

7. Bihurgailuak

Kode bitar batean adierazita dagoen zenbakia beste kode batean adierazten duen zirkuitua da, hots, zirkuituaren sarreretan kode batean idatzitako zenbaki bitarra jartzen da, eta irteeretan, zenbaki bera beste kode bitar batean idatzita agertuko da.

7.1. Diseinua

Bitar $\Rightarrow$ Gray bihurketa

Zirkuituari dagokion egia-taula, funtzio logikoak eta garapena 115. Irudian adierazten dira:

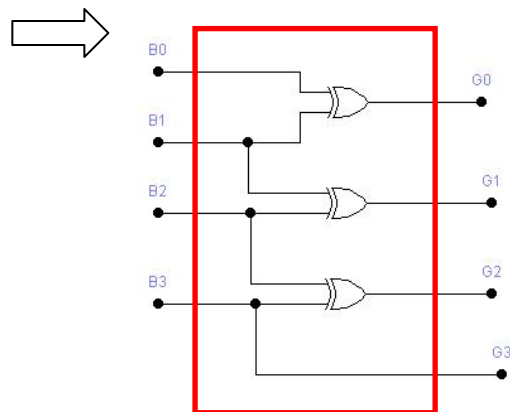
Bitar				Gray			
B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

$$G_0 = B_1 \oplus B_0$$

$$G_1 = B_1 \oplus B_2$$

$$G_2 = B_2 \oplus B_3$$

$$G_3 = B_3$$



115. Irudia

Gray $\Rightarrow$ Bitar bihurketa

Zirkuituari dagokion egia-taula, funtzio logikoak eta garapena 116. Irudian adierazten dira:

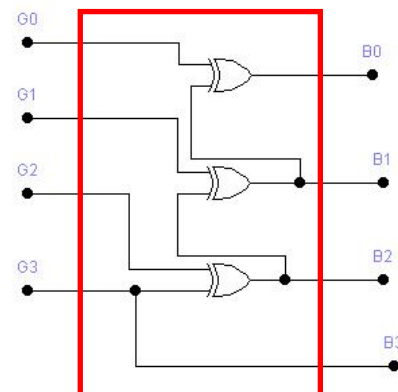
Gray				Bitar			
G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

$$B_0 = B_1 \oplus G_0$$

$$B_1 = G_1 \oplus B_2$$

$$B_2 = G_2 \oplus G_3$$

$$B_3 = G_3$$



116. Irudia

Bitar $\Rightarrow$ BCD bihurketa

Zirkuituari dagokion egia-aula 28. Taulan adierazten da:

B ₃	B ₂	B ₁	B ₀	D ₀	U ₃	U ₂	U ₁	U ₀	
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	0	1	0	2
0	0	1	1	0	0	0	1	1	3
0	1	0	0	0	0	1	0	0	4
0	1	0	1	0	0	1	0	1	5
0	1	1	0	0	0	1	1	0	6
0	1	1	1	0	0	1	1	1	7
1	0	0	0	0	1	0	0	0	8
1	0	0	1	0	1	0	0	1	9
1	0	1	0	1	0	0	0	0	16
1	0	1	1	1	0	0	0	1	17
1	1	0	0	1	0	0	1	0	18
1	1	0	1	1	0	0	1	1	19
1	1	1	0	1	0	1	0	0	20
1	1	1	1	1	0	1	0	1	21

28. Taula

Zirkuitua bi modutan gara daiteke. Alde batetik, irteerei dagozkien funtzio logikoak lor daitezke, hau da, egia-taularen informazioa Karnaugh mapara eroan irteerako funtzio logikoak murriztuta lortzeko. Bestalde, zirkuituak bete behar duen funtzioaren logika bloke-diagrama batekin azal daiteke, eta ondoren, bloke bakoitzaren diseinua garatu. Bigarren aukera hori zehazki azaldu behar da; izan ere, prozedura berria da. 28. Taulari arretaz behatuz gero, nabaria da 0tik 9ra bitarteko sarreretan, sarreraren eta irteeraren balioak berdinak direla, eta 10etik gorako sarreretan sarreraren eta irteeraren balioak desberdinak direla. Desberdintasun horrek logikaren bat jarraitzen duen aztertu behar da. Horretarako, sarrera-irteera konbinazio bikote bakoitza bitar natural kodea erabiliz

irakurtzen da, eta 6ko diferentzia dagoela ondorioztatzen da. Gertaera hori eskematikoki azalduta:

Baldin $B \in [0-9]$, orduan, $DU = B + 0$

Baldin $B \in [10-18]$, orduan, $DU = B + 6$

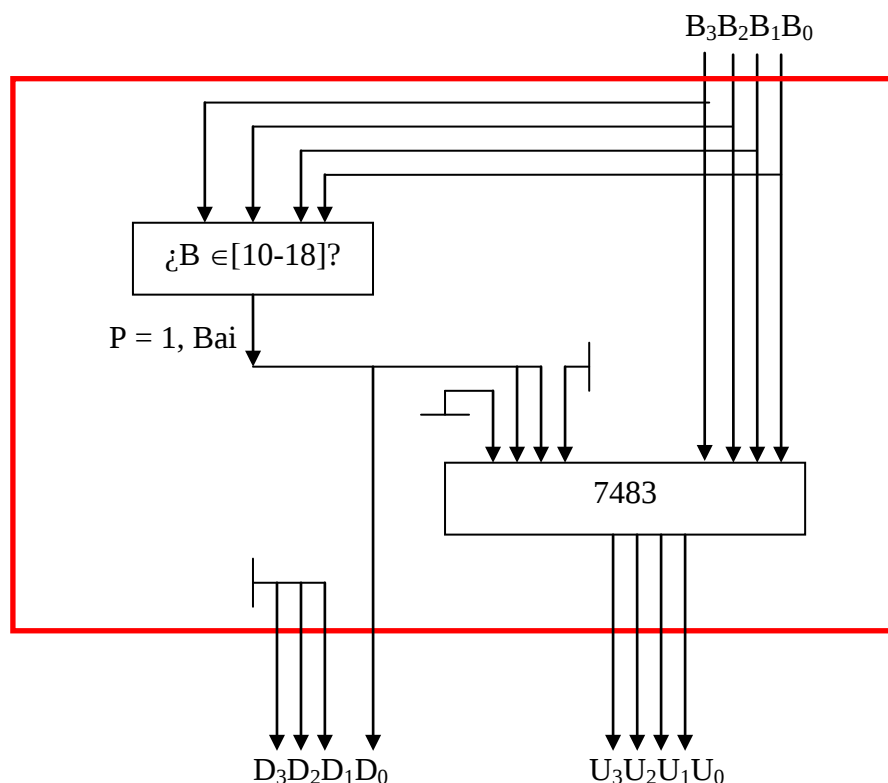
Non, DU zirkuituaren irteera izango den, eta B zirkuituaren sarrera. Hortaz, irteeraren balioa lortzeko, sarrerako balioari batugai aldakorra (0 edo 6) gehitu behar zaio. Batugai aldakorrak har ditzakeen balioen adierazpen bitarra kontuan izaten bada, konturatzen gara bi bit baino ez direla ezberdinak:

$$U_3U_2U_1U_0 = B_3B_2B_1B_0 + 0000$$

$$U_3U_2U_1U_0 = B_3B_2B_1B_0 + 0110$$

Halaber, $B \in [10-18]$ gertatzen denean ohartzen gara bit aldakorraren balioa 1 dela.

Aurreko paragrafoetan adierazitako logikarekin bat etorritz, 117. Irudian bihurgailuari dagokion bloke-diagrama egin da, non P irteerak adieraziko duen $B \in [10-18]$ baldintza, eta era berean, bit aldakorraren balioa.



117. Irudia

BCD $\Rightarrow$ bitar bihurketa.

Bihurketa hori garatzeko erabili behar den algoritmoa hurrengo lerroetan azaltzen da analitikoki.

$$BA_{\text{ham}} = B_3 B_2 B_1 B_0 A_3 A_2 A_1 A_0 \text{ BCDnat}$$

Non BA_{ham} bi digituko zenbaki hamartarra den, eta $B_3 B_2 B_1 B_0 A_3 A_2 A_1 A_0 \text{ BCDnat}$ zenbaki horren adierazpen bitarra BCD naturalean kodetua.

$$\begin{aligned} BA_{\text{ham}} &= B \cdot 10 + A = B \cdot (2^3 + 2^1) + A = B \cdot 2^3 + B \cdot 2^1 + A = \\ &= B_3 B_2 B_1 B_0 0 0 0 + B_3 B_2 B_1 B_0 0 + A_3 A_2 A_1 A_0 \end{aligned}$$

Hortaz, zirkuituak garatuko duen batuketa hau da:

$$\begin{array}{r} B_3 \ B_2 \ B_1 \ B_0 \ 0 \ 0 \ 0 \\ B_3 \ B_2 \ B_1 \ B_0 \ 0 \\ A_3 \ A_2 \ A_1 \ A_0 \\ \hline R_7 \ R_6 \ R_5 \ R_4 \ R_3 \ R_2 \ R_1 \end{array}$$

Horretarako, bi batutzaile beharrezkoak dira; beraz, 7483a erabiliz gero, eta kontuan izanda barneko carry-ak daudela, eragiketak hauek izango dira:

$$R_1 = A_0$$

$$R_2 = A_1 + B_0 \Rightarrow C_{11}$$

$$R_3 = A_2 + B_1 (+C_1) \Rightarrow C_{21}$$

$$R_4 = A_3 + B_0 + B_2 (+C_2) \Rightarrow C_{31}, C_{12}$$

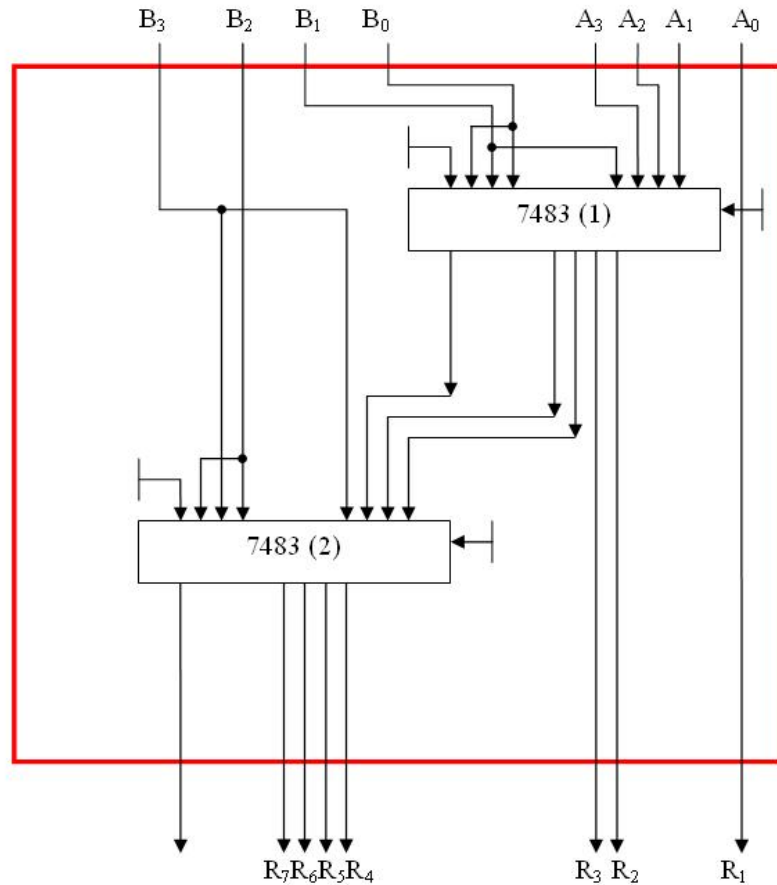
$$R_5 = B_1 (+C_{31}) + B_3 (+C_{12}) \Rightarrow C_{41}, C_{22}$$

$$R_6 = B_2 (+C_{41}) \Rightarrow C_{32}$$

$$R_7 = B_3 (+C_{32})$$

Non eragiketa bakoitzak sortzen duen bururakoa C_{ij} adierazpenaren bidez azaltzen den, eta haren azpiindizeen esanahia, batutzailea identifikatzeko, $j = 1, 2$, eta, batutzailearen barruan sortutako bururakoaren posizioa ezagutzeko, $i = 1, 2, 3, 4$ diren.

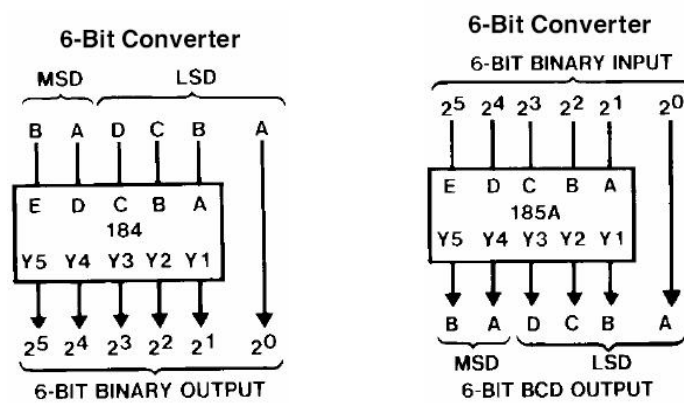
Hortaz, eragiketa horiei dagokien zirkuitua 118. Irudian azter daiteke.



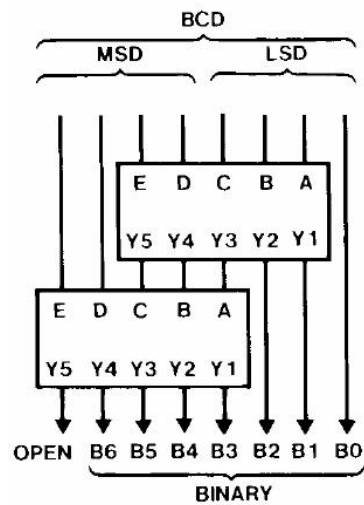
118. Irudia

7.2. Zirkuitu komertzialak: 74185 eta 74184

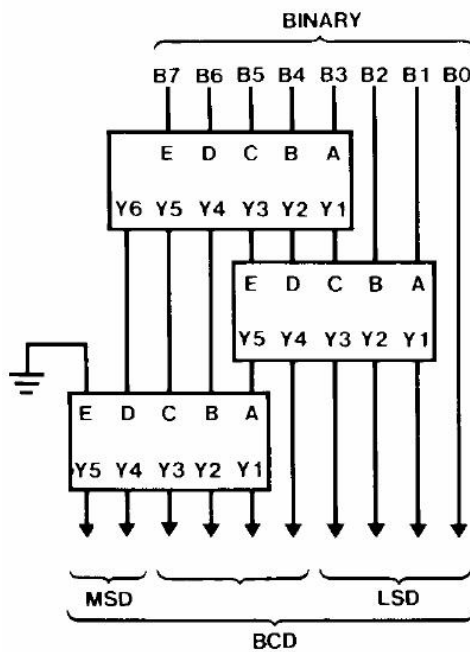
Kode-bihurgailuak garatzeko, MSI zirkuituak erabili beharrean, gailu programagarriak erabiltzen dira. Hala ere, erabilgarri diren bi MSI zirkuituak 74185 eta 74184 dira (119. Irudia), BCD eta bitar-natural kodeen arteko bihurtza egiten dutenak. Sarrerek onartzen duten zenbaki handiena baino zenbaki handiagoa bihurtu nahi bada, bi zirkuitu konbina daitezke 120. Irudian eta 121. Irudian egin den moduan. Lehenengo kasuan, BCD bi digitutik bitarrera bihurtzeko zirkuitua lortzeko, bi 74184 integratu erabili dira; bigarren kasuan, 8 biteko zenbaki bitarra BCD kodera bihurtzeko, hiru 74185 behar izan dira.



119. Irudia



120. Irudia



121. Irudia