

## 5. Deskodegailuak

Kode bitar bati dagokion zenbakia zirkuituaren sarreran adierazten denean eta konbinazio hori identifikatzen duen irteera aktibatzen denean, orduan deskodegailua deritzogun zirkuitua daukagu.

### 5.1. Diseinua

Oinarrizko bi deskodegailuen diseinua aztertzen dira hurrengo ataletan.

#### **BCD – Hamartar Deskodegailua = 4-10 Deskodegailua**

Sarreran lau bit aurkituko dira, izan ere, BCD kodean idatzitako zenbakia sartu behar da. Halaber, irteeran 10 bit izango dira, BDC kodeari dagokion konbinazio kopurua baitira. Irteerak maila baxuan aktiboak direnez eta 11 eta 15 arteko konbinazioak ezin direnez inoiz agertu zirkuituaren sarreretan, egia-taula hau da.

*Egia-taula*

Sarrerak				Irteerak
A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	0 1 2 3 4 5 6 7 8 9
0	0	0	0	0 1 1 1 1 1 1 1 1 1
0	0	0	1	1 0 1 1 1 1 1 1 1 1
0	0	1	0	1 1 0 1 1 1 1 1 1 1
0	0	1	1	1 1 1 0 1 1 1 1 1 1
0	1	0	0	1 1 1 1 0 1 1 1 1 1
0	1	0	1	1 1 1 1 1 0 1 1 1 1
0	1	1	0	1 1 1 1 1 1 0 1 1 1
0	1	1	1	1 1 1 1 1 1 1 0 1 1
1	0	0	0	1 1 1 1 1 1 1 1 0 1
1	0	0	1	1 1 1 1 1 1 1 1 1 0

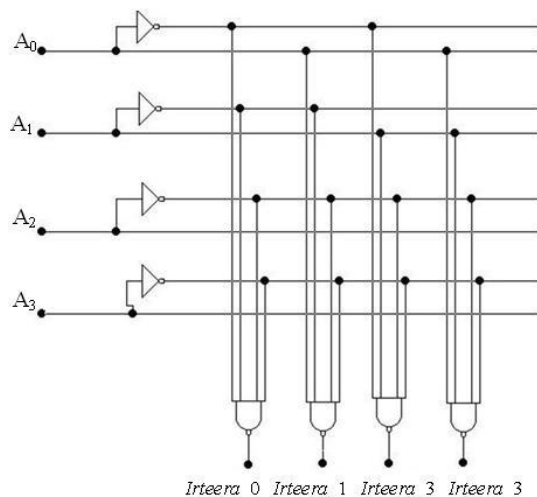
Egia-taulatik lortzen diren funtzio logikoak hauek dira:

$$Irteera\_0 = \overline{\overline{A_3 A_2 A_1 A_0}} = \overline{m_0} = M_0$$

$$Irteera\_1 = \overline{\overline{A_3 A_2 A_1 A_0}} = \overline{m_1} = M_1$$

....

Funtzio hauei dagokien zirkuitua 102. Irudian garatu da.



## 102. Irudia

### 4-16 Deskodegailua

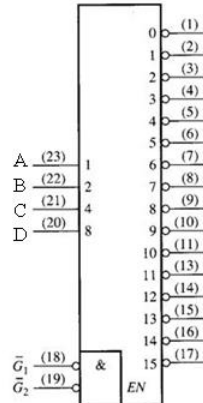
Sarreran lau bit aurkituko dira; izan ere, bitar natural kodean idatzitako zenbakia sartu behar da. Halaber, irteeran 16 bit izango dira bitar natural kodeari dagokion konbinazio kopurua baita. Irteerak maila baxuan aktiboak direnez, egia-taula hau da.

*Egia-taula*

Digu	Sarrera bitarrak $A_3A_2A_1A_0$	Deskodifikazio -funtzioa	Irteerak															
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0000	$\bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0001	$\bar{A}_3\bar{A}_2\bar{A}_1A_0$	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0010	$\bar{A}_3\bar{A}_2A_1\bar{A}_0$	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0011	$\bar{A}_3\bar{A}_2A_1A_0$	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
4	0100	$\bar{A}_3A_2\bar{A}_1\bar{A}_0$	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
5	0101	$\bar{A}_3A_2\bar{A}_1A_0$	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
6	0110	$\bar{A}_3A_2A_1\bar{A}_0$	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
7	0111	$\bar{A}_3A_2A_1A_0$	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
8	1000	$A_3\bar{A}_2\bar{A}_1\bar{A}_0$	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
9	1001	$A_3\bar{A}_2\bar{A}_1A_0$	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
10	1010	$A_3\bar{A}_2A_1\bar{A}_0$	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
11	1011	$A_3\bar{A}_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
12	1100	$A_3A_2\bar{A}_1\bar{A}_0$	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
13	1101	$A_3A_2\bar{A}_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
14	1110	$A_3A_2A_1\bar{A}_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
15	1111	$A_3A_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

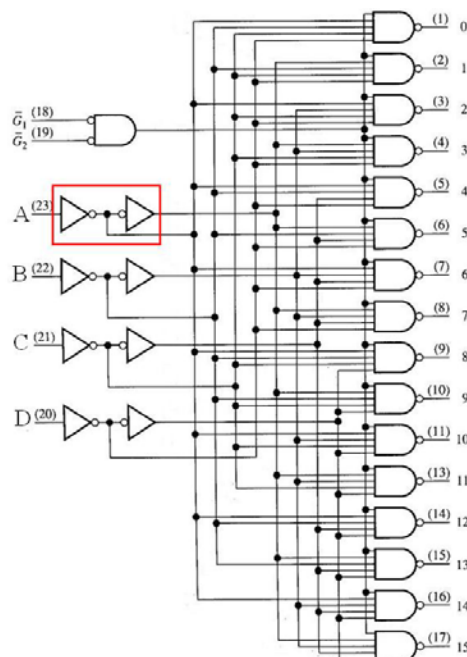
## 5.2. Zirkuitu komertzialak: 7442 eta 74154

74154 zirkuituari dagokion kapsulatua 103. Irudian erakusten da, eta beraren diagrama logikoa 104. Irudian azter daiteke. Han, deskodifikatzeko lau sarrera agertzen dira, eta beste bi, zirkuitua gaitzeko. Irteerak 16 dira.



103. Irudia

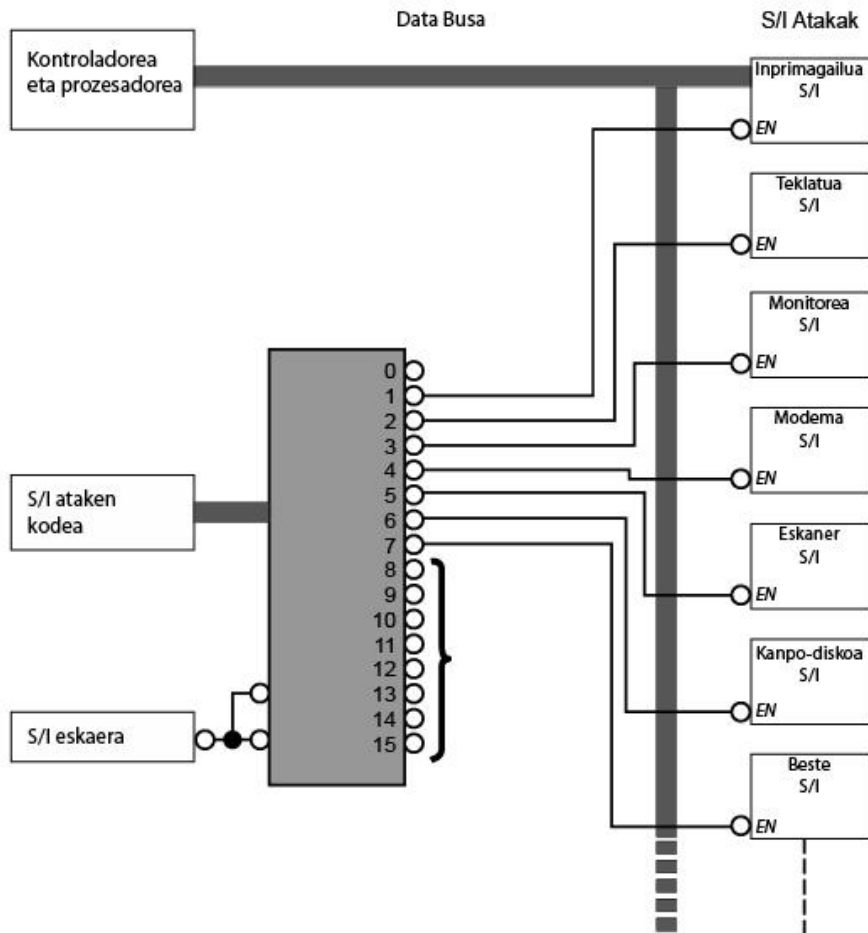
Deskodegailuaren sarrerak deskodegailura konektatzeaz gainera, beste gailuetara konekta daitezke. Hortaz, sarreraren *Fan-out-a* betetzen dela segurtatzeko, aipagarria da alderantzikagailu bikoitzaren erabilera (gorriz 104. Irudian). Alderantzikagailu bikoitzik gabe, Ai sarrera bakoitzak 7 sarrera kitzikatuko lituzke. Zenbaki handiagoak deskodifikatzeko, Ai sarrera beste deskodegailu baten sarreran konektatu behar denez, 14 ate kitzikatu beharko lituzke Ai sarrerak. Ostera, alderantzikagailu bikoitzarekin, Ai sarrerak bi sarrera besterik ez du kitzikatzen eta *fan-out-a* gainditzeko arriskua desagertzen da.



104. Irudia

### 5.3. Aplikazioa

Konputagailuen sarrerek eta irteerak aukeratzeko, deskodegailuak erabili ohi dira. Hau da, datuak konputagailutik periferiko batera bidaltzeko, periferikoa aukeratu behar da bere S/I portua aktibatuz. Prozedura hori deskodegailuarekin egin daiteke 105. Irudian egin den moduan. Portu bakoitzak portua identifikatzen duen helbide bat esleituta dauka, hots, zenbaki bat.



105. Irudia

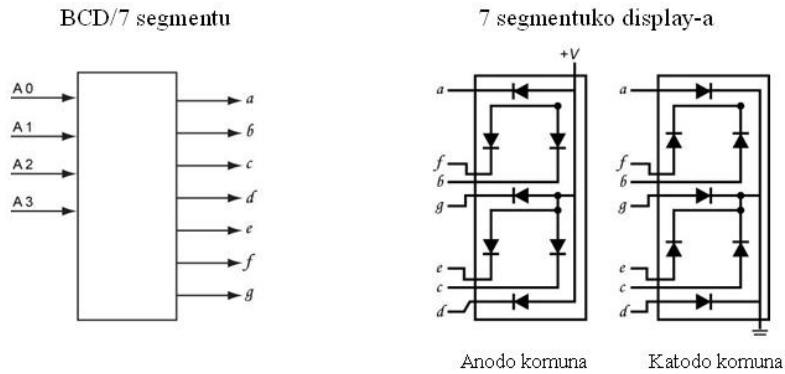
#### 5.4. VHDL deskribapena: 74138

```
library ieee;
use ieee.std_logic_1164.all;
entity decoder is
port(
entrada: in std_logic_vector(2 downto 0);
g1, g2: in std_logic;
salida: out std_logic_vector(7 downto 0)
);
end decoder;

1  architecture archidecoder of decoder is
2  begin
3  decoder: process (sarrera, g1, g2)
4  begin
5  if g2='1' then irteera<=(others=>'1');
6      elsif g2='0' and g1='0' then irteera<=(others=>'1');
7      elsif g2='0' and g1='1' then
8
9          irteera <= "01111111" when sarrera="000" else
10             "10111111" when sarrera ="001" else
11             "11011111" when sarrera ="010" else
12             "11101111" when sarrera ="011" else
13             "11110111" when sarrera ="100" else
14             "11111011" when sarrera ="101" else
15             "11111101" when sarrera ="110" else
16             "11111110" when sarrera ="111" else
17             "11111111";
18  end archidecoder;
```

### 5.5. BCD-7Segmentuko deskodegailua

BCD kodeari dagozkion konbinazioak dira integratu honen sarrerak. Bestalde, sarreran adierazitako zenbakiaren zenbaki hamartar baliokideak 7 segmentuko display batean adierazteko beharrezkoak diren 7 irteerak ditu (106. Irudia).



### 106. Irudia

#### Egia-taula

Deskodegailu honen egia-taula, 26. Taulan osatu da.

Sarrerak				Irteerak						
A3	A2	A1	A0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	0	0	1	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	0	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	x	x	x	x	x	x	x
...	...	...	...	...	...	...	...	...	...	...
1	1	1	1	x	x	x	x	x	x	x

### 26. Taula

Irteera bakoitzari dagokion funtzio logikoa hau da:

$$a = \overline{A_2} \overline{A_0} + A_2 A_0 + A_1 + A_4$$

$$b = \overline{A_0} + \overline{A_2}$$

$$c = A_2 + A_0 + \overline{A_1}$$

$$d = \overline{A_2} \overline{A_0} + \overline{A_2} A_1 + A_2 \overline{A_1} A_0 + A_1 \overline{A_0} + A_4$$

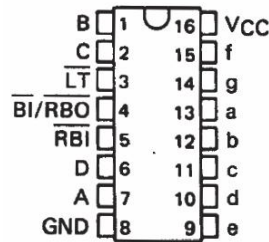
$$e =$$

$$f =$$

$$g =$$

## Zirkuitu komertziala: 7447

Zirkuitu honi dagokion kapsulatua 107. Irudian agertzen da.



### 107. Irudia

Funtzionamendu-taulari behatuz (108. Irudia), ikusten da irteerak maila baxuan aktiboak direla; hortaz, anodo komuneko display batera konektatu behar dira deskodegailu honen irteerak. Halaber nabaria da LT sarrera maila baxuan aktiboa dela eta displayaren led guztiak ondo daudela frogatzea dela beraren zeregina. Bukatzeko, aipatu beharra dago RBI sarrera eta BI/RBO hankatxoak adierazgarri ez diren zeroak ezabatzeko erabiltzen direla. Informazio hori guztia, 108. Irudian dagoen funtzionamendu-taulatik lor daiteke.

DECIMAL OR FUNCTION	INPUTS						$\overline{\text{BI}}/\overline{\text{RBO}}^\dagger$	OUTPUTS							NOTE
	$\overline{\text{LT}}$	$\overline{\text{RBI}}$	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF	1
1	H	X	L	L	L	H	H	OFF	ON	ON	OFF	OFF	OFF	OFF	
2	H	X	L	L	H	L	H	ON	ON	OFF	ON	ON	OFF	ON	
3	H	X	L	L	H	H	H	ON	ON	ON	ON	OFF	OFF	ON	
4	H	X	L	H	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON	
5	H	X	L	H	L	H	H	ON	OFF	ON	ON	OFF	ON	ON	
6	H	X	L	H	H	L	H	OFF	OFF	ON	ON	ON	ON	ON	
7	H	X	L	H	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF	
8	H	X	H	L	L	L	H	ON	ON	ON	ON	ON	ON	ON	
9	H	X	H	L	L	H	H	ON	ON	ON	OFF	OFF	ON	ON	
10	H	X	H	L	H	L	H	OFF	OFF	OFF	ON	ON	OFF	ON	
11	H	X	H	L	H	H	H	OFF	OFF	ON	ON	OFF	OFF	ON	
12	H	X	H	H	L	L	H	OFF	ON	OFF	OFF	OFF	ON	ON	
13	H	X	H	H	L	H	H	ON	OFF	OFF	ON	OFF	ON	ON	
14	H	X	H	H	H	L	H	OFF	OFF	OFF	ON	ON	ON	ON	
15	H	X	H	H	H	H	H	OFF	OFF	OFF	OFF	OFF	OFF	OFF	
BI	X	X	X	X	X	X	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
RBI	H	L	L	L	L	L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
LT	L	X	X	X	X	X	H	ON	ON	ON	ON	ON	ON	ON	4

H = high level, L = low level, X = irrelevant

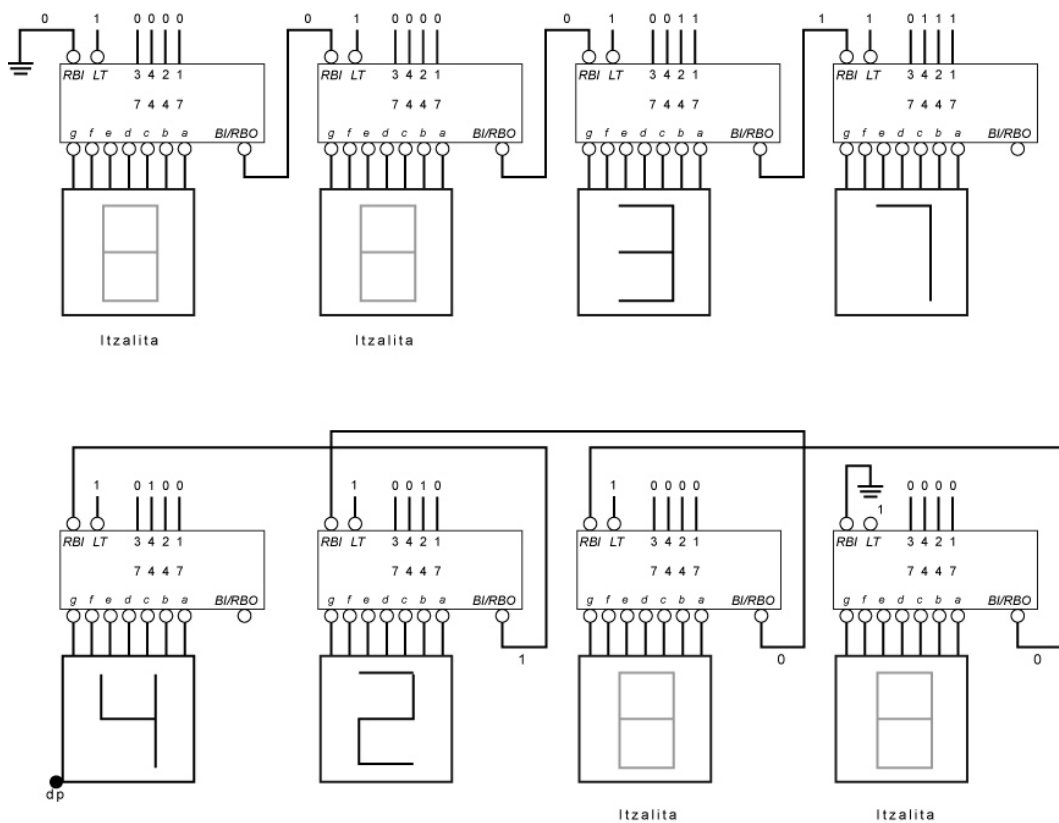
- NOTES: 1. The blanking input ( $\overline{\text{BI}}$ ) must be open or held at a high logic level when output functions 0 through 15 are desired. The ripple-blanking input ( $\overline{\text{RBI}}$ ) must be open or high if blanking of a decimal zero is not desired.  
 2. When a low logic level is applied directly to the blanking input ( $\overline{\text{BI}}$ ), all segment outputs are off regardless of the level of any other input.  
 3. When ripple-blanking input ( $\overline{\text{RBI}}$ ) and inputs A, B, C, and D are at a low level with the lamp test input high, all segment outputs go off and the ripple-blanking output ( $\overline{\text{RBO}}$ ) goes to a low level (response condition).  
 4. When the blanking input/ripple blanking output ( $\overline{\text{BI}}/\overline{\text{RBO}}$ ) is open or held high and a low is applied to the lamp-test input, all segment outputs are on.

$^\dagger \overline{\text{BI}}/\overline{\text{RBO}}$  is wire AND logic serving as blanking input ( $\overline{\text{BI}}$ ) and/or ripple-blanking output ( $\overline{\text{RBO}}$ ).

### 108. Irudia

## Aplikazioa

Adierazgarriak ez diren zeroak ezabatzeko konfigurazioak 109. Irudian azaltzen dira. Goiko konfigurazioan, zenbaki osoen ezker aldean agertzen diren zeroak ezabatzen dira, eta beheko konfigurazioan, koma hamartarraren eskuinaldeko zeroak ezabatzen dira. Zeroa ezabatzea zeroa agertzen denean deskodegailuaren irteerak desaktibatzen datza. Hori RBI sarreraren bidez lortzen da; hau da, RBI sarreraren balioaren arabera, irteerak desaktibatzen dira, edo zeroa lortzeko beharrezkoak diren irteerak aktibatzen dira. BI/RBO hankatxoaren zeroaren ezabapenari buruzko informazioa hedatzeko erabiltzen da.



109. Irudia



## 5.6. VHDL deskribapena: BCD 7/segmentu

```
library ieee;
use ieee.std_logic_1164.all;
entity aldaBCD is
port(
bcd: in bit_vector(3 downto 0);
led: out bit_vector(6 downto 0)
);
end aldaBCD;

1  architecture archialdaBCD of aldaBCD is
2  begin
3  alda: process (bcd)
4  begin
5  case bcd is
6      when "0000" => LED <= "1111110";
7      when "0001" => LED <= "1100000";
8      when "0010" => LED <= "1011011";
9      when "0011" => LED <= "1110011";
10     when "0100" => LED <= "1100101";
11     when "0101" => LED <= "0110111";
12     when "0110" => LED <= "0111111";
13     when "0111" => LED <= "1100010";
14     when "1000" => LED <= "1111111";
15     when "1001" => LED <= "1110111";
16     when others => LED <= "0000000";
17 end case;
18 end process alda;
19 end archialda;
```

### 5.7. Deskodegailuak erabiliz, funtzio logikoak garatu

Deskodegailua osatzeko erabiltzen diren ekuazio logikoei arretaz begiratu (11. Ekuazioa), erraz ikusten da funtzio logikoak garatzeko erabil daitezkeela.

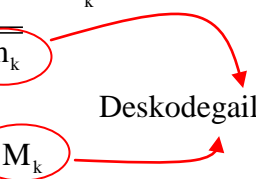
$$Irteera\_0 = \overline{\overline{A_3 A_2 A_1 A_0}} = \overline{m_0} = M_0$$

$$Irteera\_1 = \overline{\overline{A_3 A_2 A_1 A_0}} = \overline{m_1} = M_1$$

....

### 11. Ekuazioa

Hots, funtzio logikoa minterm-en arteko batuketan edo maxterm-en arteko biderketan bidez adierazten bada, funtzio horietan agertzen diren gaiak deskodegailuaren irteeren baliokideak direla ohar gaitzke.

Biderketen arteko batuketa moduko funtzio logikoa	$f(a, b, \dots, z) = m_i + m_j + \dots + m_k$ $f(a, b, \dots, z) = \overline{\overline{m_i \cdot m_j \cdot \dots \cdot m_k}}$	 <p>Deskodegailuaren irteerak</p>
Batuketan arteko biderketa moduko funtzio logikoa	$f(a, b, \dots, z) = M_i \cdot M_j \cdot \dots \cdot M_k$ $f(a, b, \dots, z) = \overline{\overline{m_i \cdot m_j \cdot \dots \cdot m_k}}$	

Hortaz, funtzio logikoaren minterm bakoitza deskodegailuaren irteera bat izango da, eta haiek, era berean, NAND atearen sarrerak. Halaber, maxterm bakoitza deskodegailuaren irteera bat izango da, eta haiek, era berean, AND atearen sarrerak.