

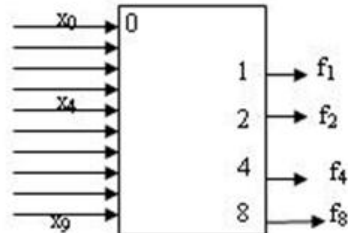
6. Kodegailua

Deskodegailuaren alderantzizko funtzioa du, hots, sarreraren bat aktibatuz gero, sarrera horri dagokion zenbaki hamartarraren adierazpen bitarra agertuko da irteeran.

6.1. Diseinua

Lehentasunik gabeko kodegailua. Adibidea: Hamartar-BCD

Aldi berean sarrera bat baino gehiago aktibatzeko aukera ez duen kodegailuari lehentasunik gabeko kodegailu deritzogu. Demagun hamar sarrera dituen kodegailua. Orduan, sarrera-konbinazio posibleak 110. Irudiaren funtzionamendu-taulan agertzen dira. Irteerako funtzio logikoak lortzeko, taula hori aztertu behar da, eta bertatik funtzioak ondorioztatu; izan ere, ezinezkoa da zortzi sarrera dituen egia-taula adieraztea eta Karnaugh mapak egitea (2^8 sarrera-konbinazio behar baitira). Hortaz, 110. Irudiaren funtzionamendu-taularen arabera esan daiteke f_8 irteera aktibatuko dela 8. edo 9. irteerak aktibatzen badira, f_4 irteera aktibatuko dela 4. edo 5. edo 6. edo 7. irteerak aktibatzen badira, etab... Esaldi horien baliokideak diren ekuazio logikoak 111. Irudian idatzi dira, eta haiei dagokien diagrama logikoa ere bertan marraztu da.



x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	f_8	f_4	f_2	f_1
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

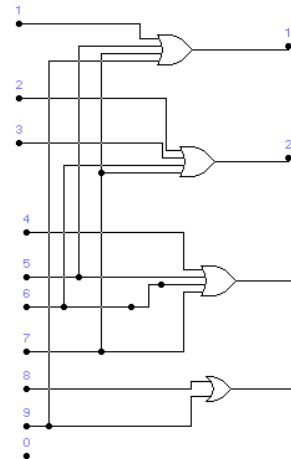
110. Irudia

$$f_8 = x_8 + x_9$$

$$f_4 = x_4 + x_5 + x_6 + x_7$$

$$f_2 = x_2 + x_3 + x_6 + x_7$$

$$f_1 = x_1 + x_3 + x_5 + x_7 + x_9$$



111. Irudia

Aldi berean sarrera bat baino gehiago aktibatzen badira, zentzurik gabeko irteera lortzen da. Jokaera hori onartzen duen kodegailua diseinatzea interesgarria litzateke kodetzailearen sarrerak aldi berean aktibatzea ariketa arrunta baita.


Lehentasunarekin lan egiten duen kodegailua. Adibidea: Hamartar_BCD

Aurreko paragrafoan adierazitako ideiarekin bat etorritz, lehentasunarekin lan egiten duen kodegailuak lehentasunik handiena (normalean, pisu esanguratsuena duen sarrera) duen sarrerari dagokion irteera soilik aktibatuko du.

Sarrera-aldagaien konbinazio posible guztiak egia-taulan adierazi nahi izanez gero, 2^{10} konbinazio idatzi beharko liriateke. Hortaz, funtzio logikoak egia-taulatik lortu beharrean, zirkuituaren funtzionamenduaren deskripziotik abiatuz lortu beharko dira, ezinbestean.

Prozedura hori errazteko, sarrera-konbinazio batzuk eta euren irteerak adierazi dira ezkerreko 27. Taulan. Eskumako taulan, irteera-konbinazio posible guztiak ematen dira.

9	8	7	6	5	4	3	2	1	0	f8	f4	f2	f1
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	1
0	0	0	0	0	0	0	0	0	1	1	0	0	1
0	0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0	1	1	0	0	1	0
0	0	0	0	0	0	0	0	1	1	1	0	1	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	1	0	1	1
0	0	0	0	0	0	0	1	0	1	0	0	1	1
0	0	0	0	0	0	0	1	0	1	1	0	1	1
0	0	0	0	0	0	0	1	1	0	0	0	1	1



f8	f4	f2	f1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

27. Taula

f₁ irteera aktibatzen da, 1, 3, 5, 7 edo 9. sarrerek lehentasuna duenean. Halaber, 1 sarrerek lehentasuna badu, 2, 4, 6 eta 8. sarrerek zero izan behar dute. Orduan, funtzionamendu hori deskribatzen duen funtzioa, hau izango da:

$$f_1 = (1.\bar{2}.\bar{4}.\bar{6}.\bar{8}) + (3.\bar{4}.\bar{6}.\bar{8}) + (5.\bar{6}.\bar{8}) + (7.\bar{8}) + 9$$

Gainerako irteeren funtzioak aurreko paragrafoan azaldutako logika bera jarraituz lortzen dira:

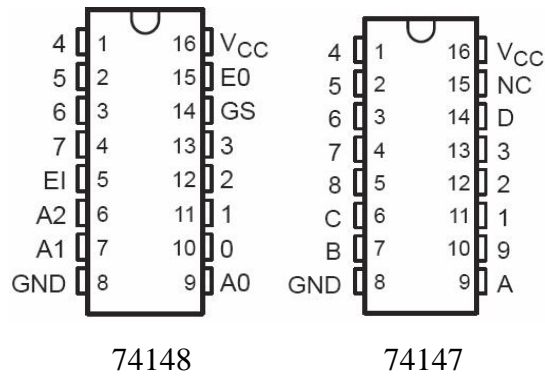
$$f_2 = (2.\bar{4}.\bar{5}.\bar{8}.\bar{9}) + (3.\bar{4}.\bar{5}.\bar{8}.\bar{9}) + (6.\bar{8}.\bar{9}) + (7.\bar{8}.\bar{9})$$

$$f_4 = (4.\bar{8}.\bar{9}) + (5.\bar{8}.\bar{9}) + (6.\bar{8}.\bar{9}) + (7.\bar{8}.\bar{9})$$

$$f_8 = 8 + 9$$

6.2. Zirkuitu komertzialak: 74148 eta 74147

Kodegailu bi horien irteerak eta sarrerak maila baxuan aktiboak dira, eta kapsulatua 112. Irudian erakusten da. 74148k, 74147ren aldean, gaitze-sarrera (EI) dauka. Halaber, beraren ezaugarri-orrian azaltzen den funtzionamendu-taula aztertuz (113. Irudia), beste bi irteeren funtzionaltasuna lor daiteke.



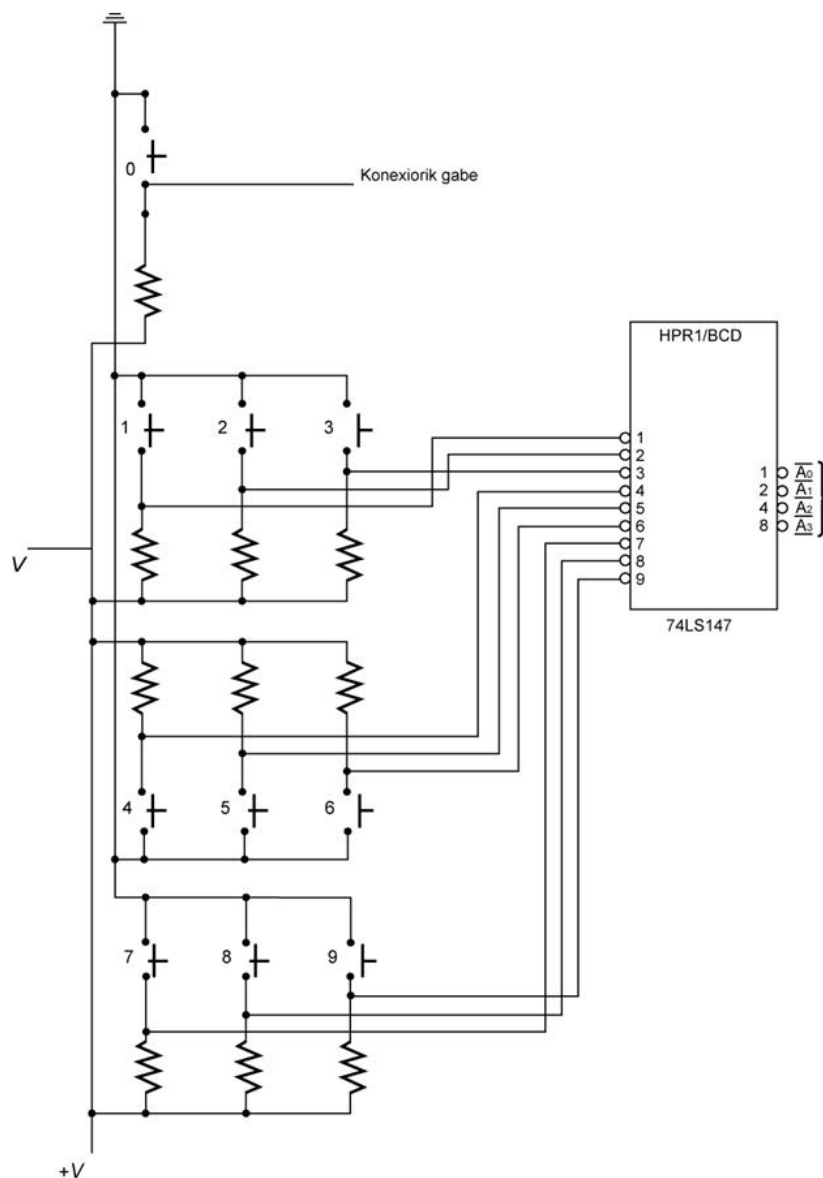
112. Irudia

EI	INPUTS								OUTPUTS				
	0	1	2	3	4	5	6	7	A2	A1	A0	GS	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	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	L	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

113. Irudia

6.3. Aplikazioa

114. Irudian zenbakizko teklatura kodetzeko adibidea azaltzen da. Tekla bakoitza, alde batetik, Vcc-ra erresistentzia baten bidez eta, bestetik, lurrera konektatzen den zirkuitu-pultsadore bat da. Horiek horrela, pultsadorea sakatzen ez bada, maila altua eragingo da pultsadore horri dagokion sarreraren, hots, desaktibazioa. Sakatzen bada, ordea, lurrarekin konexio zuzena lortzen da sarrera horretan, hau da, maila baxua edo sarreraren aktibazioa. Teklarik sakatzen ez bada, irteerak zeroa adieraziko du. Zirkuituan erabili den kodegailua lehenatasunarekin lan egiten duen kodegailua izan da.



114. Irudia

6.4. VHDL Deskribapena: 8/3

```
library ieee;
use ieee.std_logic_1164.all;
use work.std_arith.all;
entity coder is port(
barru: in std_logic_vector(7 downto 0);
kanpo : out std_logic_vector(2 downto 0);
eo    : out bit
);
end coder;
```

```
1  architecture archicoder of coder is
2  begin
3  process (barru)
4  begin
5  if std_match(barru, "1-----") then kanpo <="111";
6  eo<='0'; end if;
7  if std_match(barru, "01-----") then kanpo <="110";
8  eo<='0'; end if;
9  if std_match(barru, "001-----") then kanpo <="101";
10 eo<='0'; end if;
11 if std_match(barru, "0001----") then kanpo <="100";
12 eo<='0'; end if;
13 if std_match(barru, "00001---") then kanpo <="011";
14 eo<='0'; end if;
15 if std_match(barru, "000001--") then kanpo <="010";
    eo<='0'; end if;
    if std_match(barru, "0000001-") then kanpo <="001";
    eo<='0'; end if;
    if std_match(barru, "00000001") then kanpo <="000";
    eo<='0'; end if;
    if std_match(barru, "00000000") then kanpo <="000";
    eo<='1'; end if;
    end process;
end coder;
```