	-718 D32	-717 D33	-716 D34	-715 D35
-714 D36	-713 D37	-712 D38	-711 D39	-710 D3A	-709 D3B	-708 D3C
-707 D3D	-706 D3E	-705 D3F	-704 D40	-703 D41	-702 D42	-701 D43
-700 D44	-699 D45	-698 D46	-697 D47	-696 D48	-695 D49	-694 D4A
-693 D4B	-692 D4C	-691 D4D	-690 D4E	-689 D4F	-688 D50	-687 D51
-686 D52	-685 D53	-684 D54	-683 D55	-682 D56	-681 D57	-680 D58
-679 D59	-678 D5A	-677 D5B	-676 D5C	-675 D5D	-674 D5E	-673 D5F
-672 D60	-671 D61	-670 D62	-669 D63	-668 D64	-667 D65	-666 D66
-665 D67	-664 D68	-663 D69	-662 D6A	-661 D6B	-660 D6C	-659 D6D
-658 D6E	-657 D6F	-656 D70	-655 D71	-654 D72	-653 D73	-652 D74
-651 D75	-650 D76	-649 D77	-648 D78	-647 D79	-646 D7A	-645 D7B
-644 D7C	-643 D7D	-642 D7E	-641 D7F	-640 D80	-639 D81	-638 D82
-637 D83	-636 D84	-635 D85	-634 D86	-633 D87	-632 D88	-631 D89
-630 D8A	-629 D8B	-628 D8C	-627 D8D	-626 D8E	-625 D8F	-624 D90
-623 D91	-622 D92	-621 D93	-620 D94	-619 D95	-618 D96	-617 D97
-616 D98	-615 D99	-614 D9A	-613 D9B	-612 D9C	-611 D9D	-610 D9E
-609 D9F	-608 DA0	-607 DA1	-606 DA2	-605 DA3	-604 DA4	-603 DA5
-602 DA6	-601 DA7	-600 DA8	-599 DA9	-598 DAA	-597 DAB	-596 DAC
-595 DAD	-594 DAE	-593 DAF	-592 DB0	-591 DB1	-590 DB2	-589 DB3
-588 DB4	-587 DB5	-586 DB6	-585 DB7	-584 DB8	-583 DB9	-582 DBA
-581 DBB	-580 DBC	-579 DBD	-578 DBE	-577 DBF	-576 DC0	-575 DC1
-574 DC2	-573 DC3	-572 DC4	-571 DC5	-570 DC6	-569 DC7	-568 DC8

-567 DC9	-566 DCA	-565 DCB	-564 DCC	-563 DCD	-562 DCE	-561 DCF
-560 DD0	-559 DD1	-558 DD2	-557 DD3	-556 DD4	-555 DD5	-554 DD6
-553 DD7	-552 DD8	-551 DD9	-550 DDA	-549 DDB	-548 DDC	-547 DDD
-546 DDE	-545 DDF	-544 DE0	-543 DE1	-542 DE2	-541 DE3	-540 DE4
-539 DE5	-538 DE6	-537 DE7	-536 DE8	-535 DE9	-534 DEA	-533 DEB
-532 DEC	-531 DED	-530 DEE	-529 DEF	-528 DF0	-527 DF1	-526 DF2
-525 DF3	-524 DF4	-523 DF5	-522 DF6	-521 DF7	-520 DF8	-519 DF9
-518 DFA	-517 DFB	-516 DFC	-515 DFD	-514 DFE	-513 DFF	-512 E00
-511 E01	-510 E02	-509 E03	-508 E04	-507 E05	-506 E06	-505 E07
-504 E08	-503 E09	-502 E0A	-501 E0B	-500 E0C	-499 E0D	-498 E0E
-497 E0F	-496 E10	-495 E11	-494 E12	-493 E13	-492 E14	-491 E15
-490 E16	-489 E17	-488 E18	-487 E19	-486 E1A	-485 E1B	-484 E1C
-483 E1D	-482 E1E	-481 E1F	-480 E20	-479 E21	-478 E22	-477 E23
-476 E24	-475 E25	-474 E26	-473 E27	-472 E28	-471 E29	-470 E2A
-469 E2B	-468 E2C	-467 E2D	-466 E2E	-465 E2F	-464 E30	-463 E31
-462 E32	-461 E33	-460 E34	-459 E35	-458 E36	-457 E37	-456 E38
-455 E39	-454 E3A	-453 E3B	-452 E3C	-451 E3D	-450 E3E	-449 E3F
-448 E40	-447 E41	-446 E42	-445 E43	-444 E44	-443 E45	-442 E46
-441 E47	-440 E48	-439 E49	-438 E4A	-437 E4B	-436 E4C	-435 E4D
-434 E4E	-433 E4F	-432 E50	-431 E51	-430 E52	-429 E53	-428 E54
-427 E55	-426 E56	-425 E57	-424 E58	-423 E59	-422 E5A	-421 E5B
-420 E5C	-419 E5D	-418 E5E	-417 E5F	-416 E60	-415 E61	-414 E62
-413 E63	-412 E64	-411 E65	-410 E66	-409 E67	-408 E68	-407 E69
-406 E6A	-405 E6B	-404 E6C	-403 E6D	-402 E6E	-401 E6F	-400 E70
-399 E71	-398 E72	-397 E73	-396 E74	-395 E75	-394 E76	-393 E77
-392 E78	-391 E79	-390 E7A	-389 E7B	-388 E7C	-387 E7D	-386 E7E
-385 E7F	-384 E80	-383 E81	-382 E82	-381 E83	-380 E84	-379 E85
-378 E86	-377 E87	-376 E88	-375 E89	-374 E8A	-373 E8B	-372 E8C
-371 E8D	-370 E8E	-369 E8F	-368 E90	-367 E91	-366 E92	-365 E93
-364 E94	-363 E95	-362 E96	-361 E97	-360 E98	-359 E99	-358 E9A
-357 E9B	-356 E9C	-355 E9D	-354 E9E	-353 E9F	-352 EA0	-351 EA1
-350 EA2	-349 EA3	-348 EA4	-347 EA5	-346 EA6	-345 EA7	-344 EA8
-343 EA9	-342 EAA	-341 EAB	-340 EAC	-339 EAD	-338 EAE	-337 EAF
-336 EB0	-335 EB1	-334 EB2	-333 EB3	-332 EB4	-331 EB5	-330 EB6
-329 EB7	-328 EB8	-327 EB9	-326 EBA	-325 EBB	-324 EBC	-323 EBD
-322 EBE	-321 EBF	-320 EC0	-319 EC1	-318 EC2	-317 EC3	-316 EC4
-315 EC5	-314 EC6	-313 EC7	-312 EC8	-311 EC9	-310 ECA	-309 ECB
-308 ECC	-307 ECD	-306 ECE	-305 ECF	-304 ED0	-303 ED1	-302 ED2
-301 ED3	-300 ED4	-299 ED5	-298 ED6	-297 ED7	-296 ED8	-295 ED9
-294 EDA	-293 EDB	-292 EDC	-291 EDD	-290 EDE	-289 EDF	-288 EE0
-287 EE1	-286 EE2	-285 EE3	-284 EE4	-283 EE5	-282 EE6	-281 EE7
-280 EE8	-279 EE9	-278 EEA	-277 EEB	-276 EEC	-275 EED	-274 EEE
-273 EEF	-272 EF0	-271 EF1	-270 EF2	-269 EF3	-268 EF4	-267 EF5
-266 EF6	-265 EF7	-264 EF8	-263 EF9	-262 EFA	-261 EFB	-260 EFC
-259 EFD	-258 EFE	-257 EFF	-256 F00	-255 F01	-254 F02	-253 F03
-252 F04	-251 F05	-250 F06	-249 F07	-248 F08	-247 F09	-246 F0A
-245 F0B	-244 F0C	-243 F0D	-242 F0E	-241 F0F	-240 F10	-239 F11
-238 F12	-237 F13	-236 F14	-235 F15	-234 F16	-233 F17	-232 F18
-231 F19	-230 F1A	-229 F1B	-228 F1C	-227 F1D	-226 F1E	-225 F1F
-224 F20	-223 F21	-222 F22	-221 F23	-220 F24	-219 F25	-218 F26
-217 F27	-216 F28	-215 F29	-214 F2A	-213 F2B	-212 F2C	-211 F2D
-210 F2E	-209 F2F	-208 F30	-207 F31	-206 F32	-205 F33	-204 F34
-203 F35	-202 F36	-201 F37	-200 F38	-199 F39	-198 F3A	-197 F3B
-196 F3C	-195 F3D	-194 F3E	-193 F3F	-192 F40	-191 F41	-190 F42
-189 F43	-188 F44	-187 F45	-186 F46	-185 F47	-184 F48	-183 F49
-182 F4A	-181 F4B	-180 F4C	-179 F4D	-178 F4E	-177 F4F	-176 F50
-175 F51	-174 F52	-173 F53	-172 F54	-171 F55	-170 F56	-169 F57
-168 F58	-167 F59	-166 F5A	-165 F5B	-164 F5C	-163 F5D	-162 F5E
-161 F5F	-160 F60	-159 F61	-158 F62	-157 F63	-156 F64	-155 F65
-154 F66	-153 F67	-152 F68	-151 F69	-150 F6A	-149 F6B	-148 F6C
-147 F6D	-146 F6E	-145 F6F	-144 F70	-143 F71	-142 F72	-141 F73
-140 F74	-139 F75	-138 F76	-137 F77	-136 F78	-135 F79	-134 F7A
-133 F7B	-132 F7C	-131 F7D	-130 F7E	-129 F7F	-128 F80	-127 F81

-126 F82	-125 F83	-124 F84	-123 F85	-122 F86	-121 F87	-120 F88
-119 F89	-118 F8A	-117 F8B	-116 F8C	-115 F8D	-114 F8E	-113 F8F
-112 F90	-111 F91	-110 F92	-109 F93	-108 F94	-107 F95	-106 F96
-105 F97	-104 F98	-103 F99	-102 F9A	-101 F9B	-100 F9C	-99 F9D
-98 F9E	-97 F9F	-96 FA0	-95 FA1	-94 FA2	-93 FA3	-92 FA4
-91 FA5	-90 FA6	-89 FA7	-88 FA8	-87 FA9	-86 FAA	-85 FAB
-84 FAC	-83 FAD	-82 FAE	-81 FAF	-80 FB0	-79 FB1	-78 FB2
-77 FB3	-76 FB4	-75 FB5	-74 FB6	-73 FB7	-72 FB8	-71 FB9
-70 FBA	-69 FBB	-68 FBC	-67 FBD	-66 FBE	-65 FBF	-64 FC0
-63 FC1	-62 FC2	-61 FC3	-60 FC4	-59 FC5	-58 FC6	-57 FC7
-56 FC8	-55 FC9	-54 FCA	-53 FCB	-52 FCC	-51 FCD	-50 FCE
-49 FCF	-48 FD0	-47 FD1	-46 FD2	-45 FD3	-44 FD4	-43 FD5
-42 FD6	-41 FD7	-40 FD8	-39 FD9	-38 FDA	-37 FDB	-36 FDC
-35 FDD	-34 FDE	-33 FDF	-32 FE0	-31 FE1	-30 FE2	-29 FE3
-28 FE4	-27 FE5	-26 FE6	-25 FE7	-24 FE8	-23 FE9	-22 FEA
-21 FEB	-20 FEC	-19 FED	-18 FEE	-17 FEF	-16 FF0	-15 FF1
-14 FF2	-13 FF3	-12 FF4	-11 FF5	-10 FF6	-9 FF7	-8 FF8
-7 FF9	-6 FFA	-5 FFB	-4 FFC	-3 FFD	-2 FFE	-1 FFF
0 000						

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	*[NULL]*	32	20	*[SPACE]*	64	40	@	96	60	`
1	1	*[START OF HEADING]*	33	21	!	65	41	A	97	61	a
2	2	*[START OF TEXT]*	34	22	"	66	42	B	98	62	b
3	3	*[END OF TEXT]*	35	23	#	67	43	C	99	63	c
4	4	*[END OF TRANSMISSION]*	36	24	$	68	44	D	100	64	d
5	5	*[ENQUIRY]*	37	25	%	69	45	E	101	65	e
6	6	*[ACKNOWLEDGE]*	38	26	&	70	46	F	102	66	f
7	7	*[BELL]*	39	27	'	71	47	G	103	67	g
8	8	*[BACKSPACE]*	40	28	(	72	48	H	104	68	h
9	9	*[HORIZONTAL TAB]*	41	29	)	73	49	I	105	69	i
10	A	*[LINE FEED]*	42	2A	*	74	4A	J	106	6A	j
11	B	*[VERTICAL TAB]*	43	2B	+	75	4B	K	107	6B	k
12	C	*[FORM FEED]*	44	2C	,	76	4C	L	108	6C	l
13	D	*[CARRIAGE RETURN]*	45	2D	-	77	4D	M	109	6D	m
14	E	*[SHIFT OUT]*	46	2E	.	78	4E	N	110	6E	n
15	F	*[SHIFT IN]*	47	2F	/	79	4F	O	111	6F	o
16	10	*[DATA LINK ESCAPE]*	48	30	0	80	50	P	112	70	p
17	11	*[DEVICE CONTROL 1]*	49	31	1	81	51	Q	113	71	q
18	12	*[DEVICE CONTROL 2]*	50	32	2	82	52	R	114	72	r
19	13	*[DEVICE CONTROL 3]*	51	33	3	83	53	S	115	73	s
20	14	*[DEVICE CONTROL 4]*	52	34	4	84	54	T	116	74	t
21	15	*[NEGATIVE ACKNOWLEDGE]*	53	35	5	85	55	U	117	75	u
22	16	*[SYNCHRONOUS IDLE]*	54	36	6	86	56	V	118	76	v
23	17	*[ENG OF TRANS. BLOCK]*	55	37	7	87	57	W	119	77	w
24	18	*[CANCEL]*	56	38	8	88	58	X	120	78	x
25	19	*[END OF MEDIUM]*	57	39	9	89	59	Y	121	79	y
26	1A	*[SUBSTITUTE]*	58	3A	:	90	5A	Z	122	7A	z
27	1B	*[ESCAPE]*	59	3B	;	91	5B	[	123	7B	{
28	1C	*[FILE SEPARATOR]*	60	3C	<	92	5C	\	124	7C	|
29	1D	*[GROUP SEPARATOR]*	61	3D	=	93	5D	]	125	7D	}
30	1E	*[RECORD SEPARATOR]*	62	3E	>	94	5E	^	126	7E	~
31	1F	*[UNIT SEPARATOR]*	63	3F	?	95	5F	_	127	7F	*[DEL]*