

Dekodirnik, kodirnik, demultiplekser in multiplekser

Digitalna vezja

Miha Moškon

miha.moskon@fri.uni-lj.si

<https://fri.uni-lj.si/en/about-faculty/employees/miha-moskon>

Strukturalna vezja

Dekodirnik / demultiplekser

Kodirnik

Multiplekser

Znotraj družine 7400: MSI vezja (10 do 1000 tranzistorjev) – podobni primeri: števec, register

Dekodirnik $n/2^n$

Izbira enega izmed 2^n izhodov na podlagi n vhodov.

Mintermski izhod – določa indeks minterma

Vsak izhod je določen z enačbo minterma

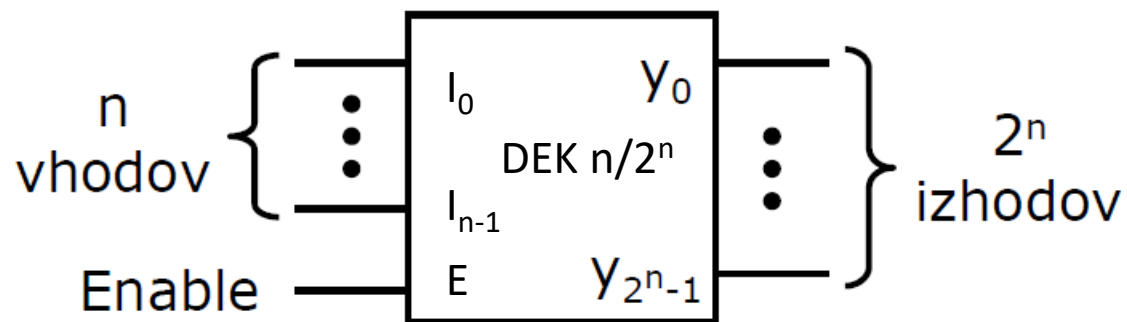
Dodatno: krmilni (enable) vhod

Dekodirnik $n/2^n$

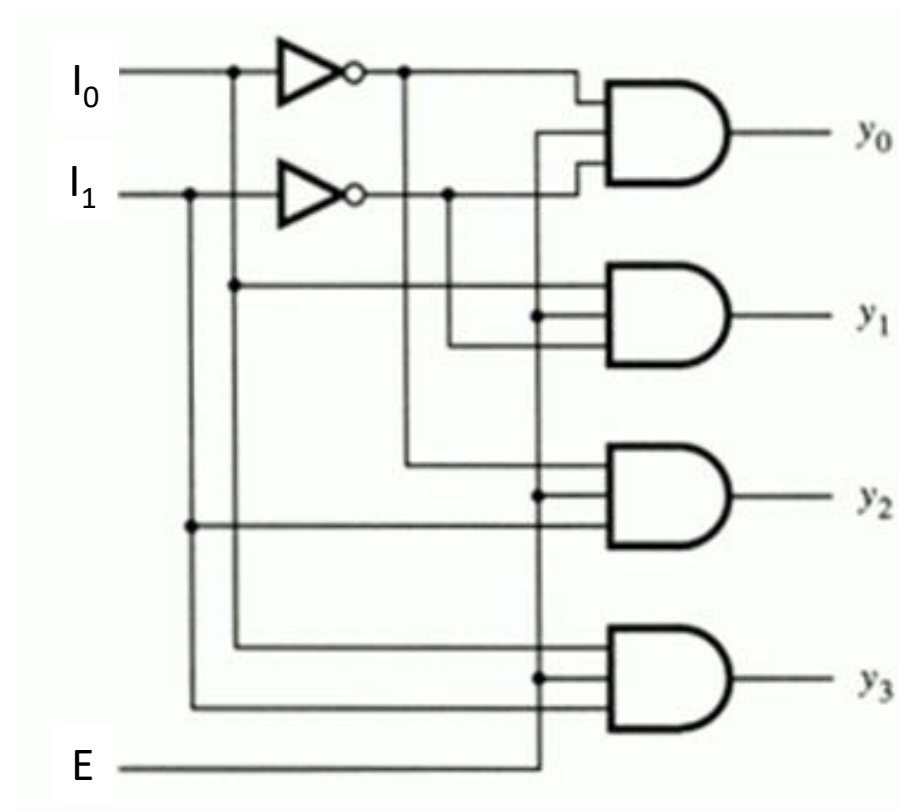
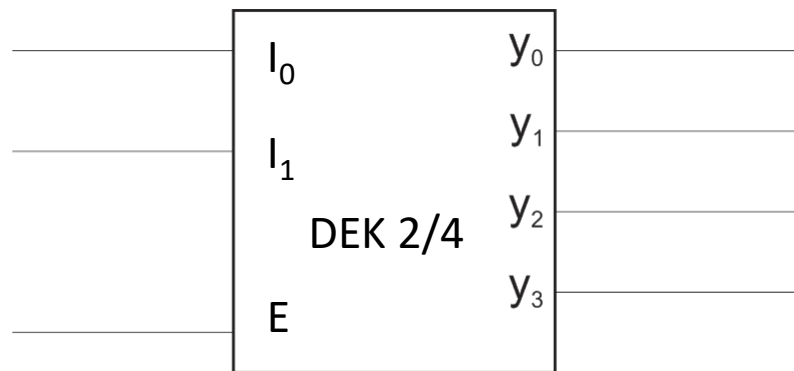
Vedno je aktiven največ en izhod (mintermski izhodi – izhod določa minterm)

n izbirnih vhodov + (enable vhod) $\rightarrow 2^n$ izhodov

- enable = '0': vsi izhodi so neaktivni (v stanju '0')
- enable = '1': izbran natanko en izhod v odvisnosti od ostalih vhodov

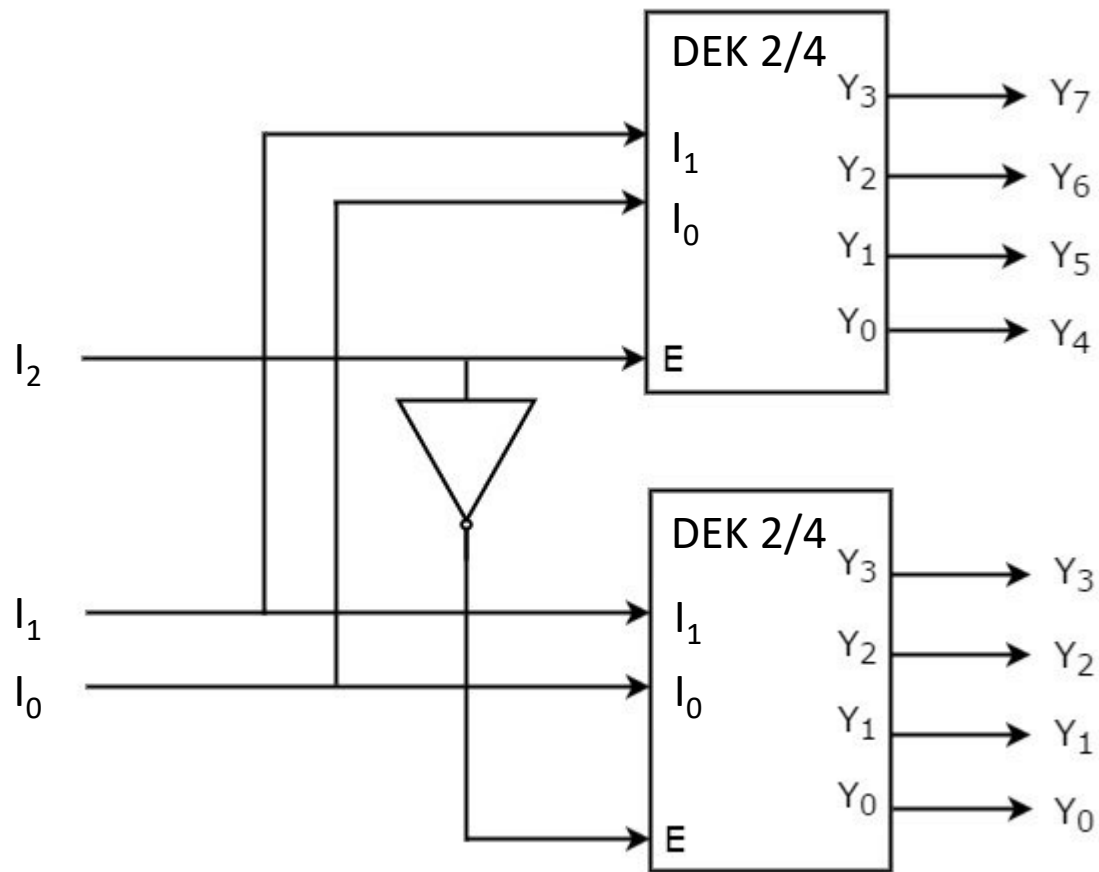


Dekodirnik 2/4



E	I_1	I_0	Y_0	Y_1	Y_2	Y_3
0	?	?	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Dekodirnik 3/8

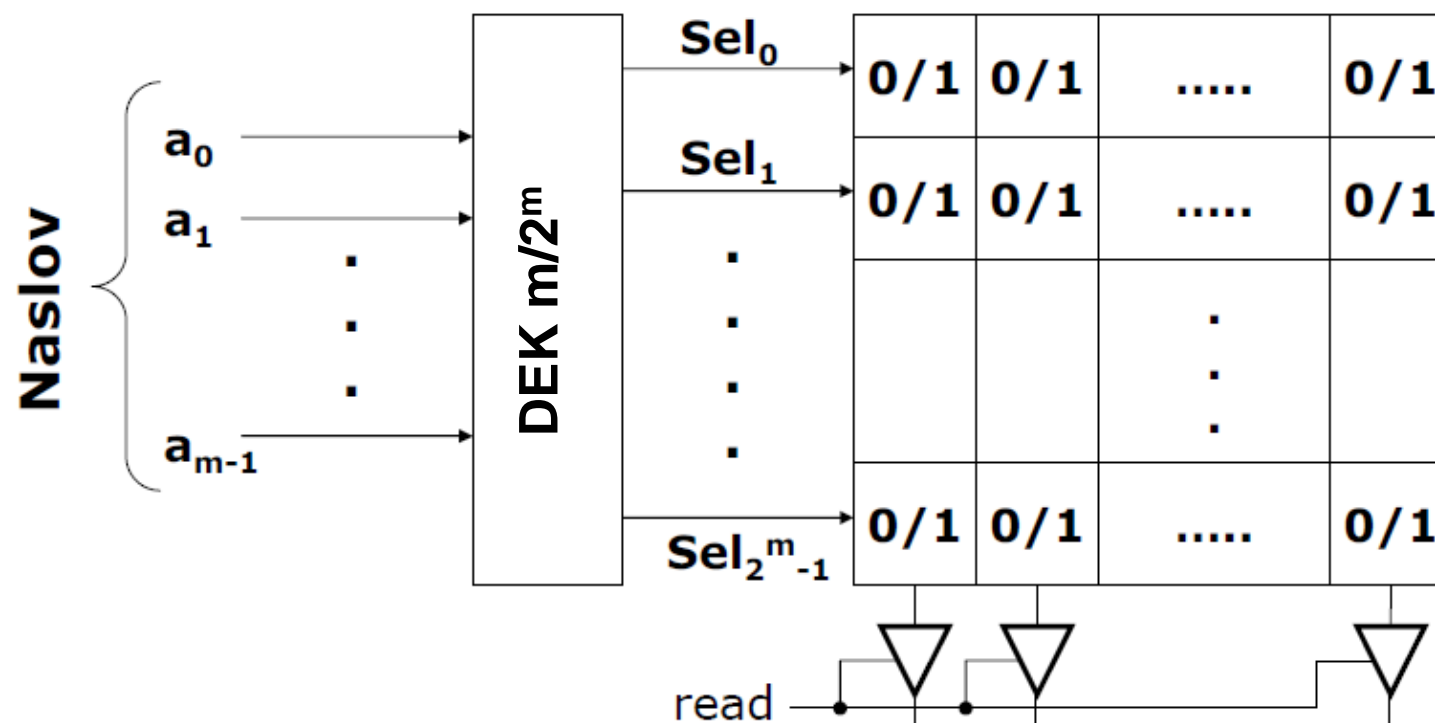


Dekodirnik $n/2^n$

Primer uporabe:

dekodiranje naslova v pomnilniku

izbira ene izmed 2^n pomnilniških besed na podlagi vrednosti na n-bitnem naslovnem vodilu

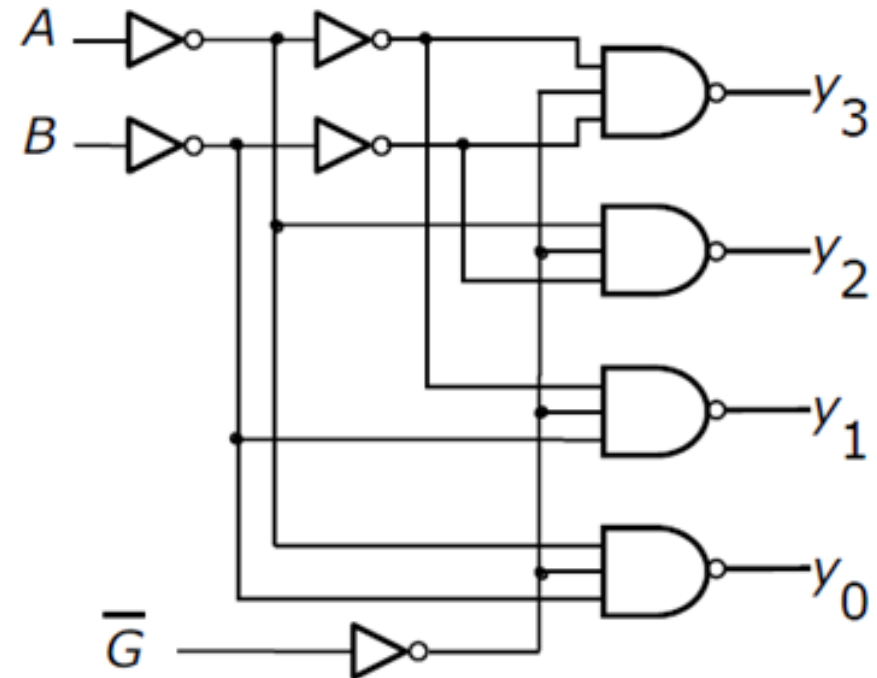
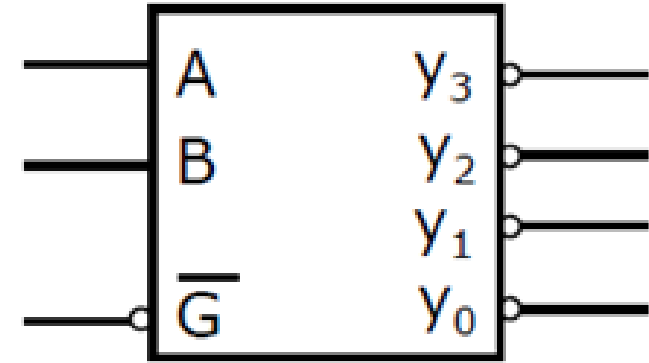


74139

- 2 x 2/4 dekodirnik (2 to 4)
- neinvertirajoči izbirni vhodi
- invertirajoči enable in izhodi

Inputs			Outputs			
Enable	Select		Y0	Y1	Y2	Y3
G	B	A				
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = High Level, L = Low Level, X = Don't Care



74138

- 3/8 dekodirnik
- 3 enable vhodi
 - 2 invertirana ($G2 = G2A + G2B$)
 - 1 neinvertiran ($G1$)
- 3 neinvertirani izbirni vhodi
- invertirani izhodi

Inputs					Outputs							
Enable		Select										
G1	G2 (Note 8)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

H = High Level, L = Low Level, X = Don't Care

Note 8: $G2 = G2A + G2B$

Kodirnik $n/2^n$

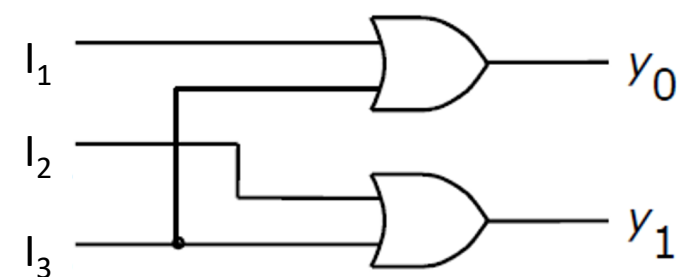
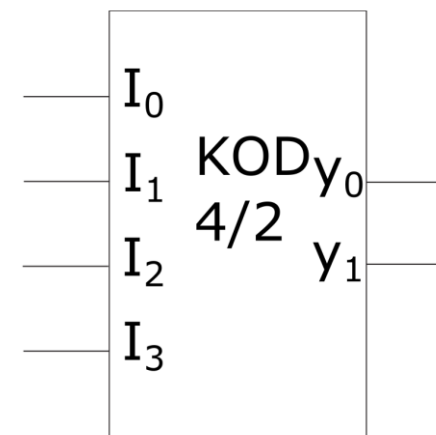
Obratna funkcija od dekodirnika – mintermski vhod preslikamo v minterm

Problem: kaj, če je aktivnih več vhodov?

Prioritetni kodirnik

4/2 binarni kodirnik

I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1



Deluje samo v primeru, da je naenkrat aktiven samo en vhod. Zato se uporablja prioritetni kodirnik!

Kodirnik prioritete 4/2

Naj ima l_3 najvišjo prioriteto izhod z določa kdaj nobeden od vhodov ni aktiven

l_3	l_2	l_1	l_0	y_1	y_0	z
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	?	0	1	1
0	1	?	?	1	0	1
1	?	?	?	1	1	1

74148

- 8/3 prioritetni kodirnik (I_7 ima najvišjo prioriteto)
- Invertirajoči vhodi in izhod

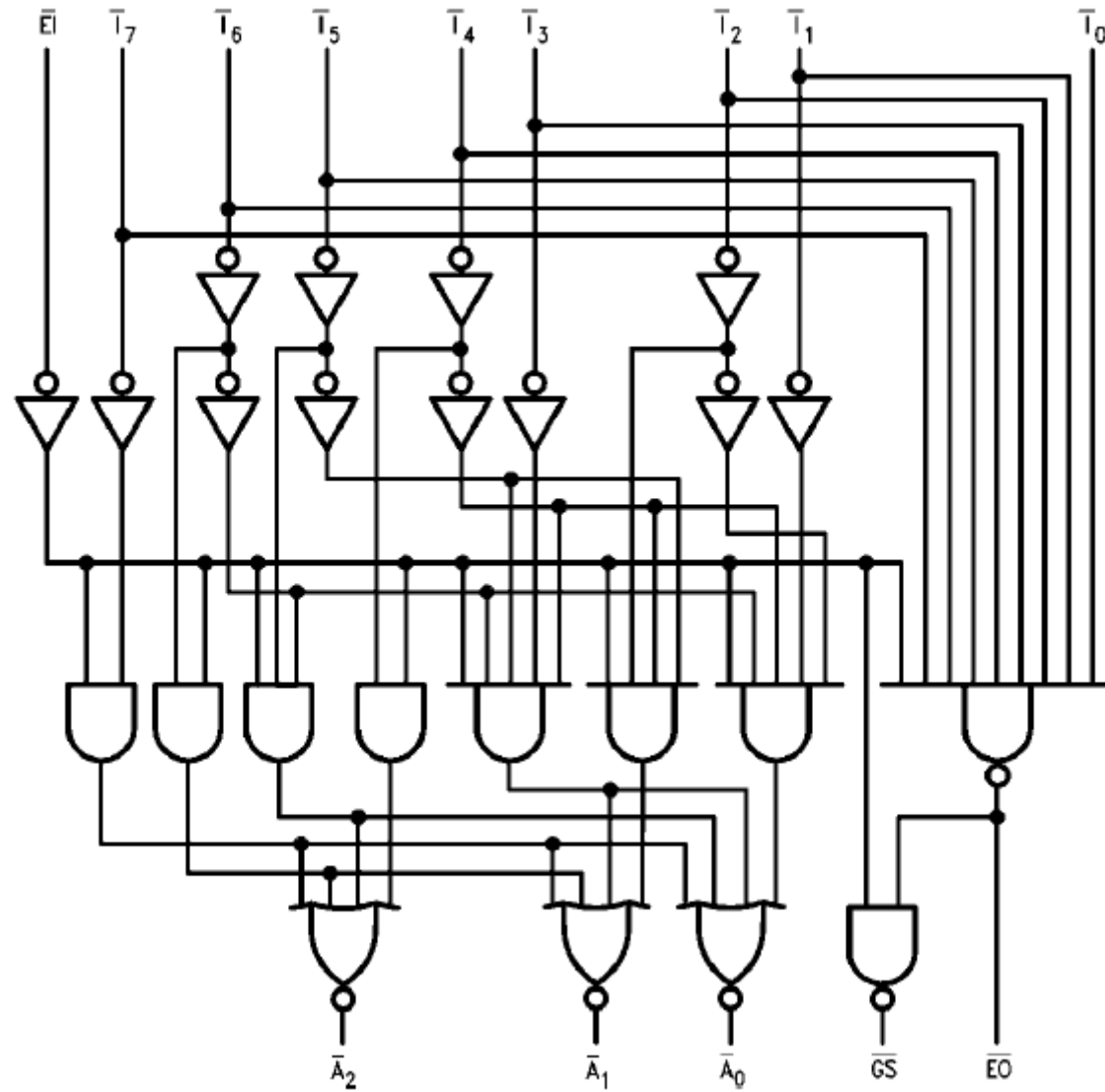
GS: veljaven izhod
(*group select*)

EO: vsi vhodi neaktivni
(*z*) (*enable output*)

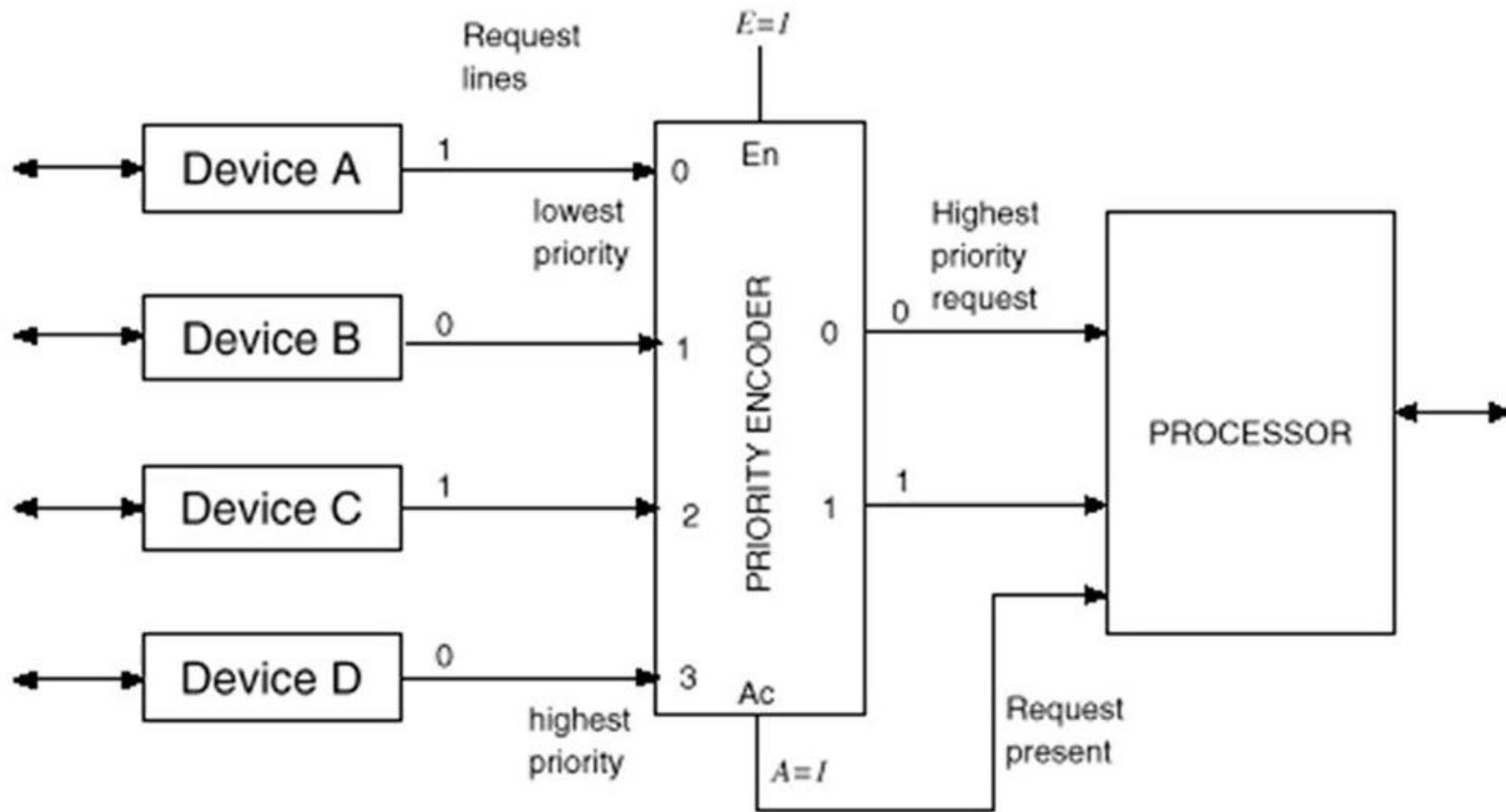
Inputs									Outputs				
\overline{EI}	\overline{I}_0	\overline{I}_1	\overline{I}_2	\overline{I}_3	\overline{I}_4	\overline{I}_5	\overline{I}_6	\overline{I}_7	\overline{GS}	\overline{A}_0	\overline{A}_1	\overline{A}_2	\overline{EO}
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	H	L	L	H
L	X	X	X	X	X	L	H	H	L	L	H	L	H
L	X	X	X	L	H	H	H	H	L	L	L	H	H
L	X	X	L	H	H	H	H	H	L	H	L	H	H
L	X	L	H	H	H	H	H	H	L	L	H	H	H
L	L	H	H	H	H	H	H	H	L	H	H	H	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

74148

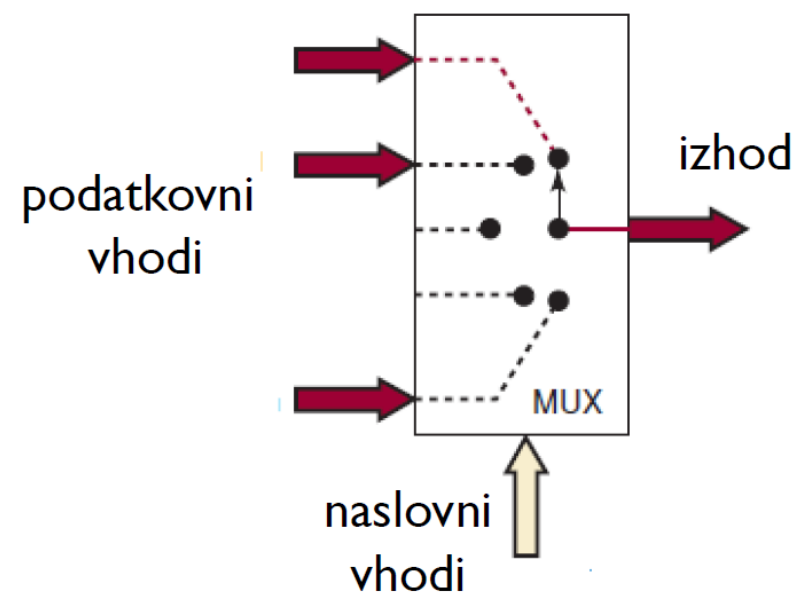
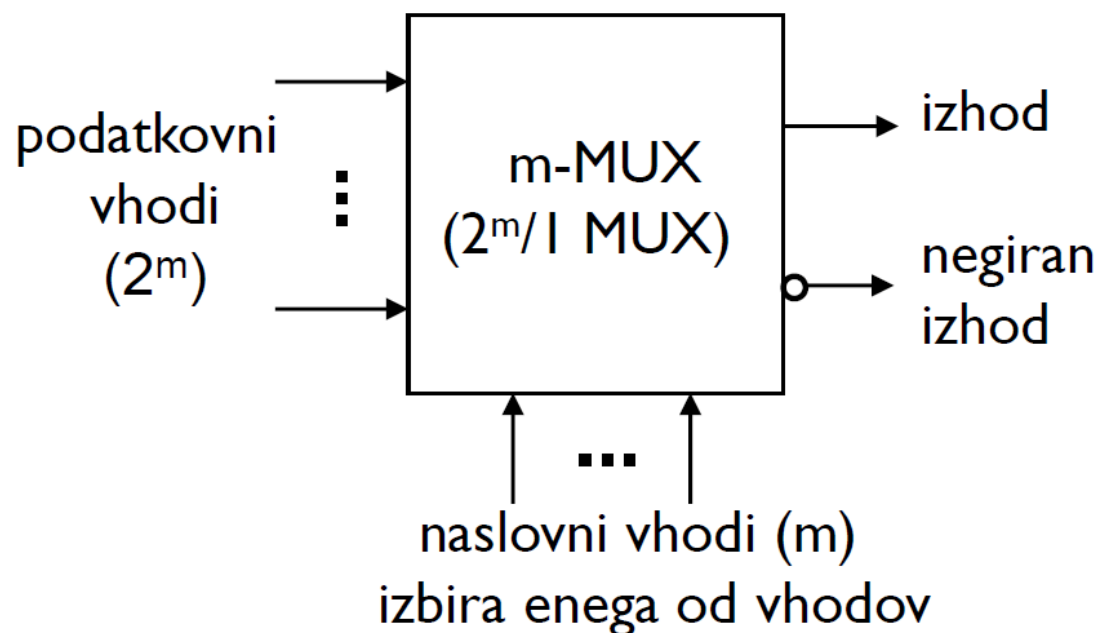


Kodirnik prioritete: primer uporabe (prekinitvene zahteve)



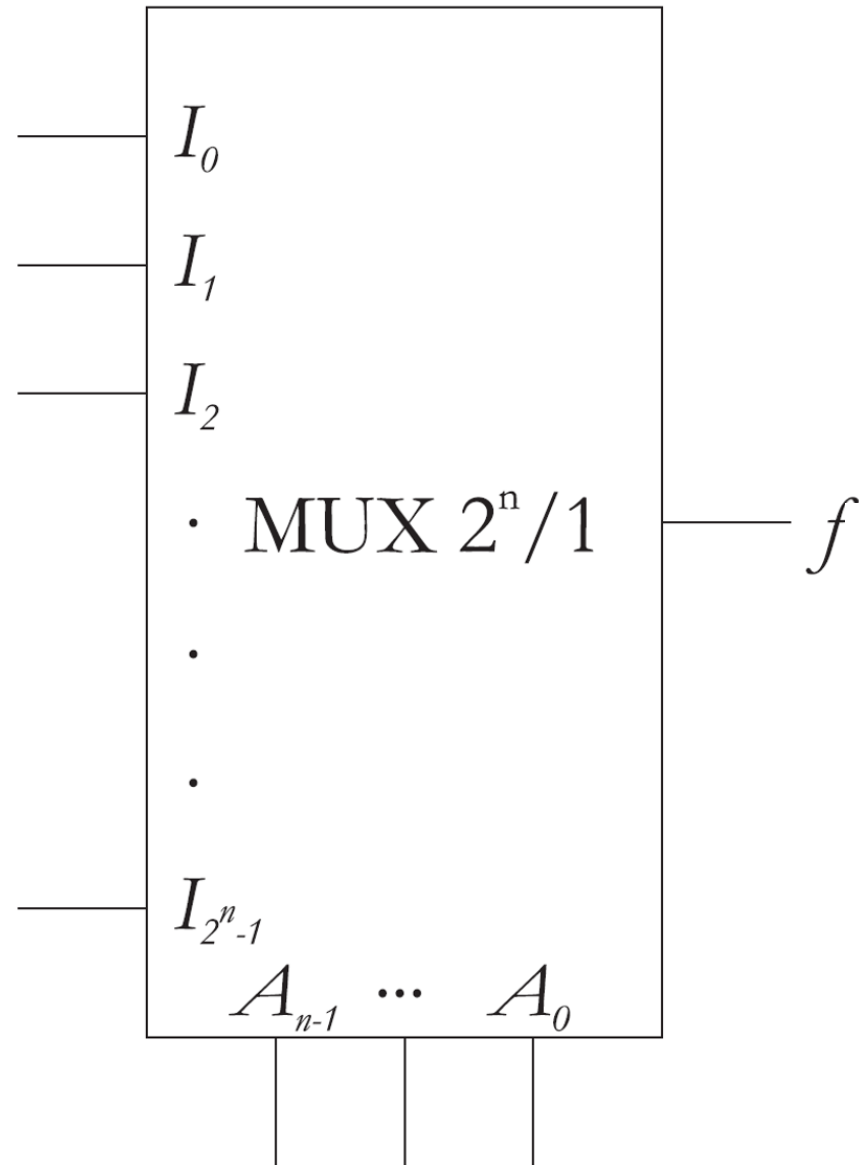
Multiplekser $2^n/1$

Preslikava enega izmed 2^n podatkovnih vhodov v 1 podatkovni izhod na podlagi n naslovnih vhodov in



Multiplekser $2^n/1$

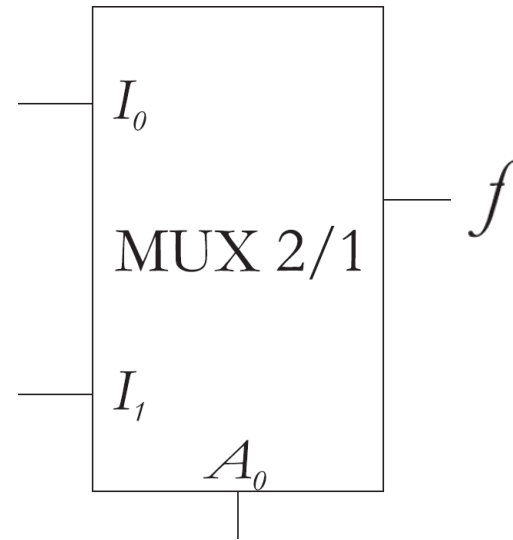
A_{n-1}	A_{n-2}	...	A_0	f
0	0	...	0	I_0
0	0	...	1	I_1
·	·	...	·	·
·	·	...	·	·
·	·	...	·	·
1	1	...	0	I_{2^n-2}
1	1	...	1	I_{2^n-1}



MUX 2/1

A_0	f
0	I_0
1	I_1

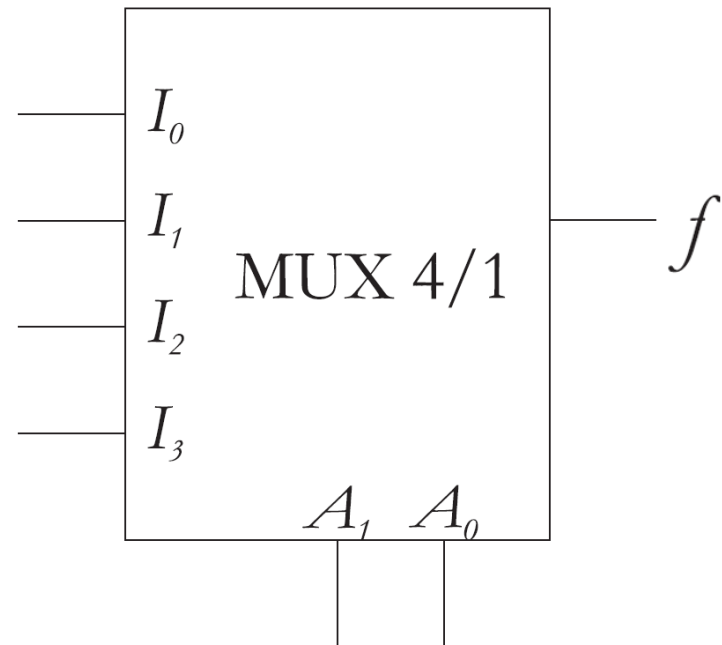
$$f = \overline{A_0}I_0 \vee A_0I_1$$



MUX 4/1

A_1	A_0	f
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

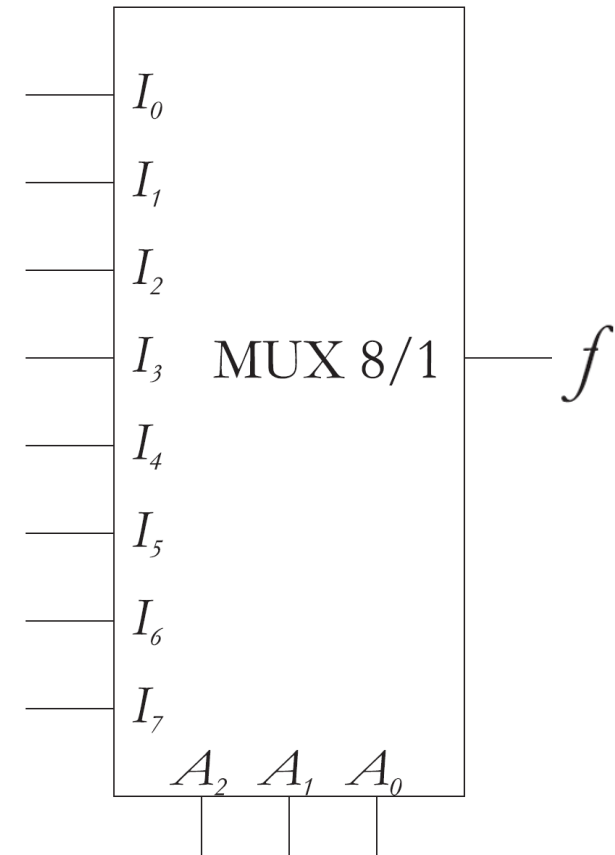
$$f = \bar{A}_1 \bar{A}_0 I_0 \vee \bar{A}_1 A_0 I_1 \vee A_1 \bar{A}_0 I_2 \vee A_1 A_0 I_3$$



MUX 8/1

A_2	A_1	A_0	f
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

$$f = \bar{A}_2 \bar{A}_1 \bar{A}_0 I_0 \vee$$
$$\bar{A}_2 \bar{A}_1 A_0 I_1 \vee$$
$$\bar{A}_2 A_1 \bar{A}_0 I_2 \vee$$
$$\bar{A}_2 A_1 A_0 I_3 \vee$$
$$A_2 \bar{A}_1 \bar{A}_0 I_4 \vee$$
$$A_2 \bar{A}_1 A_0 I_5 \vee$$
$$A_2 A_1 \bar{A}_0 I_6 \vee$$
$$A_2 A_1 A_0 I_7$$



Ločenje (Shannova dekompozicija)

Z ločenjem funkcije po eni spremenljivki pravilnostno tabelo razbijemo na dva dela:

- del, v katerem je spremenljivka enaka 0
- del, v katerem je spremenljivka enaka 1

Ločenje in multiplekser

Z vezavo vhodne spremenljivke na naslovne vhode multiplekserja 2/1 lahko funkcijo po tej spremenljivki ločimo – tabelo razbijemo na dva dela

Na podatkovne vhode moramo potem vezati t.i. funkcijske ostanke

Če uporabimo večji multiplekser (npr. 4/1), funkcijo ločimo po več (npr. 2) spremenljivkah naenkrat – pravilnostno tabelo razbijemo na več delov

Izvedba funkcij z multiplekserji

Več možnosti – glede na velikost multiplekserjev in kompleksnost funkcije (število vhodnih spremenljivk n)

- **Trivialna izvedba:** število vhodnih spremenljivk je enako številu naslovnih vhodov multiplekserja; primer: $n=3$, MUX 8/1
- **Optimalna izvedba:** število vhodnih spremenljivk je za ena večje kot število naslovnih vhodov multiplekserja; primer: $n=2$, MUX 4/1
- **Kaskadna izvedba:** kaskadna vezava več multiplekserjev, ko je število vhodnih spremenljivk več kot za ena večje od števila naslovnih vhodov; primer: $n=3$, MUX 2/1.
- **Minimalna izvedba:** najmanjše število multiplekserjev določene velikosti.

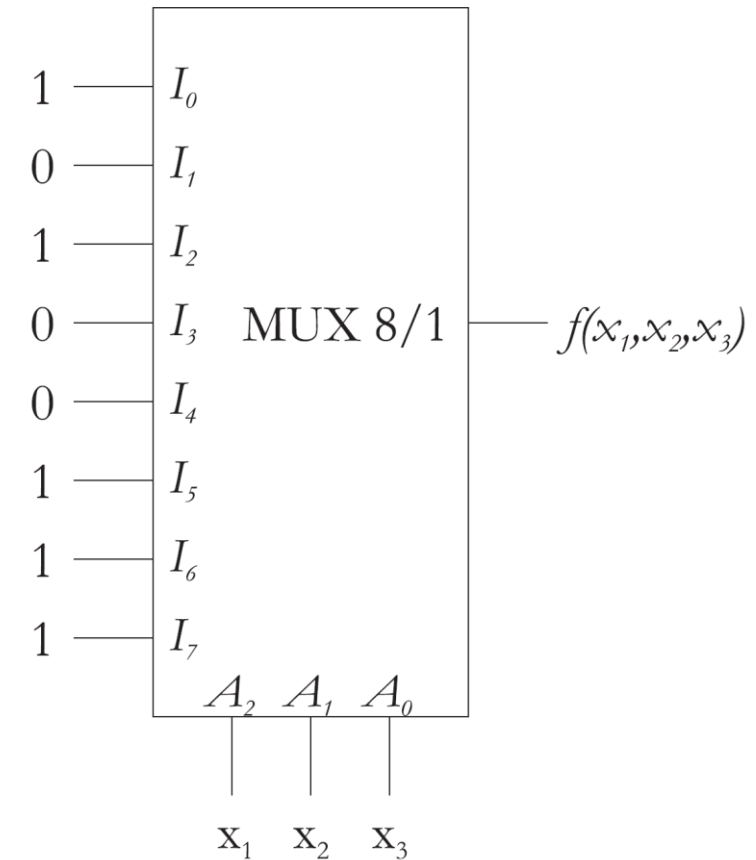
Trivialna izvedba funkcij z multiplekserji

Primer: $\vee^3(0, 2, 5, 6, 7)$ z *MUX* 8/1

3 vhodne spremenljivke, 3 adresni vhodi

Vhodne spremenljivke vežemo kar na adresne vhode, funkcijske vrednosti pa na podatkovne vhode

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Optimalna izvedba funkcij z multiplekserji

Uporabimo multiplekserjev in negatorje

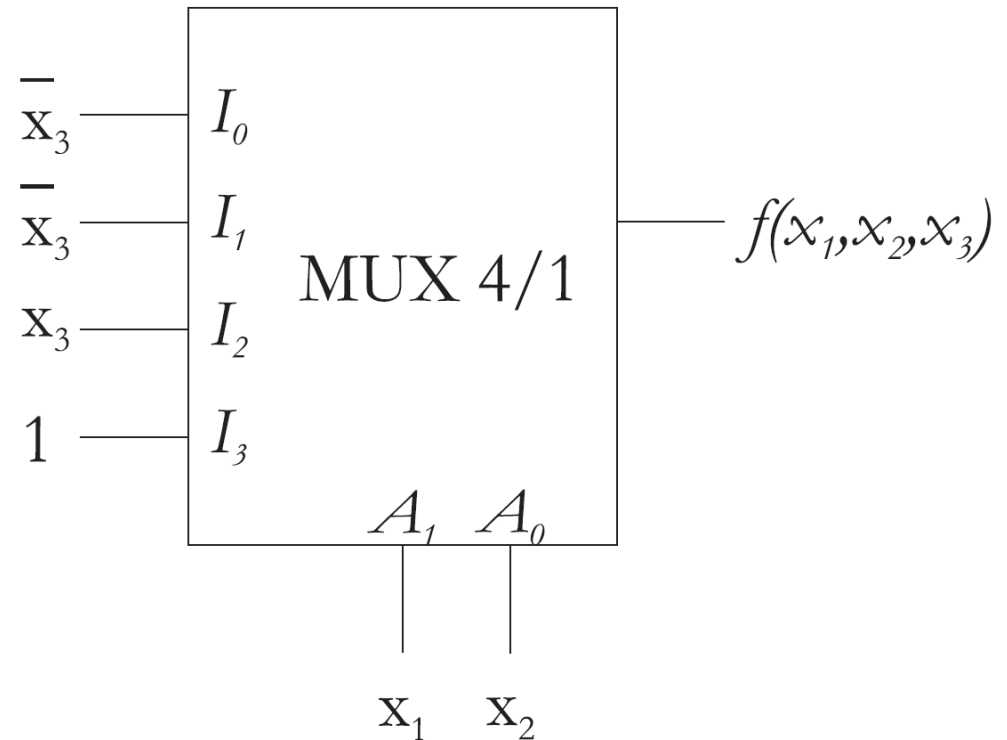
Funkcijo ločimo preko $n-1$ spremenljivke in funkcijske ostanke izrazimo s kontantami 0, 1 in negacijo spremenljivke, ki je ostala

Optimalna izvedba funkcij z multiplekserji

Primer: $\vee^3(0, 2, 5, 6, 7)$ z MUX 4/1

Ločenje preko x_1 in x_2 :

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	
0	0	0	1	$I_0 = \bar{x}_3$
0	0	1	0	
0	1	0	1	$I_1 = \bar{x}_3$
0	1	1	0	
1	0	0	0	$I_2 = x_3$
1	0	1	1	
1	1	0	1	$I_3 = 1$
1	1	1	1	



Kaskadna/minimalna izvedba funkcij z multiplekserji

Ni univerzalnega recepta

Iščemo spremenljivk(o/e), preko kater(e/ih) lahko funkcijo čim lepše ločimo.

Iščemo čimmanjši nabor multiplekserjev, s katerimi lahko funkcijo realiziramo.

Izrazimo ostanke.

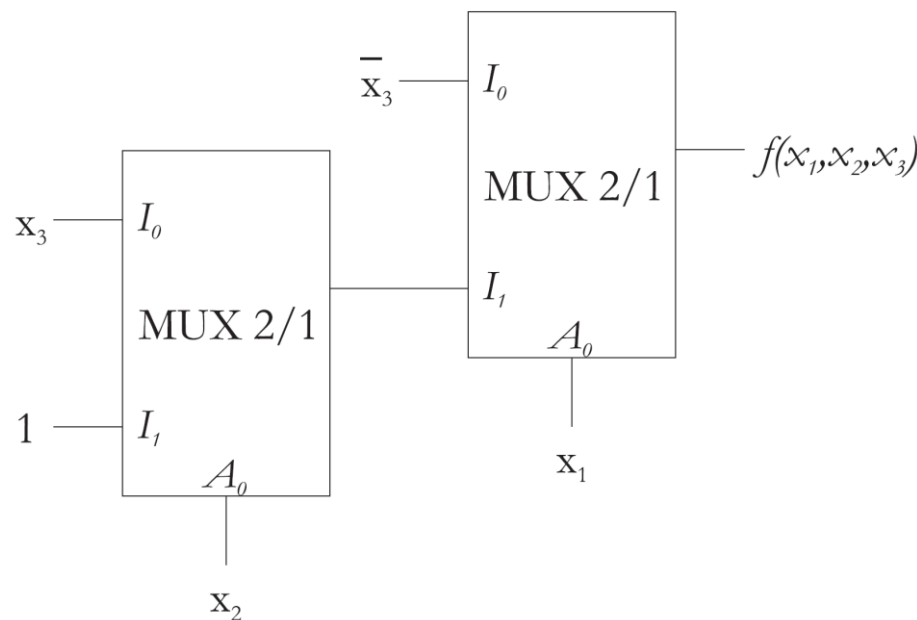
Če je potrebnih več multiplekserjev, te vežemo kaskadno – ostanke po potrebi ponovno ločimo.

Kaskadna izvedba funkcij z multiplekserji

Primer: $V^3(0, 2, 5, 6, 7) \approx MUX\ 2/1$

Najprej ločimo preko spremenljivke x_1 . Zgornji del lahko izrazimo z negacijo x_3 . Spodnji del moramo ločiti še enkrat \rightarrow preko x_2 .

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

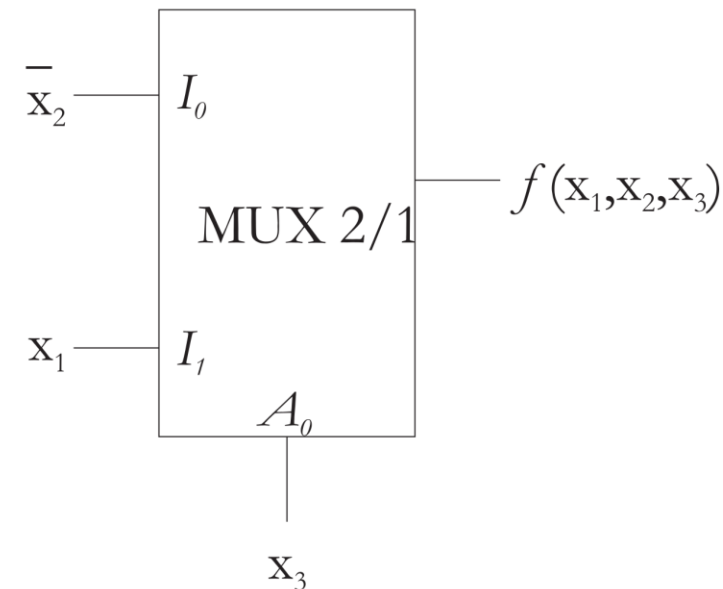


Minimalna izvedba funkcij z multiplekserji

Primer: $\vee^3(0, 4, 5, 7)$ z MUX 2/1

Najlepši rezultat dobimo, če funkcijo ločimo preko spremenljivke x_3 .

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	
0	0	0	1	\bar{x}_2
0	0	1	0	x_1
0	1	0	0	\bar{x}_2
0	1	1	0	x_1
1	0	0	1	\bar{x}_2
1	0	1	1	x_1
1	1	0	0	\bar{x}_2
1	1	1	1	x_1



Demultiplekser $1/2^n$

Dekodirnik, pri katerem si kontrolni vhod interpretiramo kot podatkovnih vhod

Podatkovni vhod preslika na en izhod v odvisnosti od stanja izbirnih vhodov

A_1	A_0	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1