

a1.2 BINARNE OPERACIJE - Bulova algebra

U matematici, binarna operacija ili dijadička operacija predstavlja pravilo za kombinovanje dva elementa (koji se nazivaju operandi) da bi se proizveo treći element. Binarna operacija je operacija ariteta dva (aritet = broj operanada).

Formalno posmatrano, binarna operacija na skupu S , je preslikavanje elemenata dekartovog proizvoda $S \times S$ u skup S :

$$f: S \times S \rightarrow S$$

pri čemu je $S \in (X_1, X_2, \dots, X_n)$ domen operacije a n je aritet operacije. Pošto je rezultat izvođenja operacije na paru elemenata S ponovo element S (kodomen je identičan domenu), operacija se naziva zatvorenom (ili internom) binarnom operacijom nad S .

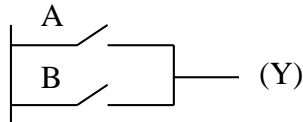
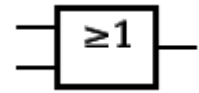
Ukoliko su elementi X_1 i X_2 bivalentni, odnosno ukoliko sadrže samo dva člana (vrednosti), binarna operacija se naziva Bulovom operacijom (pa shodno tome binarna algebra Bulovom algebrom), po matematičaru koji je formulisao njene osnovne postavke: Boole, George (1854), *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*. Ali, ovde ne treba zaboraviti ime velikog naučnika, matematičara tvorca diferencijalnog računa, ali takodje i čoveka koji je medju prvima radio na algebarskim formalizmima u oblasti logike, Gottfried Wilhelm Leibniz, koji je živeo na prelazu izmedju sedamnaestog i osamnaestog veka (1646 - 1716, Hanover, Nemačka).

Osnovu Bulove algebre sačinjava skup od 3 logičke operacije: logičko sabiranje (disjunkcija), logičko množenje (konjukcija) i negacija ili komplement. Logičko sabiranje i logičko množenje su operacije diadičkog tipa, odnosno podrazumevaju kombinovanje dve promenljive da bi generisale treću promenljivu. Negacija je unarna operacija.

Logičko sabiranje – matematička operacija disjunkcije – Ili logička funkcija, odnosno T-norma

$$A + B = Y \dots Y = \max(A, B)$$

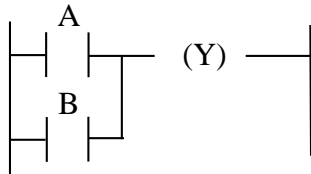
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



simbol prema
ANSI/IEEE Std 91-1984
and its supplement
ANSI/IEEE Std 91a-1991

simbol prema
IEC 60617-12

Standard IEC 60617-12 u Evropi obuhvaćen kroz EN 60617-12:1999



Description:

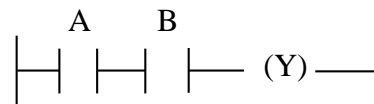
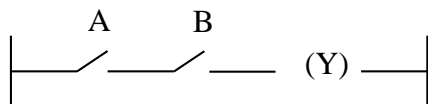
IEC 60617-1 Part 1: General information, general index. Cross-reference tables IEC 60617-2 Part 2: Symbol elements, qualifying symbols and other symbols having general application IEC 60617-3 Part 3: Conductors and connecting devices IEC 60617-4 Part 4: Basic passive components IEC 60617-5 Part 5: Semiconductors and electron tubes IEC 60617-6 Part 6: Production and conversion of electrical energy IEC 60617-7 Part 7: Switchgear, controlgear and protective devices IEC 60617-8 Part 8: Measuring instruments, lamps and signalling devices IEC 60617-9 Part 9: Telecommunications: Switching and peripheral equipment IEC 60617-10 Part 10: Telecommunications: Transmission IEC 60617-11 Part 11: Architectural and topographical installation plans and diagrams IEC 60617-12 Part 12: Binary logic elements IEC 60617-13 Part 13: Analogue elements

Pri čemu uvek važi: $A + 0 = A$, $A + 1 = 1$, $A + A = A$ (idempotencija), $A + \neg A = 1$

Logičko množenje – matematička operacija konjukcije – I logička funkcija , odnosno T-konorma

$$A \cdot B = Y \dots Y = \min(A, B)$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

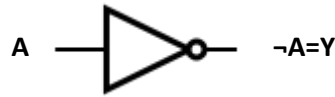


pri čemu uvek važi: $A \cdot 0 = 0$, $A \cdot 1 = A$, $A \cdot A = A$ (idempotencija), $A \cdot \neg A = 0$

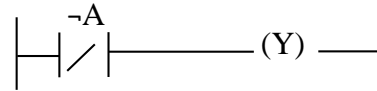
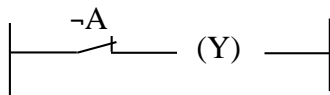
Komplement – matematička operacija negacije

$$\neg A = Y$$

A	Y
0	1
1	0



Pri čemu uvek važi: $\neg(\neg A) = A$



Zakoni i teoreme Bulove algebre

Zakon komutacije: $A + B = B + A$

$$A \cdot B = B \cdot A$$

Zakon asocijacije: $A + (B + C) = (A + B) + C$

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

Zakon distribucije: $A \cdot (B + C) = A \cdot B + A \cdot C$

$$A + B \cdot C = (A + B) \cdot (A + C)$$

Zakon apsorpcije: $A + A \cdot B = A$ dokaz: $A + A \cdot B = A \cdot (1 + B) = A \cdot (1) = A$

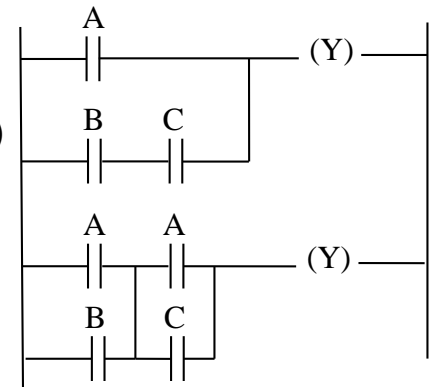
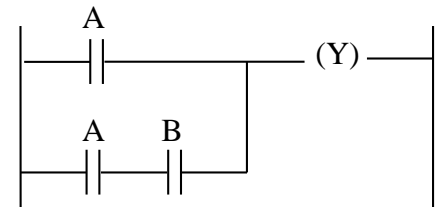
$$A \cdot (A + B) = A$$
 dokaz: $A \cdot (A + B) = A \cdot A + A \cdot B = A + A \cdot B = A \cdot (1 + B) = A$

Zakon eliminacije (adsorpcije): $A + \neg A \cdot B = A + B$

$$A \cdot (\neg A + B) = A \cdot B$$

$$A + A'B = (A + A')(A + B) = 1(A + B) = A + B \text{ (pošto je } A + BC = (A + B)(A + C))$$

$$A(A' + B) = AA' + AB = 0 + AB = AB$$



De Morganova teorema

$$\neg(A + B) = \neg A \cdot \neg B$$

$$\neg(A \cdot B) = \neg A + \neg B$$

De Morganova teorema ima veliki praktični značaj u sintezi logičkih funkcija i logičkih kola. U zavisnosti od primenjene tehnologije, često je u praksi jednostavnije realizovati NI i NILI kolo nego njihove afirmativne ekvivalente ili funkciju izraziti u disjunktivnoj ili konjuktivnoj formi. De Morganova teorema pruža mogućnost da se kompletna Bulova algebra može svesti na dva

alternativna para: komplement i logičko sabiranje, ili komplement i logičko množenje. De Morganova teorema dozvoljava prevodjenje afirmativne u negirane, odnosno komplementarne logički ekvivalentne forme.

Širi kontekst – generalizacija binarne algebre

Prethodno razmatrani skup binarnih funkcija (I, ILI, NE), odnosno Bulovih binarnih logičkih operatora, nije kompletan. On se može generalizovati, polazeći od univerzalne algebre koja na sistematičan način ispituje algebarske strukture i operacije. Odatle se izvodi potpuni skup Bulovih binarnih operacija drugog ariteta, odnosno ${}^n f_i = {}^n f_i(x_0, \dots, x_{n-1})$, $n=2$, odakle sledi šesnaest binarnih funkcija drugog reda ${}^2 f_i(x_0, x_1)$, $i=(0, 2^{2^n}-1) = (0, 15)$, odnosno 16 binarnih funkcija. Taj potpuni, kanonični skup funkcija se zbog preglednosti navodi tabelarno, i glasi:

x_0	x_1	${}^2 f_0$	${}^2 f_1$	${}^2 f_2$	${}^2 f_3$	${}^2 f_4$	${}^2 f_5$	${}^2 f_6$	${}^2 f_7$	${}^2 f_8$	${}^2 f_9$	${}^2 f_{10}$	${}^2 f_{11}$	${}^2 f_{12}$	${}^2 f_{13}$	${}^2 f_{14}$	${}^2 f_{15}$
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Odmah se uočava da ranije definisana konjunkcija ili presek, $x_0 \wedge x_1$, u ovoj kanoničnoj seriji se pojavljuje kao operacija označena sa ${}^2 f_8$, disjunkcija ili unija, $x_0 \vee x_1$, kao operacija označena sa ${}^2 f_{14}$, dok se operacija isključivo ILI, odnosno simetrična razlika, $x_0 \oplus x_1$, pojavljuje kao operacija označena sa ${}^2 f_6$.

Operacija negacije $\neg x_0$, odnosno operacija binarnog komplementa je unarna operacija i ona se pojavljuje kao operacija označena sa ${}^1 f_1$ u kanoničnom skupu operacija prvog ariteta:

x_0	${}^1 f_0$	${}^1 f_1$	${}^1 f_2$	${}^1 f_3$
0	0	1	0	1
1	0	0	1	1

takodje i kao ${}^2 f_5$ skupu operacija drugog ariteta.

U cilju formalne kompletnosti, prethodno navedene tablice zahtevaju uvodjenje operacije nultog ariteta, odnosno operacije konstanti. Ta tablica glasi:

${}^1 f_0$	${}^1 f_1$
0	1

Dodatne informacije o binarnoj algebri na sledećim sajtovima:

- https://en.wikipedia.org/wiki/Boolean_algebras_canonically_defined
- https://en.wikipedia.org/wiki/Logic_gate
- https://en.wikipedia.org/wiki/Boolean_function
- https://en.wikipedia.org/wiki/Boolean_algebra
- [https://en.wikipedia.org/wiki/Boolean_algebra_\(structure\)](https://en.wikipedia.org/wiki/Boolean_algebra_(structure))
- https://en.wikipedia.org/wiki/Logical_conjunction

Primer TERNARNE odnosno TRIVALENTNE logike, samo kao informacija za one koji žele da saznaju više:

U ternarnoj odnosno trovalentnoj logici, logička promenljiva uzima tri vrednosti:

prva vrednost = F (False), odnosno negacija, ili simbol -1, ili samo simbol -

druga vrednost = T (Truth), odnosno afirmacija, ili simbol +1, ili samo simbol +

treća vrednost = U (Undefined), neodređeno, ili nepoznato, ili možda (Maybe), ili simbol oznaka 0, ili simbol 1/2, ili simbol ?

Zapravo, ternarna osnova daje više mogućih sistema označavanja, koji se dalje mogu povezati sa tehnologijom njihove praktične primene:

Ternary computing deals with three discrete states, but the ternary digits themselves can be defined in different ways, according to Connelly:

Unbalanced Trinary — {0, 1, 2}

Fractional Unbalanced Trinary — {0, 1/2, 1}

Balanced Trinary — {-1, 0, 1}

Unknown-State Logic — {F, ?, T}

Trinary Coded Binary — {T, F, T}

“Perhaps the prettiest number system of all is the balanced ternary notation,” writes Donald Knuth in an excerpt from his book “The Art of Computer Programming.”

Odatle dalje slede kombinacione tablice. Za razumevanje, I i ILL operatora, dovoljno je da se za I primeni funkcija min, a za ILL funkcija max (sve postaje vrlo logično kada se primeni balansirani ternarni model).

NOT(A)		AND(A, B)			OR(A, B)			
A	¬A	A ∧ B	B		A ∨ B	B		
A	¬A		F	U	T	F	U	T
F	T		F	F	F	F	F	F
U	U	A	U	F	U	U	U	U
T	F		T	F	U	T	T	T

Po analogiji sa bitom koji je jedinica informacionog sadržaja binarnih sistema, kod ternarnih sistema jedinica informacionog sadržaja se naziva trit.

Inače, ternarni brojni sistem ima tri cifre: (0, 1, 2), a osnova brojnog sistema je 3.

Ternarni brojni sistem je matematička osnova za izgradnju ternarnih računskih mašina, koje za razliku od binarnih imaju niz prednosti u odnosu na binarne, zbog veće reprezentativnosti ternarnog brojnog sistema, odnosno takozvane informacione gustine (data density). Ovakve mašine zahtevaju drugačiju fizičku realizaciju u slučaju poluprovodničke tehnologije. Kada se kao tehnološka osnova koristi optika, onda se tri stanja mogu modelirati tako što se nulto stanje izvodi kao odsustvo svetlosti, a druga dva stanja korišćenjem polarizovane svetlosti, medjusobno rotirane za 90 stepeni (dve optički ortogonalne ravni). Ternarni sistem kao osnova kompjutera je nedovoljno istražen. Nesumnjivo ima veći kapacitet od binarnog, a tehnološki prihvatljiviji od decimalnog.

SETUN (ime prvog funkcionalnog ternarnog računara razvijenog pedesetih godina u SSSRu, kao univerzitetski projekat koji je realizovan 1958 godine pod vodjstvom briljantnog matematičara Николай Петрович Брусенцов (Nikolay Petrovich Brusentsov), (dobio naziv po rečici Сетунь koja protiče pored Moskve) used balanced ternary logic, {-1, 0, 1} as noted above. That's the common approach to ternary, and it's also found in the work of Jeff Connelly and Jessica Tank. "Perhaps the prettiest number system of all is the balanced ternary notation," writes Donald Knuth in an excerpt from his book "The Art of Computer Programming."

Brian Hayes is also a big fan of ternary. "Here I want to offer three cheers for base 3, the ternary system. ... They are the Goldilocks choice among numbering systems: When base 2 is too small and base 10 is too big, base 3 is just right."

One of Hayes' arguments for the virtues of base-3 is that it is the closest numbering system to base-e, "the base of the natural logarithms, with a numerical value of about 2.718." With mathematical prowess, the essayist Hayes explains how base-e (if it were practical) would be the most economical numbering system. It is ubiquitous in nature. And I clearly remember these words from Mr. Robertson, my high school chemistry teacher: "God counts by e." / $e^{i\pi} + 1 = 0$

U prilog trovalentne logike i uopšte kvantizacije neke veličine ide i činjenica da mi najčešće opisujemo veličine kroz tri vrednosti: malo, srednje, veliko. Može i kroz dve, malo i veliko. Može i sa pet lingvističkih promenljivih: malo, srednje malo, srednje, srednje veliko, veliko. Ali, priroda je pokazala da je dobar kompromis između preciznosti i kompleksnosti, ternarna diskretizacija. Ova tvrdnja se lako dokazuje u jednom širem kontekstu polivalentne logike – FAZI Logike. Ali, to je posebna tema koja bitno prevazilazi tematski okvir ovog kursa.

Using Kleene and Priest Logic Rules to Create Ternary Logic Tables

<https://community.wolfram.com/groups/-/m/t/1140640>

<https://www.techopedia.com/why-not-ternary-computers/2/32427>

a1.3 KOMBINACIONI SISTEMI

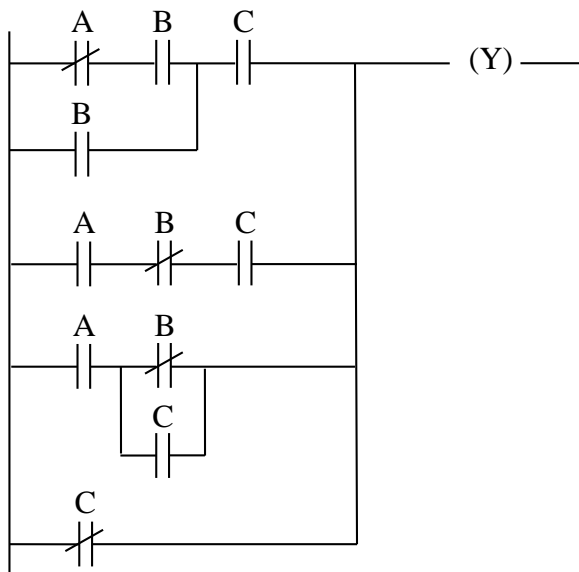
Osnovnim logičkim operatorima formiraju se logičke funkcije kombinacionog tipa. Logička vrednost ovih funkcija zavisi samo od trenutne vrednosti ulaznih logičkih promenljivih. Kombinacioni sistemi su statički sistemi.

a1.3.1 Logičke funkcije

Primenom osnovnih logičkih operatora nad proizvoljnim skupom nezavisnih logičkih promenljivih formiraju se složene logičke funkcije. Primer složene logičke funkcije izvedene nad skupom od tri logičke promenljive:

$$y = f(A, B, C) = (\bar{A}B + B)C + A\bar{B}C + A(\bar{B} + C) + \bar{C}$$

Ovakve složene logičke funkcije nazivaju se polinomalnim formama.



Sa teorijskog i praktičnog aspekta značajna je jedna posebna klasa složenih logičkih funkcija koje nastaju ograničenom primenom logičkih operatora. To su takozvane kanonične normalne forme, koje se sreću u svoja dva osnovna oblika: 1) disjunktivna normalna forma i 2) konjuktivna normalna forma.

1) Disjunktivna normalna forma (DNF):

Ova kanonična forma sastoji se iz **sume članova dobijenih isključivo primenom I operatora**:

$$y = f(A, B, C) = ABC + AC + \overline{A}BC + \overline{B}C$$

Ako je ova suma sastavljena isključivo od **članova potpune forme**, odnosno od članova koji se sastoje iz (a) svih promenljivih nekog usvojenog skupa logičkih promenljivih, (b) nekomplementiranih ili komplementiranih, (b) bez ponavljanja, onda se radi o savršenoj disjunktivnoj normalnoj formi SDNF koja se zapisuje kao:

$$y = f(X_1, \dots, X_n) = \sum_{i=0}^{2^n-1} b_i P_i \quad (4)$$

gde koeficijent b_i ima vrednost 0 ili 1, čime se određuje prisutnost određenog člana u logičkoj funkciji. Potpuni član dobijen primenom I operatora naziva se minterm. SDNF kanonična normalna forma je svaka složena logička funkcija koja je iskazana kao logička suma mintermova.

Svaka logička funkcija može se prevesti u njen SDNF oblik. Postupak prevodjenja u SDNF oblik ostvaruje se tako što se funkcija primenom osnovnih logičkih operatora dovodi na DNF oblik, a zatim se svaki od članova dovodi na potpunu formu primenom sledeće transformacije:

$$AB = AB(C + \overline{C}) = ABC + AB\overline{C}$$

pri čemu važi:

$$(C + \overline{C}) = 1 \text{ i } AB \cdot 1 = AB \text{ odnosno:}$$

$$AB = AB \cdot 1 = AB \cdot (C + \overline{C}) = ABC + AB\overline{C}$$

Primenom ove transformacije, prethodno navedena logička jednačina u DNF obliku se prevodi na njen SDNF oblik:

$$\begin{aligned} y = f(A, B, C) &= ABC + AC + \overline{A}BC + \overline{B}C = \\ &= ABC + AC(B + \overline{B}) + \overline{A}BC + (A + \overline{A})\overline{B}C = \\ &= ABC + ABC + \overline{A}BC + \overline{A}B\overline{C} + ABC + \overline{A}B\overline{C} = ABC + \overline{A}BC + \overline{A}B\overline{C} + \overline{A}B\overline{C} \end{aligned}$$

Ukoliko se mintermovi zamene njihovim decimalnim ekvivalentom primenjujući sledeće preslikavanje:

\overline{CBA}	0
$\overline{C}BA$	1
$\overline{C}\overline{B}A$	2
$\overline{C}B\overline{A}$	3
$C\overline{B}A$	4
$C\overline{B}\overline{A}$	5
$CB\overline{A}$	6
CBA	7

onda se polazna funkcija može da zapiše na kompaktan način kao zbir decimalnih ekvivalenata mintermova trećeg reda:

$$y = f(A, B, C) = ABC + AC + \overline{A}BC + \overline{B}C = ABC + \overline{A}BC + \overline{A}B\overline{C} + \overline{A}B\overline{C} = \sum_3(2, 3, 5, 7)$$

2) Konjuktivna normalna forma (KNF):

Ova kanonična forma se sastoji se iz proizvoda članova dobijenih isključivo primenom ILI operatora:

$$y = f(A, B, C) = (A + B + \overline{C}) \cdot (A + C) \cdot \overline{B}$$

Ako je ovaj proizvod sastavljen samo od članova potpune forme, odnosno od članova koji se sastoje iz svih promenljivih nekog usvojenog skupa logičkih promenljivih, onda se radi o savršenoj konjuktivnoj normalnoj formi SKNF koja se zapisuje:

$$y = f(X_1, \dots, X_n) = \prod_{i=0}^{2^n-1} (b_i + P_i) \quad (5)$$

gde koeficijent b_i ima vrednost 0 ili 1, čime se određuje prisutnost određenog člana u logičkoj funkciji. Potpuni član dobijen primenom ILI operatora naziva se maxterm. SKNF kanonična normalna forma je svaka složena logička funkcija koja je iskazana kao logički proizvod maxtermova.

a1.3.2 Sinteza logičkih funkcija

Sinteza logičkih funkcija se može ostvariti na dva načina: 1) intuitivnim postupkom i 2) formalnim postupkom.

U inženjerskoj praksi, definisanje neke logičke funkcije po pravilu započinje lingvističkim opisom. Sinteza logičke funkcije je prevodjenje lingvističkog opisa u njemu ekvivalentnu binarnu logičku jednačinu.

Intuitivni postupak se primenjuje u jednostavnim slučajevima, odnosno kada je broj logičkih promenljivih mali i kada su logičke relacije u postavci zadatka očigledne. U svim ostalim slučajevima

primenjuje se formalni postupak koji je baziran na kombinacionim tablicama. Ovaj postupak se sastoji u tome da se za konkretan skup ulaznih i izlaznih logičkih promenljivih formira odgovarajuća kombinaciona tablica koja ima sledeću strukturu:

DE	X _n	...	X ₂	X ₁	f(X ₁ ,...,X _n)
0	0	...	0	0	0 / 1
1	0	...	0	1	0 / 1
...
(2 ⁿ)-1	1	...	1	1	0 / 1

Sa leve strane unose se ulazne promenljive (X_i) a sa desne izlazna promenljiva y=f(X₁,...,X_n). Vrednosti ulaznih promenljivih se unose sledeći prirodni binarni kod. U prvoj koloni upisuje se decimalni ekvivalent sa ciljem jednostavnijeg korišćenja tablice. Prema tome, svaka vrsta odgovara jednom mintermu. Za svaki minterm upisuje se vrednost izlaza logičke funkcije, 0 ili 1, u zavisnosti od zadatkom definisanih uslova. Na ovaj način formira se (2ⁿ)-1 vrsta, gde je sa n označen broj ulaznih promenljivih. SDNF oblik složene logičke funkcije koja u analitičkom obliku opisuje zadatkom definisane uslove funkcionisanja logičkog sistema formuliše se primenom relacije (4).

Primer:

Sistem za izmenu alata robota sastoji se iz (a)magacina sa dva gnezda i (b)mehaničkog izmenjivača pogonjenog pneumatskim cilindrom, koji je vezan za terminalnu ploču robota.

Na prvom gnezdu magacina postoji senzor X2 koji registruje prisustvo alata. Analogno, drugo gnezdo magacina je opremljeno senzorom X3.

Mehanički izmenjivač je opremljen senzorom X4 koji registruje prisustvo alata na terminalnoj ploči robota.

Da bi se izvršila sekvenca izmene mora da bude ispunjen sledeći uslov: (a)alat mora da bude prisutan na terminalnoj ploči robota i (b) da bar jedno od dva gnezda bude slobodno. Alternativno, sekvenca izmene je moguća i u slučaju kada alat nije prisutan na terminalnoj ploči ali je bar jedno od dva gnezda zauzeto. Sekvenca izmene aktivira se signalom X1 koji je aktivan sve dok se ne izvrši izmena.

Izvršiti sintezu upravljanja rada pneumatskog ventila mehaničkog izmenjivača Y.

Rešenje:

Formira se kombinaciona tablica

DE	X4	X3	X2	X1	Y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

Zatim se formira logička jednačina kojom se upravlja elektromagnetnim ventilom za aktuaciju pneumatskog cilindra izmenjivača alata i to u SDNF obliku:

$$y = f(X_1, X_2, X_3, X_4) = X_1 X_2 \bar{X}_3 \bar{X}_4 + X_1 \bar{X}_2 X_3 \bar{X}_4 + X_1 X_2 X_3 \bar{X}_4 + X_1 \bar{X}_2 \bar{X}_3 X_4 + X_1 X_2 \bar{X}_3 X_4 + X_1 \bar{X}_2 X_3 X_4 = \sum_4 (3,5,7,9,11,13)$$

a1.3.3 Optimizacija logičkih funkcija

SDNF oblik logičke funkcije je ekstremni slučaj u smislu kompleksnosti. Ovaj oblik logičke funkcije je moguće učiniti jednostavnijim.

Jednostavniji oblik je moguće ostvariti pre svega minimizacijom, odnosno pronalaženjem najprostijeg DNF oblika koji je ekvivalentan polaznoj logičkoj funkciji iskazanoj u SDNF obliku.

Postupak minimizacije logičkih funkcija je vrlo delikatan inženjerski problem. Opšte rešenje je postavljeno u okviru teorije konačnih automata i bazirano je na konceptu particije skupa stanja. Praktična primenljivost ovog rešenja je vrlo složena i zato su razvijeni alternativni pristupi koji su jednostavniji.

Najefikasniji pristup je baziran na takozvanoj Vejč-Karno tablici [1], [2]. Vejč-Karno tablica za slučaj tri (na primer: X_1 , X_2 i X_3) i četiri (na primer: X_1 , X_2 , X_3 i X_4) promenljive sastoji se iz 8 i 16 polja respektivno. Svako polje odgovara jednom mintermu. Mintermovi su tako raspoređeni da obezbeđuju logičku susednost. Logička susednost postoji ne samo u unutrašnjosti tablice već i po njenim obodima. U topološkom smislu, VK tablicu treba razumeti kao površ razvijenog torusa, prikazanu u ravni papira. Logička susednost je postignuta primenom Gray-ovog koda. Decimalni ekvivalenti ovako raspoređenih mintermova upisani su u polja tablice radi lakšeg korišćenja i odgovaraju kodnim grupama koje se dobijaju iz binarnih vrednosti ulaznih promenljivih navedenih uz prvu vrstu i prvu kolonu.

X_3	00	01	11	10	X_2X_1
0	0	1	3	2	
1	4	5	7	6	

X_4X_3	00	01	11	10	X_2X_1
00	0	1	3	2	
01	4	5	7	6	
11	12	13	15	14	
10	8	9	11	10	

Logička funkcija izražena u SDNF formi unosi se u tablicu tako što se prisustvo minterma označava jedinicom, a odsustvo nulom.

U posebnom slučaju, kada logički uslovi postavljenog zadatka fizički zabranjuju postojanje nekog određenog minterma (fizička realizacija konkretnog minterma ne može se ostvariti u regularnim uslovima rada sistema), takvo stanje se označava nekim trećim simbolom, na primer sa X.

Minimizacija se ostvaruje tako što se prisutni mintermovi obuhvataju konturama, pri čemu konture moraju da zadovolje uslov obuhvata (2^n) mintermova (1, 2, 4, 8, ...). **Neophodno je da se obuhvate svi prisutni mintermovi i to sa najmanjim brojem najvećih kontura.** Kontura ne sme da sadrži ni jedan nepostojeći minterm. Redundansa, odnosno prisustvo nekog minterma u više od jedne konture je dozvoljena. Konture mogu da budu uslovno diskontinualne u smislu prividnog gubitka prostorne susednosti do koje dolazi zbog svodjenja toroidne topološke forme tablice na njenu ekvivalentnu razvijenu ravansku formu.

Kada se identifikuje skup kontura kojima se obuhvataju svi mintermovi u smislu prethodno navedenih uslova, svaka kontura se zamenjuje njenom analitičkom vrednošću, tako što se zadržavaju samo one ulazne promenljive čije se vrednosti ne menjaju unutar konture. Ukoliko kontura zahvata jedan minterm, njena analitička vrednost je jednaka tom mintermu. Ukoliko kontura zahvata dva minterma, njena vrednost se iskazuje članom iz koga je eliminisana jedna ulazna promenljiva i to **ona promenljiva čija se vrednost menja unutar konture**, odnosno za onu promenljivu koja se pojavljuje u obliku ($X \cdot A + X \cdot A'$). U slučaju konture koja zahvata četiri minterma, eliminišu se dve ulazne logičke promenljive i tako dalje.

Praktična primenljivost Vejč-Karno tablica ograničena je dimenzionalnošću logičke jednačine. Njihova racionalna primena moguća je za slučajeve koji imaju do 5 ulaznih promenljivih. Ipak, najčešće se kao granica uzimaju četiri ulazne promenljive. U slučajevima koji imaju veći broj ulaznih promenljivih primenjuje se **dekompozicija složenog sistema** na podsisisteme manje dimenzionalnosti.

Za fizičku realizaciju minimalna DNF forma po pravilu **nije optimalno rešenje**. Optimalno rešenje uključuje i neke tehničke aspekte. U završnoj fazi procesa optimizacije se dobijeni minimalni DNF oblik najčešće ponovo vraća na faktorizovanu formu, koja omogućava dalje uprošćavanje kroz smanjenje potrebnog broja logičkih elemenata, ili realizaciju nekom specifičnom kombinacijom ili vrstom logičkih elemenata.

Primer: Za slučaj prethodno razmatranog primera izmenjivača alata robota, važi sledeće:

X_4X_3	00	01	11	10	X_2X_1
00	0	1	3	2	$k_1 = x_4' x_2 x_1$
01	4	5	7	6	
11	12	13	15	14	$k_2 = x_3 x_2' x_1$
10	8	9	11	10	$k_3 = x_4 x_3' x_1$

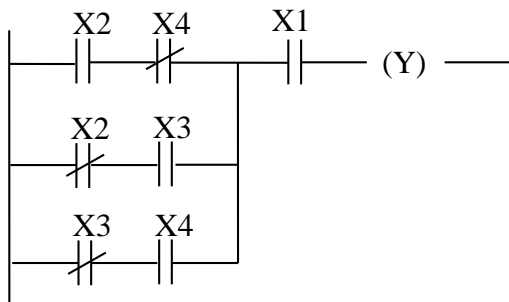
$$y = f(X_1, X_2, X_3, X_4) = X_1 X_2 \bar{X}_3 \bar{X}_4 + X_1 \bar{X}_2 X_3 \bar{X}_4 + X_1 X_2 X_3 \bar{X}_4 + X_1 \bar{X}_2 \bar{X}_3 X_4 + X_1 X_2 \bar{X}_3 X_4 + X_1 \bar{X}_2 X_3 X_4 = \sum_3 (3, 5, 7, 9, 11, 13)$$

Svi mintermovi mogu da se obuhvate sa tri konture drugog reda, odnosno, polazna SDNF forma svodi se na njoj ekvivalentnu minimalnu DNF:

$$y_{\min} = f(X_1, X_2, X_3, X_4) = k_1 + k_2 + k_3 = X_1 X_2 \bar{X}_4 + X_1 \bar{X}_2 X_3 + X_1 \bar{X}_3 X_4$$

Daljim algebarskim uobličavanjem minimalne DNF dobija se još jednostavnija formulacija polazne logičke jednačine:

$$y_{\min 1} = f(X_1, X_2, X_3, X_4) = X_1 (X_2 \bar{X}_4 + \bar{X}_2 X_3 + \bar{X}_3 X_4)$$



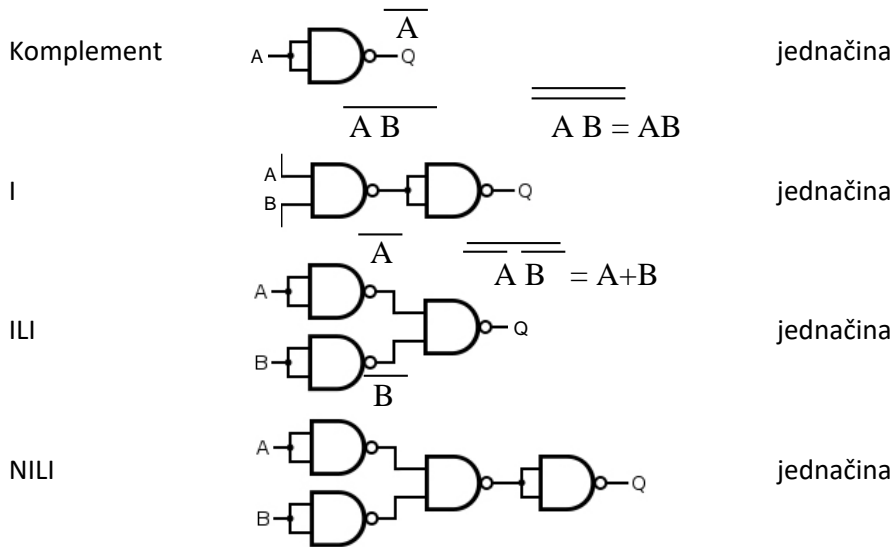
Postupak optimizacije nije samo značajan za izgradnju logičkih (prekidačkih) mreža, već on ima značaj i kod programiranja mikrokontrolera, gde u određenim situacijama ograničeni memorijski resursi ili zahtevi brzine rada nameću potrebu generisanja minimalnog programskog koda za realizaciju određene logičke funkcije. Minimalni programski kod istovremeno pojednostavljuje otkrivanje grešaka u kodiranju ili postavci problema. Sa aspekta programiranja, optimalno rešenje je minimalna DNF forma.

1. Edward W. Veitch, 1952, "A Chart Method for Simplifying Truth Functions", Transactions of the 1952 ACM Annual Meeting, ACM Annual Conference/Annual Meeting "Pittsburgh", ACM, NY, pp. 127-133.
2. Maurice Karnaugh, November 1953, The Map Method for Synthesis of Combinational Logic Circuits, AIEE Committee on Technical Operations for presentation at the AIEE summer General Meeting, Atlantic City, N. J., June 15-19, 1953, pp. 593-599.

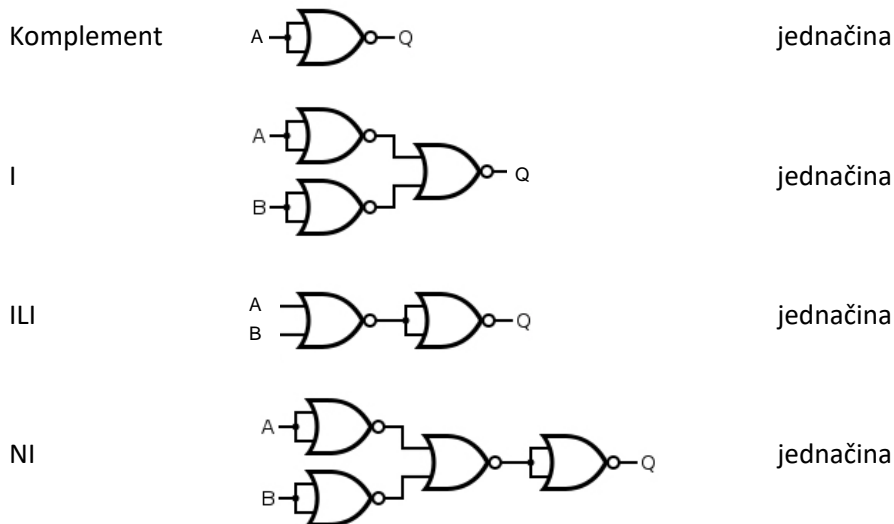
a1.3.4 Konverzija logičkih funkcija

Pod konverzijom logičke funkcije podrazumevamo njeno svodjenje na ekvivalentni oblik koji se sastoji iz određene vrste logičkih operatora. Najčešće su to NI ili NILI operatori, odnosno svodjenje skupa osnovnih logičkih operatora na jedan univerzalni operator.

NI operator kao univerzalni logički operator:



NILI operator kao univerzalni logički operator:



In practice, **the cheapest gate to manufacture** is usually the NAND gate. [Charles Sanders Peirce](#) (1880) showed that NAND gates alone (or alternatively NOR gates alone) can be used to reproduce the functions of all the other logic gates, but his work on it was unpublished until 1935. The first published proof was by [Henry M. Sheffer](#) in 1913.

Na primer, neka je data logička funkcija iskazana minimalnim skupom logičkih operatora, I / ILI /

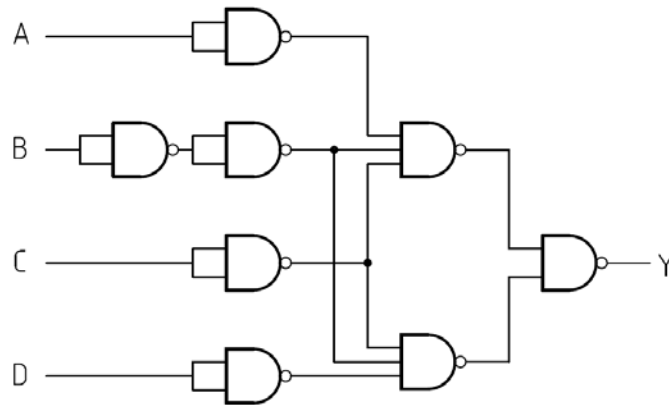
NE:

$$y = f(A, B, C, D) = \overline{A} \overline{B} \overline{C} + \overline{B} \overline{C} \overline{D}$$

Potrebno je konvertovati ovu funkciju u ekvivalentan oblik koji se može realizovati samo NI kolima.
 Primenom De Morganove teoreme važi:

$$y = \overline{\overline{\overline{ABC} + \overline{BCD}}} = \overline{\overline{ABC} \cdot \overline{BCD}}$$

Dobijena logička funkcija prikazana je na slici ispod.



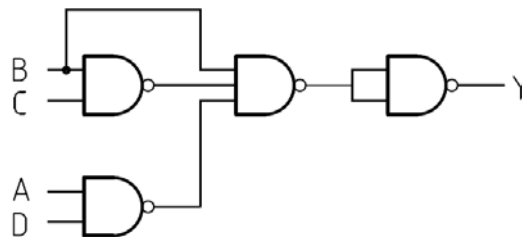
Povoljnije rešenje se može dobiti ukoliko se polazna funkcija dovede na sledeći oblik:

$$y = f(A, B, C, D) = \overline{ABC} + \overline{BCD} = \overline{BC}(\overline{A} + \overline{D})$$

odakle dalje sledi:

$$y = \overline{\overline{\overline{BC}(\overline{A} + \overline{D})}} = \overline{\overline{BC} + \overline{\overline{A} + \overline{D}}} = \overline{\overline{BC} + AD} = \overline{BC} \cdot \overline{AD} = B \cdot \overline{BC} \cdot \overline{AD}$$

čime se dolazi do značajno jednostavnijeg rešenja, jer se eliminiše inverzija ulaznih promenljivih i umesto toga invertuje izlazna promenljiva.



a1.3.5 Kombinacioni digitalni moduli

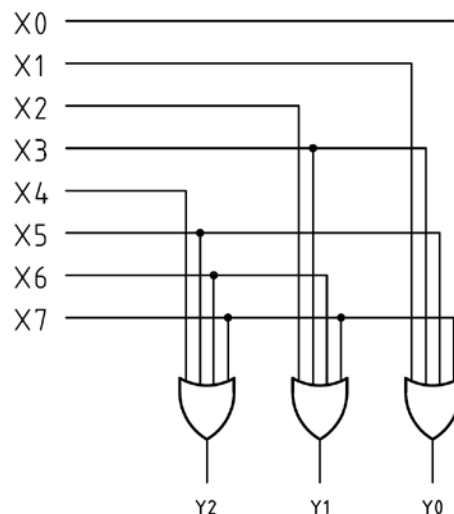
a1.3.5.1 Koder (na engleskom: encoder)

Koder je kombinaciona digitalna mreža sa više ulaza i više izlaza koja je tako sintetisana da za svaku ulaznu promenljivu generiše jednoznačni izlaz koji je kombinacija ulaznih promenljivih. Koderi se primenjuju na svim ulazima jednog digitalnog sistema gde se iz analognog fizičkog domena prelazi u apstraktni digitalni domen. Tipičan primer je numerička ili alfanumerička tastatura koja u svom sklopu mora da ima koder preko koga se obezbedjuje da se za svaki aktivirani taster na izlazu generiše jedinstveni binarni kod, na primer ASCII kod koji odgovara tom tasteru.

Primer potpunog kodera sa osam ulaznih i tri izlazna kanala:

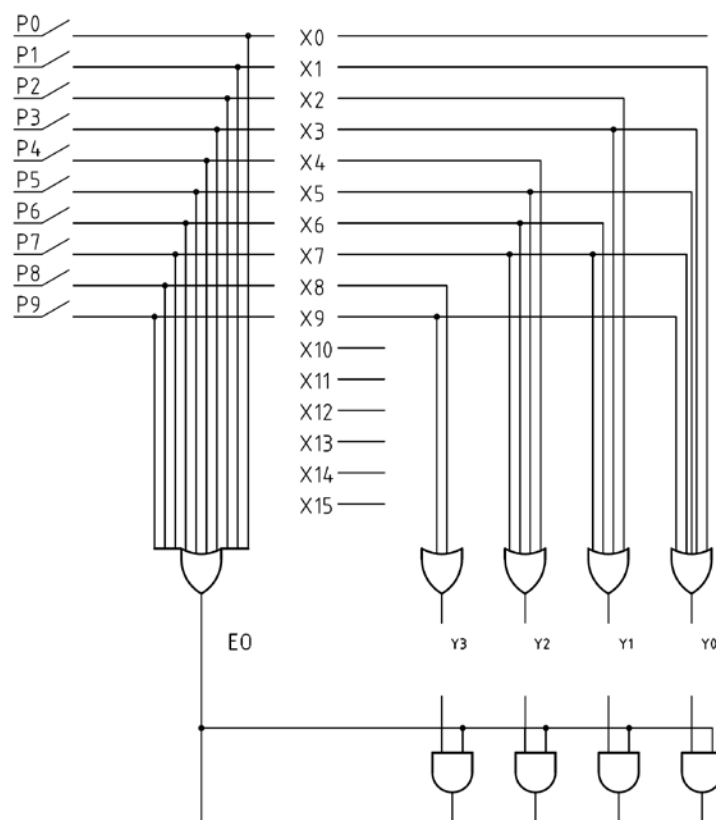
X7	X6	X5	X4	X3	X2	X1	X0	Y2	Y1	Y0	DE
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1	1
0	0	0	0	0	1	0	0	0	1	0	2
0	0	0	0	1	0	0	0	0	1	1	3
0	0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	0	0	0	0	1	0	1	5
0	1	0	0	0	0	0	0	1	1	0	6
1	0	0	0	0	0	0	0	1	1	1	7

Sinteza logičke funkcije izlaza je u ovom slučaju očigledna, a izvodi se direktno iz kombinacione tabele primenom logičkih ILI kola.



Za unos decimalnih brojeva, na primer, neophodan je koder sa 10 ulaznih i 4 izlazna kanala.
 Analogno prethodnom slučaju, kombinaciona tablica za sintezu ovog koda glasi:

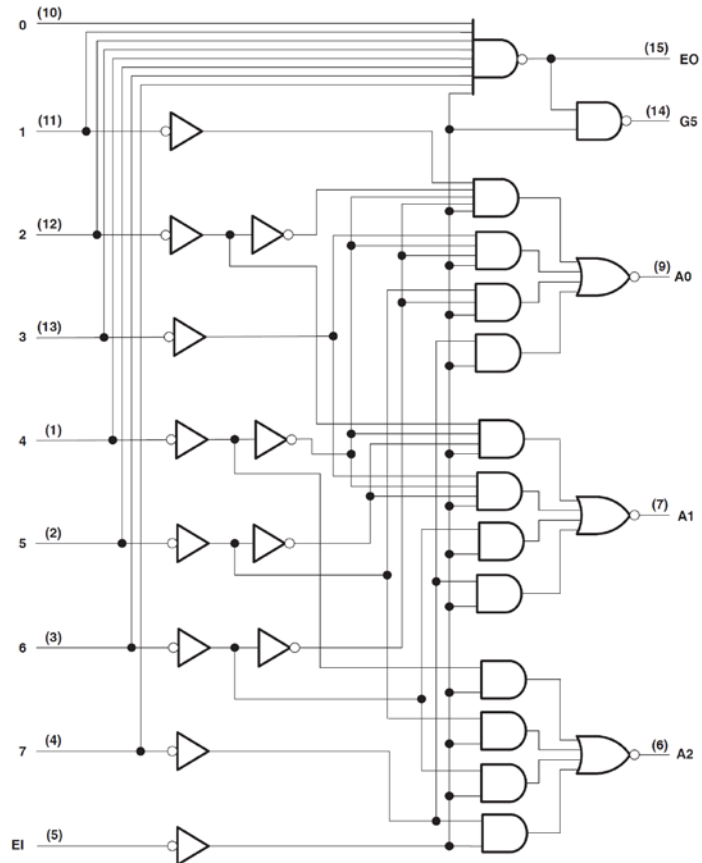
X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	Y3	Y2	Y1	Y0	DE
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0	2
0	0	0	0	0	0	1	0	0	0	0	0	1	1	3
0	0	0	0	0	1	0	0	0	0	0	1	0	0	4
0	0	0	0	1	0	0	0	0	0	0	1	0	1	5
0	0	0	1	0	0	0	0	0	0	0	1	1	0	6
0	0	1	0	0	0	0	0	0	0	0	1	1	1	7
0	1	0	0	0	0	0	0	0	0	1	0	0	0	8
1	0	0	0	0	0	0	0	0	0	1	0	0	1	9



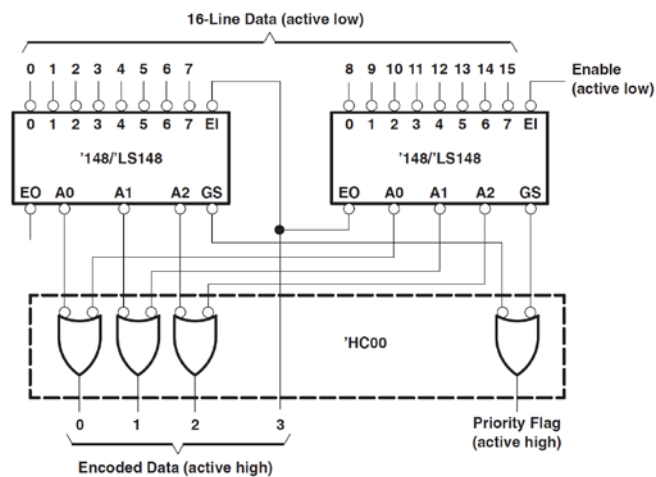
Dodatak sklopa dozvole izlaza (EO) omogućava da se odvoji fizička struktura ulaza od logičke. Ovo je samo najosnovnija modifikacija koja omogućava elementarnu funkcionalnu kompletnost i fizičku

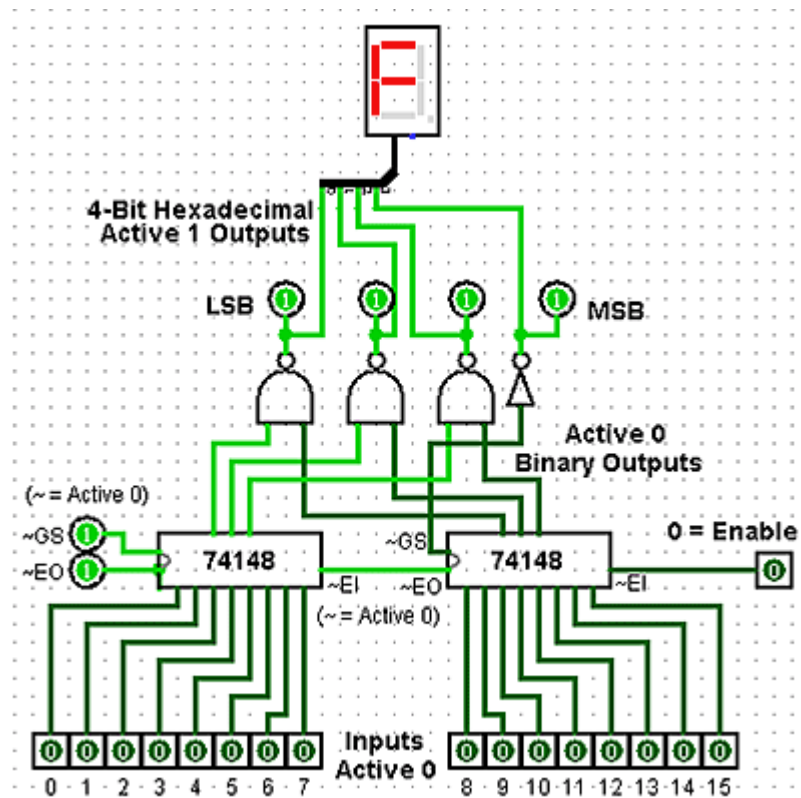
izvodljivost istovremeno. Realni uslovi primene za ovu vrstu sklopova zahtevaju dodatne elemente koja rešavaju različite tehničke detalje od velikog praktičnog značaja. Sklop koder sa 8 ulaza i 3 izlaza koji je baziran na primeni binarnih logičkih funkcija (I, ILI, NE) može se pronaći u funkcionalnom bloku 74HC148 koji se izvodi u obliku integrisanog kola/čipa sa 16 kontakata. Ovaj koder je tako izveden da se može medjusobno serijski sprežati (konkatenacija) čime se povećava broj fizičkih ulaza koji se kodira. Njegova logička šema je sledeća:

'148, 'LS148 logic diagram (positive logic)



Pin numbers shown are for D, J, N, NS, and W packages.





Primer primene kola 74HC148 za pobudu sedmosegmentnog pokazivača, kapaciteta 16 cifara (praktično, heksadecimalni koder). Ovaj primer prikazuje konkretnu namenu kodera kao sklopa koji eksternu fizičku pobudu, najčešće nekog HMI interfejsa, pretvara u binarne kodne grupe koje se uvode u neki digitalni sistem.

Binary Encoders

[Digital Electronics Module 1 \(Number Systems\)](#) described a number of different binary codes that are used to perform a range of functions in digital circuits. Mathematics, graphics, data manipulation and physical control systems are among many of the functions that are carried out using binary data, and each of these uses may require binary data arranged in various forms of binary codes. For example text may be represented by an ASCII code (American standard Code for Information Interchange), in which each letter, number or symbol is represented by a 7-bit binary code. Decimal numbers in a calculator may be sent to a numeric display using [BCD \(Binary Coded Decimal\)](#). Notice that the word 'code' appears in each of these titles, and a binary code differs from normal binary because it is arranged in a particular way to suit a given purpose.

Priority Encoders

Binary Encoders generally have a number of inputs that must be mutually exclusive, i.e. only one of the inputs can be active at any one time. The encoder then produces a binary code on the output pins, which changes in response to the input that has been activated.

Priority Encoding

Because it is always possible when using input switches that more than one input may be active at a single time, most encoders of this type feature 'priority encoding' where, if more than one input is made active at the same time, the output will select only the most significant active input. For example, if 6 and 7 are pressed together the BCD output will indicate 7. The Pinout diagram for the [74HC147 10-to-4-line priority encoder](#) from [NXP \(Philips Semiconductor\)](#), is illustrated in Fig.4.4.1.

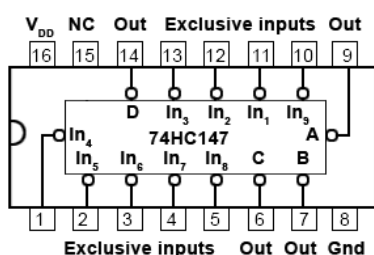


Fig. 4.4.1 74HC147 10-to-4-Line Priority Encoder

Depending on the encoding purpose, each different IC has its own particular method for solving encoding problems. For example, a simple decimal to BCD (or 10-to-4 line) encoder would be expected to have ten input pins, but in fact the 74HC147 has only 9. The tenth condition (zero) is assumed to be present because when none of the 1 to 9 input pins is active, this must indicate zero.

The input pins may be used to connect to switches on a decimal keypad, and the encoder would output a 4-bit BCD code, (0000₂ to 1001₂) depending on which key has been pressed, or simply to identify which one of ten input lines in a circuit is active, by outputting an appropriate number in four bit BCD code.

Chip Enable Inputs

Some other encoder ICs also feature extra inputs and outputs that allow several ICs to be connected together to achieve more flexibility in the numbers of input and output lines available. These include ENABLE inputs, (typically labelled E), which may consist of one or more input pins that need to have a particular logic level applied (usually logic 0) in order to activate the encoding action. In the absence of a correct ENABLE signal the output pins of the IC will remain in their inactive state.

Switch Bounce

One problem with combinational logic circuits is that unintended changes in output data can occur during the times when the outputs of the IC are changing. This can be due to problems such as switch contacts 'bouncing' as they close, creating rapid and unpredictable changes in logic levels for a very short time, however logic IC operate at high speed and will respond to these very fast changes.

Race Hazards

Problems can also occur due to 'race hazards' where different paths that digital signals take through a logic circuit may have different numbers of gates. For example two logic signals that change simultaneously at two circuit inputs may take different routes through the circuit before being applied to some common gate later in the circuit. However, if one signal passes through six gates for example, while the other signal passes through seven gates, each of the signals will have encountered a different total propagation delay due to the different number of gates they encountered. Therefore they will each arrive at the common gate at slightly different times, and so for a very short time an unexpected logic level may occur at that gate output.

In using combinational logic ICs such as an encoder, problems like switch bounce and race hazards must be allowed for, and one (though not necessarily the best) solution can be to temporarily make the ENABLE pin high during times when data is likely to change. This disables the encoder for a short time until the signal data has settled at its new state, so that there is no chance of errors at the output during changes of input signals.

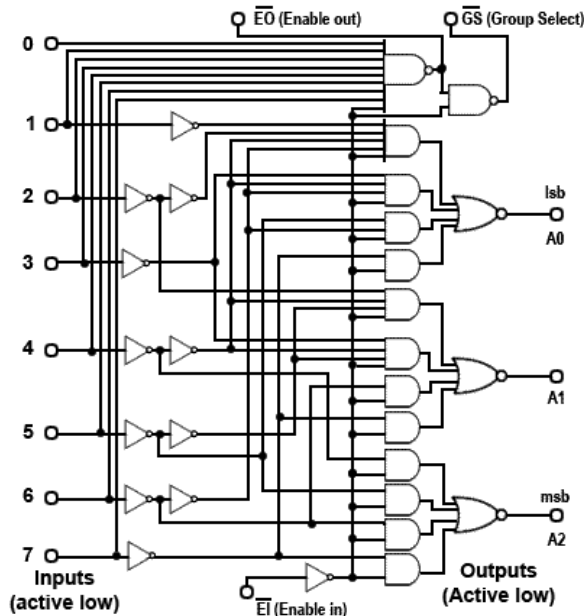


Fig. 4.4.2 74HC148 8-to-3-Line Encoder

74HC148 8-to-3-Line Encoder

The 74HC148 also uses priority encoding and features eight active low inputs and a three-bit active low binary (Octal) output. The internal logic of the 74HC148 is shown in Fig. 4.4.2

The IC is enabled by an active low Enable Input (EI), and an active low Enable output (EO) is provided so that several ICs can be connected in cascade, allowing the encoding of more inputs, for example a 16-to-6-line encoder using two 8-to-3 encoders. The CMOS 74HC148 also uses active low inputs and outputs. The operation of the 74HC148 can be seen from its truth table shown in Table 4.4.1.

Table 4.4.1													
Inputs								Outputs					
\overline{EI}	0	1	2	3	4	5	6	7	A2	A1	A0	\overline{GS}	\overline{EO}
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	L	L	L	L	H
L	X	X	L	H	H	H	H	H	L	H	L	L	H
L	X	L	H	H	H	H	H	H	H	L	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

Notice from Table 4.4.1 that the IC is only active when EI is low, and also that for each input selected by a low logic level (L), all lower value inputs indicate 'Don't Care', typical of priority encoding.

Two further outputs (GS and EO) are used for connecting additional 74HC148 ICs in cascade.

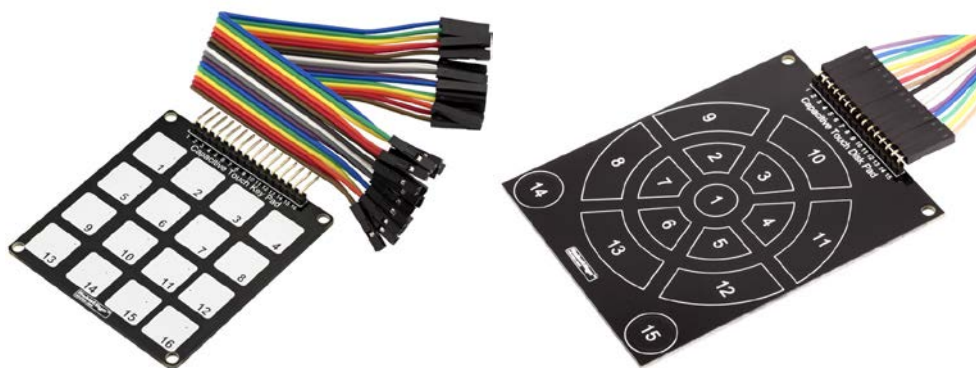
The EI input is normally used on the most significant IC and whenever an input on this IC is selected, the EO output goes high (disabling any less significant ICs), and the Group Select (GS) output goes low indicating that the group of outputs of this IC are active.

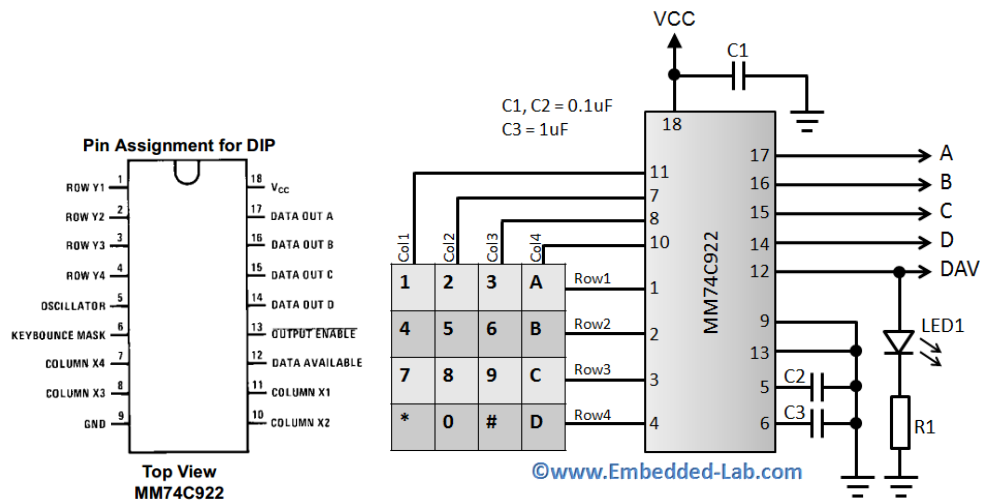
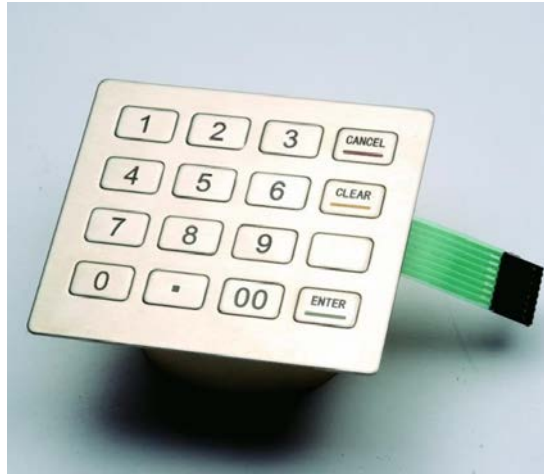
.....

U ovom kontekstu, na primer za rad sa 4x3 ili 4x4 poljima tastera navedenim na slici ispod, koriste se specijalizovana logička kola poput 74C922N ili MM74C922 kojim se pored funkcije kodiranja, istovremeno razrešava niz hardverskih specifičnosti ovog sklopa, uključujući i **problem nestabilnosti kontakta**, kao i **problem istovremenog aktiviranja dva ili više tastera**. Izlazni kanali kodera se u ovom slučaju uvode u neki digitalni mikroprocesorski sklop, koji time dobija fizički ulaznu jedinicu za unos podataka (na primer decimalnih brojeva u BCD formatu i slično).



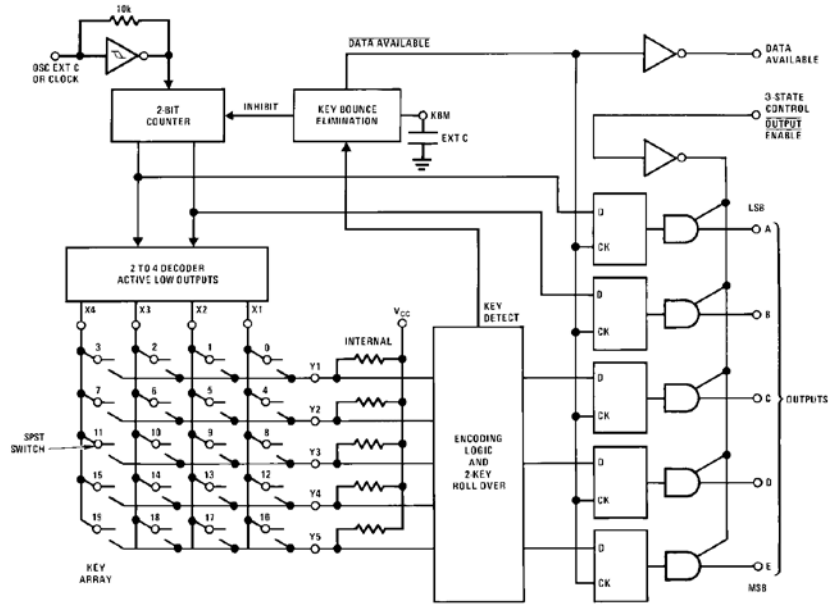
Decimalna (2x4 tastera) i heksadecimalna (4x4 tastera) tastatura





Key	D	C	B	A	Hex
1	0	0	0	0	0x0
2	0	0	1	1	0x1
3	0	0	1	0	0x0
4	0	1	0	0	0x0
5	0	1	1	1	0x1
6	0	1	1	0	0x0
7	1	0	0	0	0x0
8	1	0	1	1	0x1
9	1	0	1	0	0x0A
0	1	1	1	1	0x0F
A	0	0	0	1	0x01
B	0	1	0	1	0x05
C	1	0	0	1	0x09
D	1	1	0	1	0x0D
*	1	1	0	0	0x0C
#	1	1	1	0	0x0E

keystudio



Truth Tables

(Pins 0 through 11)

Switch Position	0	1	2	3	4	5	6	7	8	9	10	11
	Y1,X1	Y1,X2	Y1,X3	Y1,X4	Y2,X1	Y2,X2	Y2,X3	Y2,X4	Y3,X1	Y3,X2	Y3,X3	Y3,X4
D												
A A	0	1	0	1	0	1	0	1	0	1	0	1
T B	0	0	1	1	0	0	1	1	0	0	1	1
A C	0	0	0	0	1	1	1	1	0	0	0	0
O D	0	0	0	0	0	0	0	0	1	1	1	1
U E (Note 1)	0	0	0	0	0	0	0	0	0	0	0	0
T												

(Pins 12 through 19)

Switch Position	12	13	14	15	16	17	18	19
	Y4,X1	Y4,X2	Y4,X3	Y4,X4	Y5 (Note 1), X1	Y5 (Note 1), X2	Y5 (Note 1), X3	Y5 (Note 1), X4
D								
A A	0	1	0	1	0	1	0	1
T B	0	0	1	1	0	0	1	1
A C	1	1	1	1	0	0	0	0
O D	1	1	1	1	0	0	0	0
U E (Note 1)	0	0	0	0	1	1	1	1
T								

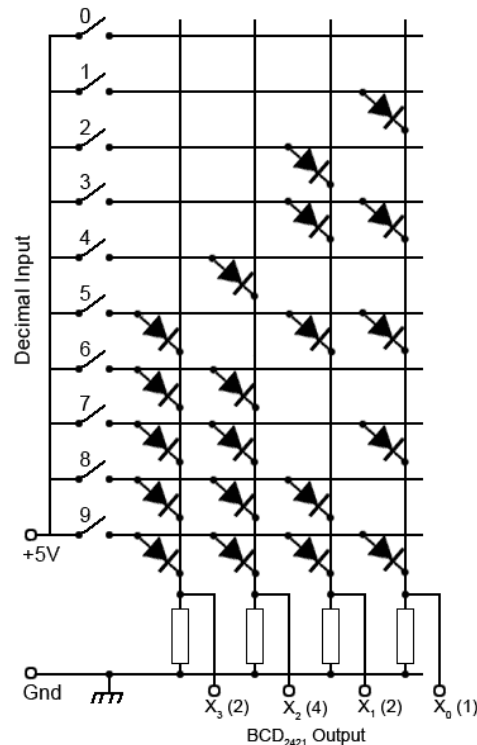
Note 1: Omit for MM74C922

.....

Fizička realizacija koderskog sklopa može da bude vrlo različita. Ovde se logičko ponašanje, koje je invarijanta, odvaja od načina na koji se ta funkcija fizički ostvaruje. Sa praktičnog aspekta, interesantna su takozvani matrični enkoderi (mada 74C922N ili MM74C922 takodje spadaju u ovu klasu).

Diode Matrix Encoders

Where encoders are needed for non-standard applications, they can also be implemented using a diode matrix, such as the decimal-to-BCD₈₄₂₁ encoder shown in Fig 4.4.4.



In this example, as any one of the ten switches is closed +5V is applied to just one of the horizontal lines. Any diode that has its anode connected to that horizontal line and its cathode connected to a vertical line (that is held at zero volts by a resistor connected to Gnd) will conduct.

When current flows through any of the resistors, the top of that resistor will be at +4.4V (i.e. +5V minus a 0.6V drop in across the diode), which will be seen by the output as logic 1.

For example if switch 6 is closed, the two diodes connected between line 6 and columns X₃ and X₂ will conduct, making outputs X₃ and X₂ logic 1 and giving a binary₂₄₂₁ output word of 1100₂ (or 2+4 =6₁₀).

This particular diode matrix will therefore give an output in BCD₈₄₂₁ code from 0000₂₄₂₁ to 1111₂₄₂₁ for closure of switches 0 to 9.

Many other output sequences are possible therefore, by using different arrangements of the diode positions.

Although the encoder circuits described in this module may be used in a number of useful encoding situations, they have some features that limit their use for realistic keyboard encoding.

- Priority encoders do not sense information from two or more keys that are pressed at the same time.
- Switches on keypads normally contact for only a brief time, these basic encoders are not able to store and remember the data input from a pressed key once it is released.
- When a switch is closed the contacts may 'bounce' giving several brief 1 and 0 logic states, when ideally there should be only one change in state for each key press.

To overcome common problems such as these, a more complex circuit (or IC) is required. These will typically have features such as key bounce elimination, built in data memory, timing control using a clock (oscillator) circuit and some ability to differentiate between two or more keys pressed at the same time. Another important feature is the ability to signal to the system that the keyboard is controlling, when a key has been pressed and new data needs to be read.

For small keypads having less than 20 keys the processing has typically been carried out by an ASIC (Application Specific Integrated Circuit) such as the [MM74C922 Keyboard Encoder](#) although this IC is now being listed as obsolete by some manufacturers, as many modern circuits, especially those with more keys, use a dedicated microprocessor or micro-controller (MCU) to carry out keyboard decoding.

.....



~	!	@	#	\$	%	^	&	*	()	-	=	←	Backspace	Num Lock	/	*	-			
Tab	Q	W	E	R	T	Y	U	I	O	P	{	}		\	7	8	9	+			
Caps Lock	A	S	D	F	G	H	J	K	L	:	"	'	↵	Enter	4	5	6	+			
Shift	Z	X	C	V	B	N	M	<	>	?	/	↵	Shift	↵	1	2	3	Enter			
Ctrl	Win Key	Alt											Alt	Win Key	Menu	Ctrl	0	.	Del		
																		Ins	Del		

Koderi za ovako složene ulazne uređaje su složeni elektronski sklopovi, najčešće programabilni, čime se obezbeđuje fleksibilnost u primeni, odnosno pruža mogućnost različitim korisnicima (proizvođači kompjuterskih komponenti) da razviju svoja rešenja na finansijski održivim osnovama, uz ugradnju nekih specifičnosti koje će ih tržišno diferencirati i tako pružiti im neku vrstu kompetitivne tržišne prednosti. Primer takvog sklopa smeštenog u posebnom čipu je SK5120 kontroler tastature koji proizvodi firma SPRINTTEK.

SK5120 FlexMatrix Keyboard Controller

Low Power User-Programmable USB Keyboard Encoder with an External PS/2 Port and KeyMouse Support
The SK5120 is a low power USB keyboard encoder with a user-programmable keyboard matrix.

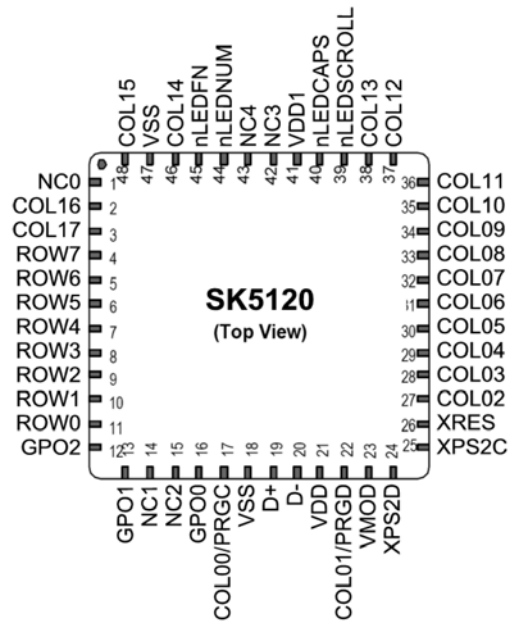
The IC can be programmed to any keyboard with four matrix tables for FN and NUMLOCK cases, so the IC is the best choice for custom keyboard solution but with an off-the-shelf IC.

The IC operates under low voltage 3.3V and can interface a low voltage (3V) PS/2 mouse such as touchpad to minimize the system power consumption.

The SK5120 scans and encodes an 8-row by 18-column matrix. The key press events are translated to keyboard and mouse report. The encoder gets matrix information from on-chip flash matrix table. Sprintek provides Windows® application FlexMatrix Editor and Programmer software to edit, download and upload the matrix table.

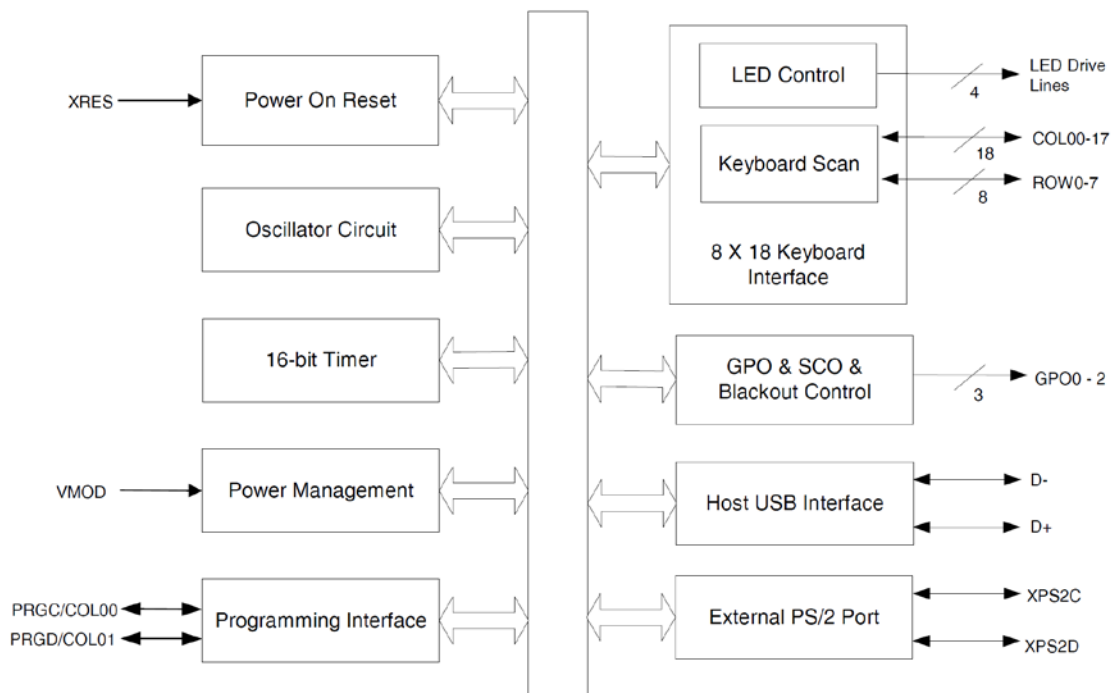
The SK5120 provides an external PS/2 port that supports hot plug and hot swap of PS/2 mouse including wheel mouse, and keyboard devices. It is ideal to work with touchpad or pointing stick PS/2 mouse encoder for new architecture laptop/netbook computers.

PIN ASSIGNMENTS

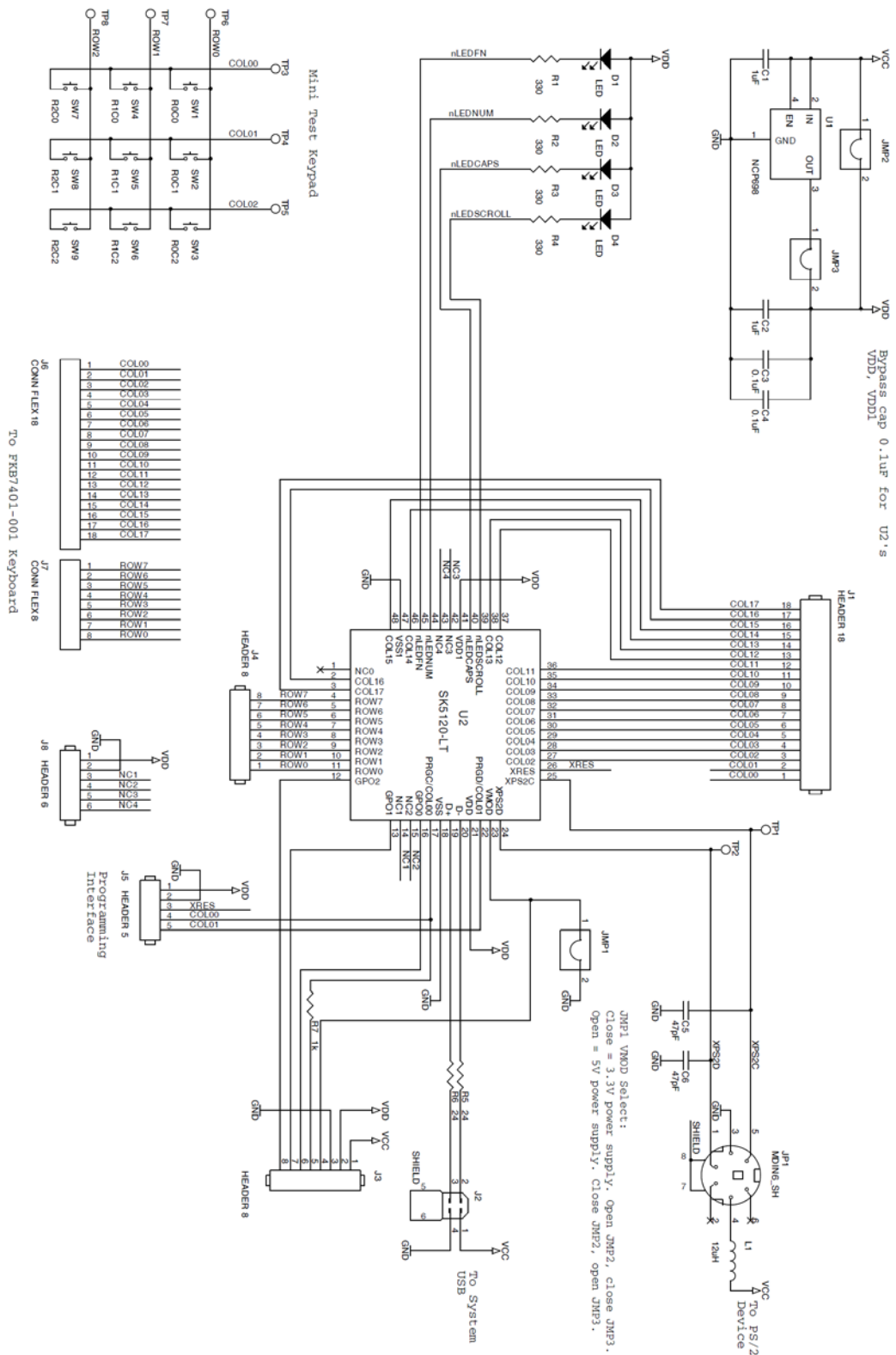


Funkcionalni blok dijagram pokazuje da je SK5120 multifunkcionalni sklop, koji pored funkcije kodiranja tastera tastature obezbedjuje niz drugih funkcija, koje omogućavaju da se tastatura kao ulazno (i delom izlazni) sklop pojavi kao uređaj. dakle fizički nezavisan i funkcionalno kompletan modul koji se dodaje digitalnom kompjuteru kako bi se obezbedila funkcija unosa alfanumeričkih karaktera i neke kontrolne funkcije za editovanje teksta, kompjutersku grafiku i slično. Ovo je samo jedno od mnogih rešenja, primer koji ima za cilj da demonstrira primenu kodera i inženjersko rešenje jednog složenog i vrlo odgovornog elektromehaničkog sklopa kao što je tastatura za unos podataka na PC računaru.

FUNCTION BLOCK DIAGRAM



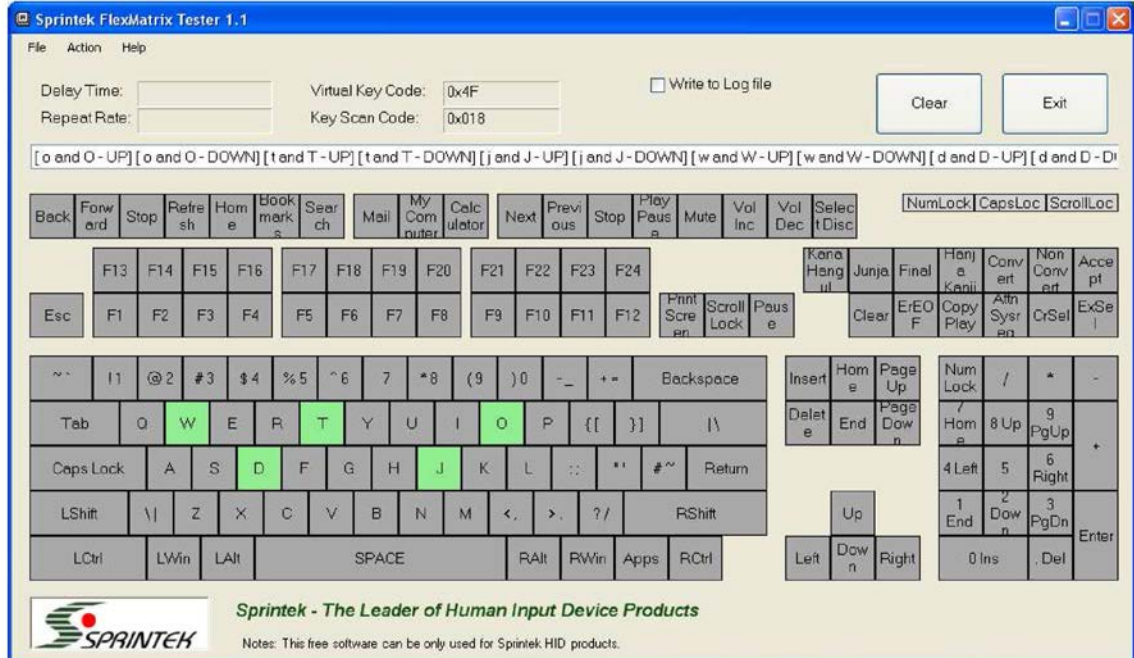
SCHEMATIC OF EVALUATION BOARD

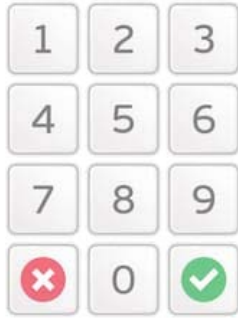


Test Keyboard Matrix

Sprintek offers a keyboard test tool to verify your keyboard design. The FlexMatrix Tester software can be downloaded from the SK5120 page on the Sprintek web site <http://www.sprintek.com/>

Here is the screen snapshot of FlexMatrix Tester software.





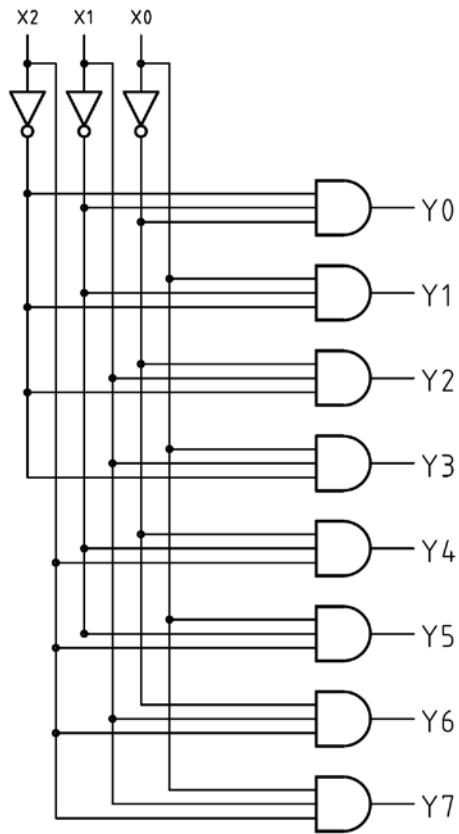
a1.3.5.2 Dekoder

Dekoder je kombinaciona digitalna mreža sa više ulaza i više izlaza koja je tako sintetisana da svaka kombinacija ulaznih promenljivih aktivira jedan i samo jedan izlaz. Dekoderi mogu biti potpuni i nepotpuni. Kod potpunih dekodera broj izlaznih kanala je jednak 2^n , gde je sa n označen broj ulaznih promenljivih. Kod nepotpunih dekodera broj izlaznih kanala je manji od 2^n .

Primer potpunog dekodera sa tri ulazna i osam izlaznih kanala:

DE	X2	X1	X0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
2	0	1	0	0	0	0	0	0	1	0	0
3	0	1	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	0	1	0	0	0	0
5	1	0	1	0	0	1	0	0	0	0	0
6	1	1	0	0	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	0

Sinteza logičke funkcije izlaza je u ovom slučaju očigledna, jer ta mreža treba da za svaku binarnu grupu na ulazu produkuje korespondentni minterm na svom izlazu. U cilju sprečavanja neregularnosti rada mreže u prelaznom režimu, kod praktičnih realizacija ulazni kanali se dopunjuju jednim kontrolnim kanalom koji ima funkciju dozvole (EN).



Obratiti pažnju na korišćenje nultih vrednosti ulaza. One se invertuju da bi se, nezavisno od stanja na ulazu, produkovala grupa od tri jedinice koje se vode na korespodentno I logičko kolo, koje dalje produkuje korespodentni izlaz.

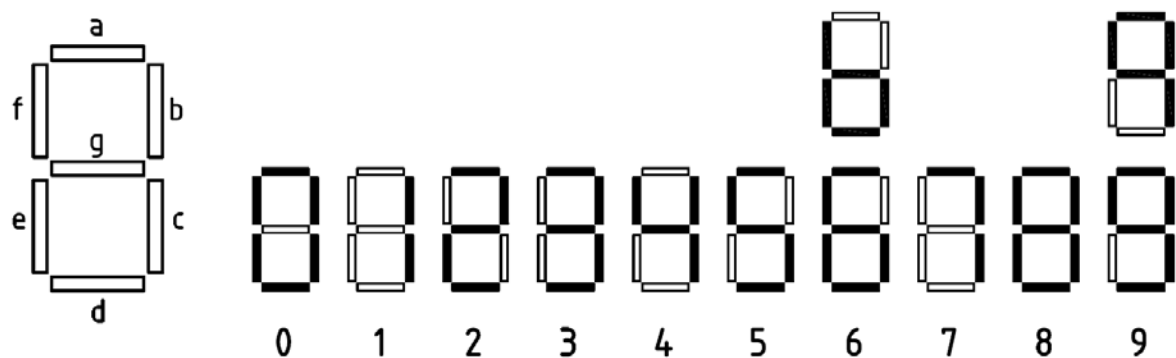
Primer komercijalnog dekodera: 74HCT138

a1.3.5.3 Konvertor koda

Konvertor koda je kombinaciona digitalna mreža sa više ulaza i više izlaza, koja prevodi jedan digitalni kod u neki drugi digitalni kod. Konvertor koda se može sintetisati kao sekvencijalno spregnuti dekoder i koder. Minimizacijom ove dve mreže dobija se jedinstvena mreža koja je jednostavnija od polazne. Sinteza konvertora koda je po pravilu složenija od sinteze koda i dekodera.

Tipičan primer konvertora koda je konvertor za konverziju binarnog u Gray-ov kod i Gray-ovog koda u binarni.

U praksi se vrlo često sreću konvertori koji prevode BCD kod u kod numeričkog sedmosegmentnog svetlosnog indikatora. Ova vrsta indikatora/pokazivača bazirana je na primeni LED dioda (poluprovodnički elementi koji emituju svetlost). Sastoji se iz 7 segmenata, geometrijski raspoređenih tako da se ovim indikatorom mogu prikazati sve cifre dekadnog brojnog sistema i jedan broj slovnih karaktera. Svaki od sedam segmenata izveden je kao jedna LED dioda.



Formiranje pojedinih simbola ostvaruje se uključivanjem odgovarajuće kombinacije svetlećih segmenata. Prema tome, da bi se ovakav indikator mogao da realizuje neophodna je odgovarajuća konvertorska mreža za konvertovanje binarno kodiranih simbola (BCD kod) u simbole sedmosegmentnog pokazivača. U pitanju je dakle, jedna složena kombinaciona mreža koja ima 4 ulazna kanala i 7 izlaznih kanala. Svaki izlazni kanal posmatramo kao logičku funkciju oblika:

$$y = f(X_1, X_2, X_3, X_4) = \sum_{i=0}^{15} b_i P_i = \sum_4 b_i P_i$$

Sinteza logičkih funkcija za svaki od sedam izlaznih kanala koda izvodi se primenom kombinacione tablice. Kombinaciona tablica u ovom slučaju ima sledeći oblik:

DE	X4	X3	X2	X1	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
A	1	0	1	0	x	x	x	x	x	x	x
B	1	0	1	1	x	x	x	x	x	x	x
C	1	1	0	0	x	x	x	x	x	x	x
D	1	1	0	1	x	x	x	x	x	x	x
E	1	1	1	0	x	x	x	x	x	x	x
F	1	1	1	1	x	x	x	x	x	x	x

Na osnovu navedene kombinacione tablice može se napisati SDNF oblik funkcije izlaza, koji se primenom Vejč-Karno tablice svodi na svoju minimalnu formu. Za slučaj segmenta 'a' sledi:

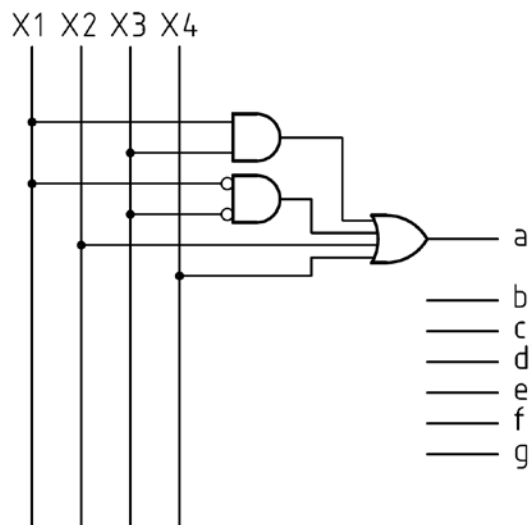
$$a = \sum_4(0,2,3,5,6,7,8,9) = k_1 + k_2 + k_3 + k_4 = X_2\bar{X}_4 + X_1X_3\bar{X}_4 + \bar{X}_2\bar{X}_3X_4 + \bar{X}_1\bar{X}_2\bar{X}_3$$

X ₄ X ₃	00	01	11	10	X ₂ X ₁
00	1	0	1	1	k ₁
01	0	1	1	1	k ₂
11	0	0	0	0	
10	1	1	0	0	k ₃
					k ₄

Postupak minimizacije izveden je uz zanemarivanje postojanja zabranjenih stanja. Svi mintermovi čiji je decimalni ekvivalent veći od devet (10, 11, 12, 13, 14 i 15) praktično predstavljaju zabranjena stanja ovog koda (ili preciznije, stanja koja se ne mogu pojaviti u regularnom radu). Ukoliko se iskoristi ova činjenica, onda je moguće postići značajno jednostavnije rešenje logičke funkcije, odnosno:

$$a = \sum_{i=0}^{15} (0, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) = k_1 + k_2 + k_3 + k_4 = X_2 + X_4 + X_1X_3 + \bar{X}_1\bar{X}_3$$

X ₄ X ₃	00	01	11	10	X ₂ X ₁
00	1	0	1	1	k ₁
01	0	1	1	1	k ₃
11	X	X	X	X	k ₂
10	1	1	X	X	k ₄



Dalje se navodi svih sedam logičkih funkcija BCD/7S koda (za segmente b, ... , g) dobijenih postupkom sinteze koji je primenjen za logičku funkciju 'a' LED segmenta (zabranjena stanja uzeta u obzir):

$$a = X_2 + X_4 + X_1X_3 + \bar{X}_1\bar{X}_3$$

$$b = \bar{X}_3 + X_1X_2 + \bar{X}_1\bar{X}_2$$

$$c = X_1 + \bar{X}_2 + X_3$$

$$d = X_4 + \bar{X}_1X_2 + X_1\bar{X}_2X_3 + \bar{X}_1\bar{X}_3 + X_2\bar{X}_3$$

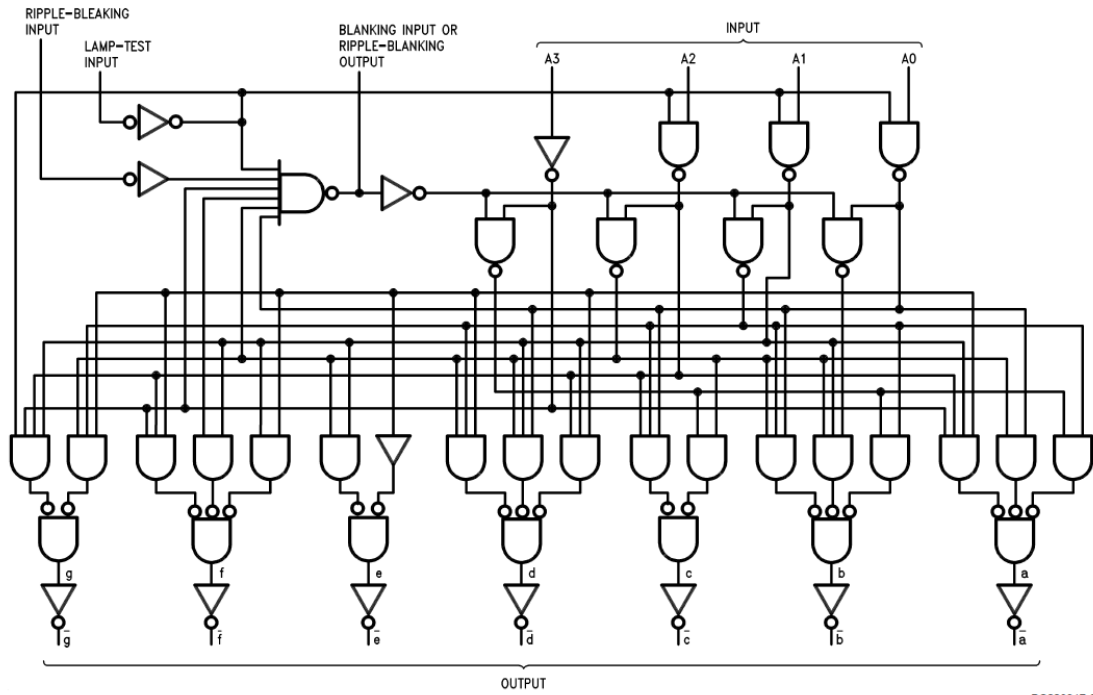
$$e = X_1\bar{X}_2 + \bar{X}_1\bar{X}_3$$

$$f = X_4 + \bar{X}_1\bar{X}_2 + \bar{X}_1X_3 + \bar{X}_2X_3$$

$$g = X_4 + \bar{X}_1X_2 + X_2\bar{X}_3 + \bar{X}_2X_3$$

BCD u 7seg konvertor koda se sreće na tržištu kao monolitni sklop zajedno sa drajverima na izlaznim kanalima za napajanje led dioda 7 segmentnog pokazivača. Na primer, **74LS47** je jedan takav proizvod koji se smešta u 16 pin DIL pakovanje.

Logic Diagram



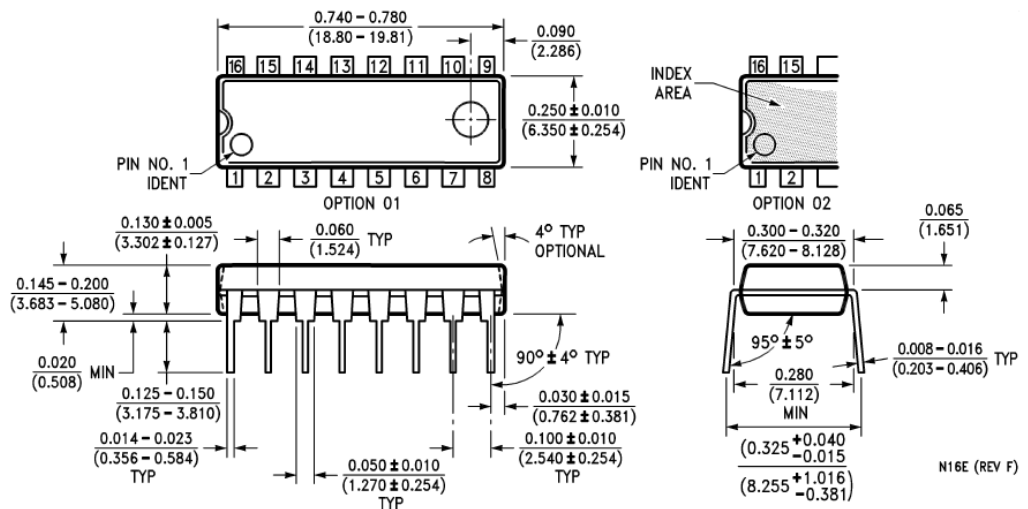
DS009817-3

Numerical Designations—Resultant Displays



DS009817-4

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



N16E (REV F)

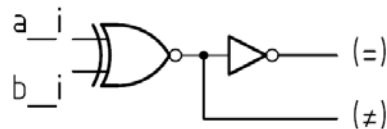
16-Lead Molded Dual-In-Line Package (N)
Order Number DM74LS47N
Package Number N16E

a1.3.5.4 Komparator

Dva broja, odnosno dve binarne grupe su medjusobno jednake ako je svaka cifra binarne grupe A jednaka binarnoj grupi B. Odatle sledi:

$$a_i \cdot b_i + \bar{a}_i \cdot \bar{b}_i = \overline{a_i \oplus b_i} = 1$$

odakle dalje sledi logička realizacija:



Simbol \oplus (XOR) označava binarnu operaciju koja se naziva ISKLJUČIVO ILI i biće detaljno objašnjen kasnije.

a	b	R	
0	0	1	=
0	1	0	>
1	0	0	<
1	1	1	=

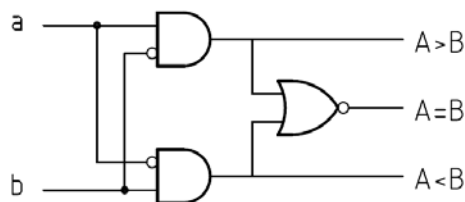
U slučaju komparacije brojeva, univerzalni komparator upoređuje brojeve A i B i kao izlaz generiše signale $A > B$, $A = B$ i $A < B$. Za jednobitni binarni broj važi:

$$a \cdot \bar{b} = 1 \rightarrow A > B$$

$$a \oplus b = 1 \rightarrow A \neq B$$

$$\bar{a} \cdot b = 1 \rightarrow A < B$$

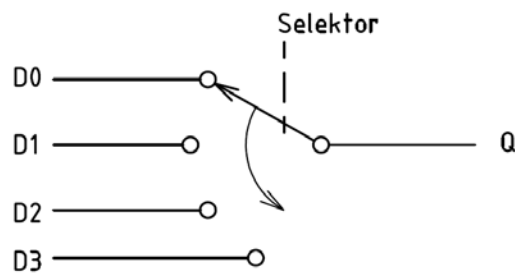
odakle sledi logička šema univerzalnog komparatora jednobitnih brojeva A i B:



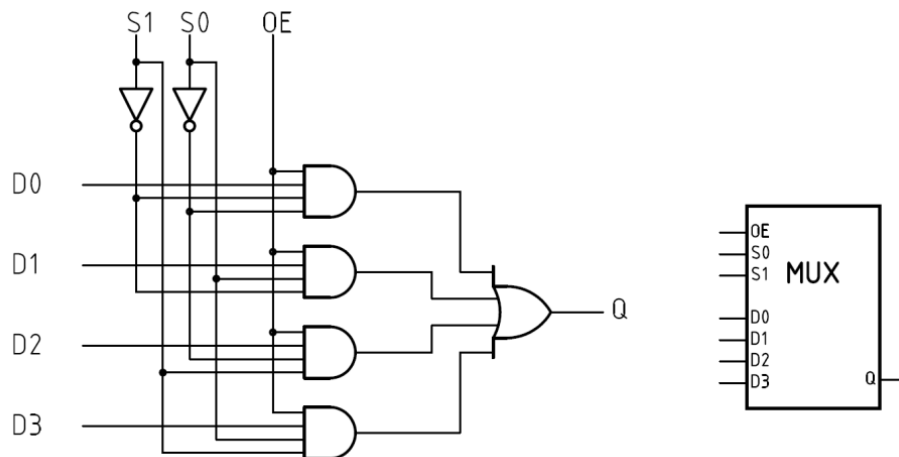
Kao komparator dva višebitna broja može se koristiti aritmetički sabirač/oduzimač brojeva. Poredjenje brojeva se ovde svodi na njihovo oduzimanje ($k=1$). Ukoliko je rezultat 0 brojevi su medjusobno jednaki. Ako se brojevi razlikuju, na osnovu bita izlaznog prenosa iz potpunog sabirača (C_{izl}) može se ustanoviti da li je $A > B$ ili $A < B$. Ovo će kasnije biti detaljno obradjeno.

a1.3.5.5 Multiplekser

Multiplekser je logička kombinaciona mreža koja ostvaruje funkciju prekidača sa više ulaza i jednim izlazom. U svakom trenutku ostvaruje se veza između jednog ulaznog kanala i izlaza. Selekcija ulaznog kanala koji će se povezati sa izlazom ostvaruje se tako što se svakom ulaznom kanalu dodaje odgovarajući kodni slog kao njegova lokalna adresa. Sklop za kodiranje ostvaruje funkciju selekcije ulaza, pa se iz tog razloga ovaj podsklop naziva selektor.



Sklop multipleksera sa četiri ulazna kanala (D3:D0) kroz koje protiču podaci (Data channels), prikazan je na slici ispod. Za kodiranje 4 kanala neophodna su dva selektorska ulaza S0 i S1. Sklop multipleksera sadrži i OE kanal koji daje dozvolu za prosledjivanje selektovanog sadržaja na izlaz multipleksera Q. Za realizaciju ovog multipleksera neophodna su četiri I i jedno ILI logičko kolo.



Multiplekser se može iskoristiti za generisanje logičkih funkcija, pri čemu selektorski ulazi dobijaju funkciju senzorskih ulaza, a Data ulazni kanali, postaju kontrolni.

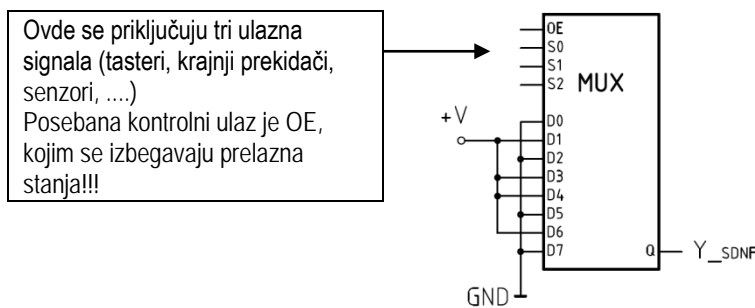
Sinteza logičke mreže se ostvaruje na vrlo jednostavan način: svakom od ulaznih kanala D_i odgovara jedan kodni slog selektorskih ulaza ($S_n:S_0$) u obliku minterma. Ukoliko u logičkoj funkciji izraženoj u SDNF obliku postoji određeni minterm, onda se na njemu korespondentni ulazni kanal dovodi logička jedinica (naponski nivo +V, odnosno Vss), u suprotnom na taj kanal se dovodi logička

nula (nulti naponski nivo, odnosno masa / GND). Primenom ovog postupka za sve mintermove realizuje se tražena logička funkcija. Dakle, multiplekser se može primeniti kao **univerzalni generator logičkih funkcija u SDNF obliku** koji na svom izlazu generiše bilo koju logičku funkciju izvedenu iz skupa mintermova generisanih njegovim selektorskim ulazima. Na primer, za logičku funkciju trećeg reda u obliku:

$$y = f(S_0, S_1, S_3) = Q_{SDNF} = S_2 S_1 \bar{S}_0 + S_2 \bar{S}_1 \bar{S}_0 + \bar{S}_2 S_1 S_0 + \bar{S}_2 \bar{S}_1 S_0 = \sum_{i=0}^7 (1,3,4,6)$$

prisutni su mintermovi sa decimalnim ekvivalentom 1, 3, 4 i 6 odakle sledi da na kanale D1, D3, D4 i D6 osmokanalnog multipleksera treba dovesti logičku jedinicu, dok na sve ostale kanale treba dovesti logičku nulu. Povezuje osmokanalnog multipleksera koji realizuje zadata logičku funkciju izvedeno je na način koji je prikazan na narednoj slici.

$$y_{SDNF} = f(S_0, S_1, S_3)$$



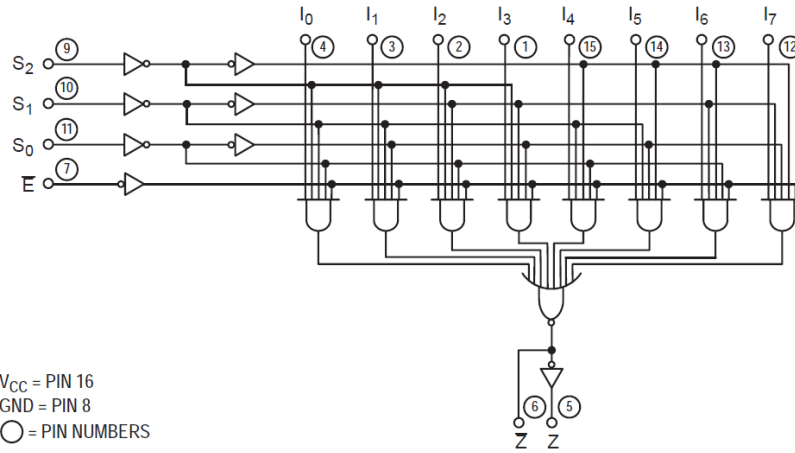
Primer multipleksera sa osam ulaznih kanala je kolo koje nosi oznaku SN74LS151, smešteno u DIP i SOIC kućištu sa 16 kontakata. Logička šema, parametri i raspored kontakata ovog kola prikazani su na slici ispod.

The TTL/MSI SN74LS151 is a high speed 8-input Digital Multiplexer. It provides, in one package, the ability to select one bit of data from up to eight sources. The LS151 can be used as a universal function generator to generate any logic function of four variables. Both assertion and negation outputs are provided.

- Schottky Process for High Speed
- Multifunction Capability
- On-Chip Select Logic Decoding
- Fully Buffered Complementary Outputs
- Input Clamp Diodes Limit High Speed Termination Effects

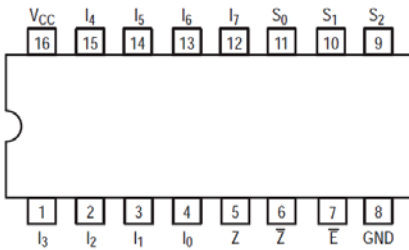
SN74LS151

LOGIC DIAGRAM



V_{CC} = PIN 16
 GND = PIN 8
 ○ = PIN NUMBERS

CONNECTION DIAGRAM DIP (TOP VIEW)



GUARANTEED OPERATING RANGES

Symbol	Parameter	Min	Typ	Max	Unit
V_{CC}	Supply Voltage	4.75	5.0	5.25	V
T_A	Operating Ambient Temperature Range	0	25	70	°C
I_{OH}	Output Current – High			-0.4	mA
I_{OL}	Output Current – Low			8.0	mA

PIN NAMES

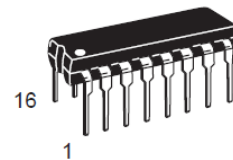
$S_0 - S_2$	Select Inputs
E	Enable (Active LOW) Input
$I_0 - I_7$	Multiplexer Inputs
Z	Multiplexer Output
Z	Complementary Multiplexer Output

LOADING (Note a)

	HIGH	LOW
$S_0 - S_2$	0.5 U.L.	0.25 U.L.
E	0.5 U.L.	0.25 U.L.
$I_0 - I_7$	0.5 U.L.	0.25 U.L.
Z	10 U.L.	5 U.L.
Z	10 U.L.	5 U.L.

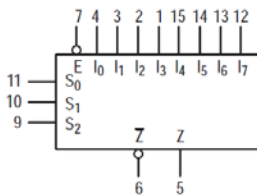
NOTES:

- 1 TTL Unit Load (U.L.) = 40 μ A HIGH/1.6 mA LOW.
- The Output LOW drive factor is 5 U.L. for Commercial (74) Temperature Ranges.

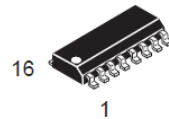


**PLASTIC
 N SUFFIX
 CASE 648**

LOGIC SYMBOL



V_{CC} = PIN 16
 GND = PIN 8

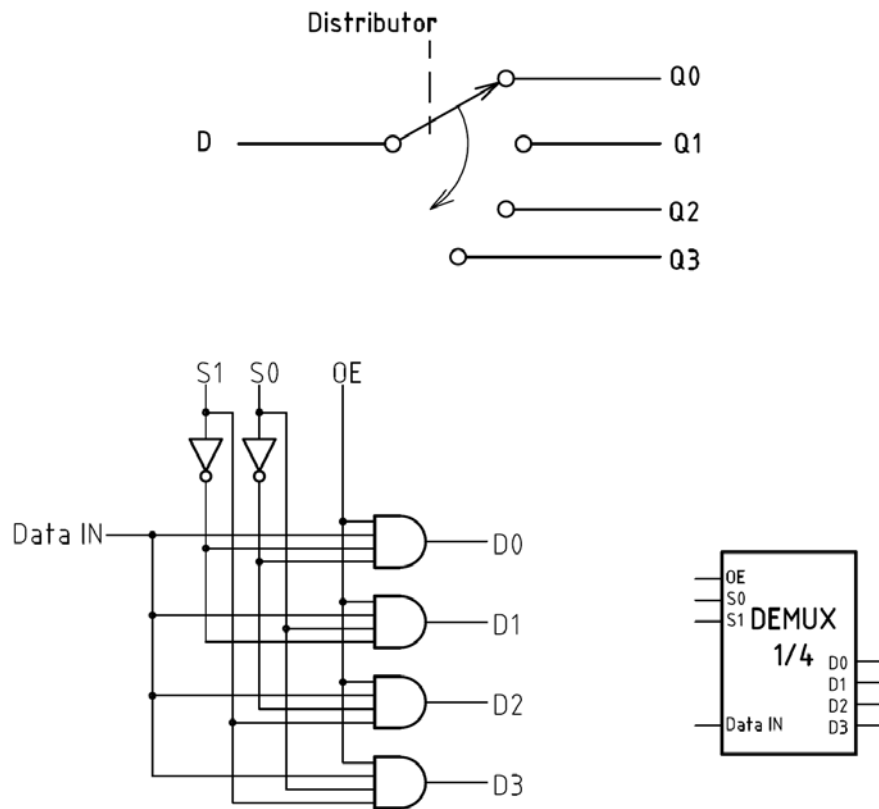


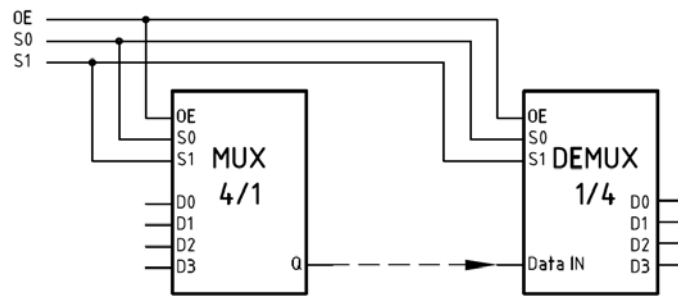
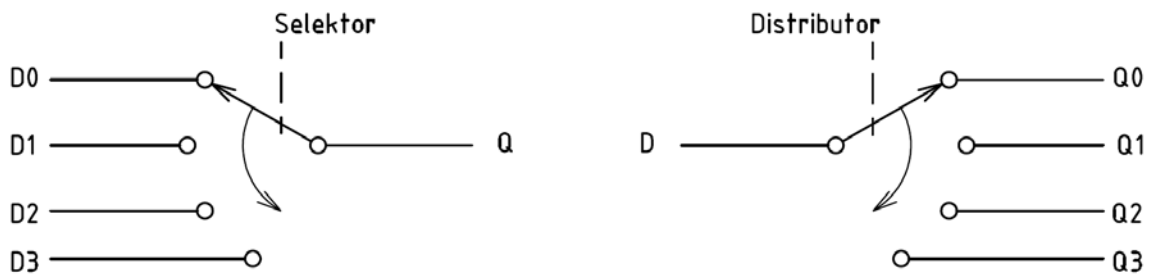
**SOIC
 D SUFFIX
 CASE 751B**

a1.3.5.6 Demultiplekser

Demultiplekser je logička kombinaciona mreža koja ostvaruje funkciju prekidača sa jednim ulazom i više izlaza. U svakom trenutku ostvaruje se veza izmedju ulaznog kanala i jednog od izlaza. Dakle, funkcija demultipleksera je suprotna funkciji multipleksera. Selekcija izlaznog kanala koji će se povezati sa ulazom ostvaruje se tako što se svakom izlaznom kanalu dodaje odgovarajući kodni slog kao njegova lokalna adresa. Sklop za kodiranje ostvaruje funkciju distribucije izlaza, pa se iz tog razloga ovaj podsklop naziva distributor.

Na slici ispod prikazana je logička šema jednog demultipleksera sa jednim ulaznim i četiri izlazna kanala.





Sprega multiplekser-demultiplekser omogućava prenos podataka na distancu **kroz jedan komunikacioni kanal u vremenskom multipleksu**. Primer ove sprege prikazan je na slici iznad. Izlazni kanal multipleksera 4/1 spojen je sa ulaznim kanalom demultipleksera 4/1. Svaki od ulaznih kanala D3:D0 propušta se kroz zajednički komunikacioni kanal prema unapred odredjenom pravilu saobraćaja i preusmerava na njemu korespodentni izlazni kanal D3:D1. Upravljanje radom ovog sklopa ostvaruje se preko zajedničkih selektorskih kanala i zajedničkog kanala dozvole prenosa. Sklop multiplekser-demultiplekser ponaša se kao komutator.

Primer demultipleksera je kolo 74HC154, koji poseduje 4 selektorska i 16 izlaznih kanala. Logička šema je prikazana na slici ispod. Ovakav sklop istovremeno služi i kao dekodeer.

