

Uvod v računalništvo (UvR)

Osnovni gradniki: binarna števila, booleanova logika in vrata

Danijel Skočaj

Univerza v Ljubljani

Fakulteta za računalništvo in informatiko

Literatura: Invitation to Computer Science, poglavje 4

v1.0

Št. leto 2013/14

Cilji predavanja

- Znati pretvarjati med desetiškimi in dvojiškimi števili
- Znati ustrezno predstavljati negativna števila
- Znati razložiti kako so števila, znaki, zvok in slike predstavljeni v računalniku
- Zgraditi tabele pravilnosti za Booleove izraze in ugotoviti kdaj so resnični alineresnični
- Opisati relacijo med Booleovo logiko in računalniškimi vezji
- Konstruirati logična vezja, analizirati preprosta vezja in določiti tabele pravilnosti
- Opisati namen in delovanje nadzornih vezij multiplekserja in dekodeerja

Uvod

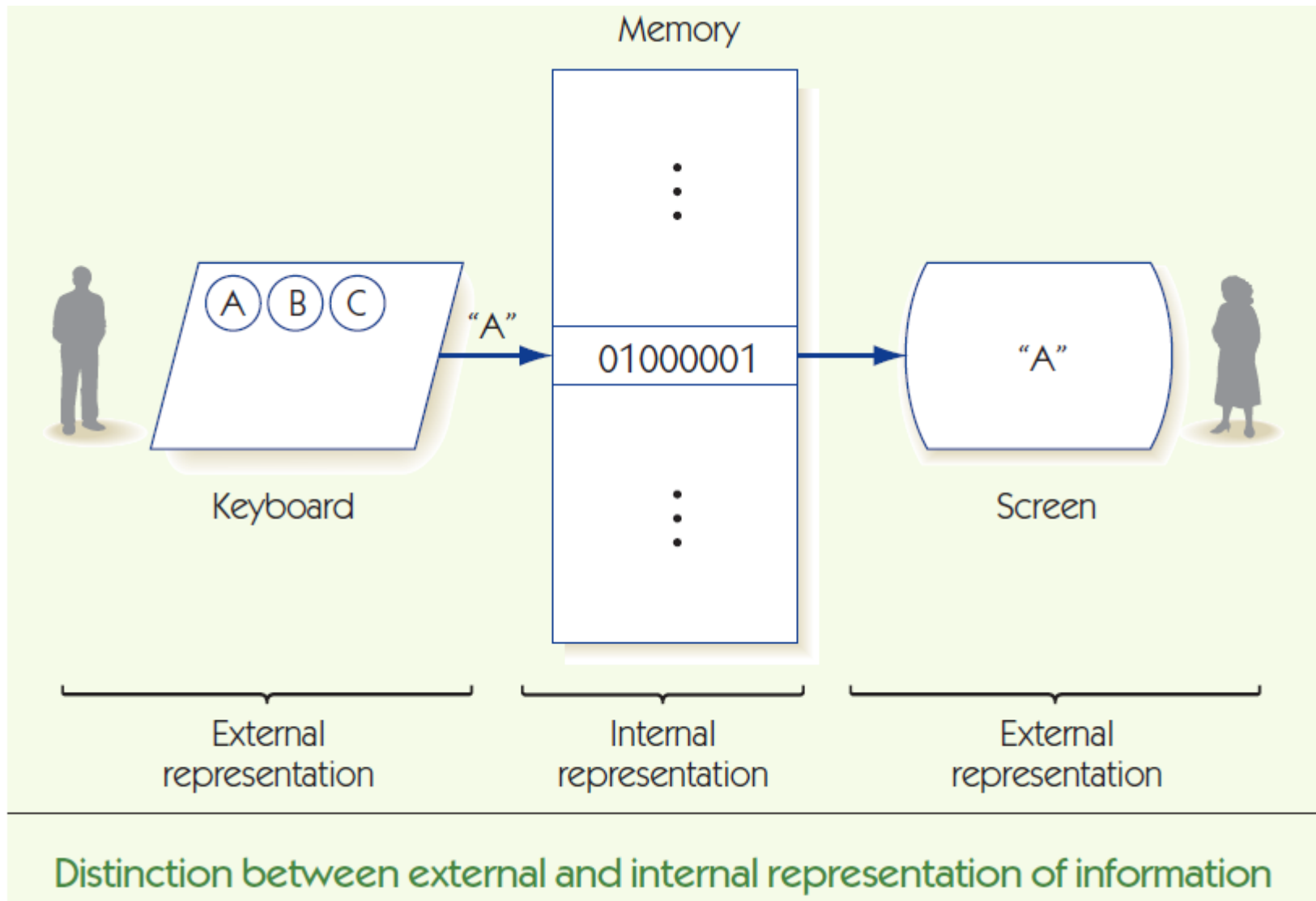
- Kako delujejo računalniki?
 - Nizkonivojsko delovanje računalnikov
- Vseh računalnikov!
 - prenosniki, osebni računalniki
 - strežniki, superračunalniki
 - igralne konzole, mobilni telefoni, MP3 predvajalniki
 - kalkulatorji
 - vgrajeni sistemi v igračah, avtomobilih, mikrovalovkah, ..



Predstavitev informacije

- Kako lahko elektronski stroj predstavi informacijo?
- Ključne zahteve:
 - jasnost
 - nedvoumnost
 - zanesljivost
- Zunanja predstavitev – prirejena človeku
 - desetiška števila
 - znaki na tipkovnici
- Notranja predstavitev – prirejena računalniku
 - dvojiška števila
 - dvojiške kode za znake

Predstavitev informacije



Dvojiški številski sistem

- Dvojiški številski sistem – osnova 2
- Desetiški sistem:
 - 10 števk: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
 - Vsako mesto ustreza potenci št. 10
 - $1943 = 1 * 10^3 + 9 * 10^2 + 4 * 10^1 + 3 * 10^0$
- Dvojiški sistem
 - 2 števki: 0, 1
 - Vsako mesto ustreza potenci št. 2
 - $1101 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 13$

Binarna števila

Binary	Decimal	Binary	Decimal
0	0	10000	16
1	1	10001	17
10	2	10010	18
11	3	10011	19
100	4	10100	20
101	5	10101	21
110	6	10110	22
111	7	10111	23
1000	8	11000	24
1001	9	11001	25
1010	10	11010	26
1011	11	11011	27
1100	12	11100	28
1101	13	11101	29
1110	14	11110	30
1111	15	11111	31

Binary-to-decimal conversion table

Dvojiški številski sistem

- Pretvarjanje iz dvojiškega v desetiški sistem
 - seštej potence števila 2, kjer je števka 1
- Pretvarjanje iz desetiškega v binarni sistem
 - število deli z dva dokler se da in si zapomni ostanke
- Fiksna dolžina binarnih števil – maksimalno število, ki je lahko še predstavljeno
 - aritmetični preliv (ang. arithmetic overflow)
- Dvojiško seštevanje:
 - $0+0=0$, $0+1=1$, $1+0=1$, $1+1=0$ in ena dalje

Negativna binarna števila

- Notacija predznak in vrednost
 - prvi bit za predznak, ostali za vrednost
 - $+5 = 0101$, $-5 = 1101$
 - $0 = 0000$ and 1000 – dve ničli!
 - precej nepraktično
 - dvoumna predstavitev
 - težje računanje
- Dvojiški komplement
 - za negativna števila obrni vsak bit in dodaj ena
 - $+5 = 0101$, $-5 = 1010 + 1 = 1011$
 - $0 = 0000$, $-0 = 1111 + 1 = 0000$ – ena ničla!
 - precej bolj praktično
 - nedvoumna predstavitev
 - lažje računanje
 - seštevanje in odštevanje: se vrtimo po številskem krogu

Števila s plavajočo vejico

- Znanstvena notacija
 - Znanstvena notacija z osnovo 10: 1.35×10^{-5}
 - Osnova 2: $3.25_{10} = 11.01_2 = 1.101 \times 2^1$
- Mantisa in eksponent: $\pm M \times 2^{\pm E}$
 - Pretvorimo število v znanstveno notacijo
 - normaliziramo
 - pomaknemo mantiso za binarno vejico in ustrezno povečamo eksponent
- Primer:
 - predznak in vrednost
 - 16 bitov: mantisa: 1+9 bitov, eksponent 1+5 bitov
 - $5,75_{10} = 101.11_2 = 101.11 \times 2^0 = ,10111 \times 2^3$
 - **0101110000000011**

Predstavitev znakov

- Preslikava znakov v binarne kode
- ASCII (8 bitov, 256 znakov)
- Unicode (16 bitov, 65.536 znakov)
- Primer: UvR = 01010101 01110110 01010010

Keyboard Character	Binary ASCII Code	Integer Equivalent	Keyboard Character	Binary ASCII Code	Integer Equivalent
(blank)	00100000	32	P	01010000	80
!	00100001	33	Q	01010001	81
"	00100010	34	R	01010010	82
#	00100011	35	S	01010011	83
\$	00100100	36	T	01010100	84
%	00100101	37	U	01010101	85
&	00100110	38	V	01010110	86
'	00100111	39	W	01010111	87
(00101000	40	X	01011000	88
)	00101001	41	Y	01011001	89
*	00101010	42	Z	01011010	90
+	00101011	43	[01011011	91
,	00101100	44	\	01011100	92
-	00101101	45]	01011101	93
.	00101110	46	^	01011110	94
/	00101111	47	_	01011111	95
0	00110000	48	`	01100000	96
1	00110001	49	a	01100001	97
2	00110010	50	b	01100010	98
3	00110011	51	c	01100011	99
4	00110100	52	d	01100100	100
5	00110101	53	e	01100101	101
6	00110110	54	f	01100110	102
7	00110111	55	g	01100111	103
8	00111000	56	h	01101000	104
9	00111001	57	i	01101001	105
:	00111010	58	j	01101010	106
;	00111011	59	k	01101011	107
<	00111100	60	l	01101100	108
=	00111101	61	m	01101101	109
>	00111110	62	n	01101110	110
?	00111111	63	o	01101111	111
@	01000000	64	p	01110000	112
A	01000001	65	q	01110001	113
B	01000010	66	r	01110010	114
C	01000011	67	s	01110011	115
D	01000100	68	t	01110100	116
E	01000101	69	u	01110101	117
F	01000110	70	v	01110110	118
G	01000111	71	w	01110111	119
H	01001000	72	x	01111000	120
I	01001001	73	y	01111001	121
J	01001010	74	z	01111010	122
K	01001011	75	{	01111011	123
L	01001100	76	:	01111100	124
M	01001101	77]	01111101	125
N	01001110	78	~	01111110	126
O	01001111	79			

ASCII conversion table

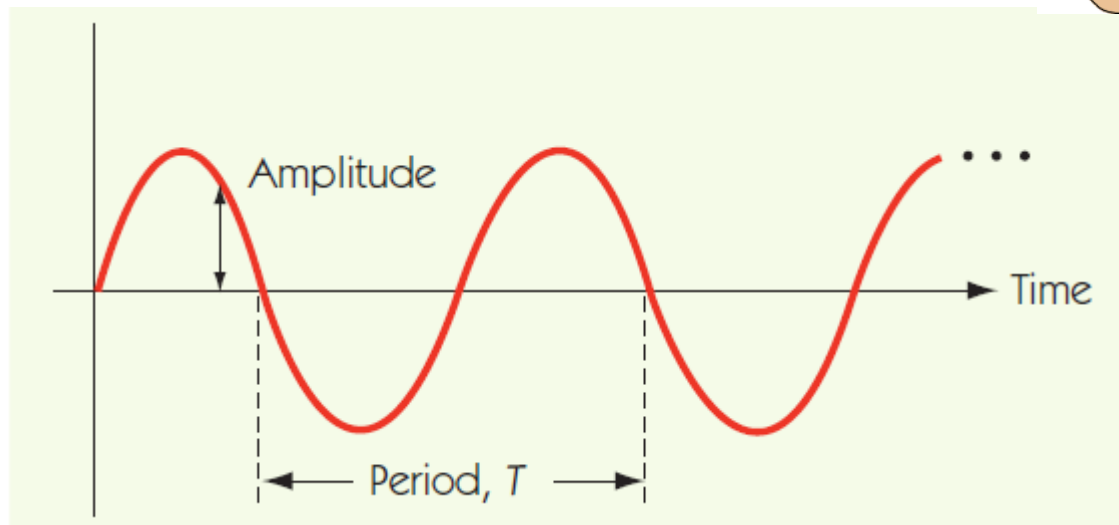
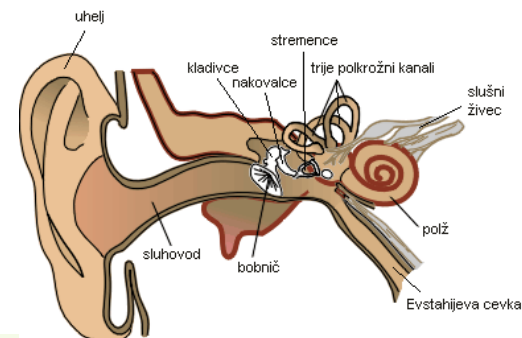
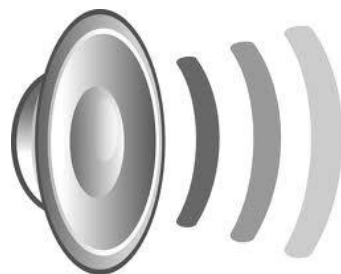
Predstavitev podatkov

- Dve pojavni obliki informacije
 - Analogni ali zvezni obseg neskončno število vrednosti
 - Digitalni ali diskretni obseg končno št. vrednosti



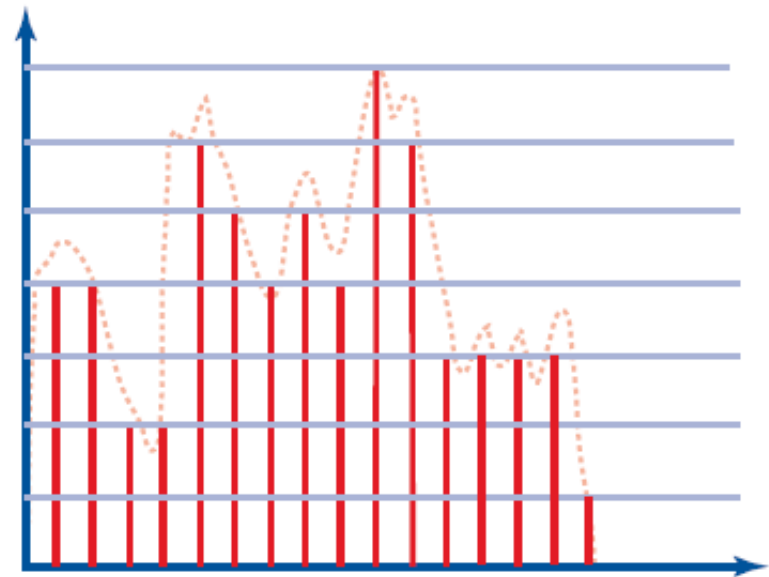
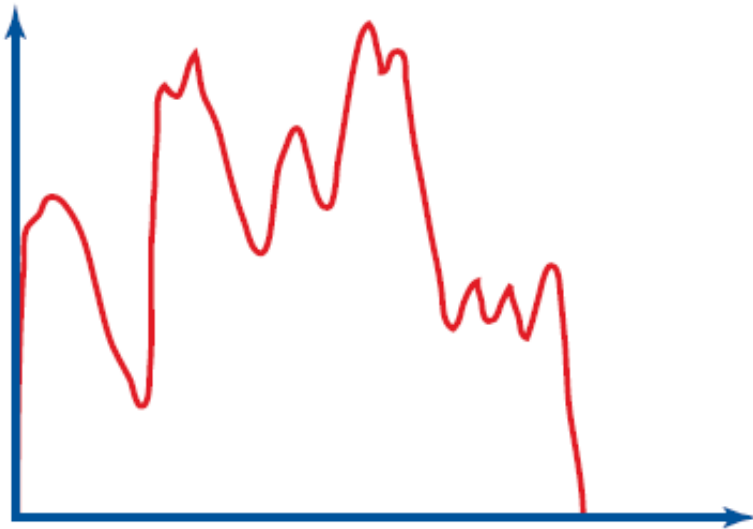
Predstavitev zvoka

- Zvok je analogni pojav
 - amplituda
 - perioda
 - frekvenca



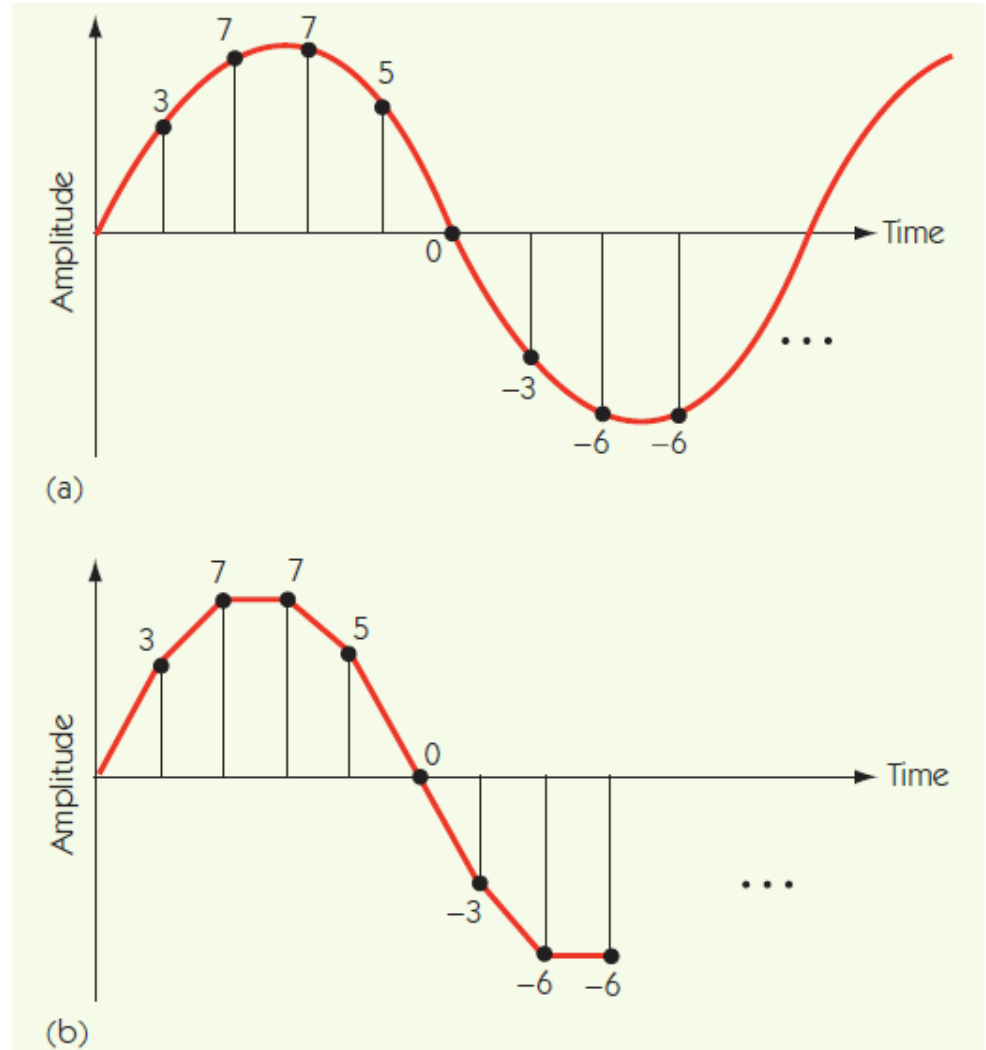
Predstavitev zvoka

- Analogno-digitalna pretvorba
 - vzročenje
 - frekvenca vzorčenja
 - št. vzorcev/sek
 - kvantizacija
 - bitna globina
 - št. nivojev



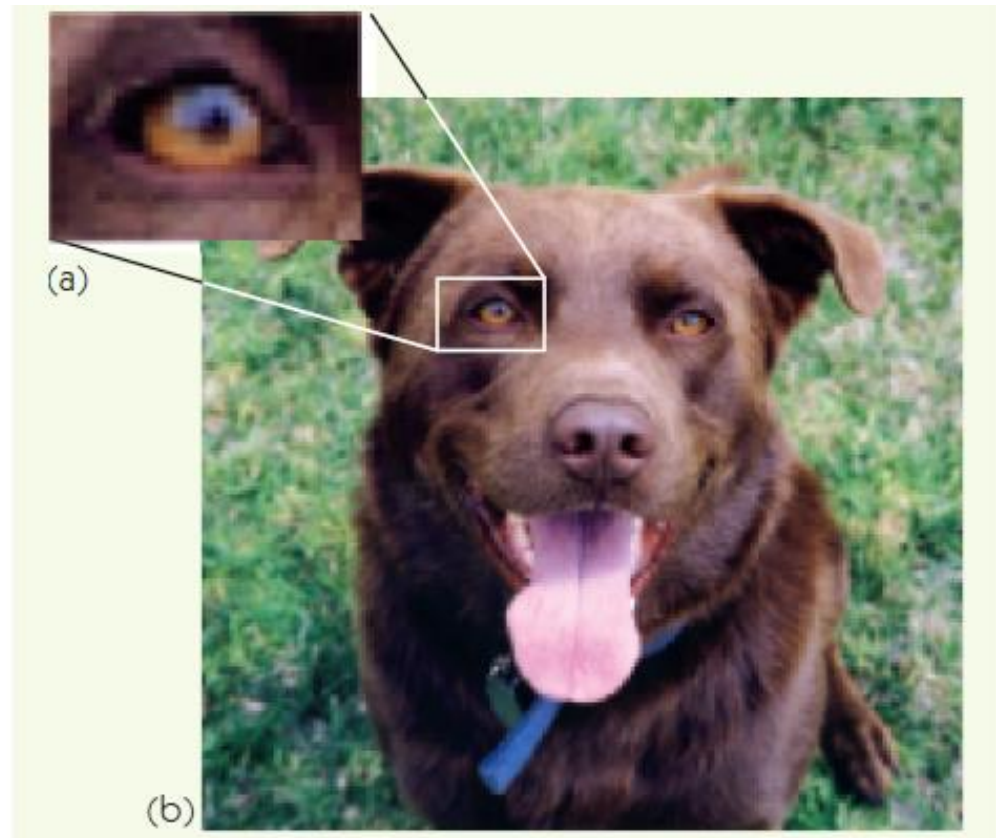
Predstavitev zvoka

- Reprodukcijska
 - približek z uporabo shranjenih diskretnih vzorcev
 - digitalno-analogni konverzija
- CD kvaliteta:
 - 44.1 kHz
 - 16 bit
- Audio formati
 - WAV, AAC, WMA, MIDI
 - MP3



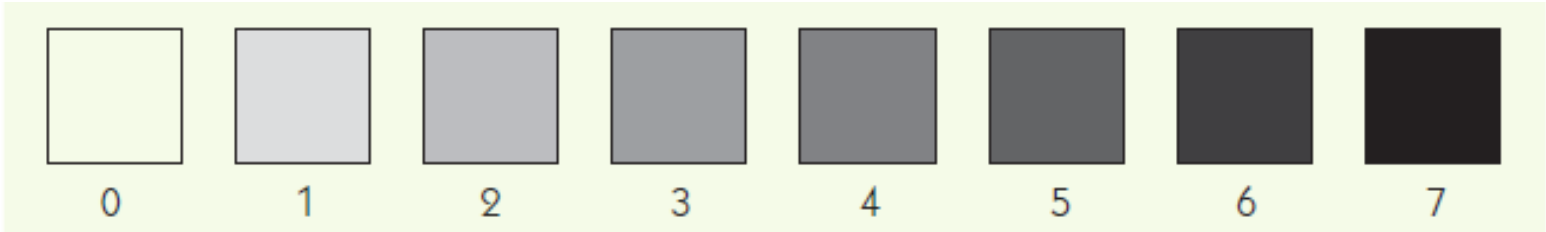
Predstavitev slike

- Vzorčenje slike
 - shranjevanje barve ali intenzitete na diskretnih intervalih v dveh dimenzijah
 - slikovni elementi (piksli)
- Ločljivost slike
 - št. slikovnih elementov
- Barvna globina
 - št. barvnih (intenzitetnih) nivojev
- Rastrske slike
 - 2D matrika števil!



Predstavitev slike

- Intenzitene slike



- Barvne slike (RGB)

- kombinacija rdeče, zelene in modre barve
- en (ali več) bytov na barvni kanal
- barvne palete

- Video: 25 slik/sek

- Ogromne količine podatkov => kompresija

Stiskanje podatkov

- Kliko prostora potrebujemo, da shranimo:
 - 300 stranski roman
 - 60 sekund zvočnega posnetka
 - slika velikosti 4.000 x 3.000 slikovnih elementov
- Nujna stiskanje (kompresija) podatkov
 - brezizgubno
 - run-length encoding
 - kode s spremenljivo dolžino
 - izgubno
 - izpusti se tisti del podatkov, ki ga človek slabše zazna
 - različne stopnje stiskanja
 - kompromis med kvaliteto in velikostjo
 - JPG, MP3, MPEG,...
 - Stopnja stiskanja
 - tudi 20:1 in več

Stiskanje podatkov

- Stiskanje s kodami s spremenljivo dolžino
 - bolj pogoste znake kodiramo s krajšimi kodami

Letter	4-bit Encoding	Variable Length Encoding
A	0000	00
I	0001	10
H	0010	010
W	0011	110
E	0100	0110
O	0101	0111
M	0110	11100
K	0111	11101
U	1000	11110
N	1001	111110
P	1010	1111110
L	1011	1111111

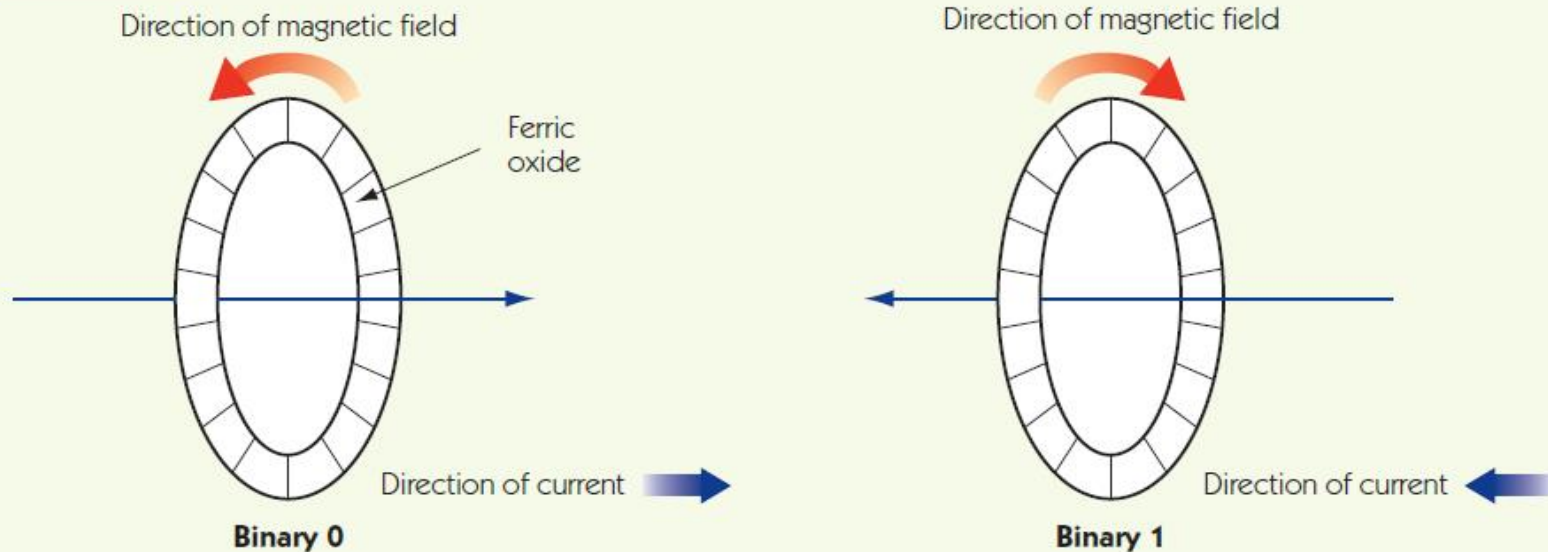
Shranjevanje binarnih podatkov

- Desetiški sistem zahteva deset stabilnih stanj
 - zelo nerobustno!
- Dvojiški sistem zahteva samo dve stabilni stanji
 - bistabilno okolje
 - zelo robustno!
- Zahteve za binarni računalnik:
 - dve stabilni energijski stanji
 - ločeni z veliko energijsko mejo
 - možno je zaznati v katerem stanju se nahaja naprava
 - možno je preklapljati med stanjema
- Različne možnosti
 - stikalo za luč vklopljeno/izklopljeno
 - napolnjeno/nenapolnjeno
 - -5V/+5V
 - pozitivno/negativno nabito
 - magnetizirano/nemagnetizirano
 - magnetizirano v smeri/v obratni smeri urinega kazalca

Shranjevanje binarnih podatkov

- Primer: magnetno jedro
 - 1955-1975
 - tok: teče/ne teče
 - magnetno polje: levo/desno

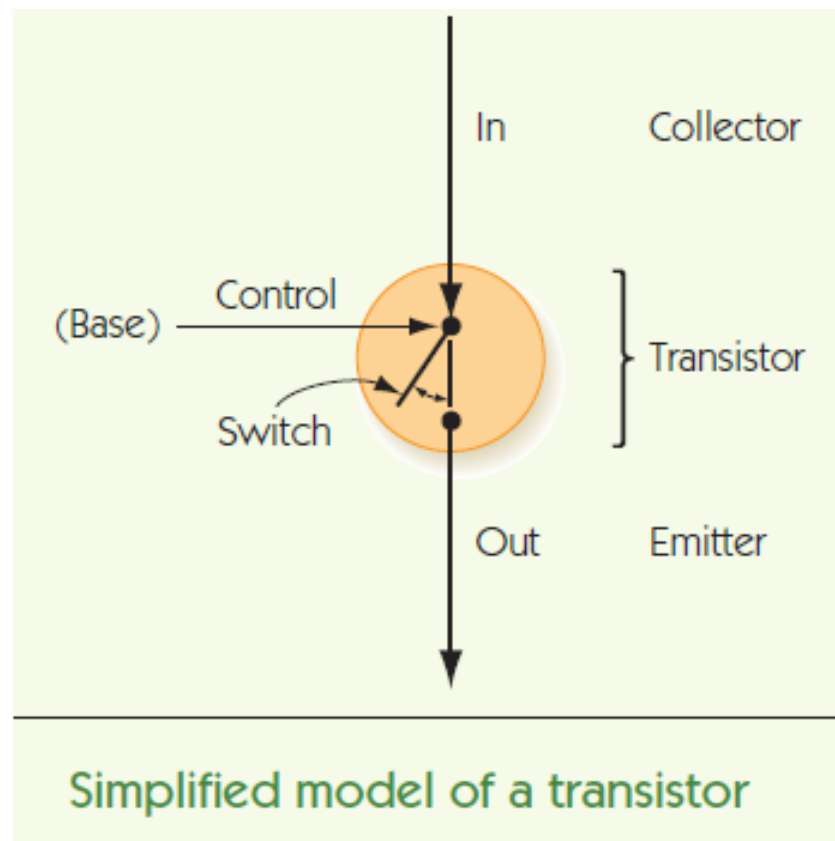
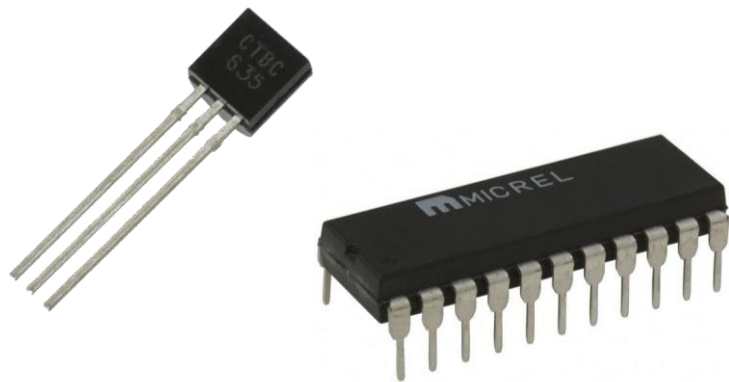
FIGURE 4.9



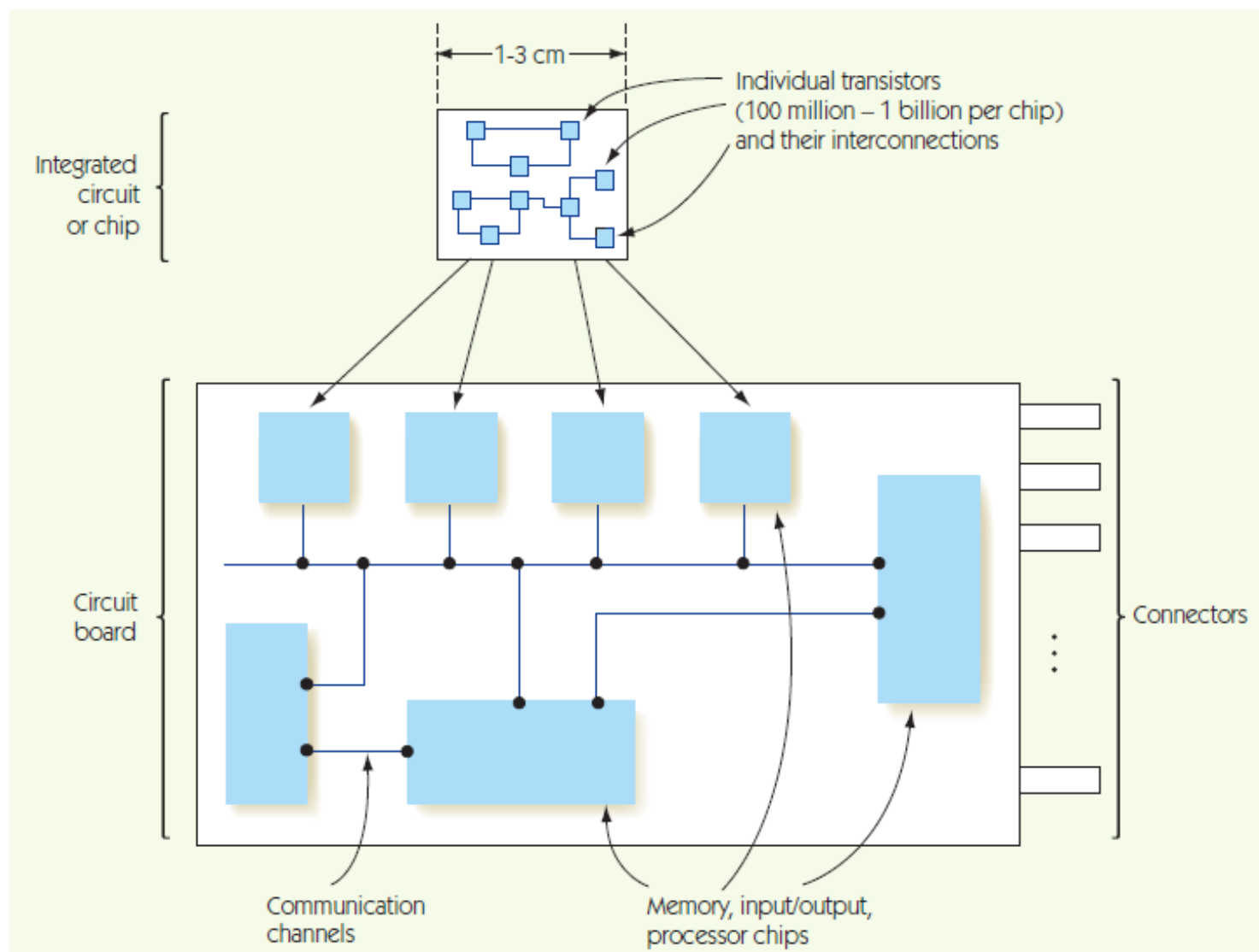
Using magnetic cores to represent binary values

Tranzistorji

- Elektronsko preklapljanje med stanjema
 - v milijardinki sekunde
- Narejen iz polprevodnikov
 - zelo majhen
 - poceni izdelava
 - GB podatkov
- Fotografsko tiskanje
 - integrirana vezja
 - čipi

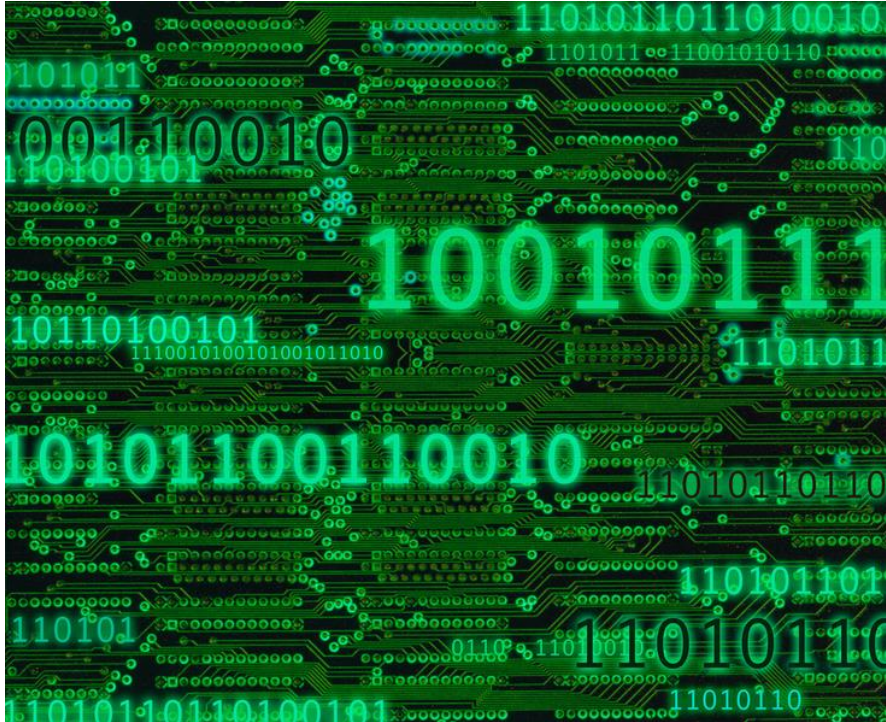


Tiskanja vezja in čipi



Booleova logika

- Pravila za delo z dvema vrednostima
 - True/false, T/F
 - 1/0



- (Logično) načrtovanje strojne opreme
 - Booleovi izrazi se lahko pretvorijo v vezja

Booleovi operatorji

- IN (AND, $\&$, \cdot , \wedge): produkt oz. konjunkcija
- ALI (OR, $|$, $+$, \vee): vsota ali disjunkcija
- NE (NOT, \sim , \neg , $\bar{}$, $'$): negacija

Inputs		Output
a	b	a AND b (also written $a \cdot b$)
False	False	False
False	True	False
True	False	False
True	True	True

Inputs		Output
a	b	a OR b (also written $a + b$)
False	False	False
False	True	True
True	False	True
True	True	True

Input	Output
a	NOT a (also written \bar{a})
False	True
True	False

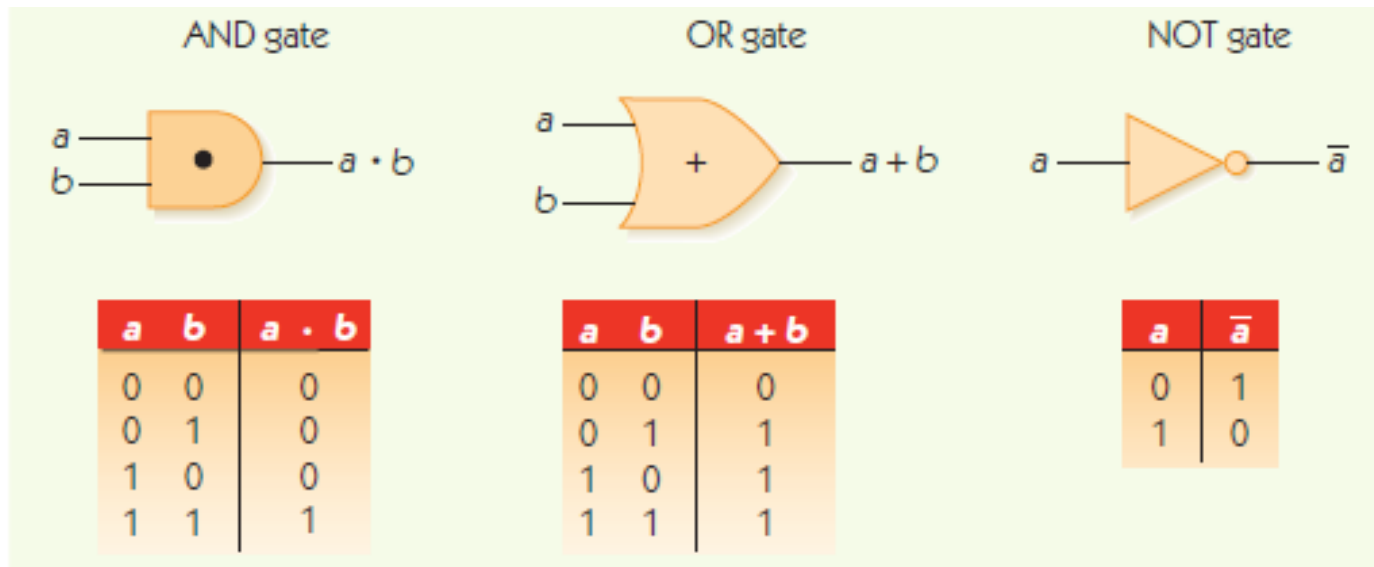
Tabela pravilnosti

- Definira rezultat (T/F) za vse možne kombinacije vhodov
- Primer: $(a \cdot b) + (a \cdot \sim b)$

a	b	$\sim b$	$(a \cdot b)$	$(a \cdot \sim b)$	$(a \cdot b) + (a \cdot \sim b)$
true	true	false	true	false	true
true	false	true	false	true	true
false	true	false	false	false	false
false	false	true	false	false	false

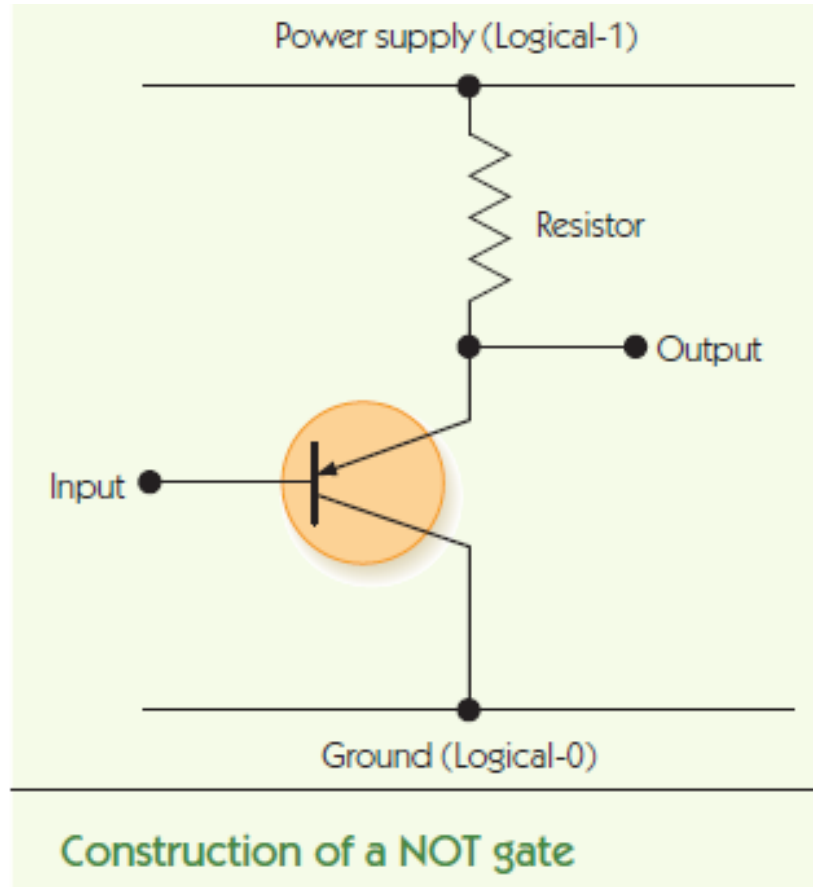
Booleova vrata

- Vrata: elektronska naprava, ki na osnovi vhodnih vrednosti proizvede izhodne vrednosti
- Vrata lahko ustrezajo Booleovim operatorjem:



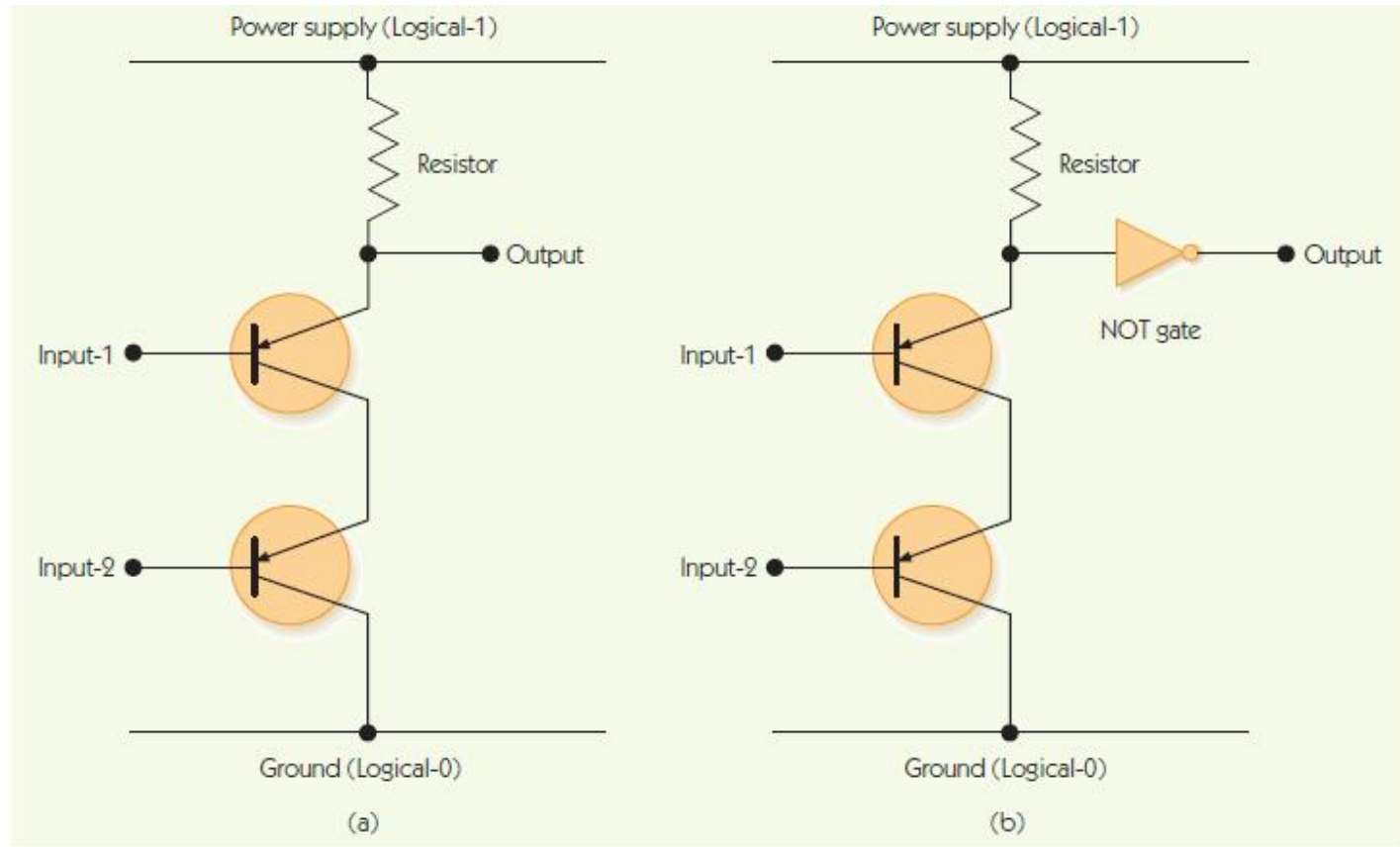
Vrata NOT

- Vrata NOT: en tranzistor



Vrata AND

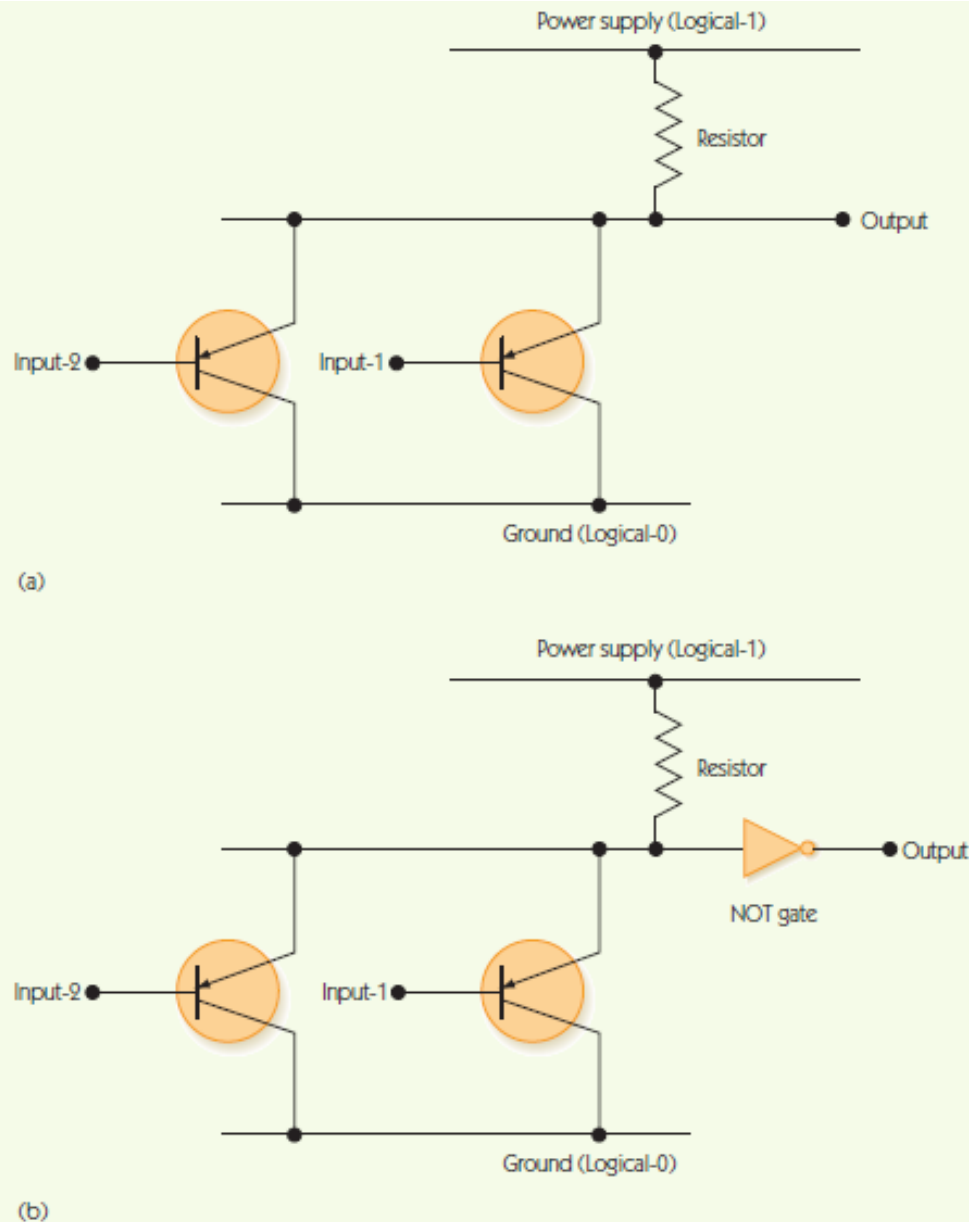
- Vrata NAND: dva tranzistorja
- Vrata AND: tri tranzistorji



Construction of NAND and AND gates
(a) A two-transistor NAND gate
(b) A three-transistor AND gate

Vrata OR

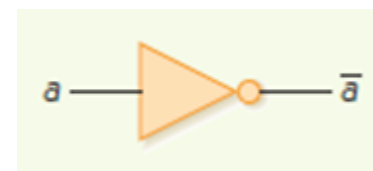
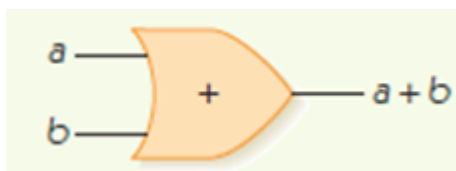
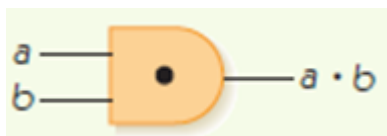
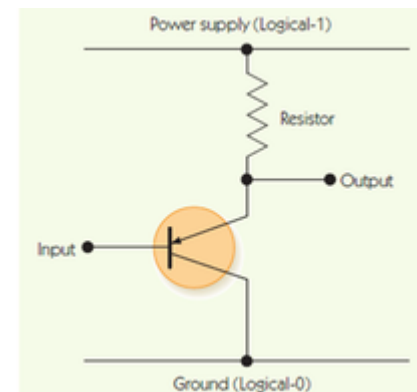
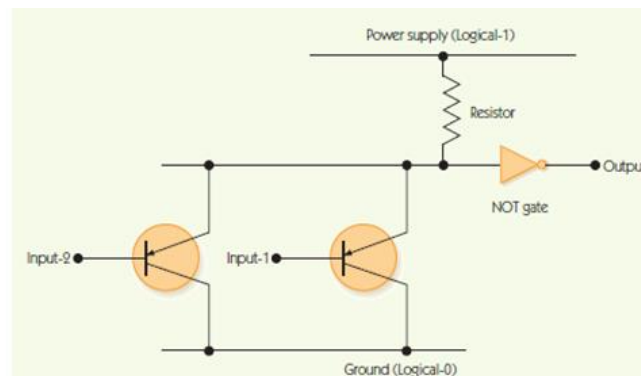
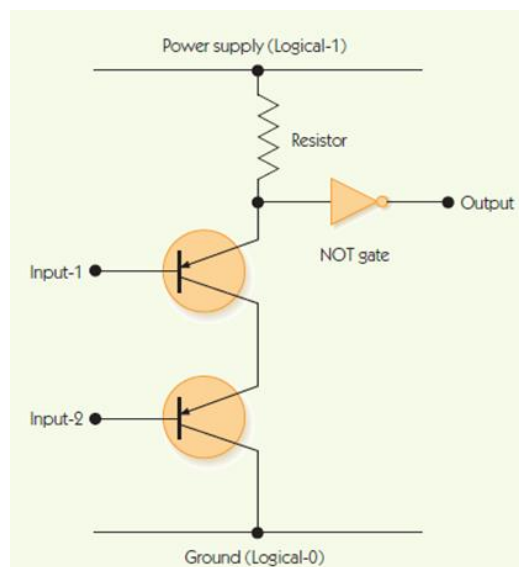
- Vrata NOR:
 - dva tranzistorja
- Vrata OR:
 - tri tranzistorji



Construction of NOR and OR gates
(a) A two-transistor NOR gate
(b) A three-transistor OR gate

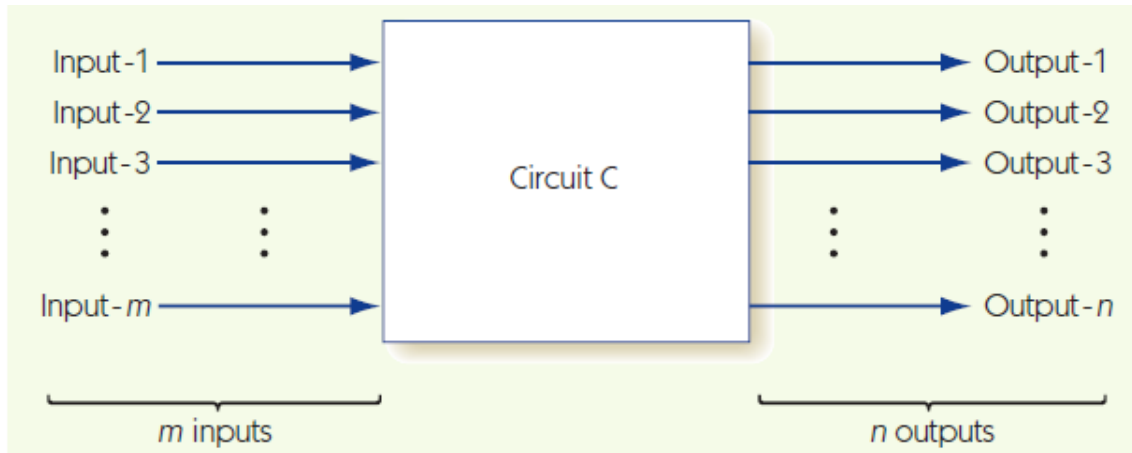
Abstrakcija

- Pri načrtovanju vezij bomo abstrahirali elektronske podrobnosti
 - načrtovali bomo z AND, OR, in NOT vrati in Booleovo logiko



Gradnja računalniških vezij

- Vezje
 - ima vhodne povezave
 - vsebuje vrata povezana s povezavami
 - ima izhodne povezave
- Izhod je odvisen samo od trenutnega vhoda
 - ni stanj



- Sekvenčna vezja vsebujejo tudi povratne zanke
 - vrednost je odvisna tudi od prejšnjega vhoda

Algoritem za konstrukcijo vezij

- Obstaja več algoritmov
- Algoritem vsota produktov
 1. Izdelaj tabelo pravilnosti
 2. Izdelaj podizraze z vrati AND in NOT
 3. Poveži podizraze z vrati OR
 4. Izdelaj diagram vezja

Algoritem vsota produktov

- Korak 1: Izdelaj tabelo pravilnosti
 - vedeti moramo kaj naj dobimo na izhodu pri določenem vhodu

a	b	c	Output1	Output2
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

Algoritem vsota produktov

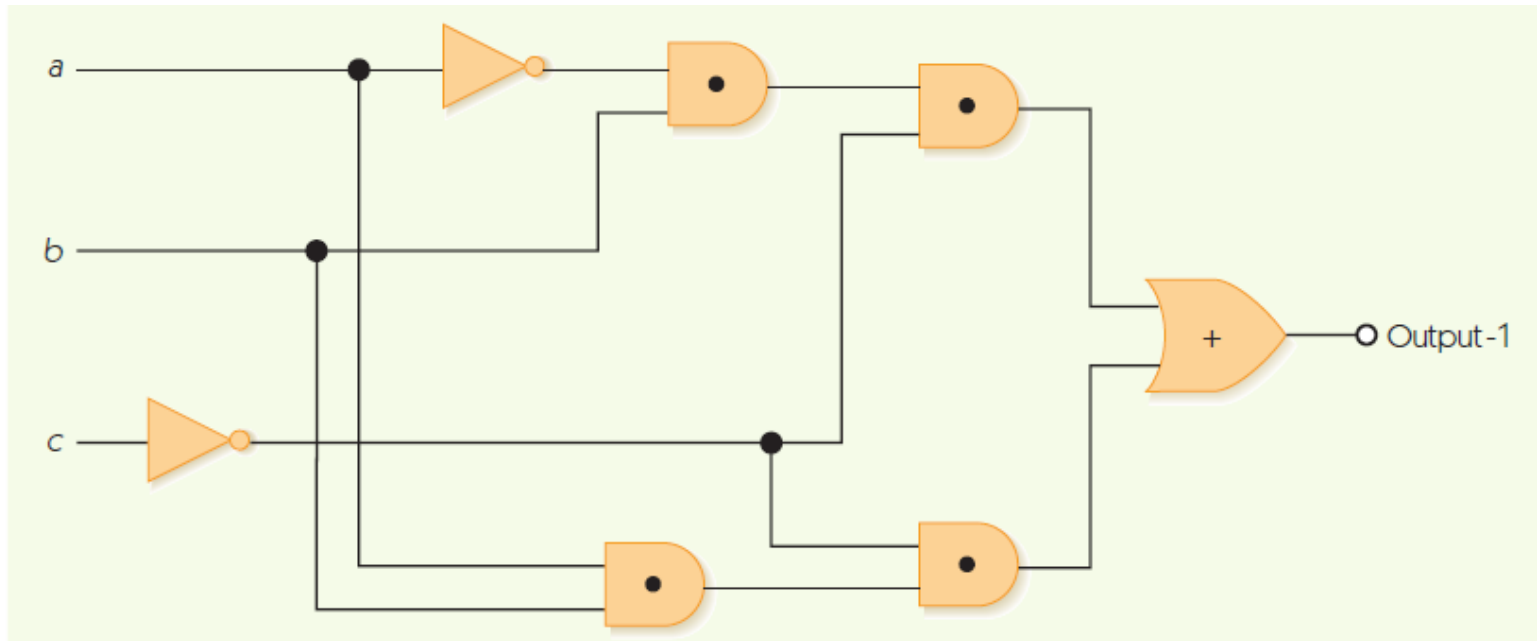
- Korak 2: Izdelaj podizraze z vrati AND in NOT
 - poišči vrstice z vrednostjo 1 (najprej gledamo za Output1)

a	b	c	Output1	Output2
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

- za vsako tako vrstico izdelaj podizraz:
 - $a = 0, b = 1, c = 0$: $(\sim a \cdot b \cdot \sim c)$
 - $a = 1, b = 1, c = 0$: $(a \cdot b \cdot \sim c)$

Algoritem vsota produktov

- Korak 3: Poveži podizraze z vrati OR
 - $(\sim a \cdot b \cdot \sim c) + (a \cdot b \cdot \sim c)$
- Korak 4: Izdelaj diagram vezja
 - iz dobljenega izraza



- Vse skupaj ponovi še za drugi izhod (Output2)
- Optimizacija vezij

Algoritem vsota produktov

1. Construct the truth table describing the behavior of the desired circuit
2. While there is still an output column in the truth table, do Steps 3 through 6
3. Select an output column
4. Subexpression construction using AND and NOT gates
5. Subexpression combination using OR gates
6. Circuit diagram production
7. Done

The sum-of-products circuit construction algorithm

Primer: Vezje za primerjanje

- Vezje CE (Compare-for-equality)
- Primerjanje dveh števil
- Zelo pogosto uporabno

- vhod: dve nepredznačeni dvojiški števili
- izhod: 1, če sta identični, 0 sicer

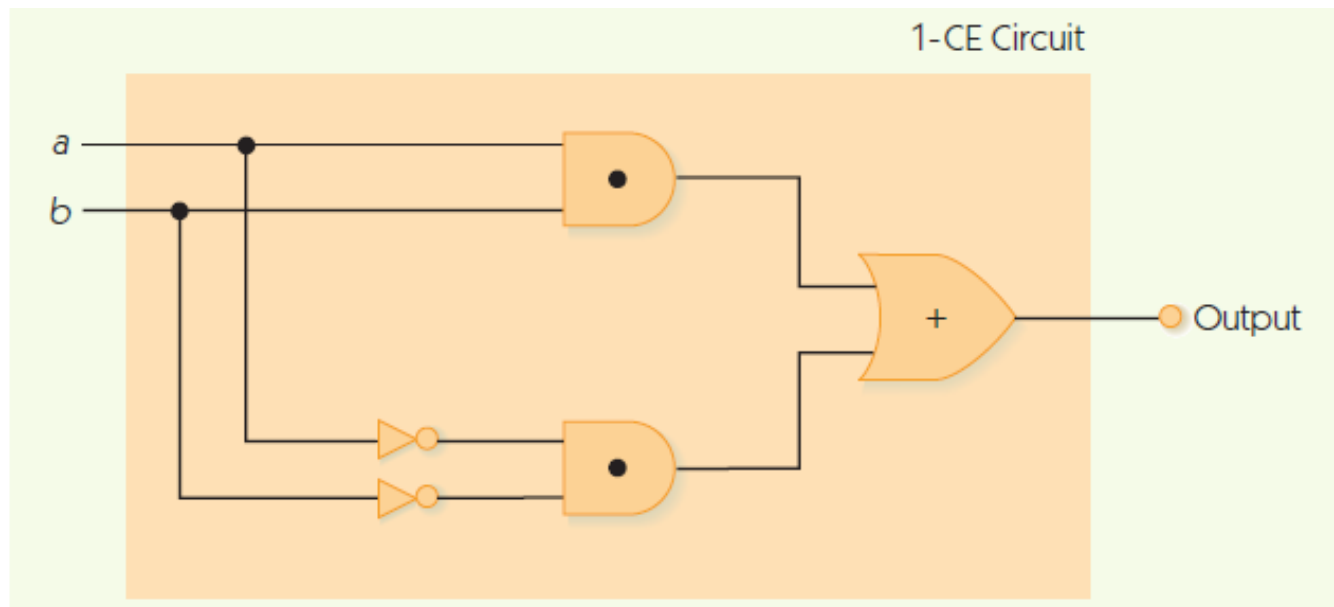
- Začni z verzijo za en bit
- Razširi jo za poljubno število bitov

Primer: Vezje za primerjanje

- Tabela pravilnosti za en bit:

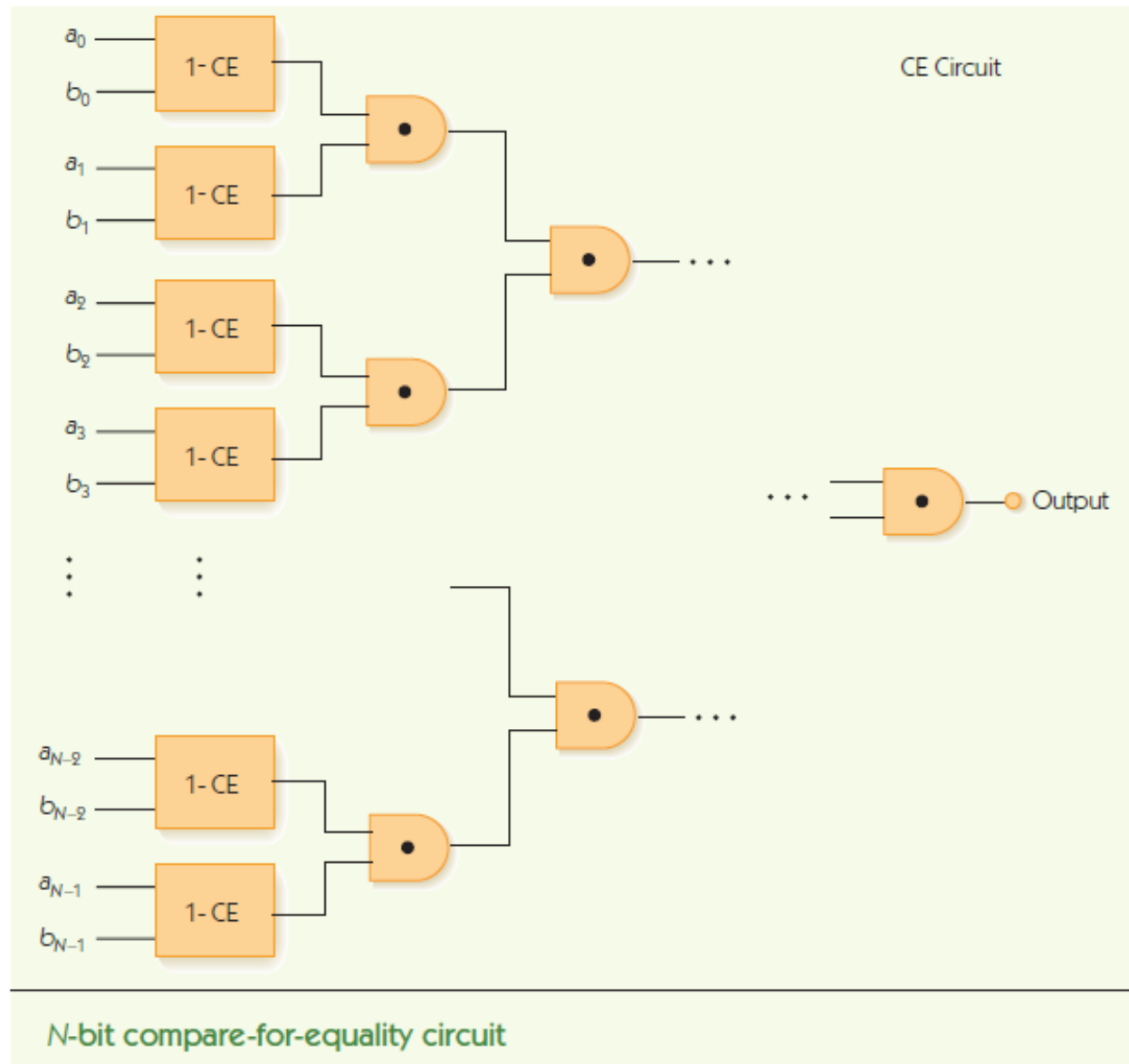
a	b	Output
0	0	1
0	1	0
1	0	0
1	1	1

- Booleov izraz: $(a \cdot b) + (\sim a \cdot \sim b)$
- Vezje za en bit:
 - 1-CE



Primer: Vezje za primerjanje

- Vezje za N bitov
- Popari istoležne bite
- Na vsakem paru poženi 1-CE
- Poveži vse skupaj z vrati AND



Primer: Seštevalnik

- Vezje za seštevanje
- Seštej dve binarni števili
- Vhod: dve nepredznačeni N-bitni dvojiški števili
- Izhod: eno nepredznačeno N-bitno dvojiško število (vsota)

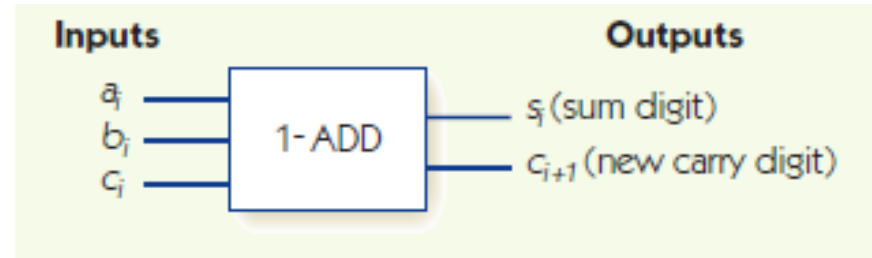
- Primer:

$$\begin{array}{r} \\ \\ + \\ \hline \\ \\ \\ \\ \\ \end{array}$$

- Začni z enobitnim seštevalnikom (1-ADD)
- Razširi ga za N bitov

Primer: Seštevalnik

- Enobitni seštevalnik 1-ADD
 - vhod: dve števili in prenos
 - izhod: vsota in prenos



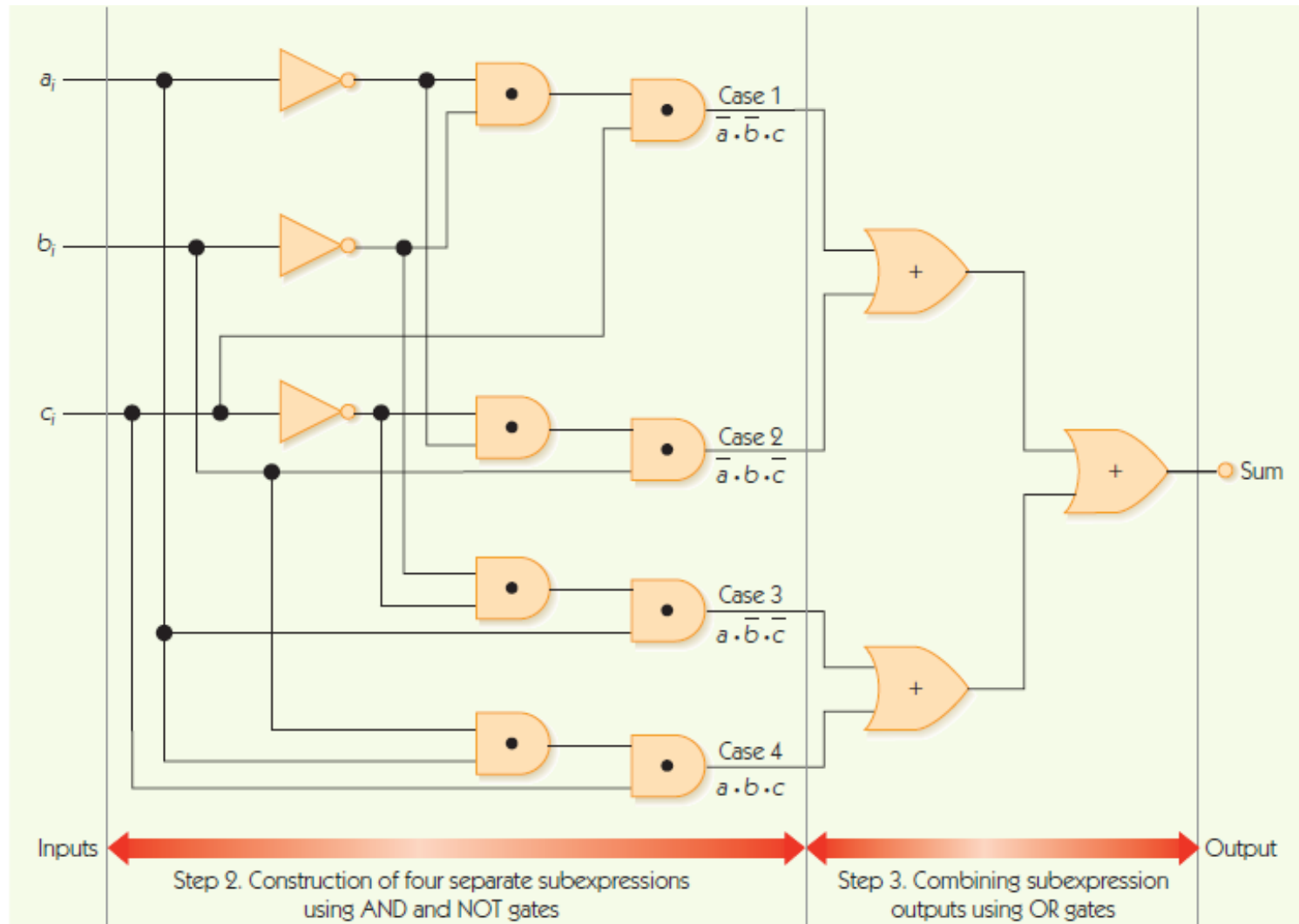
- Tabela pravilnosti

Inputs			Outputs	
a_i	b_i	c_i	s_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

- Izraza za seštevek in prenos:
 - $s_i: (\sim a_i \cdot \sim b_i \cdot c_i) + (\sim a_i \cdot b_i \cdot \sim c_i) + (a_i \cdot \sim b_i \cdot \sim c_i) + (a_i \cdot b_i \cdot c_i)$
 - $c_{i+1}: (\sim a_i \cdot b_i \cdot c_i) + (a_i \cdot \sim b_i \cdot c_i) + (a_i \cdot b_i \cdot \sim c_i) + (a_i \cdot b_i \cdot c_i)$

Primer: Seštevalnik

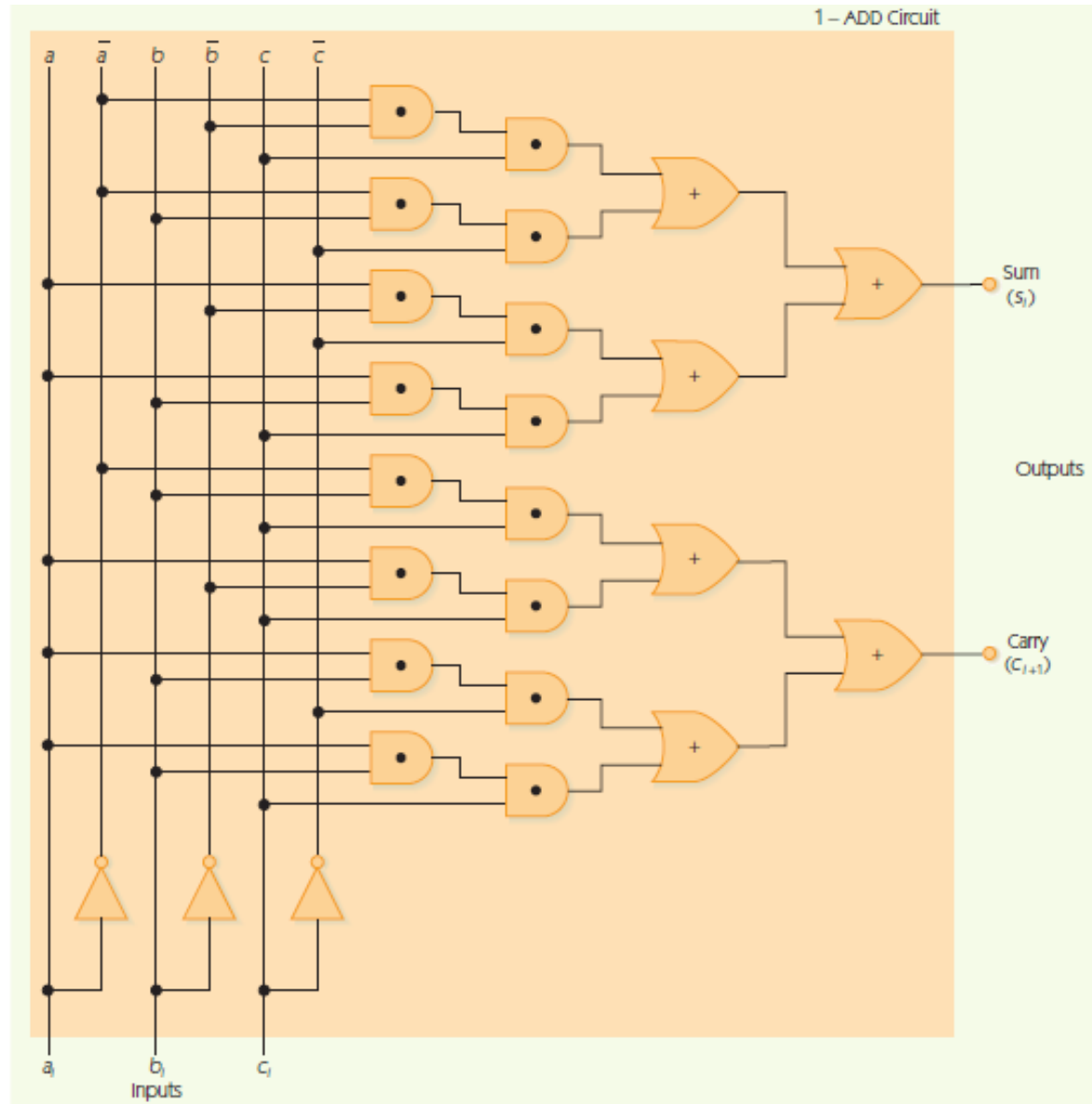
- Vezje za izhod s_i (vsota) v 1-ADD



Primer: Seštevalnik

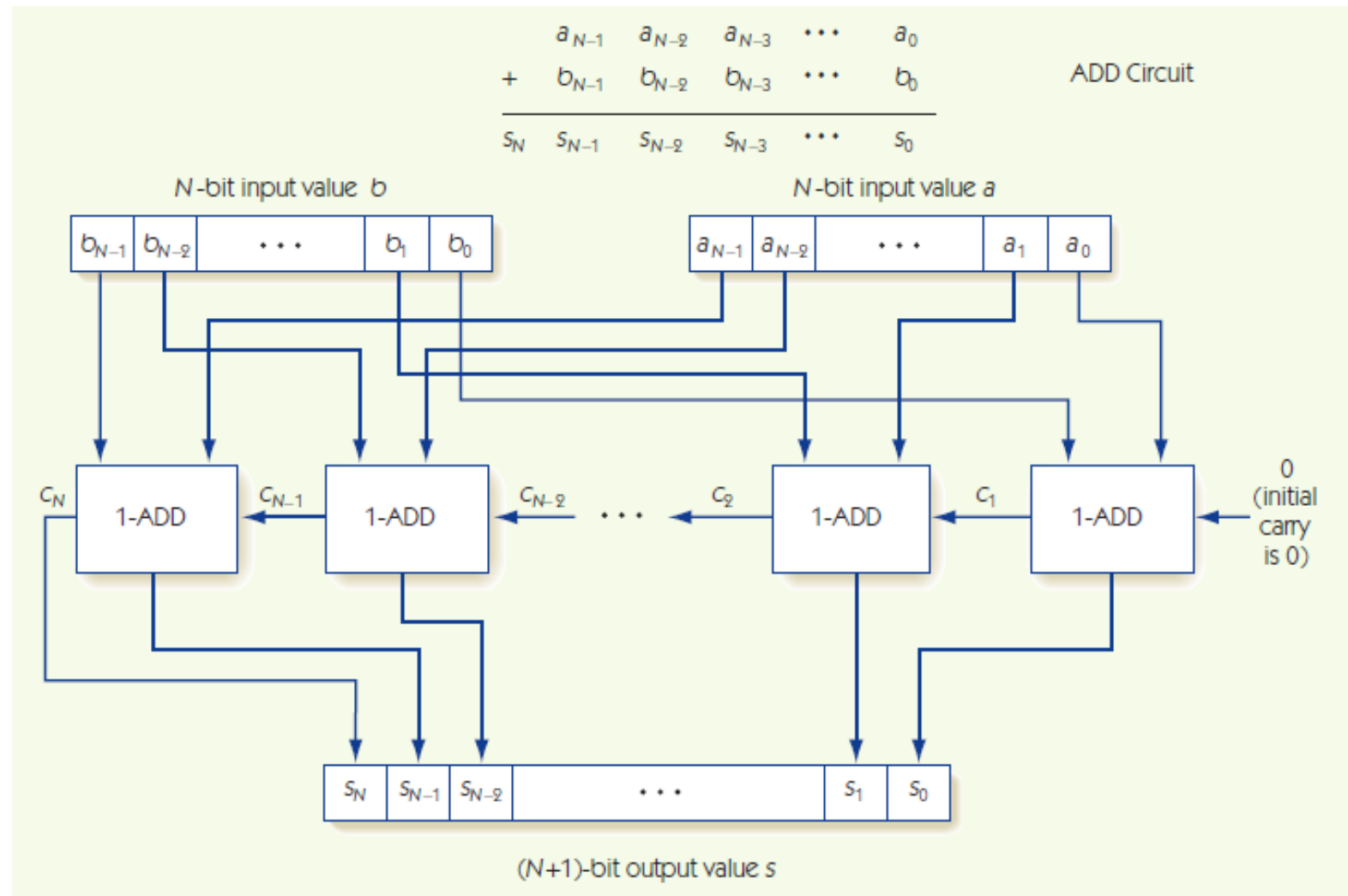
- Celotno vezje 1-ADD

- S_i
- C_{i+1}



Primer: Seštevalnik

- Celotno vezje – povežemo posamezne seštevalnike 1-ADD



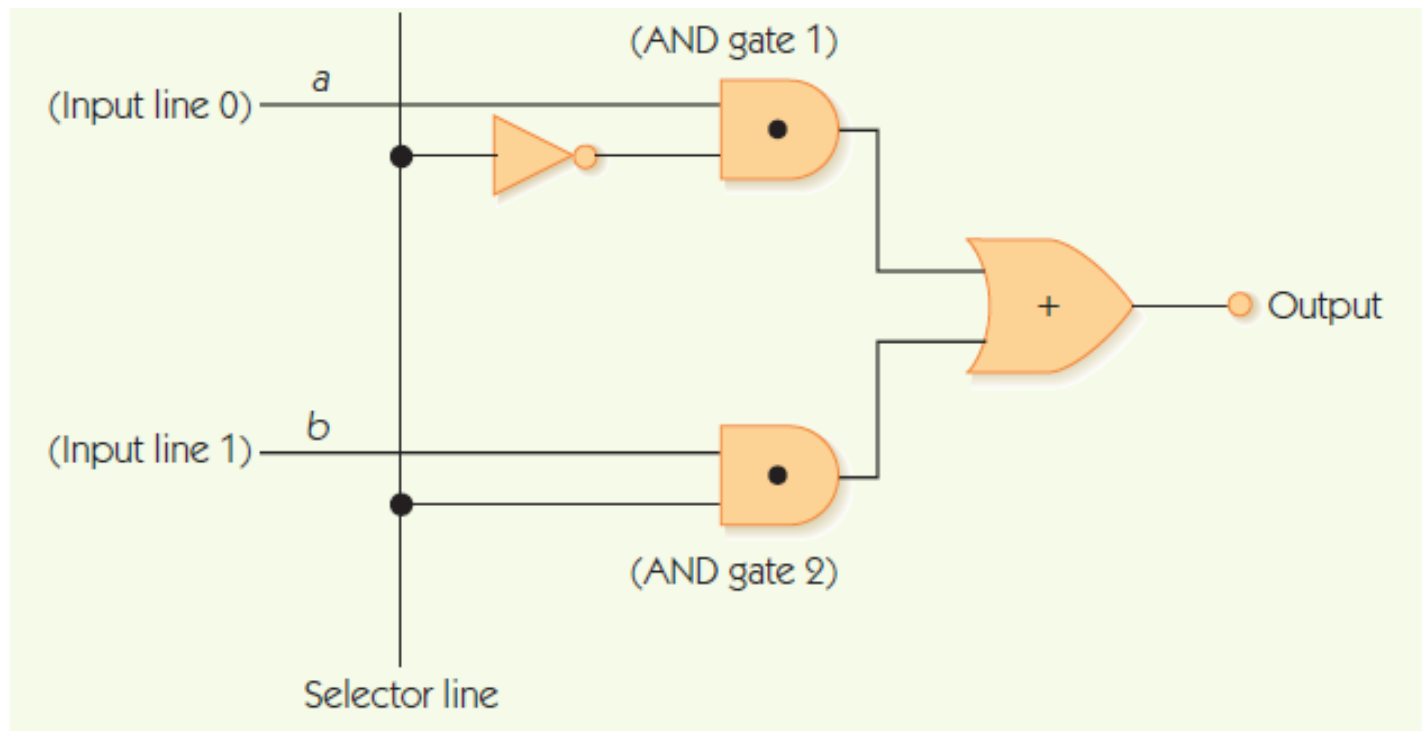
Kontrolna vezja

- Kontrolna verzija sprejemajo odločitve, določajo vrstni red operacij, izbirajo vrednosti podatkov
- Izbirnik
 - Multiplekser
 - izbere enega izmed mnogih vhodov
- Dekodirnik
 - Dekoder
 - pošlje na izhod en signal izbran z vhodom

Izbirnik

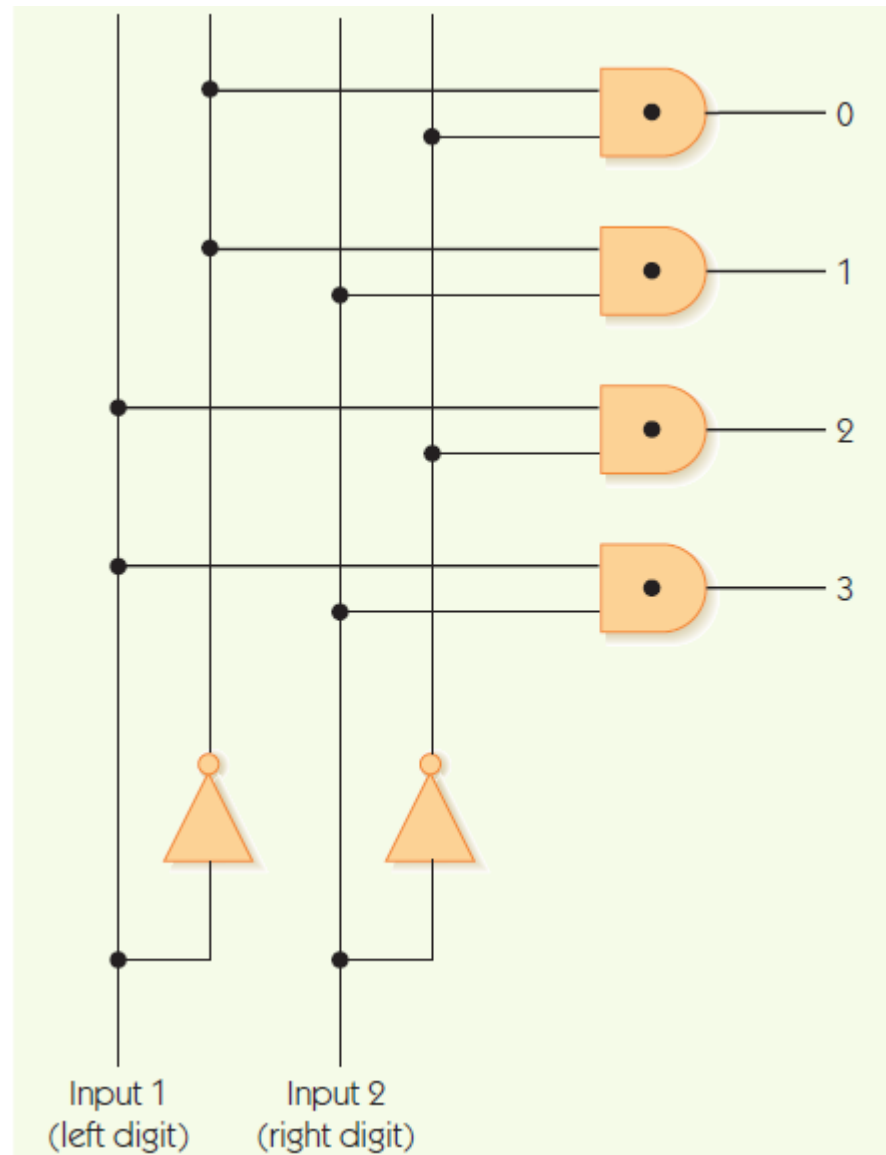
- Izbere enega izmed mnogo vhodov
 - 2^N vhodnih linij
 - N izbirnih linij
 - 1 izhodna linija
- Vsaka vhodna linija ustreza vzorcu vrednosti na izbirnih linijah
- Vrednost izbranega vhoda prenese na izhod

- Primer:
2-vhodni
MUX



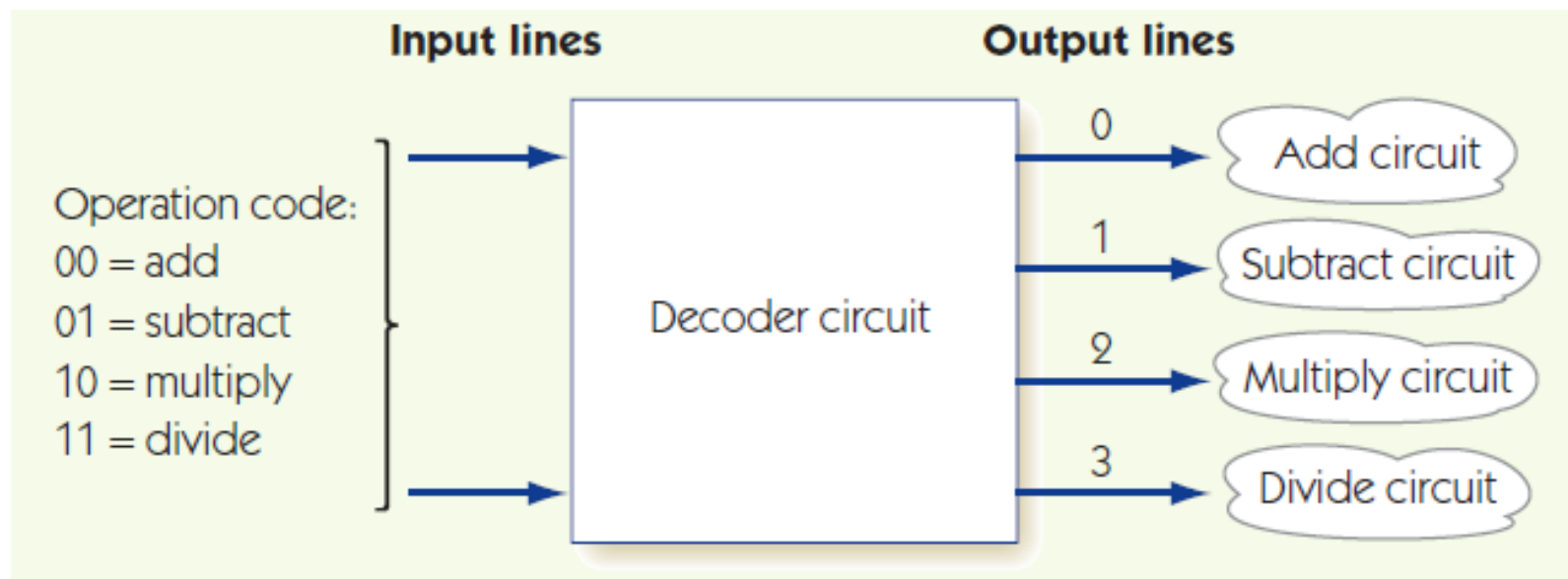
Dekodirnik

- Pošlje signal na en izhod izbran z vrednostjo na vhodnih linijah
- Vsaka izhodna linija ustreza vzorcu vrednosti na vhodnih linijah
- Izbrani izhod ima vrednost 1, vsi ostali izhodi pa 0
- Primer:
 - dekodirnik 2-v-4



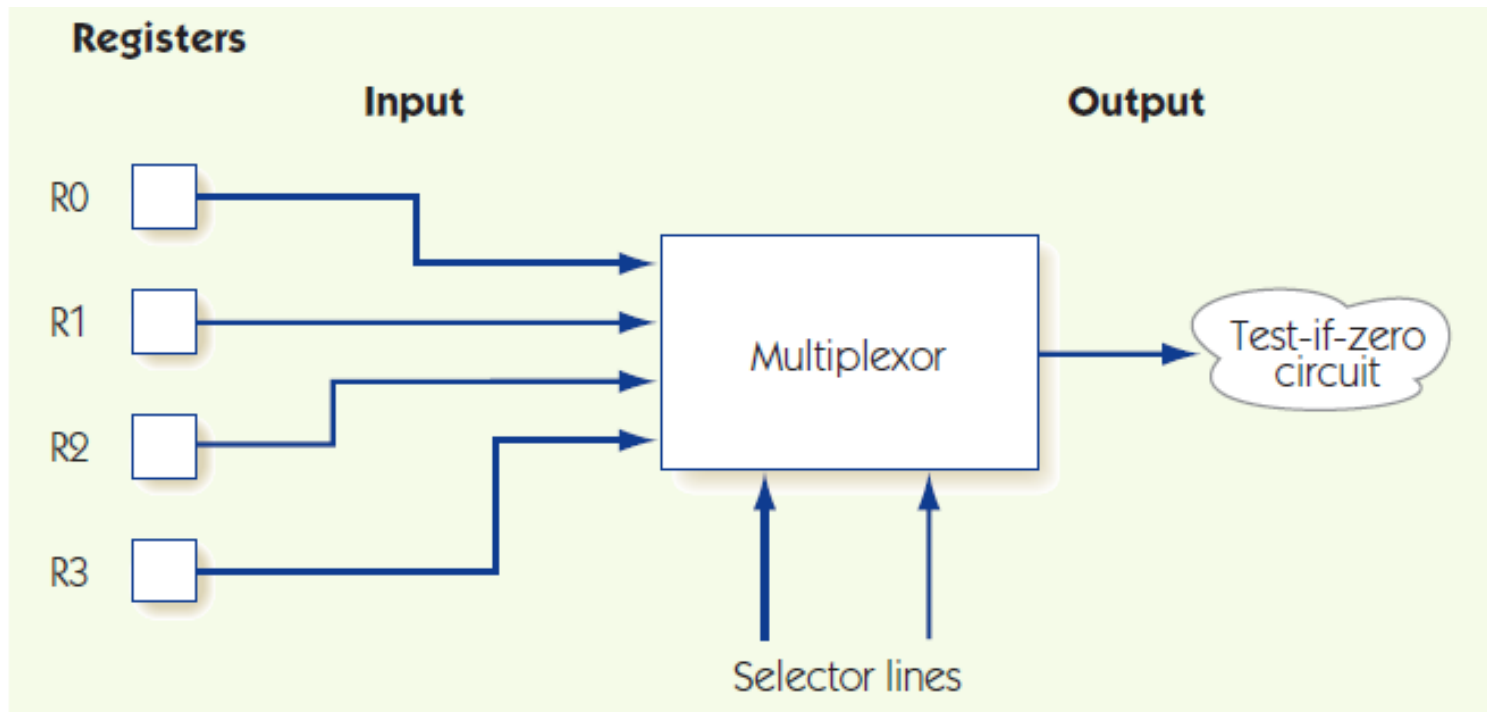
Uporaba dekodirnika

- Izbira izhoda glede na vzorec na vhodu
- Primer:
 - Izberi en aritmetični ukaz, podan s kodo za ta ukaz
 - Koda aktivira izhodno linijo, le-ta aktivira ustrezno aritmetično vezje



Uporaba izbirnika

- Izbira vrednosti podatka iz množice, kot to določa vzorec na izbirnih linijah
- Veliko podatkov pride v izbirnik, samo eden izbrani pride ven
- Primer: izbira pravega registra



- Računalniki uporabljajo binarno predstavitev, ker je robustna
- Veliko vrst podatkov lahko predstavimo (vsaj aproksimiramo) v digitalni obliki z binarnimi števili
- Booleova logika opisuje kako zgraditi in delati z izrazi, ki so bodisi pravilni ali nepravilni
- S tranzistorji lahko zgradimo logična vrata, ki se obnašajo kot Booleovi operatorji
- Z logičnimi vrati lahko gradimo vezja, ki ustrezajo Booleovim izrazom
- Vsota produktov je algoritem za načrtovanje vezij
 - iz specifikacije zgradi vezje
- Lahko zgradimo vezja za osnovne algoritmične naloge:
 - primerjave (primerjalnik)
 - aritmetiko (seštevalnik)
 - nadzor (izbirnik, dekodirnik)