

Od NAND-a do procesora

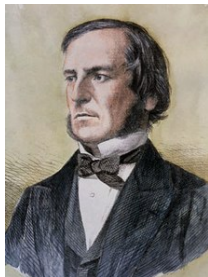
Lazar Mitrović

Matematička gimnazija

NEDELJA⁴INFORMATIKE

27. mart 2018.

- ▶ Analiza u 1. godini
 - ▶ Informatika u 1. i 2. godini
 - ▶ Računarstvo u 3.
 - ▶ Prolog u 4.
-
- ▶ ORT1, PORT, ORT2, AR, AOR1, AOR2
 - ▶ <https://www.youtube.com/watch?v=HDMkw6lnzml>



George Boole (2 November 1815 – 8 December 1864) was an English mathematician, educator, philosopher and logician. He worked in the fields of differential equations and algebraic logic, and is best known as the author of *The Laws of Thought* (1854) which contains Boolean algebra. Boolean logic is credited with laying the foundations for the information age.

Osnovne logičke operacije

operacija	simbol	novi iskaz	
<i>konjunkcija (i)</i>	$\wedge$	$p \wedge q$	$(p \text{ i } q)$
<i>disjunkcija (ili)</i>	$\vee$	$p \vee q$	$(p \text{ ili } q)$
<i>implikacija (ako ... onda)</i>	$\Rightarrow$	$p \Rightarrow q$	$(\text{ako } p \text{ onda } q)$
<i>ekvivalencija (ako i samo ako)</i>	$\Leftrightarrow$	$p \Leftrightarrow q$	$(p \text{ ako i samo ako } q)$
<i>negacija (ne)</i>	$\neg$	$\neg p$	$(\text{ne } p)$
<i>ekskluzivna disjunkcija (ili-ili)</i>	$\underline{\vee}$	$p \underline{\vee} q$	$(p \text{ ili } q \text{ ali ne oba})$

Tablice istinitosti

p	q	$p \wedge q$	$p \vee q$	$p \Rightarrow q$	$p \Leftrightarrow q$	$p \underline{\vee} q$	$\neg p$
T	T	T	T	T	T	$\perp$	$\perp$
T	$\perp$	$\perp$	T	$\perp$	$\perp$	T	$\perp$
$\perp$	T	$\perp$	T	T	$\perp$	T	T
$\perp$	$\perp$	$\perp$	$\perp$	T	T	$\perp$	T

Važnije osobine logičkih operacija

- **Komutativnost**

$$p \wedge q \Leftrightarrow q \wedge p$$

$$p \vee q \Leftrightarrow q \vee p$$

$$(p \Leftrightarrow q) \Leftrightarrow (q \Leftrightarrow p)$$

- **Idempotencija**

$$p \wedge p \Leftrightarrow p$$

$$p \vee p \Leftrightarrow p$$

- **Asocijativnost**

$$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$$

$$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$$

$$((p \Leftrightarrow q) \Leftrightarrow r) \Leftrightarrow (p \Leftrightarrow (q \Leftrightarrow r))$$

- **Apsorptivnost**

$$p \wedge (p \vee q) \Leftrightarrow p$$

$$p \vee (p \wedge q) \Leftrightarrow p$$

- **Distributivnost**

$$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$$

$$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$$

- **Involutivnost negacije**

$$\neg(\neg p) \Leftrightarrow p$$

- **Zamena za implikaciju**

$$(p \Rightarrow q) \Leftrightarrow (\neg p \vee q)$$

- **Zamena za ekvivalenciju**

$$(p \Leftrightarrow q) \Leftrightarrow ((p \Rightarrow q) \wedge (q \Rightarrow p))$$

- $\top$ je neutralni elemenat za konjunkciju:

$$(p \wedge \top) \Leftrightarrow (\top \wedge p) \Leftrightarrow p$$

- $\top$ je neutralni elemenat za ekvivalenciju:

$$(p \Leftrightarrow \top) \Leftrightarrow (\top \Leftrightarrow p) \Leftrightarrow p$$

- $\perp$ je neutralni elemenat za disjunkciju:

$$(p \vee \perp) \Leftrightarrow (\perp \vee p) \Leftrightarrow p$$

- $\top$ je nula-elemenat za disjunkciju:

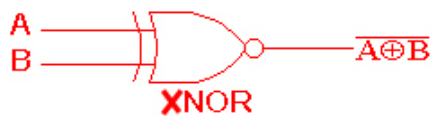
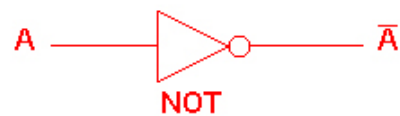
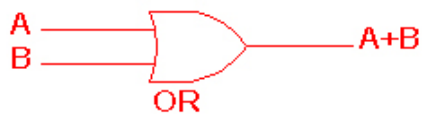
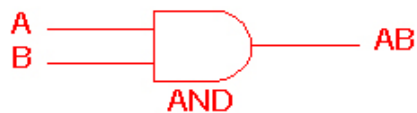
$$(p \vee \top) \Leftrightarrow (\top \vee p) \Leftrightarrow \top$$

- $\perp$ je nula-elemenat za konjunkciju:

$$(p \wedge \perp) \Leftrightarrow (\perp \wedge p) \Leftrightarrow \perp$$

Značajniji logički zakoni

- $\neg(\neg p) \Leftrightarrow p$ *zakon dvojne negacije*
- $\neg(p \wedge \neg p)$ *zakon neprotivrečnosti*
- $p \vee \neg p$ *zakon isključenja trećeg*
- $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$ *De Morganov zakon*
- $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$ *De Morganov zakon*
- $(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$ *zakon kontrapozicije*
- $(p \wedge (p \Rightarrow q)) \Rightarrow q$ *modus ponens*
- $(\neg q \wedge (p \Rightarrow q)) \Rightarrow \neg p$ *modus tolens*
- $(\neg p \Rightarrow (q \wedge \neg q)) \Rightarrow p$ *reductio ad absurdum*



AB

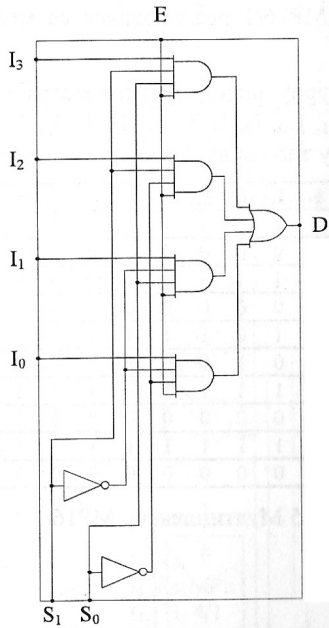
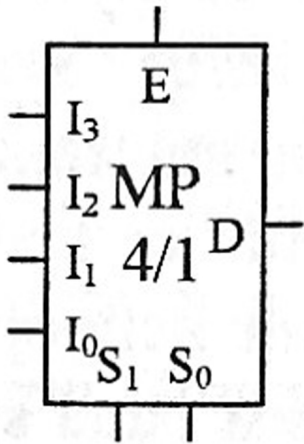
CD

	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	0	1
10	0	1	1	1

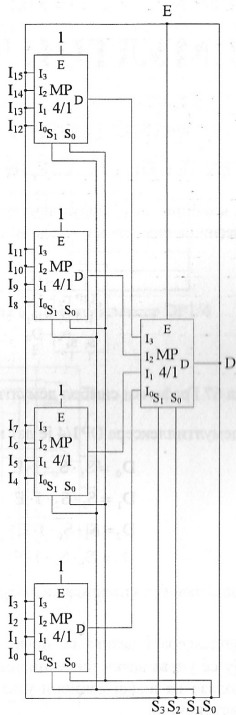
$f(A,B,C,D) = \Sigma(6,8,9,10,11,12,13,14)$

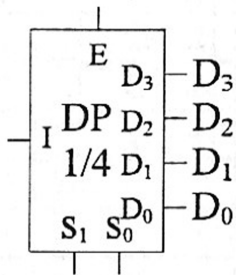
$F = AC' + AB' + BCD' + AD'$

$F = (A+B)(A+C)(B'+C'+D')(A+D')$



$$D = \overline{S_1} \cdot \overline{S_0} \cdot I_0 \cdot E + \overline{S_1} \cdot S_0 \cdot I_1 \cdot E + S_1 \cdot \overline{S_0} \cdot I_2 \cdot E + S_1 \cdot S_0 \cdot I_3 \cdot E$$



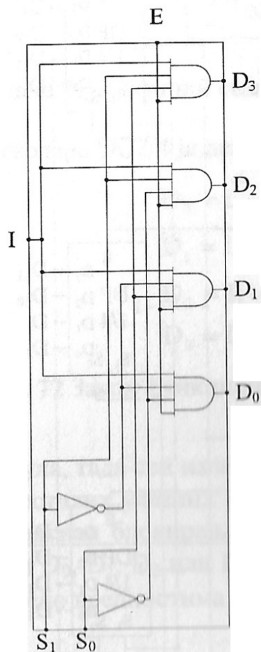


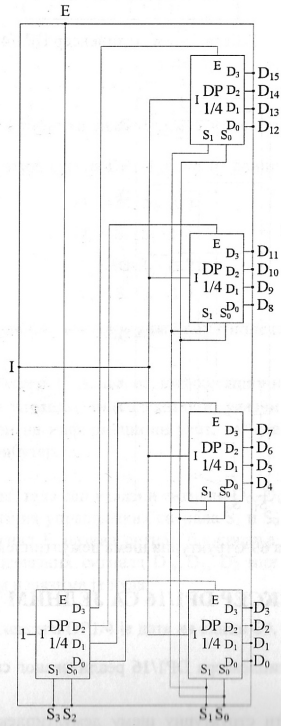
$$D_0 = \overline{S_1} \cdot \overline{S_0} \cdot I \cdot E$$

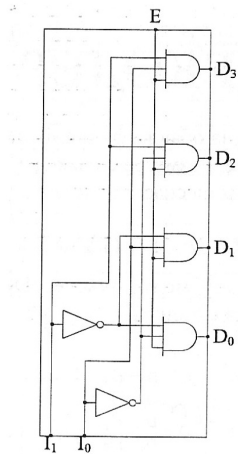
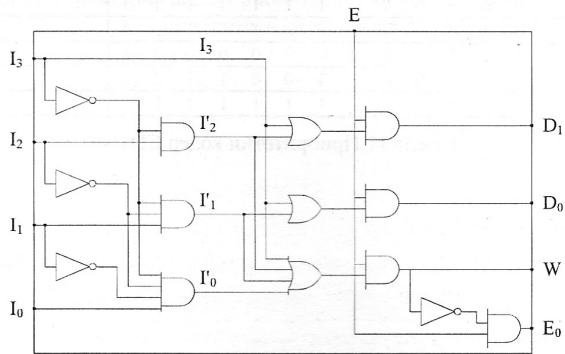
$$D_1 = \overline{S_1} \cdot S_2 \cdot I \cdot E$$

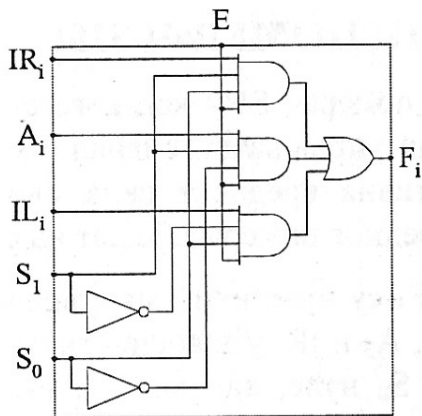
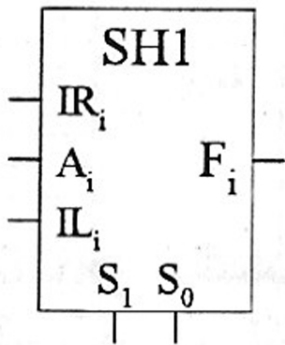
$$D_2 = S_1 \cdot \overline{S_0} \cdot I \cdot E$$

$$D_3 = S_1 \cdot S_0 \cdot I \cdot E$$



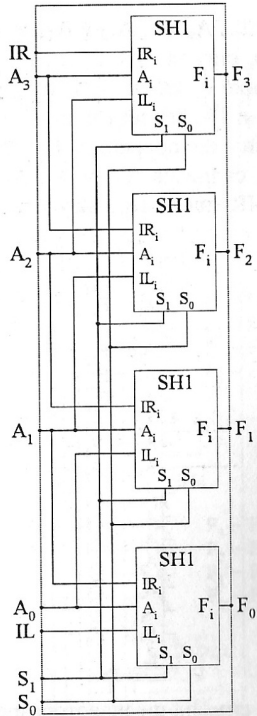


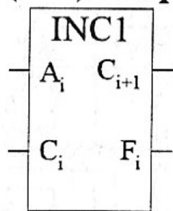




$$F_i = \overline{S_1} \cdot \overline{S_0} \cdot 0 + \overline{S_1} \cdot S_0 \cdot \overline{IL_i} + S_1 \cdot \overline{S_0} \cdot A_i + S_1 \cdot S_0 \cdot IR_i$$

$$F_i = \overline{S_1} \cdot S_0 \cdot \overline{IL_i} + S_1 \cdot \overline{S_0} \cdot A_i + S_1 \cdot S_0 \cdot IR_i$$

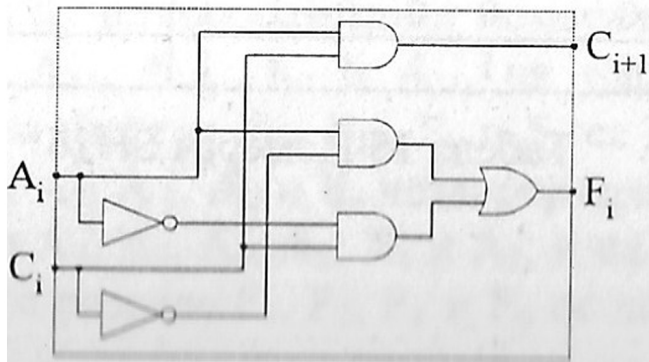


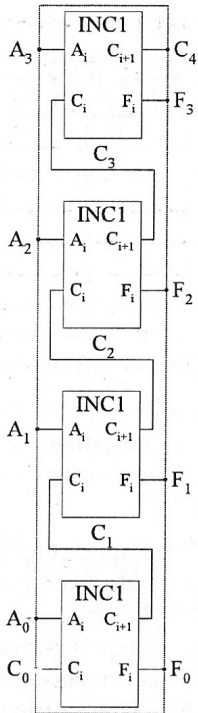


A_i	C_i	F_i	C_{i+1}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$F_i = \overline{A_i} \cdot C_i + A_i \cdot \overline{C_i}$$

$$C_{i+1} = A_i \cdot C_i$$







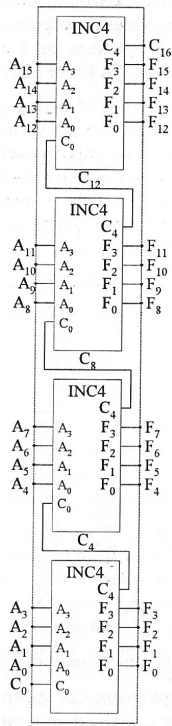
Radijator sa 16 rebara

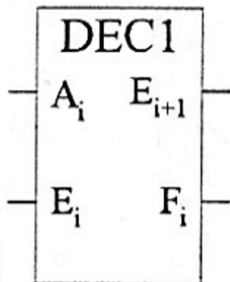
133.664 приказа

👍 1 хиљ.

💬 37

➦ дели

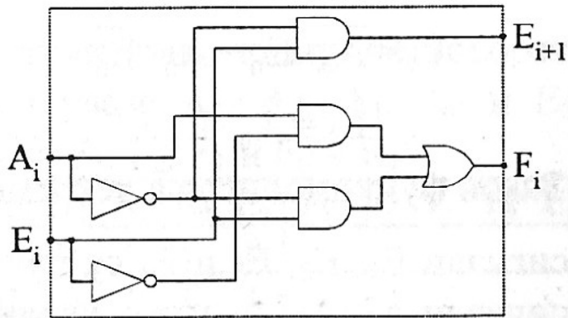


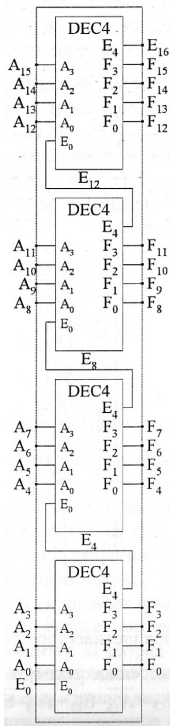


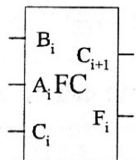
A_i	E_i	F_i	E_{i+1}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$F_i = \overline{A_i} \cdot E_i + A_i \cdot \overline{E_i}$$

$$E_{i+1} = \overline{A_i} \cdot E_i$$



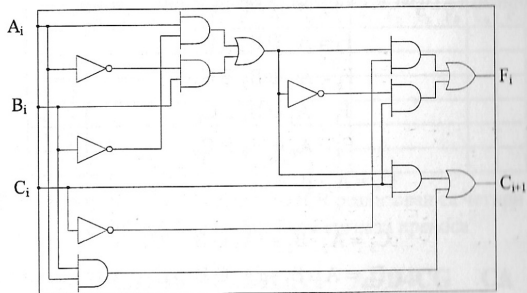


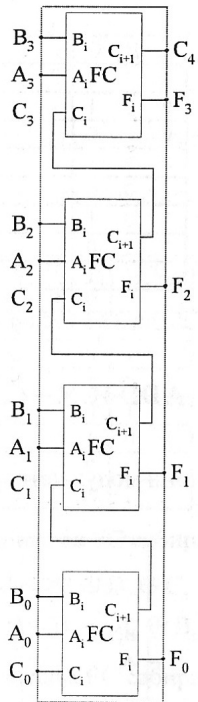


A_i	B_i	C_i	F_i	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

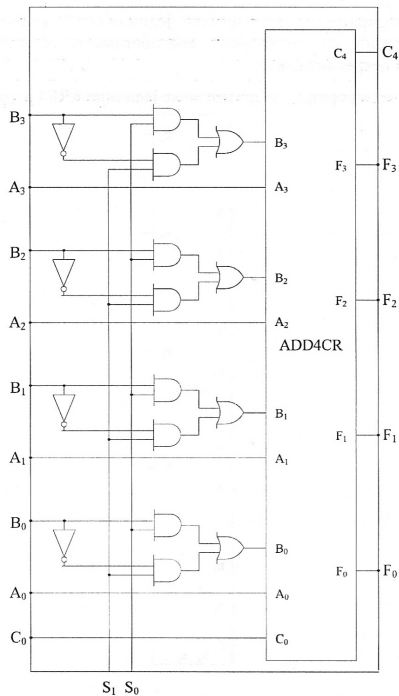
$$F_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = A_i \cdot B_i + (A_i \oplus B_i) \cdot C_i$$

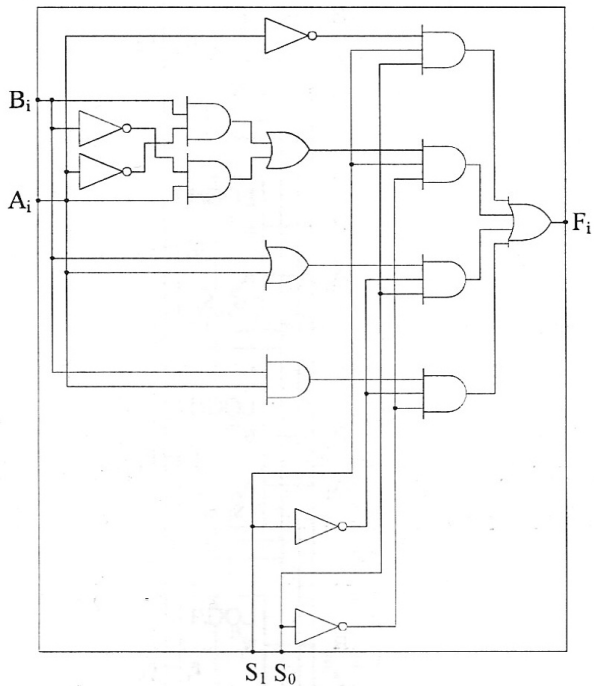




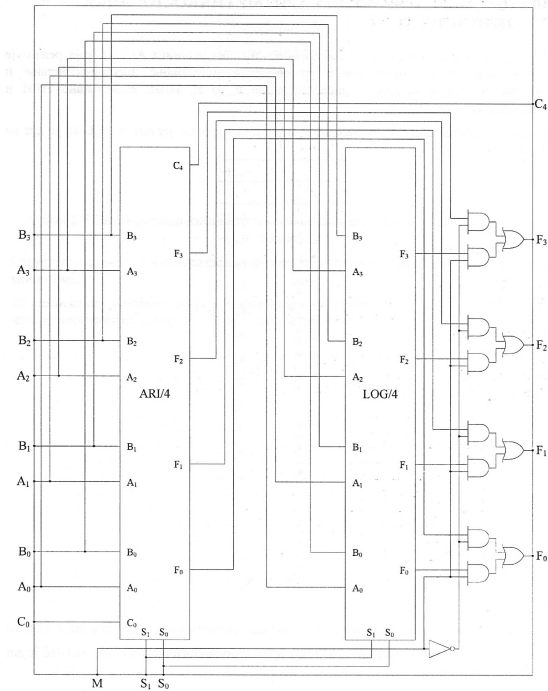
S_1	S_0	F
0	0	$A + C_0$
0	1	$A + B + C_0$
1	0	$A - B - \overline{C_0}$
1	1	$A - C_0$



S_1	S_0	F_i
0	0	$A_i \cdot B_i$
0	1	$A_i + B_i$
1	0	$A_i \oplus B_i$
1	1	$\overline{A_i}$



M	S ₁	S ₀	операција
0	0	0	$A + C_0$
0	0	1	$A + B + C_0$
0	1	0	$A - B - \overline{C_0}$
0	1	1	$A - \overline{C_0}$
1	0	0	$A_i \cdot B_i$
1	0	1	$A_i + B_i$
1	1	0	$A_i \oplus B_i$
1	1	1	$\overline{A}$



RS latch

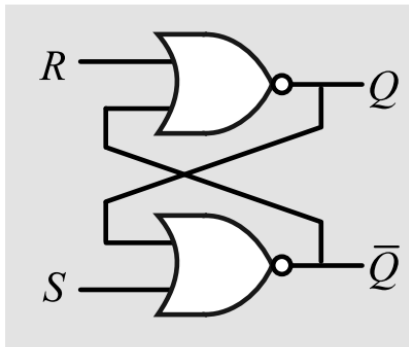
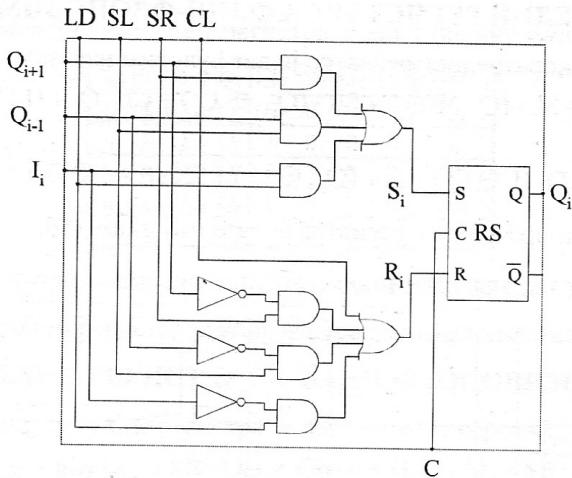


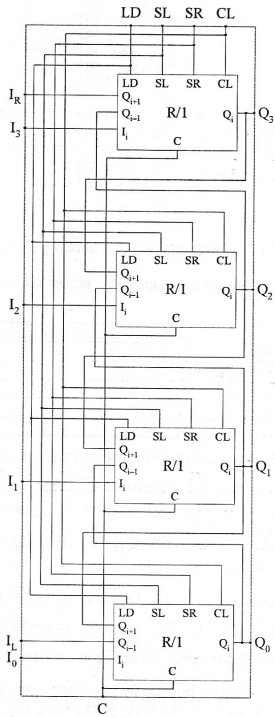
tabela preslikavanja

S	R	Q'	$\bar{Q}'$	
0	0	Q	$\bar{Q}$	<i>hold</i>
0	1	0	1	<i>reset</i>
1	0	1	0	<i>set</i>
1	1	0	0	<i>illegal</i>

$$S_i = LD \cdot I_i + SL \cdot Q_{i-1} + SR \cdot Q_{i+1}$$

$$R_i = LD \cdot \bar{I}_i + SL \cdot \overline{Q_{i-1}} + SR \cdot \overline{Q_{i+1}} + CL$$



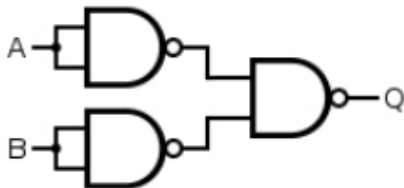


Desired OR Gate



$$Q = A \text{ OR } B$$

NAND Construction



$$= (A \text{ NAND } A) \text{ NAND } (B \text{ NAND } B)$$

Truth Table

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	1

Desired NOT Gate



$$Q = \text{NOT}(A)$$

Truth Table

Input A	Output Q
0	1
1	0

NAND Construction



$$= A \text{ NAND } A$$

Desired AND Gate

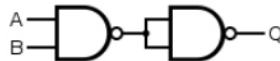


$$Q = A \text{ AND } B$$

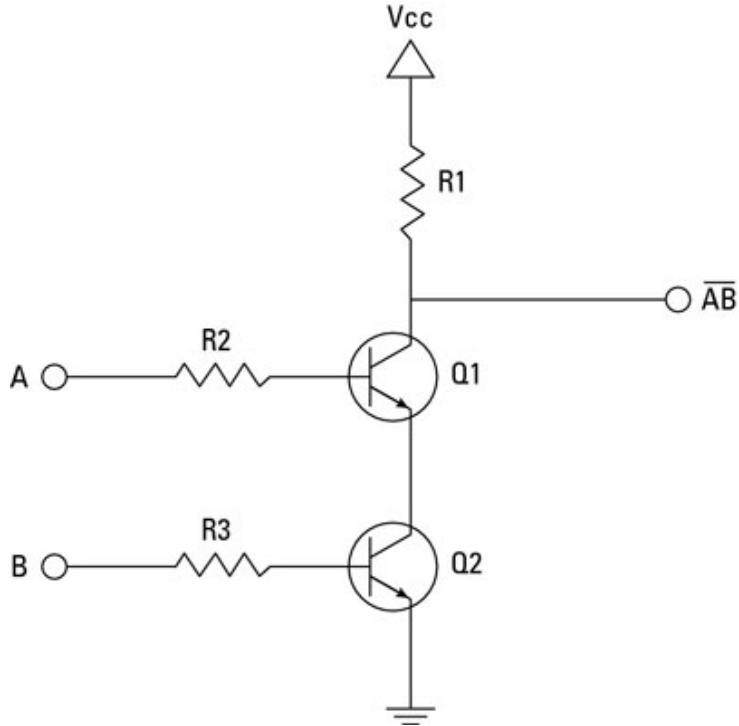
Truth Table

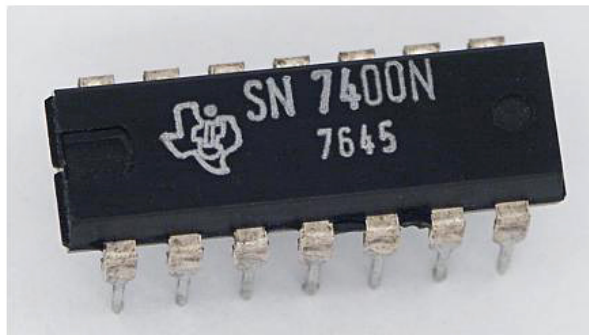
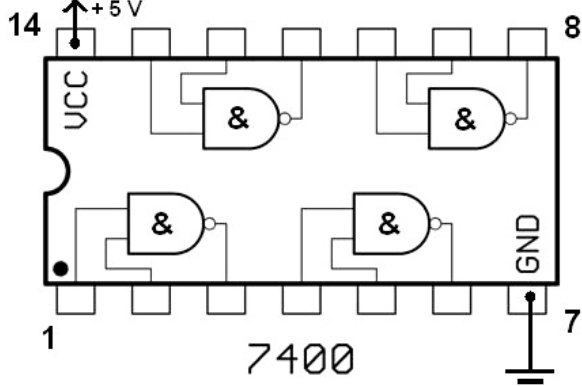
Input A	Input B	Output Q
0	0	0
0	1	0
1	0	0
1	1	1

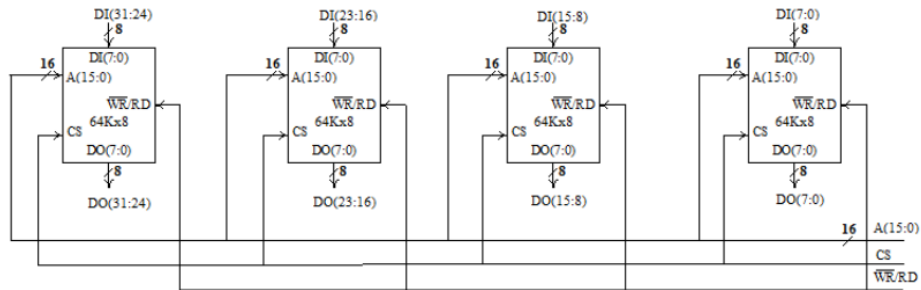
NAND Construction

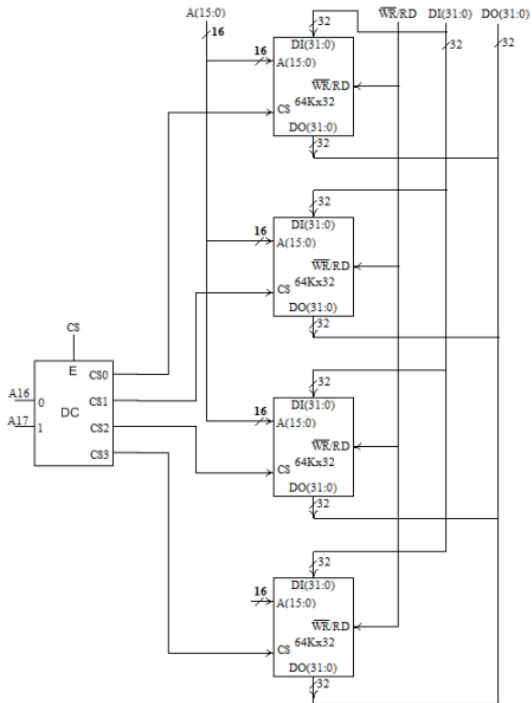


$$= (A \text{ NAND } B) \text{ NAND } (A \text{ NAND } B)$$











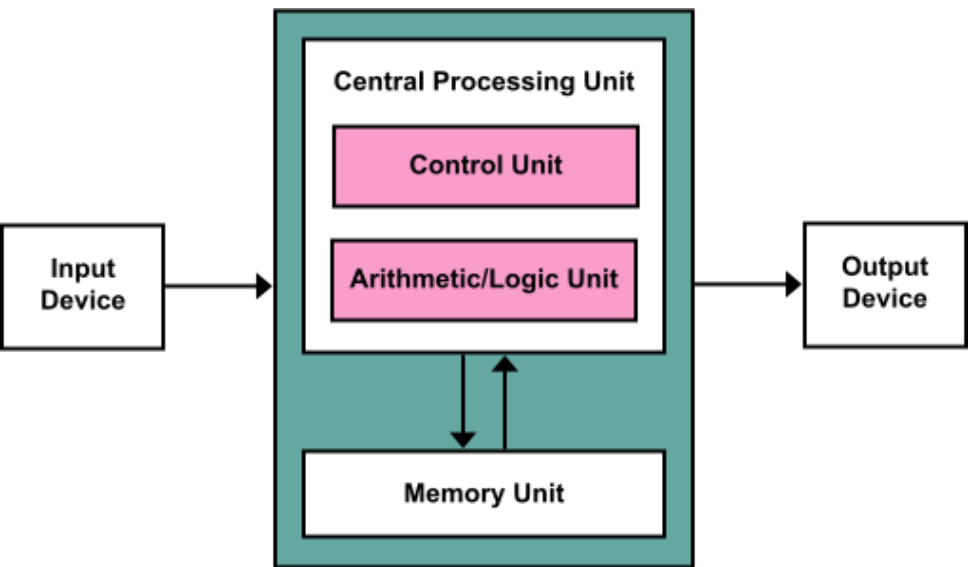
John von Neumann

(December 28, 1903 – February 8, 1957)

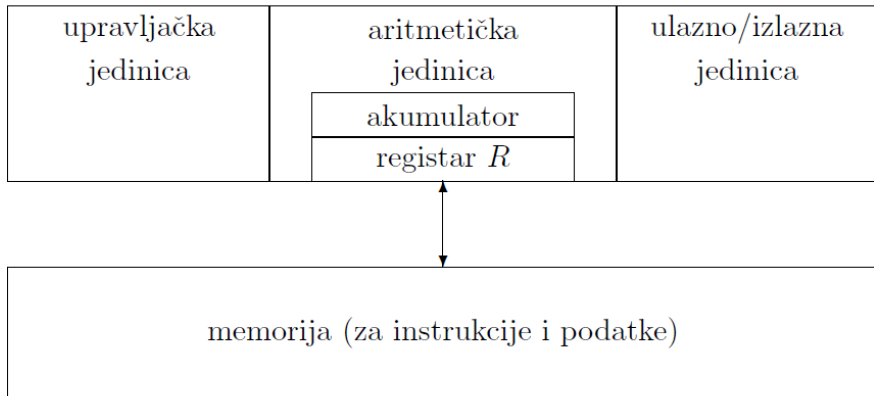
was a Hungarian-American mathematician, physicist, computer scientist, and polymath.

He made major contributions to a number of fields, including mathematics (foundations of mathematics, functional analysis, ergodic theory, representation theory, operator algebras, geometry, topology, and numerical

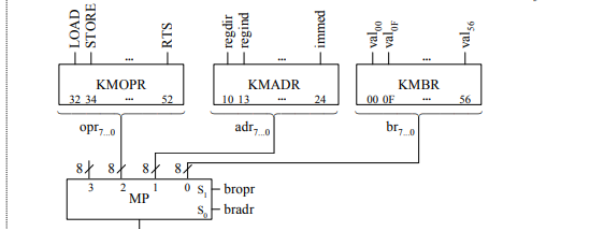
analysis), physics (quantum mechanics, hydrodynamics, and quantum statistical mechanics), economics (game theory), computing (Von Neumann architecture, linear programming, self-replicating machines, stochastic computing), and statistics.



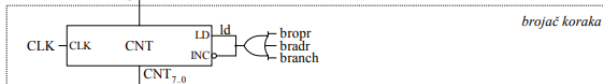
CPU



*generisanje nove vrednosti
brojača koraka*



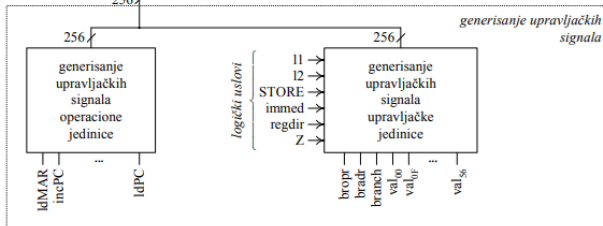
brojač koraka



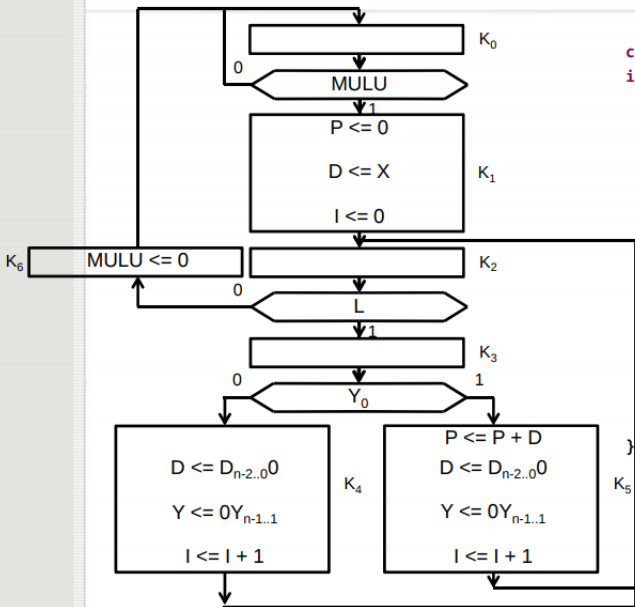
dekoder koraka



*generisanje upravljačkih
signala*



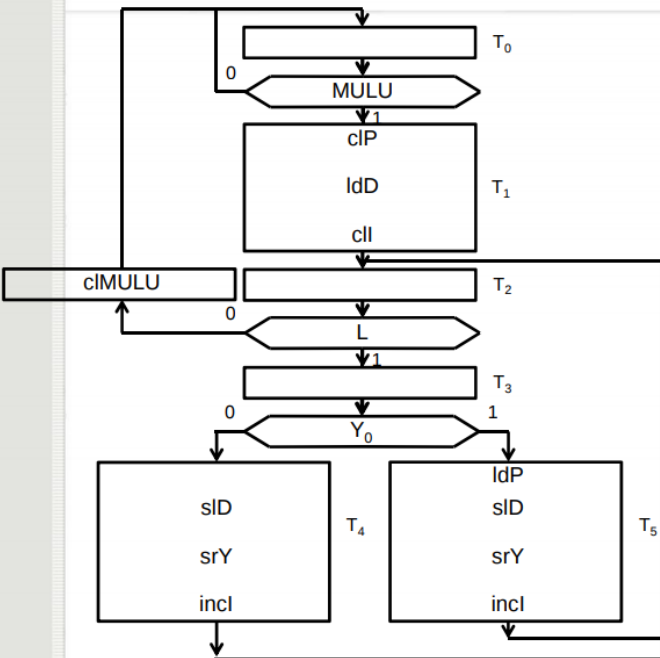
Операција множења – MULU



```

const int n = ...
int mulu(int x, int y) {
    int p = 0;
    int d = x;
    for (int i = 0; i < n; i++) {
        if ((y & 1) == 0) {
            d = d << 1;
            y = y >> 1;
        } else {
            p = p + d;
            d = d << 1;
            y = y >> 1;
        }
    }
    return p;
}
    
```

Операција множења – MULU



$$n! = n (n-1) (n-2) \dots 2 1 = \prod_{i=1}^n i$$

N=1	NF: MOV F, 1
F=2	BEQ N, 0, GOTOVO
ORG 8	PETLJA: MUL F, F, N
DALJE: IN N	SUB N, N, 1
BGT 0, N, KRAJ	BGT N, 0, PETLJA
JSR NF	GOTOVO: RTS
OUT F	
BEQ N, N, DALJE	
KRAJ: STOP	

neposredno adresiranje

LD #10h; ACC = 10h

memorijsko direktno adresiranje

LD 10h; ACC = MEM[10h] = MEM[0010h]; ACC = 00h

memorijsko indirektno adresiranje

LD (10h); ACC = MEM[MEM[10h]]; ACC = MEM[1000h]; ACC = ABh

registarsko direktno adresiranje

LD R1; ACC = R1; ACC = 10h

registarsko indirektno adresiranje

LD (R1); ACC = MEM[R1]; ACC = MEM[10h]; ACC = 00h;

registarsko indirektno adresiranje sa pomerajem

LD (R1)20h; ACC = MEM[R1+20h]; ACC = MEM[30h]; ACC = FFh;

registarsko indirektno adresiranje sa pomerajem (pomeraj 8 bita)

LD (R1)F0h; ACC = MEM[R1+FFF0h]; ACC = MEM[0000h]; ACC = 56h;

registarsko indirektno adresiranje sa pomerajem (pomeraj 16 bita)

LD (R1)F0h; ACC = MEM[R1+00F0h]; ACC = MEM[0100h]; ACC = CDh;

registarsko indirektno adresiranje sa postinkrementiranjem

LD (R1)+; ACC = MEM[R1++]; ACC = 00h, R1 = 11h (u slucaju da je operand 2B R1 = 12h)

registarsko indirektno adresiranje sa preinkrementiranjem

LD +(R1); ACC = MEM[++R1]; R1=11h; ACC = 10h

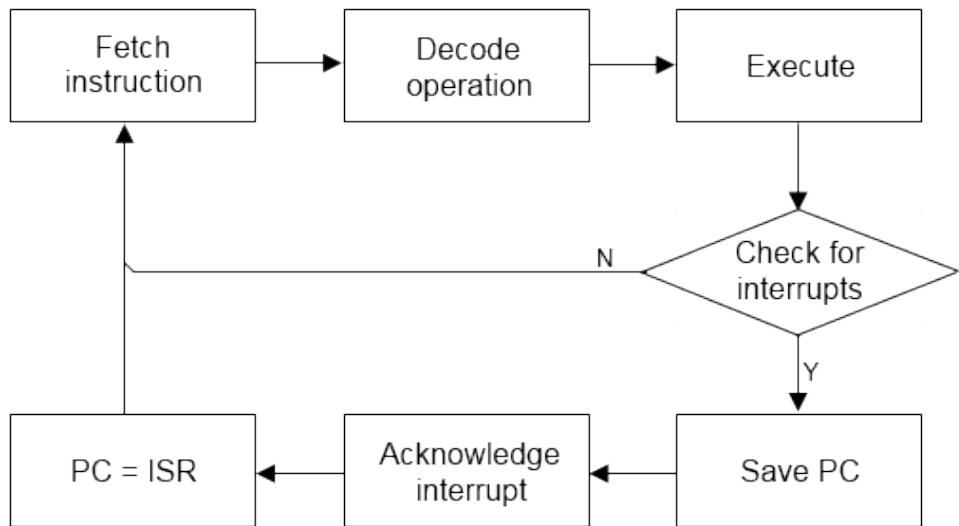
registarsko indirektno adresiranje sa postdekrementiranjem

LD (R1)-; ACC = MEM[R1--]; ACC = 00h, R1 = Fh (u slucaju da je operand 2B R1 = Eh)

registarsko indirektno adresiranje sa predekrementiranjem

LD -(R1); ACC = MEM[--R1]; R1=Fh; ACC = 78h

PROGRAM PRIMER;	
begin	
integer par1, par2;	
...	
procedure PRIM (x,y);	
begin	
integer a, b, c;	
...	
begin	PUSH BP
	MOVE SP, BP
	PUSH R1
	PUSH R2
	PUSHPSW
	SUB SP, 3
...	...
CALL PRIM (a,b);	PUSH a
	PUSH b
	JSR PRIM
	ADD SP, 2
...	...
end;	ADD SP, 3
	POPSPW
	POP R2
	POP R1
	POP BP
	RTS
end;	
...	
begin	
...	
CALL PRIM (par1,par2);	PUSH par1
	PUSH par2
	JSR PRIM
	ADD SP, 2
...	
end;	
end;	



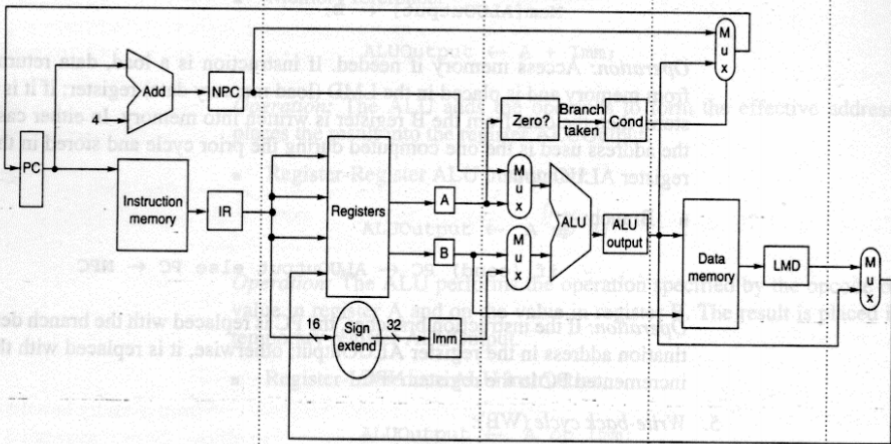
Instruction fetch

Instruction decode/
register fetch

Execute/
address
calculation

Memory
access

Write
back





- ▶ RISC
- ▶ CISC

- ▶ 8086
- ▶ 80286 (286)
- ▶ 80386 (386)
- ▶ 80486 (486)
- ▶ Pentium
- ▶ Pentium 2,3,4
- ▶ AMD64

- ▶ Spekulativno izvršavanje
- ▶ Keširanje
- ▶ Virtualizacija
- ▶ Swap
- ▶ ...

Nije kraj.



Hvala na pažnji!