LOGIKA DIGITALA ETA MIKROPROGRAMAGARRIA

ELEKTRONIKA DIGITALARI BURUZKO HITZ BATZUK 1

Helburuak

Unitate didaktiko honetan, elektronika digitalarekiko lehen hartu-emana izango duzu. Ingurune fisikoak izan ditzakeen hainbat egoeratan oinarrituta, kontrol-sistema baten eta sistema elektroniko digital baten oinarrizko egitura ikertuko dugu, eta, gainera, egitura horietako elementu bakoitzaren premia eta oinarrizko funtzioa ere bai. Prozesadore digitaletan erabilitako zenbaki-sistemak azalduko ditugu, baita sistema bakoitza zergatik erabiltzen den ere. Unitate didaktikoaren amaieran, liburu honetako gaiak zergatik ikasi behar dituzun ikusiko duzu, eta gai horiek liburuaren helburu nagusiarekin lotuko dituzu; hau da, sistema elektroniko digitalen diseinuarekin eta garapenarekin.

Laburbilduz, alderdi hauek ikasiko dituzu: sistema elektroniko digital batek zertarako balio duen, zer oinarrizko elementuk osatzen duten eta horrelako sistema bat inplementatzeko zer oinarrizko alderdi behar diren.

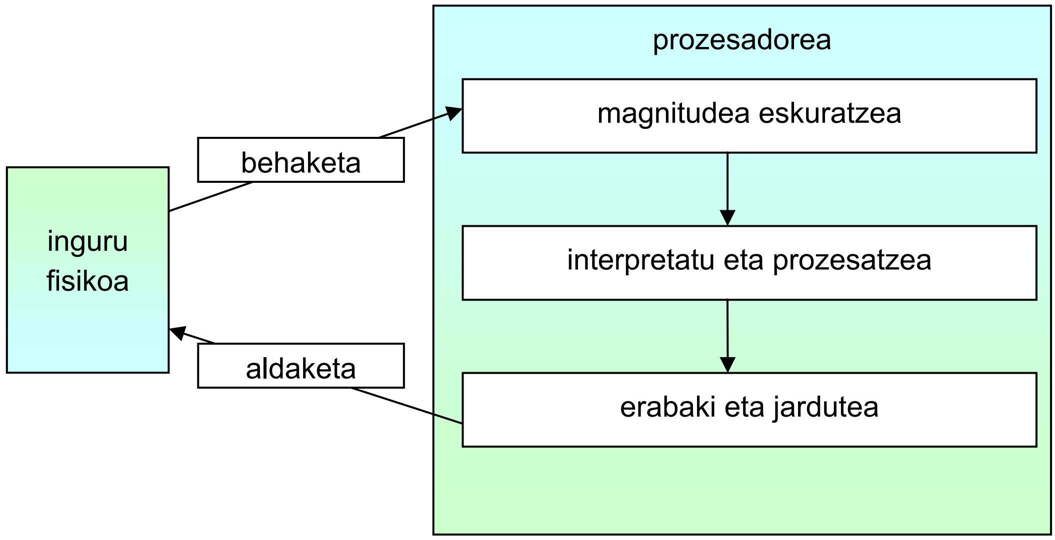
Unitate didaktiko honetan, giza adimenak darabiltzan prozeduren oinarrizko mekanismoak aztertuko ditugu, sistema elektroniko digitalak garatzeko aplikagarriak baitira. Mekanismo horiek aljebra boolearraren teoria matematikoen arabera azalduko ditugu; izan ere, sistema digitalen diseinuen kontrolerako prozesuak interpretatu, zehaztu eta azaltzeko tresnak ematen dizkigute teorema horiek. Aztertuko ditugun analisi-tresna batzuk funtzio logikoak eskuratzeko metodoak izango dira; beste batzuk, berriz, funtzio horiek sinplifikatzeko edota egia-taulak egiteko analisi-metodoak izango dira. Logika positiboa eta negatiboa ere aztertuko ditugu.

1.1 Kontrol-seinale eta -sistemak. Magnitudeen kodeketa

Teknologiari esker, ingurunea bere premietara egokitzeko gai da gizakia; izan ere, teknologiari esker, ingurunea kontrolatu eta aldatu egin dezake. Era horretan, ingurunearen ezaugarri fisikoak alda ditzake, eta, horrela, beste ezaugarri batzuk ezarri, bizirauteko edo ongizate handiagoa lortzeko.

Gaur egun, elektronika digitala dugu prozesuak kontrolatzeko teknologia nagusia. Baina, erabiltzen den kontrol-teknologia erabiltzen dela, ezin da ahaztu elektronika digitalaren xedea gizaki bat dagoen tokian makina bat jartzea eta hura ordezkatzea dela (teknologia-aurrerakuntzei esker, gainera, esparru batzuetan gizakien gaitasunak zabaldu eta are hobetu ere egin daitezke). Beraz, beharrezkoa da kontrol-prozeduren berezitasunak aztertzea, gero berezitasun horiek makinekin emulatu ahal izateko. Ondoko irudian ikus daitekeen bezala, prozedura horietan hiru elementu nagusik parte hartzen dute:

- **Sentsoreak:** elementu hauen bitartez, ingurune fisikoa aztertzen da.
- **Prozesadorea:** aztertutakoa lortu nahi den xedearekin lotuta interpretatzen du.
- **Eragingailuak:** elementu hauen bitartez, ingurunean eragin bat lortzen dugu, gero aldatzeko.



Gizakion gorputzak hiru elementu horiek ditu: zentzumenak, adimena eta beso-zangoak, hurrenez hurren.

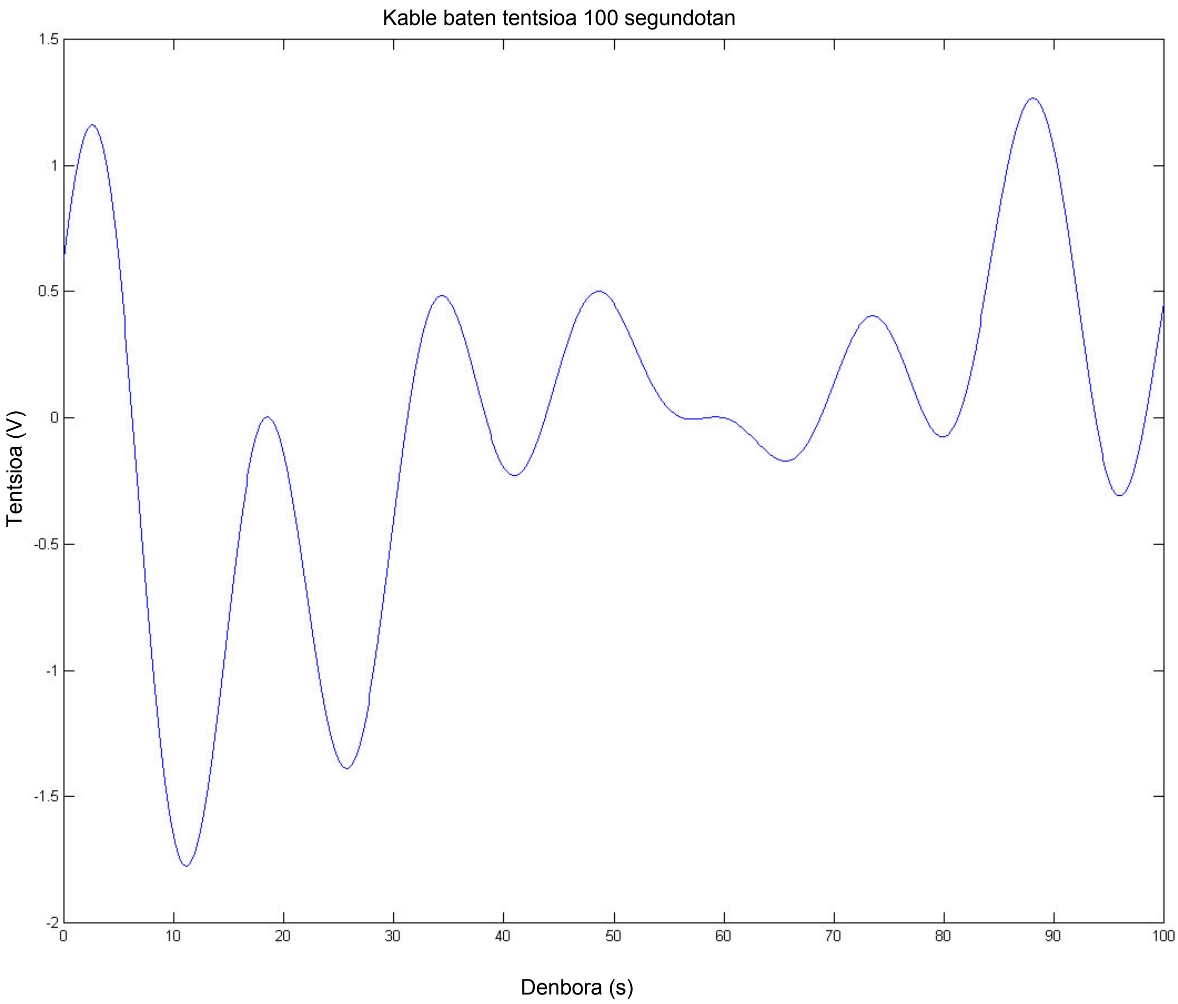
Gizakia ordezkatuko duen makina batek ere hiru elementu horiek eduki beharko ditu, eta garatu ahal izateko, lehenbizi, kontrolatu nahi diren *magnitude fisikoen ezaugarriak* ezagutu beharko dira, era horretan sentsore eta eragingailuak garatzeko; eta prozesadorea diseinatu ahal izateko, berriz*, giza adimenaren oinarrizko prozesuen ezaugarriak* ezagutu beharko ditugu. Magnitude fisikoen ezaugarriak aztertzean, ezaugarri horiek hiru irizpideren arabera sailka daitezke:

- ✓ Magnitude-neurriaren balioaren arabera. Honelako magnitudeak ditugu:
  - Magnitude analogikoak: arrazoizko marjina fisiko baten barruan, aztertutako magnitudeak edozein balio izan dezake. Magnitude analogikoak hauek dira: argitasuna, intentsitate elektrikoa, soinua, abiadura eta abar.
  - Magnitude digitalak: arrazoizko marjina fisiko baten barruan, aztertutako magnitudeak ezin du edozein balio izan, hainbat balio baimendu baizik. Balio horiek kopuru osoak izaten dira gehienetan. Magnitude digitalak hauek dira: hodi bateko zulo kopurua, ekoitzitako petrolio-upelen kopurua, Gabonetako zuhaitz batean piztutako bonbilla kopurua eta abar.
- ✓ Magnitudearen garapenaren arabera. Magnitudeak honelakoak izan daitezke:
  - Magnitude zuzenak: segidan hartutako neurrien garapena etengabeko eredu baten araberakoa da; hau da, edozein bi neurriren artean neurri infinituak daude. Magnitude zuzenak hauek dira: pila baten tentsioa hilabete batean zehar, baso batean mende batez bizi izandako otsoen kopurua; uhin baten potentzia elektromagnetikoa frekuentzia-hein batean, lursail baten altuera itsas mailarekiko eta abar.
  - Magnitude diskretuak: segidan hartutako neurrien garapena eredu diskretu baten araberakoa da; hau da, edozein bi neurriren artean neurri kopuru mugatu bat dago. Magnitude diskretuak hauek dira: urte osoan, goizeko 8:00etan izandako tenperatura; meridiano eta paralelo jakin batzuen elkarguneko koordenatu geografikoak; ekoizpen-sorta batean, lehen ehun gaileta-kaxen pisua eta abar.

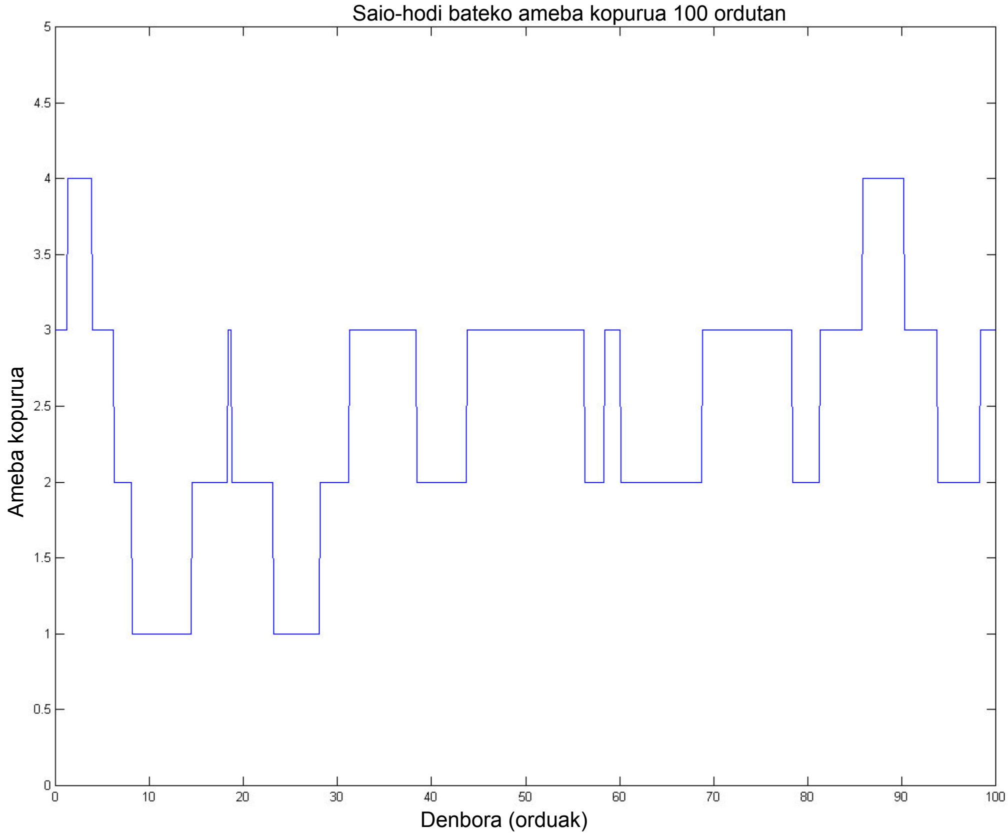
- ✓ Magnitudearen ezaugarri fisikoaren arabera. Magnitude aztergarria sorrarazten duen fenomenoaren *fisikotasuna* kontuan hartuta, magnitudeak horrela sailkatzen dira:
  - Elektromagnetikoak
  - Mekanikoak
  - Termikoak
  - Hidraulikoak
  - Pneumatikoak
  - eta abar.

Ondoko irudietan magnitude mota desberdinak ageri dira.

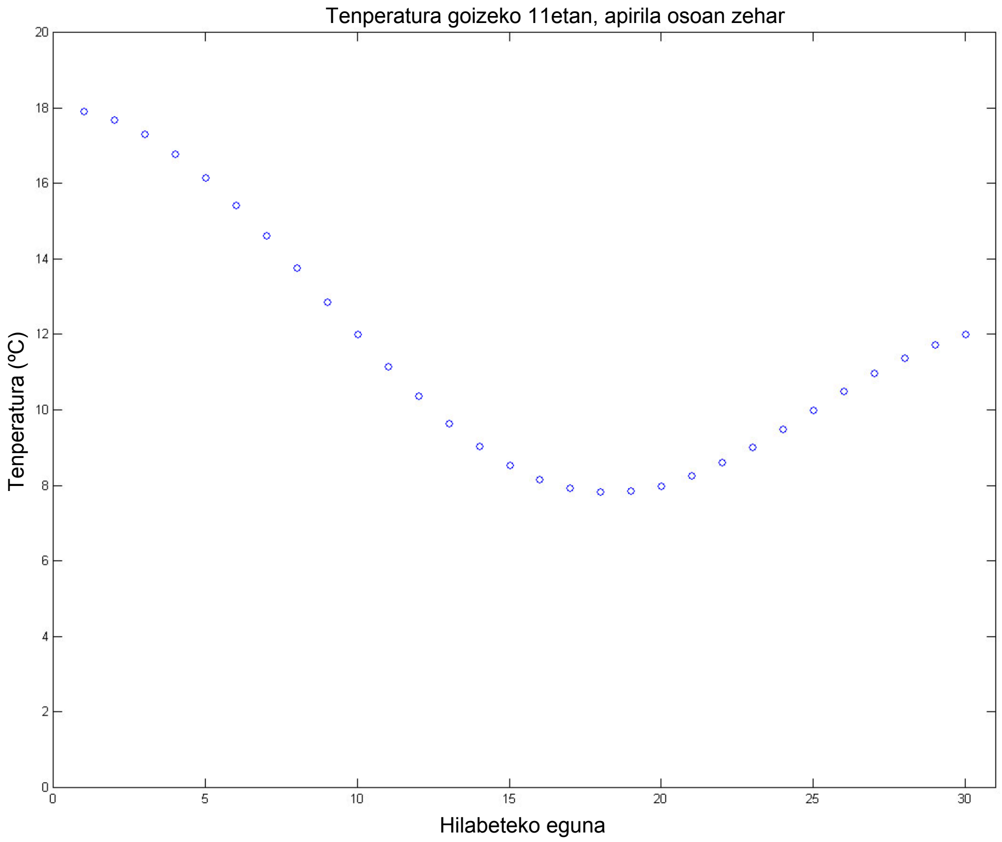
- ✓ 1. adibidea: magnitude analogiko jarraitua



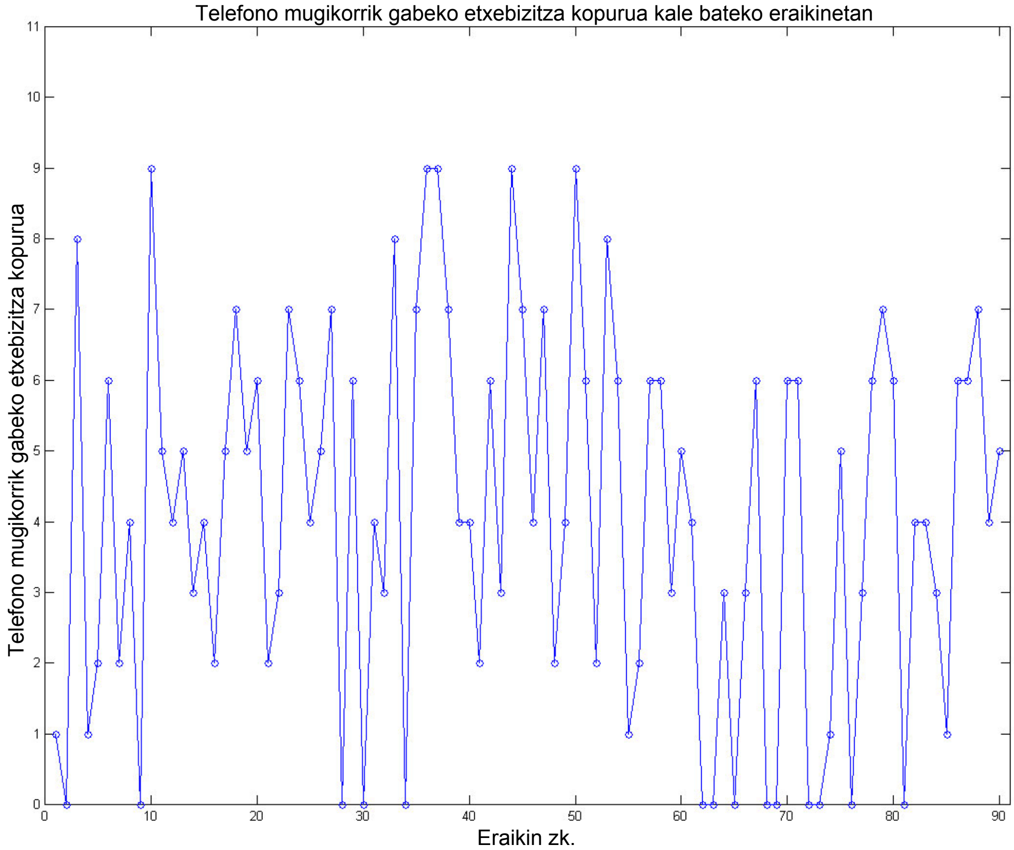
✓ 2. adibidea: magnitude digital jarraitua



✓ 3. adibidea: magnitude analogiko diskretua

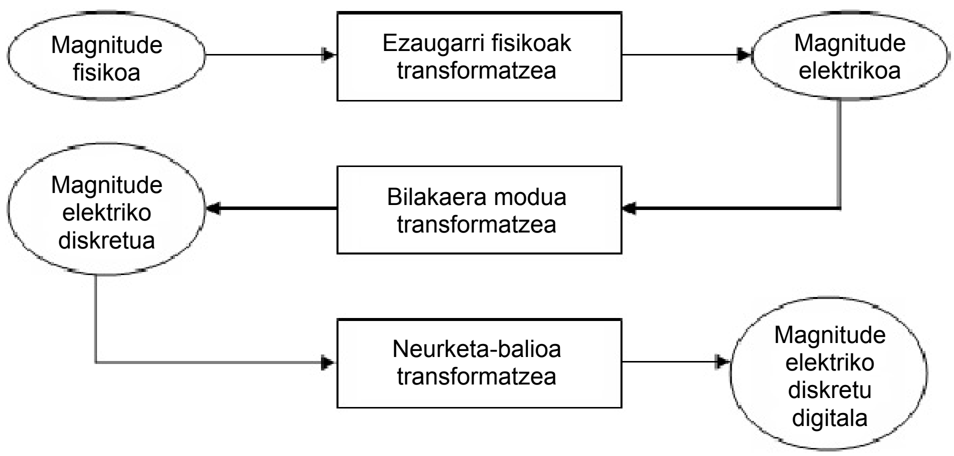


✓ 4. adibidea: magnitude digital diskretua

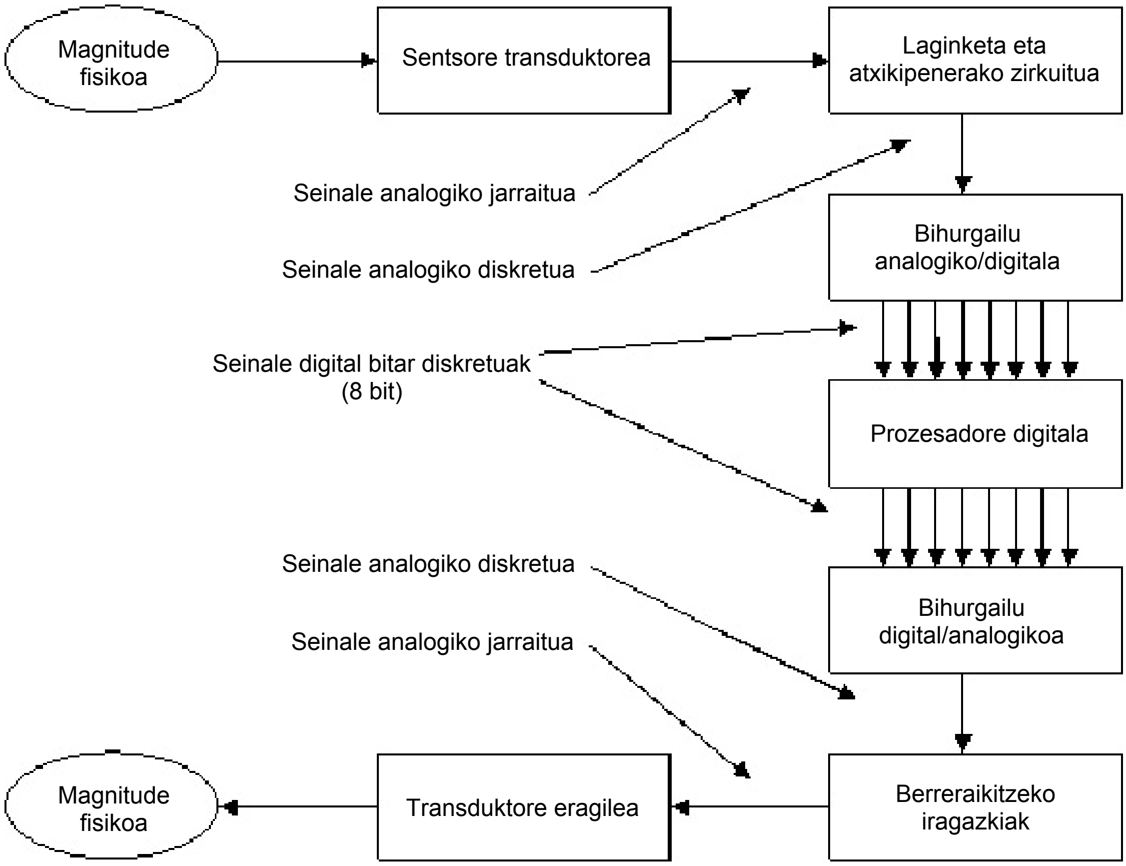


Gizakiak zentzumenak dituen bezala, prozesadore digitalak ere organoak eduki behar ditu, era horretan, kontrolatu beharreko magnitudeen jatorrizko fisikotasuna aldatu eta elementu elektroniko bihurtzeko. Organo horiek sentsore *transduktoreak* dira. Transduktoreek eroandako magnitude fisikoari *seinale* esaten zaio.

Beraz, seinaleak, magnitude fisiko izan arren, ez ditu kontrolatu nahi den magnitudearen ezaugarri fisiko berak, baina aldiz, haren balio eta garapen bera du (edo proportzionala). Magnitude fisikoak bezala, seinaleak ere analogikoak edo digitalak, zuzenak edo diskretuak izango dira. Prozesadore digitalak seinale digital diskretuekin egiten du lan. Orokorrean, azpiko irudian azaltzen den bezala, seinale elektriko digital diskretuak prozesadorera iristeko, transformazio jakin batzuk egin ohi dira.



Ondoko irudian prozesaketa digital baten barruko prozesu guztiak azaltzen dira laburtuta (adibidean, 8 bitekoa). Bloke bakoitza azpisistema elektroniko bat da, eta bakoitzak bere funtzio espezializatua betetzen du.



Oharra: Seinaleei eta sistemei buruzko informazio gehiago nahi izanez gero, ikus liburuarekin batera doan diskoan bildutako dokumentuak.

Zenbakitzeko eta kodetzeko sistemak

Seinale elektriko digitalek informazioa ematen diote prozesadore digitalari: *kodeketa* baten bitartez, kontrolatu nahi den magnitude fisikoari buruzko informazioa ematen dute. Eta *kodeketa* hain zuzen ere, hau da: informazio bat, sinbolo batek edo *kode* izeneko sinbolo-konbinazio jakin batek har ditzakeen balio desberdinak esleipen-arau jakin batzuen arabera esleitzea.

Sistema bitarra da prozesadore digitaletan erabilitako kodeketa. Jarraian sistema hori azalduko dugu, baita prozesadore digitalen diseinuan erabiltzen diren beste zenbaki-sistema batzuk ere.

Zenbaki-sistema hamartarra

Giza prozesadorearen kasuan, magnitude digitalen kodeketa-sistemarik egokiena eta erabiliena zenbaki-sistema hamartarra da. Sistema hori egokia da arrazoi hauengatik:

- ✓ Sistemak hamar sinbolo ditu, eta sinbolo horiek zenbaki arabiarrekin irudikatzen dira: {0,1,2,3,4,5,6,7,8,9}.
- ✓ Sinbolo bakoitzak forma geometriko bakuna du, eta gizakiarentzat erraza da sinbolo horiek paper batean edo beste gainazal batean marraztea. Buruan irudikatzen ere errazak dira.

- ✓ Gogoratu beharreko sinbolo kopurua (hamar) ez da oso handia, eta gizakiak ahalegin handirik egin gabe gogora ditzake denak.
- ✓ Magnitudeak kodetzeko, sinboloak konbinatu egiten dira arau erraz batzuen arabera.
- ✓ Gizakiak oso erraz interpretatu ditzake kodeketaren bitartez eskuratutako kode berriak.
- ✓ Kodeketa-sistemari esker balio erantsiak sortzen dira, eta besteak beste, errazagoa da magnitude kodetuak aljebraikoki erabiltzea.

Beste kode eta zenbaki-sistema batzuk ere badaude, baina hainbat arrazoi direla eta, ez dira sistema hamartarra bezain ongi egokitzen giza prozesadorearekin. Beraz, ez dira horrenbeste erabiltzen. Esaterako:

- ✓ Txinatar zenbaki-sistemak: sinboloen forma geometrikoa konplexua da.
- ✓ Mesopotamiar sistemak: sinbolo desberdin ugari zituzten (hirurogeita hamar inguru), eta beraz, denak gogorarazteko ahalegin handiagoa egin beharko litzateke.
- ✓ Erromatar sistema: oso zaila da sinboloak aljebraikoki erabiltzea.

Zenbaki-sistema hamartarra denok ezagutu arren, eta huskeria dirudien arren, ez legoke gaizki berriz ere aztertzea eta sistema zehazten duten puntu guztiak behar bezala interpretatzea (oharra: azalpenean bertan zehaztutakoa ez erabiltzeko, zenbakitan adierazitako kopuruak hitzez ere adieraziko dira).

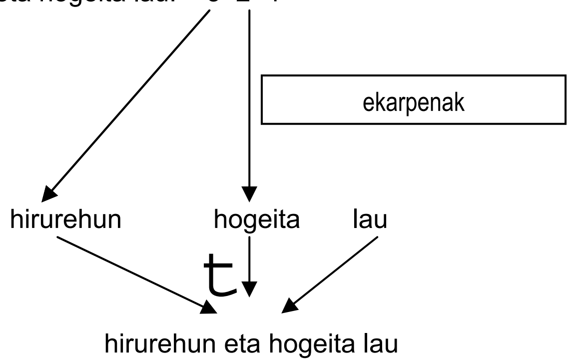
Zenbaki-sistema hamartarraren azalpena

- ✓ Hamarreko oinarria duen zenbaki-sistema da: horrek esan nahi du digitu izeneko hamar sinboloz osatutako sistema dela.

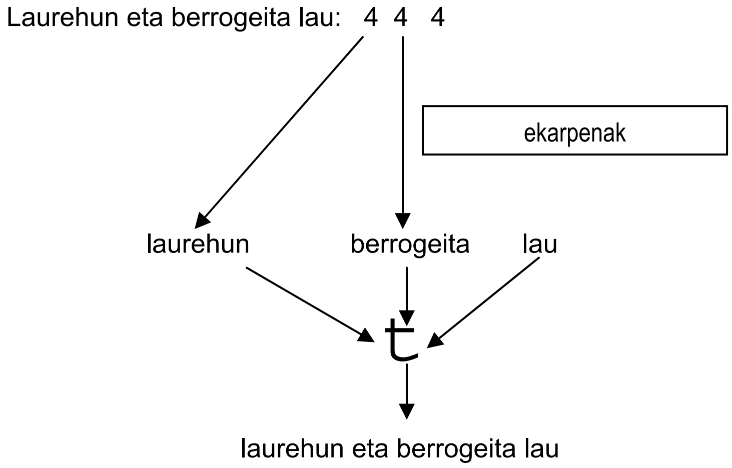
  $\{0,1,2,3,4,5,6,7,8,9\}$

- ✓ Kopuruak digitu bakar batekin edo bat baino gehiagorekin kodetzen dira, eta horren emaitza zenbaki bat da.
  - Hiru: 3
  - Hogeita lau: 24
  - Ehun eta laurogeita hemezortzi: 198
- ✓ Zenbaki batean, digitu bakoitzak kopuru bat adierazten du, eta hori, zenbaki osoak adierazitako kopuruari egiten dion ekarpena da. Zenbaki batek adierazitako kopuru osoa, zenbaki hori osatzen duten digituek adierazitako kopuruen batuketa da:

  Hirurehun eta hogeita lau: 3 2 4



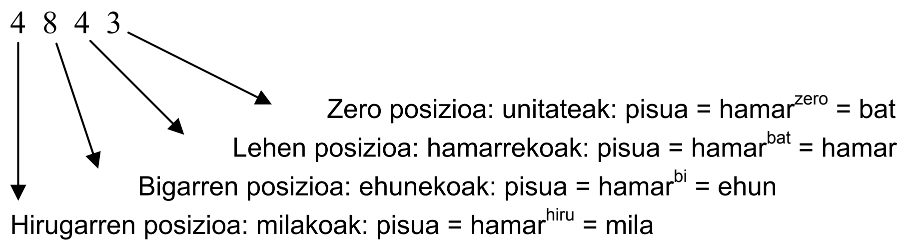
- ✓ Sistema posizionala da; hau da, digitu bakoitzak zenbakian duen posizioaren arabera, balio desberdina du. Adibidez, 4 digituak, zenbakian duen posizioaren arabera, lau, berrogei edo laurehun adieraziko du:



- ✓ Digitu bakoitzak zenbakian duen balioa kalkulatzeko, digituaren oinarrizko balioa haren lekuari dagokion pisuarekin biderkatu behar da:

Oinarrizko balioak	0: zero	1: bat	2: bi	3: hiru
	4: lau	5: bost	6: sei	7: zazpi
	8: zortzi	9: bederatzi		

- ✓ Pisuak: hamarreko oinarria duenez, n. posizioari dagokion pisua kalkulatzeko, hamar ber n egin behar da. Posizioak eskuinetik ezkerrera kontatzen dira, zerotik hasita. Gainera, posizio bakoitzak izen bat dauka: unitateak, hamarrekoak, ehunekoak eta abar. Izen horiek posizio bakoitzari dagokion pisua adierazten dute.



Beraz, zenbaki baten balioa kalkulatzeko, horrelako taula bat erabil daiteke:

Zenbakia	**Posizioa**	**Oinarrizko balioa**	**Pisua**	**Ekarpena: oinarrizko balioa x pisua**	**Batuketa**
75025	Zero	Bost	Bat	Bost	Hirurogeita hamabost mila eta hogeita bost
	Bat	Bi	Hamar	Hogei	
	Bi	Zero	Ehun	Zero	
	Hiru	Bost	Mila	Bost mila	
	Lau	Zazpi	Hamar mila	Hirurogeita hamar mila	

Sistema zehaztu dugunez, sistema hori bera erabil daiteke kopuruak adierazteko; horretarako, taula errazago hau erabil daiteke:

Zenbakia	Posizioa	Oinarrizko balioa	Pisua	Ekarpena: oinarrizko balioa x pisua	Batuketa
75025	0	5	1	5x1=5	75025
	1	2	10	2x10=20	
	2	0	100	0x100=0	
	3	5	1000	5x1000=5000	
	4	7	10000	7x10000=70000	

Azaldu berri dugun guztiak huskeria bat dirudien arren, funtsezkoa da zenbaki-sistema hamartarraren xehetasun guztiak ongi interpretatzen jakitea, aurrerago logika hori beste zenbaki-sistema batzuetan aplikatu ahal izateko. Izan ere, beste zenbaki-sistema horiek, berriak direnez, konplexuak dirudite, baina ez dira sistema hamartarra baino zailagoak. Desberdintasun bakarra da zenbaki-sistema hamartarra, eguneroko bizitzan horrenbeste erabiltzen dugunez, ia mekanikoki erabiltzen dugula, xehetasun guztiak ahaztuta, eta, ondorioz, askotan huskeria bat iruditzen zaigula. Baina, aldi berean, ia mekanikoki erabiltzeak ere argi uzten du zeinen ongi egokitzen den zenbaki-sistema hori prozesadorearen ezaugarriekin, hots, gizakiaren adimenarekin.

- **Zenbaki-sistema bitarra (bitar naturala)**

Gizakientzat zenbaki-sistema hamartarra zeinen egokia den ikusi ondoren, sistema berarekin lan egiten duen prozesadore elektroniko bat egitea egokia izan litekeela pentsa dezakegu. Horretarako, digituak adierazteko 10 sinbolo elektriko zehaztu beharko lirateke. 10 tentsio desberdin izan daitezke, esaterako: 0 V, 1 V, 2 V, ..., 9 V. adibidez, prozesadoreak 8 hari izango balitu, eta hari bakoitzean tentsio horietako edozein aplikatuko balitz, sistemak elektrikoki 00000000 eta 99999999 arteko edozein zenbaki adieraziko luke.

Hori egin daitekeen arren, ez da metodo egokiena teknologia elektronikoaren aukeren ikuspegitik, orain harira ez datozen arrazoi teknikoak direla medio. Hala ere, hamar balio desberdinekin lan egiten duen prozesadore elektroniko bat egin ordez, teknologikoki aukera egokiagoa dirudi bi balio desberdinekin lan egiten duen prozesadore bat egitea. Horrelako prozesadore batek erabilitako zenbaki-sistemak bi sinbolo izango lituzke; hau da, sistema bitarra erabiliko luke.

Baina teknologikoki bideragarria izan arren, prozesuak kontrolatzerako orduan erabilgarria ote den ere galde dezakegu, zenbaki-sistema bitarrean oinarrituta dagoenez gero. Erantzuna argi dago: bai (ordenagailuek sistema hori erabiltzen baitute). Liburu honetan, horrelako prozesadoreen oinarriak aztertuko ditugu, eta erabilgarriak direla frogatuko dugu.

Oraingoz, funtsezkoa da sistema bitarra ikastea, hori baita, hain zuzen ere, prozesadore elektroniko digitalek erabiltzen dutena. Hori ere sistema hamartarraren antzera azalduko dugu (kasu honetan, kopuruak hitzez edo sistema hamartarrarekin adieraziko dira, kasu bakoitzean komeni denaren arabera).

Zenbaki-sistema bitarraren azalpena

- ✓ Zenbaki-sistema bitarrak biko oinarria du: hau da, sistemak bi sinbolo ditu, digitu edo bit izenez ere ezagutzen direnak (ingelesezko B**I**nary digiTS terminotik sortua)

  {0,1}

- ✓ Kopuruak digitu bakar batekin edo batzuen konbinazioarekin kodetzen dira, eta emaitza bit bateko edo gehiagoko zenbakia da.
  - Bt: 1
  - Hru: 11
  - Hgeita hiru: 10111
  - Brrogeita hamabi: 110100
- ✓ Zenbaki bitar batean, digitu bakoitzak kopuru bat adierazten du, eta hori, zenbaki osoak adierazitako kopuruari egindako ekarpena da. Zenbaki batek adierazitako kopuru osoa zenbaki hori osatzen duten digituek adierazitako kopuruen batuketa da.
- ✓ Sistema posizionala da, eta horrek esan nahi du, sistema hamartarrean bezala, digitu bakoitzak balio bat izango duela zenbakian duen posizioaren arabera.
- ✓ Digitu bakoitzak zenbakian duen balioa kalkulatzeko, digituaren oinarrizko balioa eta bere lekuari dagokion pisua biderkatu behar dira.
- ✓ Onarrizko balioak:
  - 0: zero
  - 1: bat
- ✓ Psuak: biko oinarria duenez, n. posizioari dagokion pisua kalkulatzeko, 2 ber n egin behar da. Posizioak eskuinetik ezkerrera kontatzen dira, zerotik hasita.

1 1 0 1

0 posizioa: pisua = $2^0 = 1$

1. posizioa: pisua = $2^1 = 2$

2. posizioa: pisua = $2^2 = 4$

3. posizioa: pisua = $2^3 = 8$

- ✓ Pisu gutxieneko bita, esangura gutxieneko bit edo LSB ere esaten zaiona (ingelesezko Least Significant Bit hitzetik dator) eskuineko ertzean kokatutakoa da.
- ✓ Pisu gehieneko bita ezkerreko ertzean kokatutakoa da, eta esangura handieneko bit edo MSB ere esaten zaio (ingelesezko Most Significant Bit hitzetik dator).
- ✓ Beraz, zenbaki baten balioa kalkulatzeko, horrelako taula bat erabil daiteke:

zenbakia	posizioa	oinarrizko balioa	pisua=$2^{posizioa}$	ekarpena oinarrizko balioa x pisua	batuketa
1101	0	1	2^0=1	1x1=1	13
	1	0	2^1=2	0x2=0	
	2	1	2^2=4	1x4=4	
	3	1	2^3=8	1x8=8	
	←bitarra		hamartarra→		

Ikus daitekeen bezala, funtsean, sistema bitarrak eta hamartarrak mekanismo berak dituzte: izan ere, sistema posizionalak dira. Desberdintasun bakarra oinarria da; batek bikoa du, eta besteak, berriz, hamarrekoa. Hala ere, sistema bitarra ez da horren egokia giza prozesadorearentzat, aurrerago ikusiko dugun bezala, kopuruak txikiak izan arren, zenbakiak adierazteko digitu kopuru nahiko handiak behar baitira (esaterako: sistema hamartarreko 174, bitarrean 10101110 da), eta zero eta batez osatutako segida luzeak, ez dira errazak gizakiak irakurtzeko, gogoratzeko eta interpretatzeko.

Beraz, alde batetik, prozesadore elektroniko digitalak diseinatzeko, funtsezkoa da sistema bitarra erabiltzen jakitea; baina, bestetik, gizakiak hobeto menderatzen du sistema hamartarra. Beraz, beharrezkoa da sistema batetik bestera erraz igarotzeko teknikak ezagutzea.

Gainera, kopuruak bi sistemetan edukiko ditugunez, zenbaki bat sistema batean edo bestean idatzita dagoen jakiteko metodo bat ere zehaztu beharko dugu. Adibidez, 1101 zenbakia bitarra edo hamartarra izan daiteke. Zalantzarik badago, zenbakiaren azpian azpiindize bat jarriko dugu, eta bertan zein sistemakoa den zehaztuko dugu. Adibidez, 1101 sistema hamartarrean jarri nahi bada, 1101_{10} idatziko da; eta 1101 sistema bitarrean jarri nahi bada, 1101_2 idatziko da. Zalantzarik ez badago, ez dugu azpiindizerik jarri beharrik.

Bitarretik hamartarrera aldatzea

Azaldu berri dugun sistema bitarra, sistema hamartarreko zenbakiak sistema bitarrera bihurtzeko metodo bat da. Azaldu berri dugun taula horretan oinarritutako taula azkar bat erabiltzen du metodo honek: bertan, zutabe batean zenbaki bitarraren digituak jartzen dira, eta ondoan, 1 bitei dagozkien balioak. Pisuak batuz, zenbaki hamartar baliokidea lortzen da.

Sistema hamartarretik bitarrera aldatzea

Bihurketa hori egiteko hainbat metodo daude, eta denak baliokideak dira. Hemen azalduko duguna zenbaki hamartarra ondoz ondo bitan zatitzea da: zenbaki bitarra sortzeko, ondoz ondo, zenbaki hamartarra zati bi egin behar da, azken zatidura MSB gisa hartuz eta gainontzeko hondarretan gora eginez, lehen hondarrera iritsi arte; hura izango baita, hain zuzen ere, LSB.

- **Sistema hamartarraren eta bitarraren arteko oinarri-aldaketak egiteko ariketak**

Sistema hamartar eta bitarren arteko oinarri-aldaketak egiteko metodoak hauek dira:

✓ **Bitarretik hamartarrera aldatzea:** taula baten bitartez egiten da. Bertan, zutabe batean zenbaki bitarraren digituak jartzen dira, eta ondoan, 1 bitei dagozkien balioak. Pisuak batuz, zenbaki hamartar baliokidea lortzen da. Adibidez, 10101110 sistema hamartarrera aldatzeko:

– Zenbaki bitarraren digituak digituen zutabean jarri, LSB lehen lerroan, eta MSB azken lerroan.

Digituak	*Psuak*
0	
1	
1	
1	
0	
1	
0	
1	
batuketa ->	

– Pisuen zutabean, digitu bakoitzaren ondoan, kopuru hamartar hauek jarri:

- Digitua 1 bada: posizioaren araberako pisua
- Digitua 0 bada: zero

Digituak	*Psuak*
0	0
1	2
1	4
1	8
0	0
1	32
0	0
1	128
batuketa ->	

– Pisuen zutabean ageri diren kopuruak batu. Emaitza, bihurtu nahi zen zenbaki bitarraren baliokide hamartarra izango da.

Digituak	*Psuak*
0	0
1	2
1	4
1	8
0	0
1	32
0	0
1	128
batuketa ->	**174**

Beraz: $10101110_2=174_{10}$

- **Hamartarretik bitarrera aldatzea:** taula baten bitartez, zenbaki hamartarra ondoz ondo bitan zatitu. Zenbaki bitarra sortzeko, ondoz ondo, zenbaki hamartarra zati bi egiten da, azken zatidura MSB gisa hartuz eta gainontzeko hondarretan gora eginez, lehen hondarrera iritsi arte, eta hain zuzen ere, hori izango da LSB. Adibidez, 174 sistema bitarrera bihurtzeko:
  - Zenbaki hamartarra bitan zatitu, eta zatiketaren hondarra haren ondoan idatzi (1 edo 0 izango da nahitaez).

1 7 4	2			
1 6	8 7	→zatidura		
1 4				
1 4				
0	→	hondarra 0		

  - Aurreko zatiketaren zatidura hartu eta zati bi egin, berriz ere zatiketaren hondarra ondoan idatziz.

8 7	2			
8	4 3	→zatidura		
0 7				
6				
1	→	hondarra 1		

  - Aurreko zatiketaren zatidura hartu, zati bi egin, eta berriz ere zatiketaren hondarra ondoan idatzi; horrela, ondoz ondo, lortutako zatidura hori bitan zatitu ezin den arte (beraz, hondarra 1 edo 0 izango da). Egindako zatiketa guztiak horrelako taula batean idatz daitezke:

	zatidurak eta ondorengo zatikizunak	hondarrak	
	174	0	← 174/2ren hondarra (lehen hondarra)
174/2ren zatidura →	87	1	← 87/2ren hondarra
87/2ren zatidura →	43	1	← 43/2ren hondarra
43/2ren zatidura →	21	1	← 21/2ren hondarra
21/2ren zatidura →	10	0	← 10/2ren hondarra
10/2ren zatidura →	5	1	← 5/2ren hondarra
5/2ren zatidura →	2	0	← 2/2ren hondarra (azken hondarra)
(azken zatidura) 2/2ren zatidura →	1		

  - Zatiketa guztiak egin ondoren, taulako azken zatidura MSBtzat hartu, eta azken hondarretik hasita, ezkerretik eskuinera zenbaki bitarra eraikitzen joan, behetik gora hondarrak hartuz. Goitik hasita, lehen hondarra LSB izango da.

	zatidurak eta ondorengo zatikizunak	hondarrak	
	174	0	← LSB
	87	1	
	43	1	
	21	1	
	10	0	
	5	1	
	2	0	
MSB →	**1**		
		10101110	

Beraz: $174_{10} = 10101110_2$

1. adibide ebatzia

Zenbaki hamartar hauek bitarrera bihurtu: 57 eta 325.

Ebazpena

zatidurak	hondarrak
57	1
28	0
14	0
7	1
3	1
1	

$57_{10}=111001_2$

zatidurak	hondarrak
325	1
162	0
81	1
40	0
20	0
10	0
5	1
2	0
1	

$325_{10}=101000101_2$

2. adibide ebatzia

Zenbaki bitar hauek hamartarrera bihurtu: 1110011 eta 1010000111.

Ebazpena

digituak	pisuak
1	1
1	2
0	0
0	0
1	16
1	32
1	64
batuketa	115

$1110011_2=115_{10}$

digituak	pisuak
1	1
1	2
1	4
0	0
0	0
0	0
0	0
1	128
0	0
1	512
batuketa	647

$1010000111_2=647_{10}$

Bihurketa-metodo horiek ongi ikasi ondoren, gizakiak, bere ohiko zenbaki-sistema erabiliz, hots, sistema hamartarra erabiliz, aztertu eta diseinatu ahal izango ditu prozesadore elektroniko digitalak. Sistemari informazioa helarazi nahi dionean, landutako zenbaki hamartarrak sistema bitarrera bihurtu beharko ditu, hori baita makinak ulertzen duen sistema. Eta alderantziz, makinak emandako informazioa bitarrean dagoenez, hamartarrera bihurtu beharko du, errazago interpretatzeko. Hau da, prozesadore digitalekin lan egitean, egokia izango da, sistemen arteko bihurketa-metodoei esker, makinak erabiltzen dituen kopuru bitarrak sistema hamartarrera bihurtzea, eta era horretan, ez dira zeroz eta batez osatutako zerrenda luzeak interpretatzen aritu beharko, oso nahasgarriak baitira eta, gainera, interpretatzen zailak.

Hala ere, sistema bitar eta hamartarraren arteko bihurketa, erraza izan arren, ez da berehalakoa, horretarako kalkulu batzuk egin behar baitira. Trebetasun handiagoa hartzen zoazen heinean, eragiketa horiek errazago egingo dituzu, baina kopuru txikietan egin ezik, ez dituzu berehalakoan egingo. Beraz, sistema bitarra asko erabili behar den kasuetan, agian ez da aukera onena kopuru guztiak sistema hamartarrera aldatzea, sistema batetik bestera bihurtzea lan handia izan daitekeelako.

Zenbaki-sistema hamaseitarra

Sistema hamaseitarra hamaseiko oinarria duen zenbaki-sistema posizionala da; eta sistema hamartarrean bezala, gizakiarentzat askoz ere errazagoa da sistema horrekin lan egitea bitarrarekin baino. Gainera, abantaila handi bat dauka: sistema hamaseitarraren eta bitarraren arteko aldaketa egiteko ez da kalkulurik egin behar, bihurketa ia berehalakoa baita. Desabantaila bakarra da gizakia ez dagoela sistema hamaseitarra erabiltzera ohituta, baina arazo hori jardunaren poderioz konpontzen da.

Zenbaki-sistema hamaseitarraren azalpena

- ✓ Hamaseiko oinarria duen zenbaki-sistema da; horrek esan nahi du sistemak 16 sinbolo edo digitu dituela.

  {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

  Aipagarria da, sistema hamartarrean bezala, sistema hamaseitarra osatzen duten digituak 10 arabiar digituak direla, eta beste 6 digituetarako sinbolo ezagunak erabili direla: alfabetoko lehen 6 letrak, zenbaki gisa erabiliak.
- ✓ Kopuruak digitu bakar batekin edo zenbait digituren konbinazioarekin kodetzen dira, eta emaitza zenbaki bat da.
  - Hiru: 3
  - Hamar: A
  - Hogeita lau: 18
  - Bi mila bederatziehun eta berrogeita bi: B7E
- ✓ Zenbaki batean, digitu bakoitzak kopuru bat adierazten du, eta irudikatutako kopuruari zenbaki osoak egiten dion ekarpena da hori.
- ✓ Zenbaki batek irudikatutako kopuru osoa da zenbaki hori osatzen duten digituek adierazitako banakako kopuruen batuketa.
- ✓ Sistema posizionala da, eta horrek esan nahi du digitu bakoitzak balio bat duela zenbakian duen posizioaren arabera. Digitu bakoitzak zenbakian duen balioa kalkulatzeko, digituaren oinarrizko balioa eta haren lekuari dagokion pisua biderkatu behar dira.
- ✓ oinarrizko balioak:
  - 0: zero
  - 1: bat
  - 2: bi
  - 3: hiru
  - 4: lau
  - 5: bost
  - 6: sei
  - 7: zazpi
  - 8: zortzi
  - 9: bederatzi
  - A: hamar
  - B: hamaika

- C: hamabi
- D: hamahiru
- E: hamalau
- F: hamabost

✓ Pisuak: hamaseiko oinarria duenez, n. posizioari dagokion pisua 16^n izango da. Posizioak eskuinetik ezkerrera kontatzen dira, zerotik hasita.

4 D 9 A

Zero posizioa: pisua = $16^0 = 1$

Lehen posizioa: pisua = $16^1 = 16$

Bigarren posizioa: pisua = $16^2 = 256$

Hirugarren posizioa: pisua = $16^3 = 4096$

✓ Batzuetan, sistema hamaseitarra dela zehazteko, 16 azpiindizearen ordez, H letra edo $ sinboloa ere erabiltzen dira. Era horretan, 4D9A zenbaki hamaseitarra era hauetan adieraz daiteke:

$4D9A_{16}$ = 4D9AH = 4D9A$

H letra ez da digitu hamaseitar bat, digitu horiek F-ra artekoak bakarrik baitira. Zenbaki hamaseitar bat dela bakarrik adierazten du, eta ez du inolako zenbaki-baliorik.

✓ Beraz, zenbaki batek irudikatzen duen balioa kalkulatzeko, honelako taula bat erabil daiteke:

Zenbakia	Posizioa	Digitua	Oinarrizko balioa sistema hamartarrean	Pisua = $16^{posizioa}$	Ekarpena oinarrizko balioa x pisua	Batuketa
4D9A	0	A	10	1	10x1=10	19866
	1	9	9	16	9x16=144	
	2	D	13	256	13x256=3328	
	3	4	4	4096	4x4096=16384	
	← hamaseitarra			hamartarra →		

Beraz, $4D9A_{16} = 19866_{10}$

Hamaseitarretik hamartarrera aldatzea

Sistema bitarrarekin gertatzen zen bezala, sistema hamaseitarraren azalpena bera zenbaki hamaseitarrak hamartarrera bihurtzeko metodo bat da, eta horretarako, azaldu berri dugun bezalako taula bat behar da.

Hamartarretik hamaseitarrera aldatzea

Sistema hamartarretik hamaseitarrera aldatzeko metodoa sistema hamartarretik bitarrera bihurtzeko metodoaren antzekoa da; hau da, zenbaki hamartarra behin eta berriz zati hamasei egin eta, zenbaki hamaseitarra lortzeko, azken zatidura esangura handieneko digitutzat jotzea, eta zatiketen hondarretan gorantz eginez, lehen hondarra esangura gutxieneko digitutzat jotzea.

Ikus daitekeen bezala, sistema hamartarra sistema hamaseitar bihurtzea ez da berehalako kontua, eta beharrezkoa denean bakarrik egiten da. Sistema hamaseitarra erabilgarria da, baina batez ere sistema bitarrean emandako kopuruak interpretatzeko; izan ere, sistema batetik bestera igarotzea oso erraza da.

Hamaseitar-bitar harremana

Digitu hamaseitar bakoitzak 0 eta 15 arteko kopurua adierazten du, tarte hori baita digituen oinarrizko balioa. Kopuru horiek sistema bitarrean adierazi nahi izanez gero, 4 bit ere behar izaten dira, ondoko taulan ikus daitekeen bezala:

Hamaseitarra	Bitarra	Hamaseitarra	Bitarra	Hamaseitarra	Bitarra	Hamaseitarra	Bitarra
0	0000	1	0001	2	0010	3	0011
4	0100	5	0101	6	0110	7	0111
8	1000	9	1001	A	1010	B	1011
C	1100	D	1101	E	1110	F	1111

Taula horrek 0 eta 15 arteko kopuruak hamaseitarretik bitarrera edo bitarretik hamaseitarrera bihurtzeko balio du. Kopuru handiagoak bihurtzeko, oinarritzat aurreko taula hartzen da, eta metodo hauek erabiltzen dira:

Hamaseitarretik bitarrera aldatzea

Zenbaki bitarra lortzeko, digitu hamaseitarren baliokide bitarrak hartzen dira oinarritzat, eta digitu hamaseitar bakoitzaren ordez 4 biteko baliokide bitarra ezartzen da.

Bitarretik hamaseitarrera aldatzea

Bihurketa hau aurrekoaren alderantzizkoa da: LSBtik hasita, zenbaki bitarra lau biteko taldetan banatzen da, eta talde bakoitzaren baliokide hamaseitarrarekin ordezten da.

Ohar garrantzitsua

Ikus ezazue sistema hamaseitar eta bitarraren arteko bihurketa zuzenean egiten dela, digitu hamaseitar↔lau digitu bitar arteko harremanaren bitartez. Metodo hori ezin da erabili bitar eta hamartarren arteko bihurketa egiteko, eskuratutako emaitzak ez liratekeelako zuzenak izango. Hau da, zenbaki hamartar bat bitar bihurtzeko, ezin da digituz digituko metodoa erabili, lehen aipatutakoa baizik.

Sistema hamaseitarraren eta hamartar edo bitarraren arteko oinarri-aldaketak egiteko ariketak

Sistema hamaseitar eta hamartarraren arteko oinarri-aldaketak egiteko metodoak hauek dira:

- **Hamaseitarretik hamartarrera aldatzea:** taula baten bitartez, digitu hamaseitar bakoitza, horietako bakoitzaren posizioari dagokion pisurekin bidertu. Gero, digitu bakoitzaren oinarrizko balioak gehitu. Adibidez, $4D9A_{16}$ hamartarrera bihurtzea.

Zenbakia	Posizioa	Digitua	Oinarrizko balioa sistema hamartarrean	Pisua = $16^{posizioa}$	Ekarpena oinarrizko balioa x pisua	Batuketa
4D9A	0	A	10	1	10x1=10	19866
	1	9	9	16	9x16=144	
	2	D	13	256	13x256=3328	
	3	4	4	4096	4x4096=16384	
	← hamaseitarra			hamartarra →		

Beraz, $4D9A_{16}=19866_{10}$

- **Hamartarretik hamaseitarrera aldatzea:** taula batekin, zenbaki bitarra behin eta berriz zati hamasei egin. Zenbaki hamaseitarra osatzeko, azken zatidura MSB gisa hartzen da eta, gainontzeko hondarretan gora eginez, lehen hondarra LSB izango da. Esaterako: 19866_{10} hamaseitarrera bihurtzeko:

✓ Zenbaki hamartarra hamaseitan zatitu, zatiketaren hondarra ondoan idatziz (nahitaez 15 edo gutxiago izango da).

```
1 9 8 6 6 | 1 6
1 6         1 2 4 1  →zatidura
---
  3 8
  3 2
  ---
    6 6
    6 4
    ---
      2 6
      1 6
      ---
      1 0  →hondarra=10
```

✓ Aurreko zatiketaren zatidura hartu eta zati 16 egin, berriz ere zatiketaren hondarra ondoan idatziz.

```
1 2 4 1 | 1 6
1 1 2       7 7  →zatidura
-----
  1 2 1
  1 1 2
  -----
      9  →hondarra=9
```

✓ Aurreko zatiketaren zatidura hartu eta zati hamasei egin, berriz ere zatiketaren hondarra ondoan idatziz, horrela, ondoz ondo, lortutako zatidura zati 16 egin ezin den arte (beraz, 15 edo gutxiago izango da). Hurrenez hurren egindako zatiketak horrelako taula batean idatz daitezke:

	zatidurak eta ondorengo zatikizunak	hondarrak
	19866	10 ← 19866/16ren hondarra (lehen hondarra)
19866/16ren zatidura →	1241	9 ← 1241/16ren hondarra
1241/16ren zatidura →	77	13 ← 77/16ren hondarra (azken hondarra)
(azken zatidura) 77/16ren zatidura →	4	

✓ Zatiketa guztiak egin ondoren, azken zatiduraren eta hondarren kopuru hamartarrak hartu eta, digitu hamaseitarren oinarrizko balioen taularen arabera, digitu hamaseitar bihurtu.

zatidurak eta ondorengo zatikizunak	hondarrak		digitu hamaseitarrak	
19866	10	→	A	← lehen hondarra
1241	9	→	9	
77	13	→	D	← azken hondarra
azken zatidura → 4	→	→	4	

✓ Zenbaki hamaseitarra eraikitzeko, esangura handieneko digitutzat taulako beheko balioa hartu behar da (azken zatiduraren emaitza), eta gorantz jarraitzen da (azken hondarretik lehen hondarrera).

zatidurak eta ondorengo zatikizunak	hondarrak	digitu hamaseitarrak
19866	10	A ↑
1241	9	9
77	13	D
4		4
		4D9A

Beraz, 19866_{10}=4D9AH

Sistema hamaseitar eta bitarren arteko oinarri-aldaketak egiteko metodoak hauek dira:

- **Hamaseitarretik bitarrera aldatzea:** zenbaki bitarra eskuratzeko, digitu hamaseitarren lau biteko baliokide bitarrak hartzen dira oinarritzat. Esaterako: 8C6EH bitarrera bihurtzeko:

✓ Zenbaki hamaseitarraren digitu bakoitza 4 biteko baliokide bitarrarekin ordeztu.

digituak →	8	C	6	E
	↓	↓	↓	↓
4 biteko baliokide bitarra →	1000	1100	**0**110	1110

ezkerrean **0** gehitu da, zenbaki guztiak lau bitez osatuta egoteko ↑

✓ Zenbaki bitarra osatzeko, digitu hamaseitarren ordena berean digitu bakoitzaren bitak elkartu.

digituak →	8	C	6	E
	↓	↓	↓	↓
4 biteko baliokide bitarra →	1000	1100	0110	1110
	↘	↓	↓	↙
bilatutako zenbaki bitarra →	1000110001101110			

Beraz, $8C6E_{16}=1000110001101110_2$

- **Bitarretik hamaseitarrera aldatzea:** LSBtik hasita, zenbaki bitarra lau biteko taldetan banatu, eta talde bakoitza baliokide hamaseitarrarekin ordeztu. Esaterako: 10101101111101_2 hamaseitarrera pasatzeko:

✓ LSBtik hasita (eskuinetik), zenbaki bitarra lau biteko taldetan banatu. Azken taldeak lau bit ez baditu, dagoen bezala uzten da.

zenbaki bitarra →	10101101111101			
taldean jarraitu beharreko ordena →	←			
	↙	↓	↓	↘
sortutako taldeak →	10	1011	0111	1101
azken taldeak lau bit baino gutxiago izan ditzake ↑				

- ✓ Talde bakoitzeko lau biteko zenbaki bitarrean oinarrituz, talde bakoitzeko digitu bakar batez osatutako baliokide hamaseitarra eskuratu.

zenbaki bitarra →	10101101111101			
taldean jarraitu beharreko ordena →	←			
	↙	↓	↓	↘
sortutako taldeak →	10	1011	0111	1101
	↓	↓	↓	↓
baliokide hamaseitarra →	2	B	7	D

- ✓ Zenbaki hamaseitarra osatzeko, talde bakoitzeko digituak taldeen ordena berean bildu.

zenbaki bitarra →	10101101111101			
taldean jarraitu beharreko ordena →	←			
	↙	↓	↓	↘
sortutako taldeak →	10	1011	0111	1101
	↓	↓	↓	↓
baliokide hamaseitarra →	2	B	7	D
	↘	↓	↓	↙
bilatutako zenbaki hamaseitarra →	2B7D			

Beraz, 10101101111101_2=2B7DH.

1. adibide ebatzia

Zenbaki hamaseitar hauek hamartarrera bihurtu: F7 eta D0C5.

Ebazpena

- ✓ Hamartarrera aldatzeko:

zenbakia	digitua	oinarrizko balioa sistema hamartarrean	pisua	oinarrizko balioa x pisua
F7	7	7	1	7
	F	15	16	240
				batuketa: 247

$F7H=247_{10}$

zenbakia	digitua	oinarrizko balioa sistema hamartarrean	pisua	oinarrizko balioa x pisua
D0C5	5	5	1	5
	C	12	16	192
	0	0	256	0
	D	13	4096	53248
				batuketa: 53445

$DC05H=53556_{10}$

✓ Bitarrera aldatzeko:

digituak →	F	7
	↓	↓
4 biteko baliokide bitarra →	1111	0111

$F9H=11110111_2$

digituak →	D	0	C	5
	↓	↓	↓	↓
4 biteko baliokide bitarra →	1101	0000	1100	0101

$D0C5H=1101000011000101_2$

2. adibide ebatzia

Zenbaki hamartar hauek hamaseitarrera bihurtu: 236 eta 12587.

Ebazpena

zatidurak	hondarrak		hamaseitarra
236	12	→	C
14		→	E

236_{10}=ECH

zatidurak	hondarrak		hamaseitarra
12587	11	→	B
786	2	→	2
49	1	→	1
3		→	3

12587_{10}=312BH

3. adibide ebatzia

Zenbaki bitar hauek hamaseitarrera bihurtu: 1110011 eta 1010000110.

Ebazpena

zenbaki bitarra →	1110011	
	↓	↓
sortutako taldeak →	111	0011
	↓	↓
baliokide hamaseitarra →	7	3
	↓	↓
bilatutako zenbaki hamaseitarra →	73	

1110011_2=73$

zenbaki bitarra →		1010000110	
	↙	↓	↓
sortutako taldeak →	10	1000	0110
	↓	↓	↓
baliokide hamaseitarra →	2	8	6
	↘	↓	↓
bilatutako zenbaki hamaseitarra →		286	

1010000110_2=286$

Zenbaki-sistema zortzitarra

Zenbaki-sistema hamaseitarrean bezala, sistema zortzitarrari esker, kopuru bitarrak erraz erabil daitezke, sistemen arteko bihurketak ia berehalakoak baitira. Ez da sistema hamaseitarra adina erabiltzen; izan ere, oro har, zenbaki bitarrak zortzitarrera bihurtzean, luzeagoak dira, eta, sistema hamaseitarrari esker, zenbaki txikiagoak sortzen dira. Hala ere, sistema hau ere erabiltzen denez, komenigarria da ezagutzea.

Zenbaki-sistema zortzitarraren azalpena

- ✓ Zortziko oinarria duten zenbakien sistema bat da, eta horrek esan nahi du sistemak 8 sinbolo edo digitu dituela.

  {0,1,2,3,4,5,6,7}

  Aipagarria da, zenbaki-sistema zortzitarra osatzen duten digituak lehen 8 arabiar digituak direla. Beraz, zenbaki zortzitar batean ezin da 8 edo 9 digitua agertu.
- ✓ Kopuruak digitu bakar batekin edo zenbaiten konbinazioarekin kodetzen dira, eta emaitza zenbaki bat da.
  - Hiru: 3_8
  - Hamar: 10_8
  - Hogeita sei: 32_8
  - Bi mila bederatziehun eta berrogeita lau: 5576_8
- ✓ Zenbaki batean, digitu bakoitzak kopuru bat adierazten du, eta irudikatutako kopuruari zenbaki osoak egiten dion ekarpena da hori.
- ✓ Zenbaki batek irudikatutako kopuru osoa da zenbakia osatzen duten digituek adierazitako banakako kopuruen batuketa.
- ✓ Sistema posizionala da, eta horrek esan nahi du digitu bakoitzak balio bat duela zenbakian duen posizioaren arabera. Digitu bakoitzak zenbakian duen balioa kalkulatzeko, digituaren oinarrizko balioa eta haren lekuari dagokion pisua biderkatu behar dira.
- ✓ Oinarrizko balioen taula:

Digitua	Oinarrizko balioa	Digitua	Oinarrizko balioa	Digitua	Oinarrizko balioa	Digitua	Oinarrizko balioa
0	zero	1	bat	2	bi	3	hiru
4	lau	5	bost	6	sei	7	zazpi

- ✓ Pisuak: zortziko oinarria izanik, n. posizioari dagokion pisua kalkulatzeko, 8 ber n egin behar da. Posizioak eskuinetik ezkerrera kontatzen dira, zerotik hasita.

4 7 0 3

Zero posizioa: pisua = $8^0 = 1$

Lehen posizioa: pisua = $8^1 = 8$

Bigarren posizioa: pisua = $8^2 = 64$

Hirugarren posizioa: pisua = $8^3 = 512$

✓ Beraz, zenbaki baten balioa kalkulatzeko, horrelako taula bat erabil daiteke:

Zenbakia	Posizioa	Digitua	Oinarrizko balioa sistema hamartarrean	Pisua = $8^{posizioa}$	Ekarpena oinarrizko balioa x pisua	Batuketa
	0	3	3	1	3x1=3	
	1	0	0	8	0x8=0	
4703	2	7	7	64	7x64=448	2499
	3	4	4	512	4x512=2048	
	← zortzitarra			hamartarra →		

Beraz, $4703_8=2499_{10}$

Zortzitarretik hamartarrera igarotzea

Sistema hamaseitarrarekin gertatzen zen bezala, sistema zortzitarraren azalpena bera zenbaki zortzitarrak hamartar bihurtzeko metodo bat da, azaldu berri dugun bezalako taula bat behar baita horretarako.

Hamartarretik zortzitarrera aldatzea

Sistema hamartarretik zortzitarrera bihurtzeko metodoa sistema hamaseitarretik hamartarrera edo hamartarretik bitarrera bihurtzeko metodoaren antzekoa da; hau da, behin eta berriz zortzitan zatitzea: behin eta berriz zenbaki hamartarra zati zortzi egitea, azken zatidura pisu handieneko digitutzat jotzea, eta gainontzeko zatiketak jarraituz, gorantz joan eta lehen hondarra pisu gutxieneko digitutzat jotzea.

Zortzitar-bitar harremana

Digitu zortzitar bakoitzak 0 eta 7 arteko kopuru bat adierazten du, tarte hori baita digituen oinarrizko balioa. Kopuru horiek sistema bitarrean adierazi nahi izanez gero, 3 bit ere behar dira horretarako, ondoko taulan ikus daitekeen bezala:

Zortzitarra	Bitarra	Zortzitarra	Bitarra	Zortzitarra	Bitarra	Zortzitarra	Bitarra
0	000	1	001	2	010	3	011
4	100	5	101	6	110	7	111

Taula horrek 0 eta 7 arteko kopuruak zortzitarretik bitarrera edo bitarretik zortzitarrera bihurtzeko balio du. Kopuru handiagoak bihurtzeko, aurreko taula hartzen da oinarritzat, eta metodo hauek erabiltzen dira:

Zortzitarretik bitarrera aldatzea

Zenbaki bitarra lortzeko, digitu zortzitarren baliokide bitarrak hartzen dira oinarritzat, eta digitu zortzitar bakoitzaren ordez, 3 biteko baliokide bitarra ezartzen da.

Bitarretik zortzitarrera aldatzea

Bihurketa hau aurrekoaren alderantzizkoa da: LSBtik hasita, zenbaki bitarra hiru biteko taldetan banatu, eta talde bakoitza baliokide zortzitarrarekin ordezten da.

Ohar garrantzitsua

Sistema hamaseitarrean bezala, zortzitarraren eta bitarraren arteko aldaketak, zortzitar↔hiru digitu bitarren arteko harremanaren bitartez egiten dira. Berriz ere aipatu behar da, metodo hori ezin dela erabili bitar eta hamartarren arteko bihurketa egiteko, emaitzak okerrak izango bailirateke.

Sistema zortzitarraren eta hamartar edo bitarraren arteko oinarri-aldaketak egiteko ariketak

Sistema zortzitar eta hamartarraren arteko oinarri-aldaketak egiteko metodoak hauek dira:

- **Zortzitarretik hamartarrera aldatzea:** taula baten bitartez, digitu zortzitar bakoitza, ber bakoitzak duen posizioari dagokion pisua egin. Gero, digitu bakoitzaren oinarrizko balioak gehitu. Adibidez, 4703_8 hamartarrera aldatzeko.

Zenbakia	*Posizioa*	*Digitua*	*Oinarrizko balioa sistema hamartarrean*	*Pisua = $8^{posizioa}$*	*Ekarpena oinarrizko balioa x pisua*	*Batuketa*
4703	0	3	3	1	3x1=3	2499
	1	0	0	8	0x8=0	
	2	7	7	64	7x64=448	
	3	4	4	512	4x512=2048	
	← zortzitarra			hamartarra →		

Beraz, $4703_8=2499_{10}$

- **Hamartarretik zortzitarrera aldatzea:** taula batekin, zenbaki bitarra behin eta berriz zati zortzi egin: zenbaki zortzitarra sortzeko, azken zatidura MSB gisa hartzen da eta gainontzeko hondarretan gora eginez, lehen hondarra LSB izango da. Adibidez, 19866_{10} zortzitarrera aldatzeko.

  ✓ Zatidura eta hondarren taula egin:

	zatidurak eta ondorengo zatikizunak	hondarrak
	19866	2 ← 19866/8ren hondarra (lehen hondarra)
19866/8ren zatidura →	2483	3 ← 2483/8ren hondarra
2483/8ren zatidura →	310	6 ← 310/8ren hondarra
310/8ren zatidura →	38	6 ← 38/8ren hondarra (azken hondarra)
(azken zatidura) 38/8ren zatidura →	4	

✓ Zenbaki zortzitarra eraikitzeko, azken zatiduratik hasi, eta hondarrak hartuz gorantz joan, azkenetik lehenengora:

zatidurak eta ondorengo zatikizunak	hondarrak
19866	2
2483	3
310	6
38	6
4	
	46632

Beraz, $19866_{10}=46632_{8}$

Sistema hamaseitarrean bezala, sistema hamartarraren eta zortzitarraren arteko bihurketa ez da berehalakoa, eta beraz, beharrezkoa denean soilik egiten da. Sistema zortzitarra erabilgarria da, baina sistema bitarrean emandako kopuruak interpretatzeko soilik; izan ere, sistema batetik bestera igarotzea oso erraza da.

Sistema zortzitar eta bitarraren arteko oinarri-aldaketak egiteko metodoak hauek dira:

- **Zortzitarretik bitarrera aldatzea:** zenbaki bitarra lortzeko, digitu zortzitarren baliokide bitarrak, hiru bitekoak, hartzen dira oinarritzat. Esaterako: 3725_{8} bitarrera aldatzeko.

✓ Zenbaki zortzitarraren digitu bakoitza 3 biteko baliokide bitarrarekin ordeztu.

digituak →	3	7	2	5
	↓	↓	↓	↓
3 biteko baliokide bitarra →	011	111	010	101

0 ezkerrean gehitu da, zenbaki guztiak hiru bitez osatuta egoteko
↑

✓ Zenbaki bitarra osatzeko, digitu zortzitar bakoitzaren bitak elkartu, digitu hamaseitarren ordena berean.

digituak →	3	7	2	5
	↓	↓	↓	↓
3 biteko baliokide bitarra →	011	111	010	101
	↘	↓	↓	↙
bilatutako zenbaki bitarra →		011111010101		

Beraz, baliorik ez duenez, ezkerreko ertzeko zeroa kenduz (hamartarrean bezala, ezkerreko zeroak ez dira kontuan hartzen) $3725_{8}=11111010101_{2}$

- **Bitarretik zortzitarrera aldatzea:** LSBtik hasita, zenbaki bitarra hiru biteko taldetan banatu, eta talde bakoitza baliokide zortzitarrarekin ordeztu. Esaterako: 1011111101_{2} zortzitarrera aldatzeko:

✓ LSBtik hasita (eskuinetik), zenbaki bitarra hiru biteko taldetan banatu. Azken taldeak hiru bit ez baditu, dagoen bezala uzten da.

zenbaki bitarra →	1011111101			
taldean jarraitu beharreko ordena →	←			
	↙	↓	↓	↘
sortutako taldeak →	1	011	111	101

azken taldeak hiru bit baino gutxiago izan ditzake ↑

- ✓ Talde bakoitzeko hiru biteko zenbaki bitarrean oinarrituz, baliokide zortzitarra eskuratu, talde bakoitzeko digitu bakar batez osatua dagoena.

zenbaki bitarra →	1011111101			
taldean jarraitu beharreko ordena →	←			
	↙	↓	↓	↘
sortutako taldeak →	1	011	111	101
	↓	↓	↓	↓
baliokide zortzitarra →	1	3	7	5

- ✓ Zenbaki zortzitarra osatzeko, talde bakoitzeko digituak elkartu, haien ordena berean.

zenbaki bitarra →	1011111101			
taldean jarraitu beharreko ordena →	←			
	↙	↓	↓	↘
sortutako taldeak →	1	011	111	101
	↓	↓	↓	↓
baliokide zortzitarra →	1	3	7	5
	↘	↓	↓	↙
bilatutako zenbaki zortzitarra →	1375			

Beraz, $1011111101_2=1375_8$.

1. adibide ebatzia

Zenbaki zortzitar hauek hamartarrera eta bitarrera pasa: 74 eta 1564.

Ebazpena

- ✓ Hamartarrera aldatzeko:

zenbakia	digitua	oinarrizko balioa sistema hamartarrean	pisua	oinarrizko balioa x pisua
74	4	4	1	4
	7	7	8	56
				batuketa: 60

$74_8=60_{10}$

zenbakia	digitua	oinarrizko balioa sistema hamartarrean	pisua	oinarrizko balioa x pisua
1564	4	4	1	4
	6	6	8	48
	5	5	64	320
	1	1	512	512
			batuketa:	884

$1564_8=884_{10}$

- ✓ Bitarrera aldatzeko:

digituak →	7	4
	↓	↓
3 biteko baliokide bitarra →	111	100

$74_8=111100_2$

digituak →	1	5	6	4
	↓	↓	↓	↓
3 biteko baliokide bitarra →	001	101	110	100

$1564_8=1101110100_2$

2. adibide ebatzia

Zenbaki hamartar hauek zortzitarrera pasa: 236 eta 12587.

Ebazpena

zatidurak	hondarrak
236	4
29	5
3	

$236_{10}=354_8$

zatidurak	hondarrak
12587	3
1573	5
196	4
24	0
3	

$12587_{10}=30453_8$

3. adibide ebatzia

Zenbaki bitar hauek zortzitarrera pasa: 1110011 eta 1010000110.

Ebazpena

zenbaki bitarra →		1110011		
		↓	↓	↘
sortutako taldeak →		1	110	011
		↓	↓	
baliokide zortzitarra →		1	6	3
		↓	↓	↓
bilatutako zenbaki hamaseitarra →		163		

$1110011_2 = 163_8$

zenbaki bitarra →		1010000110		
	↙	↓	↓	↘
sortutako taldeak →	1	010	000	110
	↓	↓	↓	↓
baliokide hamaseitarra →	1	2	0	6
	↘	↓	↓	↙
bilatutako zenbaki hamaseitarra →		1206		

$1010000110_2 = 1206_8$

Zenbaki-sistemekin lan egiteko informatika-tresnak

Windows familiako sistema eragile guztiek, azken bertsioetan, kalkulagailu bat dute aipatu berri ditugun zenbaki-sistema guztiekin lan egiteko. Horri esker, sistema bitar digitalekin lan egitean, zenbaki-sistemen arteko bihurketak eta eragiketak asko ere errazagoak dira.

Aipatu behar da, tresna hori eskura eduki arren, garrantzitsua dela oinarri-aldaketak eskuz egiten jakitea, eta ikasitako zenbaki-sistemen xehetasunak gogoan eduki behar ditugu. Alderantziz, hasiera batean, kalkulagailua eskuz egindako eragiketen emaitzak egiaztatzeko erabili behar da, eta aurrerago, bihurketa-teknikak erabat menderatzean, lan-tresna gisa erabil daiteke.

Windows-eko kalkulagailuak mahai gaineko ohiko kalkulagailuen antzera funtzionatzen du.

Puntu honetan, sistema zortzitarraren eta hamaseitarraren aplikazioa eta erabilera ulertzeko gai izan behar duzu, lehen ikusi dugun bezala, ez baita erraza zeroz eta batez osatutako zerrenda luzeak erabiltzea, eta akatsak egitea oso erraza baita. Beraz, zifra horiek sistema hamaseitarrera edo zortzitarrera bihurtuz gero, lana askoz ere errazagoa da.

Windows-eko kalkulagailua nola erabili oinarri-aldaketak egiteko

Gehienetan, Windows-eko menu honetan egon ohi da kalkulagailua:

Hasi → Programak → Gehigarriak → Kalkulagailua

Edo bestela:

Hasi → Exekutatu

eta *Exekutatu* leihoko *Ireki* barran *kalk* idatzi eta *Onartu* sakatu.

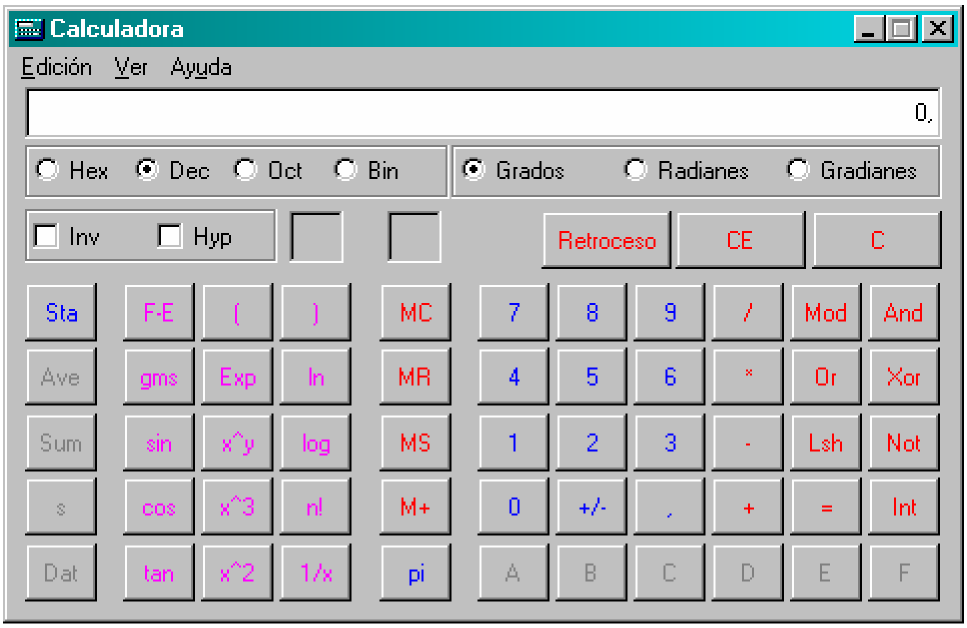
Bi aukera horietako bat egin ondoren, Windows-eko kalkulagailuaren leihoa agertuko da. Kalkulagailu horrek bi itxura izan ditzake: estandarra edo zientifikoa. Zenbaki-sistema desberdinekin lan egiteko itxura zientifikoa erabili behar da.

Hasiera batean bertsio estandarra agertzen bada, zientifikora aldatu behar da:

Ikusi → Zientifikoa

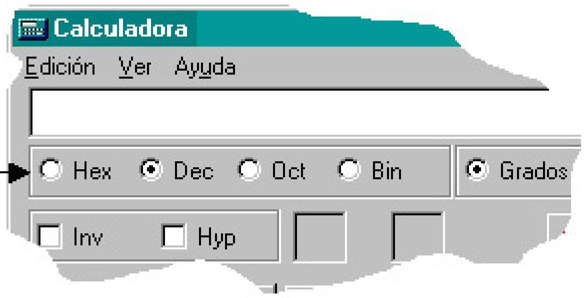
Irudian ageri den bezala, kalkulagailuko menu-barran:

Kalkulagailuak itxura hau dauka:



Puntu horretatik aurrera, kalkulagailua beste edozein bezalakoa da, baina berezitasun bakarra da zenbaki-sistema desberdinak aukeratu daitezkeela. Adibidez, irudian sistema hamartarra aukeratu da.

Zenbaketa-sistema aukeratzeko barra



Beste zenbaki-sistema bat aukeratzeko, aukeratu nahi den sistemaren ondoko zirkulu zuria sakatu behar da saguarekin.

Sistema batetik besterako bihurketak egiteko prozedura hauxe da:

- ✓ Bihurtu nahi den zenbakiaren sistema aukeratu. Adibidez, $2B7D_{16}$ zenbakia sistema bitarrera bihurtzeko, lehenbizi, sistema hamaseitarra (Hex) aukeratu. Aipagarria da A eta F bitarteko digituek, kalkulagailuaren beheko aldean kokatutakoek, sistema bitarrean gris kolorea zutela, eta hamaseitarra aukeratzean urdin kolorea hartu dutela. Kolore grisak esan nahi du digitu horiek ezin direla erabili, eta urdinak esan nahi du sistema hamaseitarrean erabil daitezkeela.
- ✓ Bihurtu nahi den zenbakia idatzi. Adibidez, 2B7D idatzi.
- ✓ Idatzitako zenbakia zein zenbaki-sistematara bihurtu nahi den aukeratu. Adibidez, sistema bitarra (Bin) aukeratu. Aipagarria da sistema bitarra aukeratzean digitu guztiak kolore griseko bihurtzen direla, bata eta zeroa izan ezik, hain zuzen ere, horiek baitira sistema bitarrean erabil daitezken bi digitu bakarrak.

Pantailan hasierako zenbakiaren baliokide bitarra agertuko da; kasu honetan, 2B7DH zenbakiarena: el $10101101111101_{2.}$

Kodetzeko beste sistema batzuk

Aurreko ataletan hainbat kodeketa-sistema azaldu ditugu: bakoitza prozesadore eta lan jakin baterako da egokia:

- ✓ Sistema hamartarra ia edozein kontabilitate-lanetarako da egokia gizakiarentzat.
- ✓ Sistema zortzitarra eta hamaseitarra prozesadore elektroniko digitalekin lan egiteko dira egokiak gizakiarentzat.
- ✓ Sistema bitarra prozesadore elektroniko digitalekin lan egiteko egokia da, izan ere, bi sinbolo desberdin bakarrik dituenez, teknologia elektronikoa erabiliz gero erraza delako elektrikoki bi tentsio desberdin adieraztea.

Aurreko ataletan azaldutako sistema bitarra ez da bi sinbolo soilik erabiltzen dituen sistema bakarra. Izan ere, kode bitar gehiago badaude, eta horiek ere prozesadore elektroniko digitaletan erabil daitezke, kode horietan zero eta bat sinboloak soilik erabiltzen baitira. Jarraian, beste kodeketa-sistema bitarren zerrenda bat emango dugu.

Zenbakizko sistema bitarrak

1 eta 0 digituetan oinarrituta, kopuruak bihurtzeko kodeketa-sistema desberdinak ezartzen dira, zenbaki-sistema bitar desberdinak lortzeko.

- ✓ Bitar naturala: aurreko atalean azaldutako sistema da.
- ✓ BCD: sistema honi esker, zenbaki hamartarrekin erraz lan egin dezakegu sistema bitarrean.
- ✓ Bitar zeinuduna: sistema honekin, berriz, kopuru positibo eta negatibo osoak adieraz daitezke.
- ✓ Bateko osagarria: sistema honekin kopuru positibo eta negatibo osoak adieraz daitezke.
- ✓ Biko osagarria: sistema honekin kopuru positibo eta negatibo osoak adierazi daitezke.
- ✓ Koma finkoa: sistema honekin zatikizko kopuru positibo eta negatiboak adieraz daitezke.
- ✓ Koma mugikorra: sistema honekin zatikizko kopuru positibo eta negatiboak adieraz daitezke.
- ✓ Gray kodea, Jhonson kodea eta abar.

Sistema horietatik, oraingoz, BCD sistema azalduko dugu.

Bitarrean kodetutako kode hamartarra BCD (Binary Coded Decimal)

BCD sistema (Bitarrean Kodetutako Hamartarra), izenak beran dioen bezala, sistema hamartarrerako kodeketa-sistema bitarra da. Hau da, zenbaki hamartar bat ematen denean, digitu hamartar bakoitza 4 biteko baliokide bitarrarekin ordezten da. Eta alderantziz, BCDn emandako zenbaki bitarra hamartar bihurtzeko, LSBtik hasi eta 4 biteko taldeak osatzen dira, eta talde bakoitza digitu hamartar baliokidearekin ordezten da.

Metodo hori sistema zortzitar eta hamaseitarrean erabilitakoaren antzekoa da, baina hamartarrera aplikatua. Beti kontuan eduki behar da, BCD zenbaki bitarra ez dela bihurtutako zenbaki hamartarraren baliokide bitar naturala.

Sistema hamartarren eta BCDaren arteko oinarri-aldaketak egiteko ariketak

Sistema hamartarraren eta BCD arteko oinarri-aldaketak egiteko metodoak hauek dira:

- **Hamartarretik BCDra aldatzea:** BCD zenbakia eskuratzeko, digitu hamartarren lau biteko baliokide bitarrak hartzen dira oinarritzat. Esaterako: 1879_{10} BCDra aldatzeko.

  - ✓ Zenbaki hamartarraren digitu bakoitza 4 biteko baliokide bitarrarekin ordeztu.

digituak →	1	8	7	9
	↓	↓	↓	↓
4 biteko baliokide bitarra →	0001	1000	**0**111	1001

0 ezkerrean gehitu da, zenbaki guztiak lau bitez osatuta egoteko ↑

- ✓ BCD zenbakia osatzeko, digitu hamartar bakoitzaren bitak elkartu, digitu hamartarraren ordena berean.

digituak →	1	8	7	9
	↓	↓	↓	↓
4 biteko baliokide bitarra →	0001	1000	0111	1001
	↘	↓	↓	↙
bilatutako BCD zenbakia →	1100001111001			

Beraz, 1879=$1100001111001_{BCD.}$

Kontuan izan ezkerreko zeroak kendu egin direla, inolako baliorik ez dutelako.

- **BCDtik hamartarrera aldatzea:** LSBtik hasita, BCD zenbakia lau biteko taldetan banatzen da, eta talde bakoitzaren baliokide hamartarrarekin ordezten da. Esaterako: 10100101110101_{BCD} hamartarrera aldatzeko:

- ✓ LSBtik (eskuinetik) hasita, BCD zenbakia lau biteko taldetan banatu. Azken taldeak lau bit ez baditu, dagoen bezala utzi.

BCD zenbakia →	10100101110101			
taldean jarraitu beharreko ordena →	←			
	↙	↓	↓	↘
sortutako taldeak →	10	1001	0111	0101
azken taldeak lau bit baino gutxiago izan ditzake ↑				

- ✓ Talde bakoitzeko lau biteko zenbaki bitarrean oinarrituz, baliokide hamartarra eskuratu, talde bakoitzeko digitu bakar batez osatua. Hori gertatzen bada, esan nahi du BCD zenbakia gaizki osatuta dagoela, eta ez dela benetako BCD zenbaki bat.

BCD zenbakia →	10100101110101			
taldean jarraitu beharreko ordena →	←			
	↙	↓	↓	↘
sortutako taldeak →	10	1001	0111	0101
	↓	↓	↓	↓
baliokide hamartarra →	2	9	7	5

- ✓ Zenbaki hamartarra osatzeko, talde bakoitzeko digituak elkartu, haien ordena berean.

BCD zenbakia →	10100101110101			
taldean jarraitu beharreko ordena →	←			
	↙	↓	↓	↘
sortutako taldeak →	10	1001	0111	0101
	↓	↓	↓	↓
baliokide hamartarra →	2	9	7	5
	↘	↓	↓	↙
bilatutako zenbaki hamartarra →	2975			

Beraz, $10100101110101_{BCD}=2975_{10}$.

1. adibide ebatzia

BCD zenbaki hauek hamartarrera bihurtu: 35 eta 5907.

Ebazpena

digituak →	3	5
	↓	↓
4 biteko baliokide bitarra →	0011	0101

$35_{10}=110101_{BCD}$

digituak →	5	9	0	7
	↓	↓	↓	↓
4 biteko baliokide bitarra →	101	1001	0000	0111

$5907_{10}=101100100000111_{BCD}$

2. adibide ebatzia

BCD zenbaki hauek hamartarrera bihurtu: 1110011 eta 1010000110.

Ebazpena

BCD zenbakia →	1110011	
	↓	↓
sortutako taldeak →	111	0011
	↓	↓
baliokide hamartarra →	7	3
	↓	↓
bilatutako zenbaki hamartarra →	73	

$1110011_{BCD}=73_{10}$

BCD zenbakia →		1010000110	
	↙	↓	↓
sortutako taldeak →	10	1000	0110
	↓	↓	↓
baliokide hamartarra →	2	8	6
	↘	↓	↓
bilatutako zenbaki hamartarra →		286	

$1010000110_{BCD}=286_{10}$

Kodetzeko sistema alfanumerikoak

Sistema horien bitartez sinbolo alfanumerikoak adierazi edo kodetu daitezke (letrak, puntuazio-zeinuak, digituak, eta abar). Kodeketa-sistema horietan, sinbolo alfanumeriko bakoitzari zenbakizko balio bat ematen zaio, taula baten arabera. Era horretan, testu idatzi bat zenbaki-segida baten bidez adieraz daiteke. Zenbakizko balioa bitarrean adieraztean, testuak prozesadore elektroniko digital batekin erabil daitezke.

Sinbolo alfanumeriko bakoitzari emandako zenbakizko balioaren arabera, kodeketa-sistema ugari zehaztu daitezke. Errealitatean, kodeketa-sistema alfanumeriko gutxi daude, eta erabiliena ASCII kode edo taula da (ingelesezko American Standard Code for Information Interchange terminotik dator; hau da, informazio-trukerako Amerikako kode estandarra). Sistema horren barne ASCII kode estandarra eta ASCII kode hedatua bereizi behar dira.

- ✓ ASCII estandarra: kopuruak kodetzeko 7 bit erabiltzen dira, beraz, 2^7=128 sinbolo alfanumeriko desberdin adierazten dira. Taulak sinbolo erabilienak biltzen ditu, eta munduan bakarra da.
- ✓ ASCII hedatua: kopuruak kodetzeko 8 bit erabiltzen dira, beraz, 2^8=256 sinbolo alfanumeriko desberdin adierazten dira. Lehen 128 kodetan, ASCII estandarreko sinbolo berak ageri dira (kodeko MSBa 0 da). Azken 128 kodeak (kodearen MSB 1 da) herrialdetako alfabetoen sinbolo bereziak dira, eta, beraz, taularen azken zati hori aldatu egiten da herrialde batetik bestera.

ASCII kodeaz gainera, beste kodeketa-sistema alfanumerikoak ere badaude, esaterako EBCDIC, baina ez dira horrenbeste erabiltzen. ASCII kodearen taulak eta beste kode batzuenak Interneten begiratu daitezke.

1.2 Proposizio-logika eta aljebra boolearra

Atal honetan adimen-prozesuari buruzko alderdi garrantzitsuenak aztertuko ditugu, era horretan, gizakiaren oinarrizko arrazoiketa emulatu dezaketen prozesadore elektroniko digitalak garatu eta kontrol-lanak egin ahal izateko.

Logika

Logikak, premisa batzuetan oinarrituta, ondorio zuzenetara garamatzaten baldintzak aztertzen ditu; hau da, *arrazoiketa-printzipio* zuzenak aztertzen ditu. Ez da zalantzan jartzen premisa horiek ziurrak diren ala ez; aitzitik, arrazoiketa baliozkoa ote den galdetzen da.

Logikak ez du bermatzen ondorioak beti zuzenak izango direnik, batzuetan oinarrizko premisa horiek okerrak izan baitaitezke. Logikak bermatzen duen gauza bakarra da premisa okerren ondorioz bakarrik sortzen direla akatsak, ez arrazoiketaren ondorioz.

Proposizio-logika

Proposizio-logikak giza logikaren dedukzio-prozesu oinarrizkoenak aztertzen ditu, besteak beste adierazpenezko proposizio-esakuneak, hain zuzen ere, hurrengo atalean azalduko ditugunak.

Proposizioa

Proposizio-esakune deklaratzailea, edo proposizioa, esaldi bat da, eta bertan adierazitakoa egia ala gezurra izan daiteke, baina ez bi gauzak.

Adierazpen-esakune hauek proposizioak dira, egia ala gezurra baitira:

- ✓ “Hiru gehi bi bost dira”. proposizioa egia da.
- ✓ “Lau gehi sei zortzi dira”. proposizioa faltsua da.
- ✓ “Bihar euria egingo du”: gaur ez dakigun arren, egia ala gezurra izango da.
- ✓ “Madril Espainian dago, eta Erroma, Frantzian”: osotasunean faltsua da.

Adierazpen-esakune hauek ez dira proposizioak, ezin baita esan egia ala gezurra diren; nahiak, aholkuak edo zenbaketak dira:

- ✓ Ikas ezazue pixka bat gehiago
- ✓ Egizu ongia ahal bezainbat
- ✓ Gora ezkonberriak
- ✓ Kaxa bat

Balio logikoa

Proposizio baten sinesgarritasun edo faltsutasuna da.

Aurreko adibidean, lehen proposizioa egia da, beraz, horren balioa ere egiazkoa da; bigarrena gezurra da, beraz, horren balioa ere faltsua da; eta hirugarrena egia ala gezurra izan daiteke, baina bihar arte ezingo dugu jakin. Laugarren proposizioaren zati bat egia da, eta bestea, gezurra. Baina proposizioa adierazita dagoen moduan, haren balio osoa faltsua da.

Balio logikoak bi sinbolorekin adierazten dira, E egiazkorako eta F faltsurako.

Aldagai logiko edo proposizionala

Proposizio baten adierazpen sinbolikoa da. Usadioz, proposizioak adierazteko sinboloak, besteak beste, **p**, **q**, **r** letra xeheak dira. Aldagai proposizionalak erosoak direlako erabiltzen dira; hau da, horien bidez, proposizioaren esakune osoa idatzi behar ez delako.

Adibidez, **p** letrak “hiru gehi bi bost dira” proposizioa adieraz dezake. Beraz, proposizio baten balio logikoa egia bada, era honetan ere adieraz daiteke:

$$p \equiv E$$

Lokailuak

Lokailua proposizioak konbinatuz proposizio berriak sortzeko mekanismoa da.

Giza lengoaian, lokailuak juntagailu edo menderagailuak dira: horiei esker, emendiozko perpausak, aurkaritzako perpausak, perpaus hautakariak eta abar sortzen dira. Lengoaiak aberatsak eta erredundanteak direnez, adierazpen batzuk formalki desberdinak izan arren, konbinazio logiko mota bera adierazten dute.

Unitate honetan aztertuko diren lokailurik arruntenak eta logikaren lengoaian horietako bakoitzari dagokion sinboloa taula honetan ageri dira:

<table>
<tr><th>Lokailua</th><th>Euskara</th><th>Sinboloa</th><th colspan="7">Adibidea</th></tr>
<tr><td>juntagailua</td><td>ETA</td><td>∧</td><td>euria eta eguzkia egiten du</td><td>→</td><td>euria ∧ eguzkia egiten du</td><td>→</td><td rowspan="3">p=euria egiten du
q=eguzkia egiten du</td><td rowspan="3">→</td><td>p ∧ q</td></tr>
<tr><td>disjuntzioa</td><td>EDO</td><td>∨</td><td>euria edo eguzkia egiten du</td><td>→</td><td>euria ∨ eguzkia egiten du</td><td>→</td><td>p ∨ q</td></tr>
<tr><td>ezetza</td><td>EZ</td><td>⌐</td><td>ez du euririk egiten</td><td>→</td><td>⌐ euria</td><td>→</td><td>⌐ p</td></tr>
</table>

Beste lokailu batzuk logikari buruzko testuetan aurkitu daitezke. Hemendik aurrera, idazkera gehiegi ez konplikatzearren, ez ditugu lokailuen sinbolo logikoak erabiliko, euskarazko **eta, edo** nahiz **ez** lokailuak baizik.

Lokailuen erabileraren arabera, bi proposizio mota bereizten dira:

Proposizio bakunak edo atomikoak: proposizio horiek ez dira beste lokailu batzuek konbinatuz sortzen. Adibidez: “Madril Espainian dago”.

Proposizio konposatu edo molekularrak: beste lokailu batzuk konbinatuz sortutako proposizioak dira. Proposizio konposatuak proposizio atomikoz edo beste proposizio konposatu batzuez osatuta egon daitezke. Baina, azken finean, proposizio molekular horiek ere proposizio atomikoz osatuta daude. Adibidez: “Madril Espainian dago, eta Erroma, Frantzian”.

- **Adierazpen logikoa**

Proposizioaren adierazpena da: proposizioen aldagai logikoek eta lokailuek osatzen dute. Proposizio atomikoen kasuan, adierazpen logikoa proposizioak adierazitako aldagai logikoa da. Adibideak:

Proposizioa	*Adierazpena*	*Azalpena*
Madril Espainian dago	p	Proposizio atomikoa da
Erroma Frantzian dago	q	Hau ere proposizio atomikoa da
Erroma ez dago Frantzian	**Ez** q	Proposizio konposatua da
Madril Espainian dago, eta Erroma ez dago Frantzian	p **eta ez** q	Hau ere proposizio konposatua da
Madril Espainian dago, eta Erroma ez dago Frantzian	r = p **eta ez** q	Proposizio konposatua da, baina, adierazpena errazteko, **r** aldagai logikoa esleitzen zaio.
Mapek gezurra diote	s	Proposizio atomikoa da
Madril Espainian dago, eta Erroma ez dago Frantzian edo mapek gezurra diote	r **edo** s = p **eta ez** q **edo** s	Proposizio konposatua da

- **Zeinu laguntzaileak**

Harreman logikoen idazkeran erabiltzen dira, ahozko hizkuntzak zenbaitetan dituen anbiguotasunak ekiditeko. Parentesiak, kortxeteak eta giltzak dira, eta adierazpen baten barruan aldagai logikoak behar bezala biltzeko erabiltzen dira. Adibidea:

Proposizioa	*Adierazpena*
Erruduna kartzelara doa	p
Erruduna hil egiten da	q
Errudunak isuna ordaintzen du	r
Erruduna kartzelara doa **eta**, hil egiten da **edo** isuna ordaintzen du	p eta (q edo r)
Erruduna kartzelara doa **eta** hil egiten da, **edo** isuna ordaintzen du	(p eta q) edo r

Azken bi proposizioetan esandakoa ia gauza bera izan arren, adierazpen logikoan ikus daitekeen bezala, bi egoerak oso desberdinak dira. Aurreko proposizioak egia badira, lehen kasuan, erruduna ziur joango da kartzelara, eta hil nahi ez badu, bertan isuna ordaindu beharko du. Bigarren kasuan, erruduna kartzelara joan eta bertan hil daiteke, edo isuna ordainduz libre gera daiteke. Beraz, argi dago; bi egoerak erabat desberdinak dira.

Logika-legeak eta dedukzio-metodoak

Aipatu berri ditugun kontzeptu eta elementuetan oinarrituta, proposizio-logikak giza arrazoiketaren mekanismoak aztertzen ditu, eta gero, lege edo arau gisa sailkatzen ditu. Adibidez, arrazoiketa logikoaren arabera, **eta** lokailuarekin osatutako proposizio konposatua egia da, baldin eta lokailu horrekin lotutako bi proposizioak egia badira, eta bietako bat edo biak faltsuak badira, proposizio konposatu hori faltsua da. Beraz, hori kontuan hartuta, lege hau sortzen da:

- ✓ **Eta** lokailuaren legea: **eta** lokailuarekin osatutako proposizio konposatua egia da, baldin eta lokailu horrekin lotutako bi proposizioak egia badira; eta bietako bat edo biak faltsuak badira, proposizio konposatu hori faltsua da.

  Adibidez, "gatza gozoa da **eta** urak busti egiten du" esatean, proposizio osoa faltsua da, **eta** lokailuarekin lotutako proposizioetako bat faltsua delako.

Halaber, **edo** lokailuari buruzko arrazoiketa logikoaren ondorioz, beste lege bat sortzen da:

- ✓ **Edo** lokailuaren legea: edo lokailuarekin osatutako proposizio konposatua egia da, baldin eta lokailu horrekin lotutako bi proposizioetako bat edo biak egia badira, eta biak faltsuak badira, proposizio konposatu hori faltsua da.

  Adibidez, "gatza gozoa da edo urak busti egiten du" esatean, proposizio osoa egia da, **edo** lokailuarekin lotutako proposizioetako bat egia baita.

Era berean, lokailu guztietarako, bai orain arte ikusi ditugunetarako nahiz beste batzuetarako ere, giza arrazoiketatik eratorritako funtzionamendu-arau bat dago, eta era horretan, logikaren oinarrizko arau edo legeak sortzen dira.

Gainera, beste harreman logiko konplexuagoak ere badaude, eta horien bitartez, proposizio hori osatzen duten proposizio bakunagoetan oinarrituta, proposizio konposatuen balio logikoa ondoriozta daiteke. Harreman horiek ere lege eta metodo deduktibo gisa sailkatzen dira.

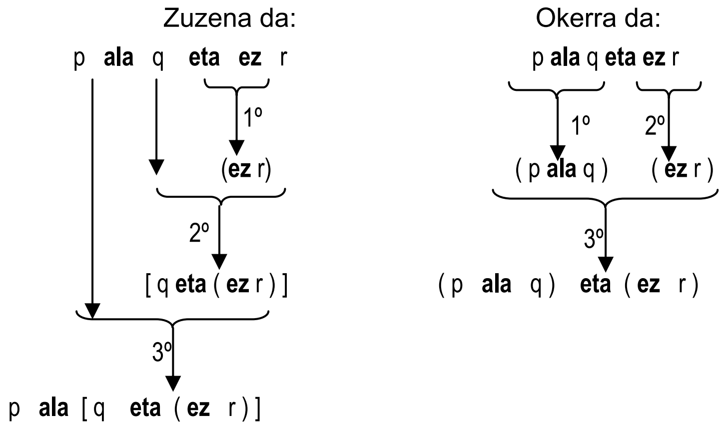
Logika-legeak aplikatuz, proposizio konposatu baten balio logiko osoa ebaluatu daiteke***.

Lehentasun-arauak

Hainbat lokailu dituen adierazpen batean, harreman logikoak zer ordenan interpretatu behar diren zehazten duten arauak dira. Interpretazioan zalantzarik balego, lehentasun gehien duten lokailuak aplikatzen dira lehenbizi. Lokailu logikoen lehentasun-ordena hau da:

Lehentasun-ordena		lokailua
handiagoa	↑	ez
		eta
gutxiago		ala

Era horretan, adibidez, “p **ala** q **eta ez** r”, adierazpenean, lehenbizi **ez** r ebaluatzen da; gero, q **eta** eta **ez** r-ren emaitza; eta azkenik, p **ala** aurreko ebaluaketaren emaitza. Ez litzateke zuzena izango lehenbizi p **ala** q ebaluatzea, eta emaitza **eta ez** r-rekin ebaluatzea.



Parentesiei esker lehentasun-ordena argitu edo aldatu daiteke, era horretan parentesi arteko terminoak ebaluatzen baitira lehenbizi.

Logikako lehentasun-arauek kalkulu aritmetikoan, hau da, batuketa, biderketa eta beste eragiketetan bezala funtzionatzen dute, beraz, aplikatzerakoan ez lukete arazo handirik sortu behar.

- **Egia-taula**

Egia-taula bat proposizio konposatu baten eta hura osatzen duten proposizioen arteko harreman logikoa azaltzeko metodoa da. Taularen alde batean proposizio konposatua osatzen duten proposizio bakunagoak ezartzen dira, eta azpian, horien balio logikoen arteko konbinazio posible guztiak. Taularen beste aldean proposizio konposatua ezartzen da, eta haren azpian, proposizio konposatua osatzen duten proposizio bakoitzaren konbinazioaren balio logikoa zehazten da.

Adibidez, p **ala** q proposizioa. Egia-taulak p eta y-k osatutako proposizio konposatua azaltzen du:

p	q	p **ala** q
E	F	F
F	E	E
E	F	E
E	E	E

P eta q azpian, bi proposizioen balio logikoen arteko konbinazio guztiak azaltzen dira; hau da, biak faltsuak dira, bata egia da eta bestea faltsua, eta alderantziz; eta biak egia dira. Konbinazio bakoitzaren ondoan, logika legeen arabera, p **ala** q adierazpenari dagokion balio logikoa zehazten da.

Aurreko taula hori metodo ona da **ala** lokailuaren legea adierazteko, lehen lege hori azaltzeko emandako definizioa baino erabilgarriagoa baita.

Era berean, hurrengo adibideak argi uzten duen bezala, logikaren oinarrizko lege bakoitza egia-taula baten bitartez adierazi daiteke.

p	q	**Eta** lokailuaren legea p **eta** q	**Ala** lokailuaren legea p **ala** q	Ezetzaren legea **ez** p
F	F	F	F	E
F	E	F	E	E
E	F	F	E	F
E	E	E	E	F

- **Adierazpen baliokideak**

Formalki desberdinak izan arren, proposizioen balioek emaitza logiko berdinera garamatzate. Beraz, bi adierazpen baliokidek egia-taula bera dute. Adibidez, aurrerago ikusiko dugu **ez** (p **eta** q) adierazpena **ez** p **ala ez** q-ren baliokidea dela.

Proposizio bateko elementu logikoak identifikatzeko ariketak

1. adibide ebatzia

Proposizio honetako elementu logikoak identifikatu:

- ✓ Proposizioa osatzen duten proposizio bakunak eta zer aldagai logiko esleitzen zaien.
- ✓ Proposizioaren adierazpen logikoa.
- ✓ Adierazpenaren egia-taula.

Emandako proposizioa: “tenperatua 20ºC baino baxuagoa da, eta tangan erregaia dago”.

Ebazpena

- ✓ Proposizioa osatzen duten proposizio bakunak eta zer aldagai logiko esleitzen zaien zehaztu: esaldia ikusita, ondorioztatzen da bi proposizio bakun daudela; eta beraz, bi aldagai zehaztuko dira:
  - p→ tenperatura 20ºC baino baxuagoa da.
  - q→ tangan erregaia dago.
- ✓ Proposizioaren adierazpen logikoa: esaldian ikusten da erabilitako lokailua **eta** dela. Beraz, adierazpen logikoa hau izango da:
  - p **eta** q
- ✓ Adierazpenaren egia-taula: bi aldagai daudenez, balio logikoen lau konbinazio posible daude: biak egia izatea, biak faltsuak izatea; bata egia eta bestea faltsua izatea, eta alderantziz. Konbinazio bakoitzerako balio logikoaren adierazpena zuzenean **eta** lokailuaren legearen arabera egiten da, ondoko taulan adierazten den bezala:

p	q	p **eta** q
F	F	F
F	E	F
E	F	F
E	E	E

Puntu honetan, logikaren lengoaia eta haren arau nahiz metodoak ikertzen jarrai dezakegu. Hala ere, aurreko ataletan aipatutakoez gain, ez ditugu lengoaia logikoaren elementu gehiago aztertuko, elektronikaren esparruan, logikako elementuak nahiko lengoaia orokorragoekin erabiltzen baitira, besteak beste aljebra boolearrarekin. Azken hori da, hain zuzen ere, hurrengo atalean aztertuko duguna.

1.3 Aljebra boolearra

Aljebra boolearra teoria matematiko bat da. Ezaugarri edo baldintza jakin batzuk betetzen dituzten multzoen propietateei buruzkoa da. Matematikan beti gertatzen den bezala, teoria abstraktu bat da, eta teoriaren baldintzetara egokitutako benetako edozein egoeran aplikatu daiteke.

Kasu honetan, proposizio-logika aljebra boolearraren ezaugarrietara oso ongi egokitzen da. Beraz, aljebra boolearraren ondorio eta emaitza guztiak zuzenean logikan aplikatu daitezke. Gainera, aljebra boolearrean erabiltzen den lengoaia oso ongi aplikatzen da prozesadore elektroniko digitaletan.

Logika, aljebra boolearra eta elektronika digitala horren ongi egokitzen dira beren artean, non etengabe esparru bateko eta besteko kontzeptu eta terminoak nahastu egiten dira. Ildo horretan, jarraian aljebra boolearraren berezko termino eta kontzeptuak azalduko ditugu, baina elektronika digitalari buruz ari garenez eta esparru horretan ohikoa den legez, termino batzuk eletronika edo logikako beste termino batzuekin ordeztuko ditugu.

Definizio formala

Edozein elementuz osatutako B multzoari buruz, bi barne-eragiketa zehazten dira. Eragiketa horiek adierazteko "+" eta "·" sinboloak erabiltzen dira. Beraz, lehen eragiketari *batuketa boolear* esaten zaio; eta bigarrenari, *biderkadura boolear*. Bi eragiketa horiek ez dira gehienetan erabiltzen ditugun batuketa eta biderkadura aritmetikoekin nahastu behar.

Barne-eragiketak direnez, Bko elementu pare bakoitzari beste B elementu bat esleitzen zaio. Hori horrela adierazten da:

$$BxB \xrightarrow{+} B \qquad\qquad B \times B \xrightarrow{\bullet} B$$
$$(a,b) \rightarrow (a+b) \in B \qquad\qquad (a,b) \rightarrow (a \cdot b) \in B$$

A eta *b* Bko elementuak dira, edo Bko elementuak ordezkatzen dituzten aldagai boolearrak.

Eragiketak zehaztuta dituen B multzoa aljebra boolearra da, baldin eta hurrengo baldintza edo postulatuak betetzen baditu:

- **1. Postulatua**. Bi eragiketak trukakorrak izatea

  B-ko elementu pare bakoitzean *(a,b)* hau egiaztatzen da:

  ✓ Batuketa trukakorra izatea:

  $$a + b = b + a$$

  ✓ Biderkadura trukakorra izatea:

  $$a \cdot b = b \cdot a$$

- **2. Postulatua.** Eragiketa bestearekiko banakorra izatea

B-ko elementu-hirukote bakoitzean (a,b,c) hau egiaztatzen da:

✓ Batuketarekiko banakorra izatea:

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

✓ Biderkadurekiko banakorra izatea:

$$a + (b \cdot c) = (a + b) \cdot (a + c)$$

- **3. Postulatua.** Elementu neutroa

B-ko edozein *a* elementurako:

✓ Batuketarako elementu neutro bat dago, zero elementua izenekoa: "0", era horretan:

$$a + 0 = a$$

✓ Biderkadurarako elementu neutro bat dago, bat elementua izenekoa: "1", era horretan:

$$a \cdot 1 = a$$

0 eta 1 elementuak B multzoan bakarrak dira, eta elementu guztietarako balio dute; hau da, B-n unibertsalak dira.

- **4. Postulatua.** Osagarri-axioma

B-ko *a* elementu bakoitzeko, B multzo berean *a*-ren osagarria izeneko elementu bat dago: *a*-ren kenketa edo *a*-ren aurkakoa, *a'*, ⌐*a*, -*a*, *a/*, a* edo $\overline{a}$ adierazita, beraz, era horretan:

✓ Batuketarekiko osagarria:

$$a + \overline{a} = 1$$

✓ Biderkadurarekiko osagarria

$$a \cdot \overline{a} = 0$$

Proposizio-logika aljebra boolear gisa

Aljebra boolearra definitu ondoren, logika- eta aljebra-elementu hauek baliokideak direla frogatu daiteke:

Proposizio-logika		Aljebra boolearra
E balio logikoa	⇦⇨	1 elementua
F balio logikoa	⇦⇨	0 elementua
eta lokailua	⇦⇨	biderkadura eragiketa: ·
ala lokailua	⇦⇨	batuketa eragiketa: +
edo lokailua	⇦⇨	elementu osagarria
proposizio edo aldagai logikoa	⇦⇨	aldagai boolearra
adierazpen logikoa	⇦⇨	adierazpen boolearra
logika-legeak	⇦⇨	aljebra postulatuak

Baliokidetasun horiek agerian uzten dute proposizio-logika entitate jakin bat dela eta aljebra boolearraren eredu matematikoaren araberakoa dela. Izan ere:

- **B multzoa**

Proposizio logiko batek har ditzakeen balio logikoen multzoa da; hau da:

B={E, F}

Multzo boolear guztietan bezala, 0 eta 1 balioek egon behar dute, eta E eta F balio logikoek bi elementu horiekin bat etorri behar dute derrigorrez. Aurrerago ikusiko dugun bezala, logika-legeen arabera, F 0rekin dator bat, eta E, 1ekin.

Elektronikaren esparruan, 0 eta 1 balio logikoei buruz hitz egiten da, baina, egia esan, ez da termino zuzena, bi termino nahasten baitira. Izan ere, 0 eta 1 balio boolearrak dira, baina horrek ez du ez arazorik ezta nahasterik ere sortu behar.

- **Aldagai boolearrak**

Proposizio edo aldagai logikoak dira, eta bakoitzak E (1) edo F (0) balio logikoa adierazten du.

Elektronikaren esparruan, aldagai boolearra eta aldagai logikoa terminoak ez dira bereizten.

- **Eragiketak**

  ✓ Batuketa boolearra: **ala** lokailuaren baliokidea da, eta, beraz, honela ere adieraz daiteke:

$$p\ o\ q \leftrightarrow p+q$$

  ✓ Biderkadura boolearra: **eta** lokailuaren baliokidea da, beraz, honela ere adierazi daiteke:

$$p\ y\ q \leftrightarrow p \cdot q$$

Elektronikaren esparruan, biderkadura logikoa, batuketa logikoa, biderkadura boolear eta batuketa boolear terminoak ez dira bereizten.

Eragiketa boolearretako lehentasun-ordena baliokide logikoen eta baliokide aritmetikoen bera da, eta, ondorioz, lehentasun-ordena aplikatzeak ez luke arazo edo nahasterik sortu behar: lehenbizi, kenketa; gero, biderkadura; eta, azkenik, batuketa. Parentesiak ordena hori aldatzeko erabiltzen dira.

- **Postulatuak**

Logika-legeen arabera, eta egia-taulen laguntzarekin, nahiko erraz frogatzen da postulatuak betetzen direla. Izan ere:

- ✓ Bi eragiketak trukakorrak izatea: bete egiten da:

$$p+q \leftrightarrow p\ o\ q \overset{\downarrow \text{ \textbf{ala} lokailuaren legeagatik}}{=} q\ o\ p \leftrightarrow q+p$$

$$p.q \leftrightarrow p\ y\ q \overset{\downarrow \text{ \textbf{eta} lokailuaren legeagatik}}{=} q\ y\ p \leftrightarrow q \cdot p$$

- ✓ Eragiketa distributiboa izatea: bete egiten da:

$$p \cdot (q+r) \leftrightarrow p\ y\ (q\ o\ r) \overset{\downarrow \text{ Logika-legeen arabera, berdina da}}{=} (p\ y\ q) o (p\ y\ r) \leftrightarrow (p \cdot q)+(p \cdot r)$$

$$p+(q \cdot r) \leftrightarrow p\ o\ (q\ y\ r) \overset{\downarrow \text{ Logika-legeen arabera, berdina da}}{=} (p\ o\ q)\ y\ (p\ o\ r) \leftrightarrow (p+q) \cdot (p+r)$$

- ✓ Elementu neutroak: batuketa eta biderkadurarako elementu neutroak 0↔F eta 1↔E dira hurrenez hurren. Izan ere:

$$p+0 \leftrightarrow p\ o\ F = p \qquad\qquad p \cdot 1 \leftrightarrow p\ y\ V = p$$

- ✓ Osagarri-axioma: B-ko edozein elementurako osagarria dago, kasu honetan, bi osagarri ditu soilik. Beraz:

$\overline{0} = 1$ Izan ere: $0+\overline{0} = 0+1 \leftrightarrow F\ o\ V = V \leftrightarrow 1$

Eta: $0 \cdot \overline{0} = 0 \cdot 1 \leftrightarrow F\ y\ V = F \leftrightarrow 0$

$\overline{1} = 0$ Izan ere: $1+\overline{1} = 1+0 \leftrightarrow V\ o\ F = V \leftrightarrow 1$

Eta: $1 \cdot \overline{1} = 1 \cdot 0 \leftrightarrow V\ y\ F = F \leftrightarrow 0$

Aipatu berri dugun guztiak kontuan hartuta, osagarri boolearra **ez** lokailu logikoarekin lotu daiteke, eta beraz, lokailua aplikatuz gero eskuratutako emaitza osagarri boolearra da. Hau da, baliokideak dira:

$$no\ p \leftrightarrow \overline{p}$$

Elektronikaren esparruan, **ez** lokailuari edo osagarri boolearraren baliokideari osagarri logiko esaten zaio. Berriz ere terminoak nahastu egiten dira, baina horrek ez du arazo handirik sortzen.

Ikusi berri dugun bezala, proposizio-logika aljebra boolearra da. Hemendik aurrera, kontzeptu logikoak maneiatzeko aljebra boolearraren tresnak erabiliko ditugu, elektronika digitalaren esparruan, tresna horiek logikaren lengoaia formala baino egokiagoak baitira.

Aljebra boolearraren teoremak

Aljebra boolearraren postulatuetan oinarrituta, aldagai boolearren arteko harreman batzuk ondorioztatu daitezke. Harreman horiek interesgarriak dira, adierazpen logikoak sinplifikatzeko aplikatzen baitira. Adierazpen bat sinplifikatzeak, formalki baliokidea izan arren, bakunagoa den beste adierazpen bat eskuratzea esan nahi du.

Jarraian harreman boolear horietako batzuk aipatuko ditugu, teorema gisa. Ez ditugu teoremen froga-penik egingo, liburuaren helburutik kanpo geratzen baita lan hori.

1. Teorema. Elementu menderatzaile-teorema

$$a+1=1$$
$$a\cdot 0=0$$

2. Teorema. Idenpotentzia-teorema

$$a+a=a$$
$$a\cdot a=a$$

3. Teorema. Inboluzio-legea edo ukapen bikoitzarena

$$\overline{\overline{a}}=a$$

4. Teorema. Xurgapen-legea

$$a\cdot(a+b)=a$$
$$a+(a\cdot b)=a$$

5. Teorema. Bigarren xurgapen-legea

$$a\cdot(\overline{a}+b)=a\cdot b$$
$$a+(\overline{a}\cdot b)=a+b$$

6. Teorema. Elkartze-legea

$$(a+b)+c=a+(b+c)=a+b+c$$
$$(a\cdot b)\cdot c=a\cdot(b\cdot c)=a\cdot b\cdot c$$

7. Teorema. De Morganen legeak

$$\overline{a+b}=\overline{a}\cdot\overline{b}\quad generalizando:\quad \overline{(a_1+a_2+\ldots+a_n)}=\overline{a}_1\cdot\overline{a}_2\cdot\ldots\cdot\overline{a}_n$$
$$\overline{a\cdot b}=\overline{a}+\overline{b}\quad generalizando:\quad \overline{(a_1\cdot a_2\cdot\ldots\cdot a_n)}=\overline{a}_1+\overline{a}_2+\ldots+\overline{a}_n$$

Adierazpen logikoak sinplifikatzeko ariketak

Aljebra boolearraren teorema eta postulatuetan oinarrituta, aldagai boolearren arteko harreman batzuk ezartzen dira. Harreman horiek interesgarriak dira adierazpen logikoak sinplifikatzeko aplikatzen baitira. Adierazpen bat sinplifikatzeak formalki baliokidea izan arren, bakunagoa den beste adierazpen bat eskuratzea esan nahi du.

Aljebra boolearra erabiliz, ez dago sinplifikatzeko arau finkorik, eta aplikatutako teoremen arabera, adierazpen bat ala bestea sortu daiteke. Ahalik eta adierazpen sinplifikatuena lortzeko, kasu bakoitzean teorema egokiena aplikatzen jakin behar da, eta hori eskarmentu kontua da soilik. Gainera, bide desberdinetatik emaitza berdina lor dezakegu.

1. adibide ebatzia

Adierazpen hau sinplifikatu:

$$a \cdot b \cdot \overline{c} + a \cdot b \cdot c$$

Ebazpena

Bi terminoetan $a{\cdot}b$ faktore komuna atereaz:

$$a \cdot b \cdot \overline{c} + a \cdot b \cdot c = a \cdot b \cdot (c + \overline{c})$$

Parentesiari osagarri-axioma aplikatuz:

$$a \cdot b \cdot (c + \overline{c}) = a \cdot b \cdot 1 = a \cdot b$$

2. adibide ebatzia

Adierazpen hau sinplifikatu:

$$a \cdot b \cdot \overline{c} + a \cdot b \cdot c + \overline{a} \cdot \overline{b} \cdot c + \overline{a}$$

Ebazpena

Lehen eta bigarren terminoari 1. adibideko emaitza aplikatuz, eta hirugarren eta laugarren terminoei xurgapen-legea aplikatuz gero, emaitza hau lortzen da:

$$[a \cdot b \cdot \overline{c} + a \cdot b \cdot c] = a \cdot b \downarrow \qquad \downarrow (\overline{a} \cdot \overline{b} \cdot c + \overline{a}) = \overline{a}$$

$$[a \cdot b \cdot \overline{c} + a \cdot b \cdot c] + (\overline{a} \cdot \overline{b} \cdot c + \overline{a}) = a \cdot b + \overline{a}$$

Bigarren xurgapen-legea aplikatuz:

$$a \cdot b + \overline{a} = \overline{a} + b$$

Funtzio boolearrak

Proposizio-logikari buruzko atalean aipatu ez ditugun logikaren lege eta dedukzio-metodo asko, askoz ere errazago interpretatzen dira aljebra boolearraren lengoaia erabiliz, eta bereziki funtzio boolearrak erabiliz. Kasu honetan ere, elektronikaren esparruan boolear funtzioei funtzio logiko ere esaten zaie, baina horrek ez du arazorik sortzen.

Funtzio logiko edo boolearraren definizioa

Matematikoki, *f* funtzio boolearra B-ko n. ordenako biderkadura kartesiar bat da, bere burua bider B egindakoa. Hori horrela adierazten da:

$$f : (BxBx\ldots xB) \rightarrow B$$
$$(a_1, a_2, \ldots, a_n) \xrightarrow{f} f(a_1, a_2, \ldots, a_n) \in B$$

Beste era batean esanda, funtzio logikoa edo funtzio boolearra, beste proposizio edo balio logikoetan oinarrituta, proposizio bati balio logiko bat esleitzeko harremana matematikoki adierazteko era da.

Adibidez, “tenperatura igo behar da hotza egiten badu” esaten badugu, “tenperatura igo behar da” proposizioaren balio logikoa “hotza egiten badu” proposizioaren araberakoa izango da. “Hotza egiten du” egia bada, “tenperatura igo behar da” ere egia izango da; eta “hotza egiten du” faltsua bada, “tenperatura igo behar da” ere faltsua izango da.

Funtzio baten elementuak

Mendeko proposizioak eta proposizio nagusiak ere aldagai logiko bidez adierazten direla kontuan hartzen badugu, elektronika digitalaren esparruan elementu hauek bereizten dira, nahiz eta batzuetan termino bera bi aldiz erabili:

- ✓ **Funtzioa:** lehen aipatutako aplikazio matematikoaz gain, beste proposizioen mendekoa den proposizioa haien funtziotzat jotzen da, edo funtzioa dela esaten da soilik. Matematikoki, mendeko aldagaia da.
- ✓ **Aldagaiak:** mendeko aldagaia (funtzio) ere aldagaia izan arren, testuinguru honetan, funtzioaren aldagaiei edo funtzioaren proposizio ala aldagai logikoei esaten zaie aldagai. Matematikoki, aldagai askeak dira.
- ✓ **Adierazpena:** funtzio bati balio bat esleitzeko adierazpen logikoa; hau da, aldagaiak konbinatzen dituena da funtzioaren adierazpena. Adierazpenetan, funtzioaren aldagaiak eragile logiko edo eragile boolear bidez konbinatzen dira.

Adibidez, "atea ireki behar da sarreran autorik badago edo etengailua sakatzen badute" esaldian, proposizio hauek identifikatu daitezke, bakoitza aldagai logiko batekin adierazita:

- ✓ *f*→ atea ireki behar da
- ✓ *a*→ sarreran autorik badago
- ✓ *b*→etengailua sakatzen badute

Esakunearen esanahia ikusita ondorioztatzen da, *f*-ren balio logikoa *a* eta *b* balio logikoen araberakoa dela; beraz, lehen esan dugun bezala:

- ✓ $f \rightarrow$ funtzioa
- ✓ *a, b* $\rightarrow$ aldagaiak

f-k *a* eta *b*-kiko duen menpekotasuna horrela adierazten da:

$f(a,b) \rightarrow$ esan nahi du *f a* eta *b-ren araberakoa dela*

f , *a* eta *b*-ren arteko menpekotasuna zehazten duen funtzioaren adierazpen logikoa hau izango da:

$$f(a,b) = a \text{ o } b \rightarrow \text{adierazpen logikoa}$$

Edo aljebra boolearraren lengoaia erabiliz, adierazpen boolearra horrelakoa da:

$$f(a,b) = a + b \rightarrow \text{adierazpen boolearra}$$

Hemendik aurrera, orokorrean adierazpen boolearra erabiliko dugu soilik, eta forma logikoa alde batera utziko dugu, nahiz eta ohitura dela-eta, adierazpen boolearrari adierazpen logiko ere esan.

- **Funtzio baten egia-taula**

Funtzio bat adierazpen logiko batek zehazten duenez, adierazpen hori hari dagokion egia-taularen bitartez ere adierazi daiteke. Era horretan, azken adibidea, ateari buruzkoa: *f*(*a*,*b*)=*a*+*b*, era honetan ere adierazi daiteke:

a	*b*	*f=a+b*
0	0	0
0	1	1
1	0	1
1	1	1

Adibide horretan bezala, hemendik aurrera, balio boolearrak (1 eta 0) erabiliko dira, eta balio logikoak (E eta F) alde batera utziko ditugu.

- ✓ Egia-taularen konbinazioak

  Batak eta zeroak erabiliz, funtzio baten n aldagaien balio logikoen konbinazio desberdinek n bitez osatutako zenbaki bitar bat osatzen dutela imajinatu dezakegu. Egia-taulan n aldagaien balioen konbinazio posible guztiak agertzen direnez, n bitez osatutako zenbaki bitar guztiak ere agertuko dira.

  Balio logikoen konbinazioak zenbaki bitar gisa interpretatzean, egia-taula batean jasotako konbinazio posible guztiak oso erraz osatu daitezke. Horretarako, aldagaiak ordena jakin batean ezarri behar dira, eta bit guztiak bitarrean zenbatu, 0koak eta 1ekoak, aldagai bakoitzak bit bat adierazten duela pentsatuz.

- ✓ Egia-taulako sarrerak

  Metodo horren bitartez, egia-taula bateko lerro bakoitza oso erraz identifikatu daiteke. Taulako lerro bakoitzari sarrera esaten zaio, eta hari esleitutako zenbaki bitarraren bidez identifikatzen da, edo bestela, errazago, zenbaki bitar horren baliokide hamartar edo hamaseitarraren bidez. Era horretan, esaterako, aurreko adibideko taulako 2. sarrera a=1 eta b=0 konbinazioa da, $10_2=2_{10}$ baita.

Funtzio motak

Jatorrizko funtzioak

Funtzio logiko bakunenak jatorrizko funtzioak dira, eta lokailu logikoen bitartez osatzen dira. Beraz, era horretan **eta** funtzioari, **ala** funtzioari nahiz **ez** funtzioari buruz hitz egin dezakegu. Horiez gain, identitate-funtzioa ere zehaztu daiteke, funtzio hori zehazten duen aldagai bakarraren funtzio bera baita: $f(a)=a$.

Elektronika digitalaren esparruan, jatorrizko funtzioak ingelesezko izenarekin erabiltzen dira. Ondoko taulan jatorrizko funtzioak ageri dira. Parentesi artean ingelesezko izena ageri da, baita funtzio bakoitzaren funtzionamendua zehazten duen egia-taula ere.

<table>
<tr><th colspan="10">Jatorrizko funtzioak</th></tr>
<tr><th colspan="2">Identitate funtzioa</th><th colspan="2">Kenketa (NOT)</th><th colspan="3">Batuketa edo ala (OR)</th><th colspan="3">Biderkadura edo eta (AND)</th></tr>
<tr><td>a</td><td>f=a</td><td>a</td><td>$f=\overline{a}$</td><td>a</td><td>b</td><td>f=a+b</td><td>a</td><td>b</td><td>f=a·b</td></tr>
<tr><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>

Funtzio elementalak

Elektronika digitalaren esparruan, ohikoak diren harreman logiko bakunen arabera zehaztutako funtzioak dira. Horietako bat **ala** esklusiboa edo batuketa esklusiboa da. Horrek esan nahi du funtzioa egia izango dela, baldin eta funtzio horretako bi aldagaietako bat soilik egia bada; eta faltsua izango dela bi aldagaiak faltsuak ala *biak egia badira.*

Funtzio horrek horrelako egoerak islatzen ditu, adibidez: kutxazain automatiko batean “ordainketa ongi egin da, baldin eta 10 euroko billete bat ala 5 euroko bi billete eman badira". Hauek dira egoera posibleak: billeterik eman ez bada (taulako 1. kasua), ordainketa ez da ongi egin. 10 euroko billete bat bakarrik eman bada (2. kasua), ordainketa ongi egin da. 5€ euroko bi billete eman badira (3. kasua), ordainketa ongi egin da. Azkenik, 10 euroko billete bat eta 5 euroko bi billete eman badira (4. kasua), guztira 20 euro ematen dira, eta beraz, ordainketa ez da ongi egin, 10 euro bakarrik eman behar baitziren.

	10 euroko billete bat eman da	*5 euroko bi billete eman dira*	*Ordainketa ongi egin da*
1. kasua	0	0	0
2. kasua	0	1	1
3. kasua	1	0	1
4. kasua	1	1	0

Ala esklusibo edo batuketa esklusiboaren funtzio logikoak berezko zeinu edo eragilea du, eta aljebra boolearrean honela adierazten da:

a ala esklusiboa b $\leftrightarrow a \oplus b$

$\oplus \rightarrow$ *batuketa esklusiboa*

Sinplifikazioak egiteko, baliokidetasun hauek erabilgarriak dira:

$$a \oplus b = a \cdot \overline{b} + \overline{a} \cdot b$$

$$\overline{a \oplus b} = a \cdot b + \overline{a} \cdot \overline{b}$$

Jatorrizko funtzioetan oinarrituta, kenketaren eragilearekin konbinatuta, beste funtzio elemental batzuk sortzen dira, hau da, **eta** ezeztatua, **ala** ezeztatua eta abar.

Elektronika digitalaren esparruan, funtzio elementalak ere ingelesezko izenarekin erabiltzen dira. Ondoko taulan funtzio elementalak ageri dira. Parentesi artean ingelesezko izena ageri da, baita funtzio bakoitzaren funtzionamendua zehazten duen egia-taula ere.

Funtzio elementalak											
Batuketa edo ala esklusiboa (XOR)			***Ala ezeztatua (NOR)***			***Eta ezeztatua (NAND)***			***Batuketa edo ala esklusibo ezeztatua(XNOR)***		
a	b	$f = a \oplus b$	a	b	$f = \overline{a + b}$	a	b	$f = \overline{a \cdot b}$	a	b	$f = \overline{a \oplus b}$
0	0	0	0	0	1	0	0	1	0	0	1
0	1	1	0	1	0	0	1	1	0	1	0
1	0	1	1	0	0	1	0	1	1	0	0
1	1	0	1	1	0	1	1	0	1	1	1

Funtzio konposatuak

Jatorrizko funtzioek eta funtzio elementalek baino harreman konplexuagoak zehazten dituzte, eta jatorrizko funtzio ala funtzio elementalen arteko konbinazioek osatzen dituzte; esaterako:

$f(a,b,c) = \overline{a \oplus b} \cdot c$.

Funtzio logikoak identifikatzeko ariketak

Beste proposizio edo balio logiko batzuetan oinarrituta proposizio bati balio logiko bat esleitzeko harremana matematikoki adierazteko era da funtzio logikoa.

Sistema elektroniko digitalak diseinatzeko orduan, diseinatu beharreko sistema deskribatzen duen esakunea hartu eta haren funtzio logikoak identifikatzea da lehen pausoa. Funtzioak identifikatzeko, pauso hauek eman behar dira:

- ✓ Funtzioa identifikatu
- ✓ Funtzioaren aldagaiak identifikatu
- ✓ Funtzioaren esakunea zehazten duen harreman logikoa ala haren egia-taula identifikatu

1. adibide ebatzia

Hurrengo esakunean definitutako funtzioa eta aldagaiak identifikatu, eta haren adierazpena ala egia-taula zehaztu:

“Hiru kidek osatutako epaimahai batek gai bati buruzko aldeko ala aurkako epaia eman du. Epaia kide bakoitzak emandako botoen gehiengoak zehazten du”.

Ebazpena

Kasu honetan, funtzioak eta aldagaiak ez dira esakunean esplizituki ematen. Esakunea horrela interpretatu behar da (beste interpretazio batzuk ere egin daitezke):

- ✓ Funtzioa: beste proposizio batzuen mende dagoen proposizio bakarra epaimahaiak emandako epaia da, kide bakoitzak emandako botoaren araberakoa baita. Beraz, funtzioa horrela adierazi daiteke:
  - f → epaia aldekoa da, eta epaimahaiko kideen botoen arabera, egia edo faltsua izan daiteke.
- ✓ Aldagaiak: aurreko funtzioa epaimahaiko hiru kideen botoen araberakoa denez, hiru aldagai egongo dira, eta aldagai horiek horrela adierazi daitezke:
  - *a* → lehen kideak aldeko botoa eman du
  - *b* → bigarren kideak aldeko botoa eman du
  - *c* → hirugarren kideak aldeko botoa eman du

  Aldagai bakoitza, kide bakoitzak emandako botoaren arabera, egia ala faltsua izan daiteke.

  (Oharra: kontuan hartu beste interpretazioak ere egin daitezkeela, adibidez funtzio gisa “epaia aurkako da” hartzea; edo aldagai gisa “lehen kideak aurkako botoa eman du”; eta horiek ere egia ala faltsuak izan daitezke. Edonola ere, interpretazioa bakoitzaren esku geratzen da, eta garrantzitsuena ebazpen osoan irizpide bera mantentzea da).
- ✓ Harreman logikoa: kasu honetan, ez da erraza funtzioa eta aldagaiak lotzen dituen adierazpena esakunean aurkitzea. Hala ere, funtzioaren egia-taula bat egin daiteke, epaimahaiaren boto posible guztiak eta kasu bakoitzeko funtzioaren balioa (emandako botoa) zehaztuz. Hiru kide dira, eta bakoitzak bi boto mota eman ditzake: aldekoa (aldagaia=1 egia) edo aurkakoa (aldagaia=0 faltsua). Beraz, guztira zortzi boto-konbinazio daude. Zortzi konbinazioak lortzeko, taula honetako metodoa erabiliko da; hau da, egia-taulako sarreretan sistema bitarrean kontatuko da, eta bakoitzean hiru bit jarriko dira:

sarrerak	*a*	*b*	*c*	*f(a,b,c)*	
0	0	0	0	0	Aurkako epaia: gehiengo osoak aurkako botoa eman du
1	0	0	1	0	Aurkako epaia: aurkako 2 boto eta aldeko boto 1 eman dira
2	0	1	0	0	Aurkako epaia: aurkako 2 boto eta aldeko boto 1 eman dira
3	0	1	1	1	Aldeko epaia: aldeko 2 boto eta aurkako boto 1 eman dira
4	1	0	0	0	Aurkako epaia: aurkako 2 boto eta aldeko boto 1 eman dira
5	1	0	1	1	Aldeko epaia: aldeko 2 boto eta aurkako boto 1 eman dira
6	1	1	0	1	Aldeko epaia: aldeko 2 boto eta aurkako boto 1 eman dira
7	1	1	1	1	Aldeko epaia: gehiengo osoak aldeko botoa eman du

2. adibide ebatzia

Etxe bateko alarma batek, leiho batean V1 sentsorea du, eta beste batean V2 sentsorea; atean, berriz, P sentsorea du. Sentsoreak leihoak ala atea irekitzean aktibatzen dira. Gainera, alarmak A kontrol-terminal bat dauka, alarma zaintza moduan aktibatu edo desaktibatzeko. Horrez gain, sistemak S irteera-sistema dauka alarma-egoerak jakinarazteko. Alarma zaintza moduan aktibatuta dagoenean, sistema hori atea edo leihoren bat irekitzean aktibatu beharko da.

Sentsoreak lotzen dituzten funtzioak, aldagaiak eta harreman logikoa zehaztu.

Ebazpena

- ✓ Funtzioa: Esakunean ageri den funtzio bakarra S irteera-terminalaren egoera da, aktibatuta ala desaktibatuta egon baitaiteke. Hori zehazteko leihoak eta atea irekita ote dauden ikusi behar da, baita alarma zaintza moduan jarrita dagoen ala ez. S=1 funtzioak aktibatuta dagoela, eta beraz, alarma-egoera dagoela esan nahi badu, S=0 funtzioak desaktibatuta dagoela esan nahi du; eta beraz, ez dagoela alarma-egoerarik.
- ✓ Aldagaiak: Funtzioa V1, V2, P eta A aldagaien menpean dago; hau da, sentsoreak aktibatuta ote dauden eta alarma zaintza moduan aktibatuta ote dagoen ikusi behar da funtzioa zehazteko. Adibidez, V1=1 bada, leihoko sentsorea aktibatuta dago; eta V1=0 bada, desaktibatuta dago.
- ✓ Harreman logikoa: kasu honetan, adierazpen logikoa esakunetik zuzenean ondorioztatu daiteke:

$$S(A,V1,V2,P) = A \cdot (V1 + V2 + P)$$

Adierazpen kanonikoak

Funtzio bat hainbat aldagaien menpe dagoenez, honako kontzeptuak zehaztu dira:

- ✓ **Minterm** edo biderkadura-termino kanonikoa: biderkadura bat da, eta bertan aldagai guztiak biderkatuta ageri dira, bai zuzenean edo ezeztatuta, baina behin bakarrik. Adibidez, *f*(*a*,*b*,*c*,*d*) funtzioan:

Mintermak dira	Ez dira mintermak	
$a \cdot b \cdot c \cdot d$	$a \cdot b \cdot c$	*d* aldagaia falta da
$a \cdot \overline{b} \cdot c \cdot d$	$\overline{a} \cdot c \cdot \overline{d}$	*b* aldagaia falta da
$a \cdot \overline{b} \cdot c \cdot \overline{d}$	$c \cdot \overline{d}$	*a* eta *b* aldagaiak falta dira
$\overline{a} \cdot \overline{b} \cdot c \cdot \overline{d}$	$a \cdot b \cdot \overline{b} \cdot c \cdot \overline{d}$	*b* aldagaia bi aldiz ageri da

- ✓ **Maxterm** edo batuketa-termino kanonikoa: batuketa horretan aldagai guztiak biderkatuta ageri dira, bai zuzenean edo ezeztatuta, baina behin bakarrik. Adibidez, f(a,b,c,d) funtzioan:

Maxtermak dira	Ez dira maxtermak	
$a + b + c + d$	$a + b + d$	*c* aldagaia falta da
$a + \overline{b} + c + d$	$\overline{a} + c + \overline{d}$	*b* aldagaia falta da
$a + \overline{b} + c + \overline{d}$	$c + \overline{d}$	*a* eta *b* aldagaiak falta dira
$\overline{a} + \overline{b} + \overline{c} + \overline{d}$	$a + b + \overline{b} + c + \overline{d}$	*b* aldagaia bi aldiz ageri da

- **Adierazpen kanonikoak:** funtzio batek izan ditzakeen adierazpen baliokide guztien artean, bi adierazpeni adierazpen kanoniko esaten zaie, eta ezaugarri hauek dituzte:
  - *Batuketa-adierazpen kanonikoa:* mintermen batuketaz osatuta dago soilik. Esaterako:

    $$f(a,b,c,d) = a \cdot b \cdot c \cdot d + a \cdot b \cdot \overline{c} \cdot d + a \cdot \overline{b} \cdot c \cdot \overline{d} + a \cdot \overline{b} \cdot \overline{c} \cdot \overline{d}$$

  - *Biderkadura-adierazpen kanonikoa:* maxtermen biderkaturaz osatuta dago soilik. Esaterako:

    $$f(x,y,z) = (x \cdot y \cdot \overline{z}) + (x \cdot \overline{y} \cdot \overline{z}) + (\overline{x} \cdot \overline{y} \cdot z)$$

Adierazpen kanonikoak nola erabili

Funtzio baten era kanonikoa interesgarria da kasu hauetarako:

- *Gailu konfiguragarriak erabiltzeko:* aurrerago, gai horri dagokion unitatean ikusiko dugun bezala, PLD's, PAL, GAL eta beste gailu elektroniko digital konfiguragarriekin lan egitean, komenigarria da funtzioak era kanonikoan adieraztea.
- *Egia-taula egiteko:* funtzio bat era kanonikoan ematean, egia-taula zuzenean egiten da, taulako sarrera bakoitzeko adierazpen logikoak ebaluatu beharrik gabe.

Horretarako metodoa minterm bakoitza (edo adierazpen kanonikoa batuketa ala biderkadura bat bada, maxterm bakoitza) zenbaki bitar gisa interpretatzen da, zenbaki bitar bakoitzarekin taulako sarrera bat eginez:

- Mintermetarako: ezeztatu gabeko aldagaiek 1 adierazten dute, eta ezeztatutakoek, 0. Adibidez: ($a \cdot \overline{b} \cdot c$) 101_2=5_{10} sarrerari dagokiona da.
- Maxtermetarako: ezeztatu gabeko aldagaiak 0 adierazten dute, eta ezeztatutakoek, 1. Adibidez: ($a + \overline{b} + c$) 010_2=2_{10} sarrerari dagokiona da.

Taula egiteko, funtzioari, haren adierazpen kanonikoaren arabera, honako balioak esleitzen zaizkio:

- Batuketa-adierazpena: mintermek adierazitako sarreretan 1 balioa, eta gainontzeko sarreretan 0 balioa.
- Biderkadura-adierazpena: maxtermek adierazitako sarreretan 0 balioa, eta gainontzeko sarreretan 1 balioa.
- *Funtzioaren adierazpen laburtua egiteko:* aipatu berri dugunarekin lotuta, funtzio bat forma kanonikoan ematen denean, hura laburtuta adierazteko notazioa erabil daiteke:
  - Batuketa-adierazpen kanonikoa: batuketa-eragilearen sinboloarekin adierazten da: Σ
  - Biderkadura-adierazpen kanonikoa: biderkadura-eragilearen sinboloarekin adierazten da: Π
  - Eragilearen oinean, funtzioak zenbat aldagai dituen zehaztuko da.
  - Eragilearen barnean, mintermek edo maxtermek (kasuaren arabera) adierazitako taulako sarreren zenbakiak idatziko dira, sistema hamartarrean edo hamaseitarrean. Esaterako:

$$f(a,b,c,d) = \overline{a} \cdot \overline{b} \cdot c \cdot d + \overline{a} \cdot b \cdot \overline{c} \cdot \overline{d} + \overline{a} \cdot b \cdot c \cdot d + a. \cdot \overline{b} \cdot c \cdot \overline{d} + a \cdot b \cdot \overline{c} \cdot d = \sum\nolimits_4 (3,4,7,A,D)$$

$$g(x,y,z) = (\overline{x} + \overline{y} + z) \cdot (x + \overline{y} + \overline{z}) \cdot (x + y + z) = \prod\nolimits_3 (6,3,0)$$

Adierazpen kanonikoa nola lortu

- *Egia-taulan oinarrituta:* aurreko atalean aipatutako metodoaren alderantzizkoa da. Beraz, adierazpen kanonikoan oinarrituta, egia-taula egiteko:
  - Batuketa-adierazpena: funtzioak 1 balio duten sarreren bitartez, mintermak eskuratu.
  - Biderkadura-adierazpena: funtzioak 0 balio duten sarreren bitartez, maxtermak eskuratu.
- Adierazpen ez-kanoniko batean oinarrituta: Prozedura bi pausotan egin behar da:
  - Aljebra boolearraren teoremen bitartez, adierazpen ez-kanonikoa garatu, biderkaduren (ziur asko ez-kanonikoak) batuketaz osatutako adierazpena edo batuketen biderkaduraz osatutakoa lortu arte.
  - Lortutako termino bakoitza garatu, termino kanoniko bat osatu arte. Horretarako, osagarri-axiomaren postulatua erabiliz, falta diren aldagaiak gehitu.

Ariketak funtzioen adierazpen kanonikoekin

1. adibide ebatzia

Funtzio honen egia-taula egin, eta notazio laburtuan adierazi:

$$f(a,b,c,d)=\overline{a}\cdot\overline{b}\cdot\overline{c}\cdot d+\overline{a}\cdot\overline{b}\cdot c\cdot d+\overline{a}\cdot b\cdot\overline{c}\cdot\overline{d}+a\cdot\overline{b}\cdot\overline{c}\cdot d+a\cdot\overline{b}\cdot c\cdot\overline{d}+a\cdot b\cdot c\cdot d$$

Ebazpena

- Notazio laburtua:

$$f(a,b,c,d)=\overline{a}\cdot\overline{b}\cdot\overline{c}\cdot d+\overline{a}\cdot\overline{b}\cdot c\cdot d+\overline{a}\cdot b\cdot\overline{c}\cdot\overline{d}+a\cdot\overline{b}\cdot\overline{c}\cdot d+a\cdot\overline{b}\cdot c\cdot\overline{d}+a\cdot b\cdot c\cdot d$$

	↓	↓	↓	↓	↓	↓
Aldagaiak 1arekin ordezkatu, eta ezeztatutakoak, 0arekin, batuketa-adierazpena delako	0001	0011	0100	1001	1010	1111
	↓	↓	↓	↓	↓	↓
	1	3	4	9	*A*	*F*

$$f(a,b,c,d)=\sum\nolimits_4(1,3,4,9,A,F)$$

✓ Egia-taula:

– Mintermek zehaztutako sarrerei 1 balioa esleitu:

	sarrerak	a	b	c	d	f(a,b,c,d)
	0	0	0	0	0	
→	1	0	0	0	1	1
	2	0	0	1	0	
→	3	0	0	1	1	1
→	4	0	1	0	0	1
	5	0	1	0	1	
	6	0	1	1	0	
	7	0	1	1	1	
	8	1	0	0	0	
→	9	1	0	0	1	1
→	A	1	0	1	0	1
	B	1	0	1	1	
	C	1	1	0	0	
	D	1	1	0	1	
	E	1	1	1	0	
→	F	1	1	1	1	1

– Gainontzeko sarreretan 0 jarri:

sarrerak	a	b	c	d	f(a,b,c,d)
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
A	1	0	1	0	1
B	1	0	1	1	0
C	1	1	0	0	0
D	1	1	0	1	0
E	1	1	1	0	0
F	1	1	1	1	1

2. adibide ebatzia

Unitate honetako auto-ebaluazio ataleko 7. ariketako egia-taulan emandako funtzioaren batuketa eta biderkadura-adierazpen kanonikoa eskuratu.

Ebazpena

Hau da egia-taula, eta bertan oinarrituta mintermak (1ak) edo maxtermak (0ak) eskuratu daitezke:

a	b	c	d	f(a,b,c,d)	mintermak	maxtermak
0	0	0	0	0		$a+b+c+d$
0	0	0	1	0		$a+b+c+\overline{d}$
0	0	1	0	0		$a+b+\overline{c}+d$
0	0	1	1	0		$a+b+\overline{c}+\overline{d}$
0	1	0	0	0		$a+\overline{b}+c+d$
0	1	0	1	1	$\overline{a}\cdot b\cdot\overline{c}\cdot d$	
0	1	1	0	1	$\overline{a}\cdot b\cdot c\cdot\overline{d}$	
0	1	1	1	1	$\overline{a}\cdot b\cdot c\cdot d$	
1	0	0	0	0		$\overline{a}+b+c+d$
1	0	0	1	0		$\overline{a}+b+c+\overline{d}$
1	0	1	0	0		$\overline{a}+b+\overline{c}+d$
1	0	1	1	1	$a\cdot\overline{b}\cdot c\cdot d$	
1	1	0	0	1	$a\cdot b\cdot\overline{c}\cdot\overline{d}$	
1	1	0	1	1	$a\cdot b\cdot\overline{c}\cdot d$	
1	1	1	0	1	$a\cdot b\cdot c\cdot\overline{d}$	
1	1	1	1	1	$a\cdot b\cdot c\cdot d$	

Batuketa-adierazpen kanonikoa:

$$f(a,b,c,d)=\sum_4(5,6,7,11,12,13,14,15)=$$
$$=\overline{a}\cdot b\cdot\overline{c}\cdot d+\overline{a}\cdot b\cdot c\cdot\overline{d}+\overline{a}\cdot b\cdot c\cdot d+a\cdot\overline{b}\cdot c\cdot d+a\cdot b\cdot\overline{c}\cdot\overline{d}+a\cdot b\cdot\overline{c}\cdot d+a\cdot b\cdot c\cdot\overline{d}+a\cdot b\cdot c\cdot d$$

Biderkadura-adierazpen kanonikoa:

$$f(a,b,c,d)=\prod 4(0,1,2,3,4,8,9,10)=$$
$$=(a+b+c+d)\cdot(a+b+c+\overline{d})\cdot(a+b+\overline{c}+d)\cdot(a+b+\overline{c}+\overline{d})\cdot$$
$$.(a+\overline{b}+c+d)\cdot(\overline{a}+b+c+d)\cdot(\overline{a}+b+c+\overline{d})\cdot(\overline{a}+b+\overline{c}+d)$$

3. adibide ebatzia

Funtzio honen adierazpen kanonikoa eskuratu:

$$f(a,b,c)=\overline{a}\cdot\overline{b}+a\cdot b+\overline{a\cdot c}+a\cdot\overline{c}$$

Ebazpena

Kasu honetan, funtzioa ia biderkadura ez-kanonikoen batuketa gisa adierazita dago. Hirugarren terminoari De Morgan teorema aplikatuz, erabat garatzen da:

$$f(a,b,c)=\overline{a}\cdot\overline{b}+a\cdot b+\overline{a\cdot c}+a\cdot\overline{c}=\overline{a}\cdot\overline{b}+a\cdot b+\overline{a}+\overline{c}+a\cdot\overline{c}$$

Termino bakoitza kanonikoa izan dadin, falta den aldagaia gehituko da era honetan:

$$f(a,b,c) = \underbrace{\overline{a}\cdot\overline{b}}_{\text{c falta da}\downarrow} + \underbrace{a\cdot b}_{\text{c falta da}\downarrow} + \underbrace{\overline{a}}_{\text{b eta c falta dira}\downarrow} + \underbrace{\overline{c}}_{\text{a eta b falta dira}\downarrow} + \underbrace{a\cdot\overline{c}}_{\text{b falta da}\downarrow}$$

$$f(a,b,c) = \overline{a}\cdot\overline{b}\cdot(c+\overline{c}) + a\cdot b\cdot(c+\overline{c}) + \overline{a}\cdot(b+\overline{b})\cdot(c+\overline{c}) + \overline{c}\cdot(a+\overline{a})\cdot(b+\overline{b}) + a\cdot\overline{c}\cdot(b+\overline{b})$$

Aipagarria da, $(x+\overline{x})$ motako terminoak biderkatzean ez dela balioa aldatzen, osagarriaren postulatuaren ondorioz, 1ez biderkatzea bezala baita, eta beraz, elementu neutroen postulatuaren arabera, balioak ez baitira aldatzen.

Parentesi bakoitzari banakaria aplikatuz, termino kanonikoak lortzen dira:

$$\begin{aligned} f(a,b,c) &= \overline{a}\cdot\overline{b}\cdot(c+\overline{c}) + a\cdot b\cdot(c+\overline{c}) + \overline{a}\cdot(b+\overline{b})\cdot(c+\overline{c}) + \overline{c}\cdot(a+\overline{a})\cdot(b+\overline{b}) + a\cdot\overline{c}\cdot(b+\overline{b}) = \\ &= \overline{a}\cdot\overline{b}\cdot c + \overline{a}\cdot\overline{b}\cdot\overline{c} + a\cdot b\cdot c + a\cdot b\cdot\overline{c} + \overline{a}\cdot b\cdot c + \overline{a}\cdot b\cdot\overline{c} + \overline{a}\cdot\overline{b}\cdot c + \overline{a}\cdot\overline{b}\cdot\overline{c} + \overline{c}\cdot a\cdot b + \overline{c}\cdot a\cdot\overline{b} + \\ &+ \overline{c}\cdot\overline{a}\cdot b + \overline{c}\cdot\overline{a}\cdot\overline{b} + a\cdot\overline{c}\cdot b + a\cdot\overline{c}\cdot\overline{b} \end{aligned}$$

Azkenik, errepikatutako terminoak kentzeko, idenpotentzia-legea aplikatuz, adierazpen kanonikoa lortzen da:

$$f(a,b,c) = \overline{a}\cdot\overline{b}\cdot c + \overline{a}\cdot\overline{b}\cdot\overline{c} + a\cdot b\cdot c + a\cdot b\cdot\overline{c} + \overline{a}\cdot b\cdot c + \overline{a}\cdot b\cdot\overline{c} + a\cdot\overline{b}\cdot\overline{c} = \sum_3(0,1,2,3,4,6,7)$$

Prozesadore digital elektronikoetan aplikatzea

Lehen esan dugun bezala, behaketa batzuetan eta funtzionamendu-baldintza batzuetan oinarrituta jarduketa batzuk sortzea da kontrol-sistema baten helburua. Adibidez, aparkaleku bateko atea kontrolatzeko (atea irekitzeko), autoak behatuz eta etengailu bat sakatuz, jarduera zehatz bat sorrarazi nahi da. Gainera, baldintzak behatzeaz gain, funtzionamendu-baldintza bat ere egon daiteke, esaterako, behatutako baldintza batek baiezko emaitza izan dezake. Hori guztia adierazpen honetan laburbil daiteke: “atea ireki behar da sarreran autorik badago edo etengailua sakatzen badute”.

Orain arte egindako ariketetan susmatu dugun bezala, kontrol-sistema bakunenak, esaterako atearenak, funtzio logiko baten bidez adieraz daitezke, harreman hauek zehaztuta:

- ✓ Funtzioa: kontrolatu behar den elementua da.
- ✓ Aldagaiak: sistema kontrolatzeko egiten diren behaketak dira.
- ✓ Adierazpena: funtzionamendu-baldintzak dira.

Beraz, kontrol-prozesua funtzio logiko baten bidez adierazten da. Atearen kasuan, aldagai hauek zehazten dira:

- ✓ P: atea ireki behar da $\rightarrow$ funtzioa
- ✓ C: sarreran autorik badago $\rightarrow$ aldagaia
- ✓ I: etengailua sakatzen dute $\rightarrow$ aldagaia

Kontrol-prozesua zehazten duen funtzio logikoa hau da:

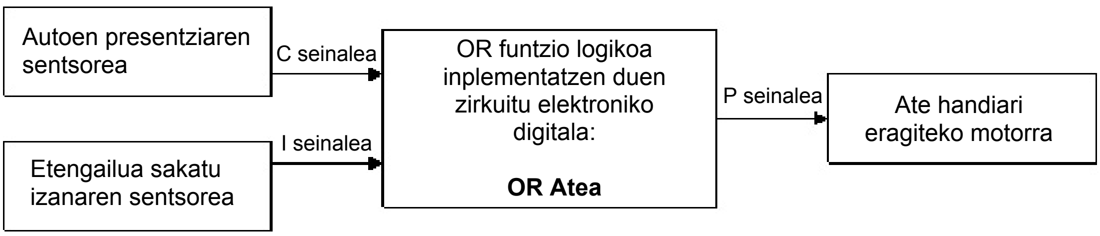
$$P(C,I) = C + I$$

Atearen kontrolatzailea gizakia bada, funtzio logiko hori izango du buruan. Gizakiaren ordez sistema elektroniko digital bat ezarri nahi bada, pauso hauek eman beharko dira:

- ✓ C eta I aldagaien balio logikoak adierazteko seinale elektrikoak igortzeko sentsoreak jarri behar dira. Hau da, sentsore batek, autoak sarreran daudenean (C=1), 5 V emango ditu, eta 0 V autoak daudela faltsua bada (C=0). Beste sentsore batek 5 V emango ditu etengailua sakatzen dutenean (I=1), eta 0 V etengailua sakatu dutela faltsua denean (I=0). Modu horretan sortutako seinaleak, hurrenez hurren C eta I seinaletzat jo daitezke, horrela haien zentzua hobeto identifikatzeko.
- ✓ Atea mugitzeko motor eragingailu bat jarri behar da, tentsio-linea batekin kontrolatutakoa. Kontrol-linean, adibidez ateari 5 V-eko seinalea aplikatzen bazaio, atea ireki egingo da, eta 0 V-eko seinalea aplikatzen bazaio, itxi egingo da (adibidez, malguki batek tira egiten diolako). Atea kontrolatzen duen seinalea P izendatu daiteke, era horretan P kontrol-funtzioarekin identifikatzeko.
- ✓ Kontrol-funtzio logikoa zehazten duen harremanaren arabera, C eta I seinaleen bitartez P seinalea balio egokiekin igortzen duen zirkuitu elektronikoa ezarri behar da. Kasu honetan, zirkuitu elektronikoak batuketa ala OR funtzio logikoaren arabera funtzionatu beharko du. Badago horrelako zirkuitu bat OR *atea* izena du, eta OR funtzioaren taularen arabera funtzionatzen du:

C seinalearen balioa	I seinalearen balioa	P seinalerako sortutako balioa
0 V	0 V	0 V
0 V	5 V	5 V
5 V	0 V	5 V
5 V	5 V	5 V

5 V-eko tentsioak 1 balioa adierazten du, eta 0-V-ekoak, 0a.



Orokorrean, kontrol-sistema bat funtzio logiko baten bidez adierazi daitekeenean, elektronika digitalari esker, aldagai logikoen balioak adierazten dituzten seinale elektrikoen bitartez, funtzio horren portaera aplikatzeko beharrezko zirkuituak sortzen dira. Orain arte ikasitakoa kontuan hartuta, zirkuitu horiek diseinatzeko prozesua honakoa da:

- ✓ Kontrol-sistemako esakuna edo haren berezitasunak oinarritzat hartuta, aldagaiak eta funtzioak identifikatzea.
- ✓ Esakunea edo berezitasunak oinarritzat hartuta, egia-taula osatzea.
- ✓ Egia-taula oinarritzat hartuta, funtzioaren adierazpen kanonikoa osatzea.

- ✓ Adierazpen kanonikoan oinarrituta, aljebra boolearraren teoremak erabilita, adierazpen sinplifikatua lortzea.
- ✓ Kontrol-sistema zehazten duen funtzio sinplifikatua eskuratu ondoren, zirkuituan aplikatzea, elektronika digitalak ematen dituen ate eta zirkuituen bidez.

Hurrengo unitateetan prozesu hori sinplifikatzeko teknikak ikasiko ditugu.

Kontuan hartu behar da aldagaien balio logikoak ez direla sistemaren araberakoak; horren ordez, inguruak zehazten ditu, eta beraz, edozein balio har dezakete. Bestalde, funtzioak kontrol-sistemak sortutako balioak hartu behar ditu, eta, ondorioz, sistemaren erantzukizuna da funtzioak egoera guztietarako balio egokiak izatea.

Baita ere, ezin dugu ahaztu, seinale digital bakoitzaren atzean aldagai logiko bat dagoela; eta beraz, proposizio bat, batzuetan ahoz, ohiko lengoaian adieraztea zaila izan arren.

Hurrengo unitateetan funtzio logikoak aplikatzeko zirkuitu elektroniko digital mota desberdinak aztertuko ditugu, eta era horretan, kontrolatzaile elektroniko digitalak garatzen eta kontrol-sistemak zehazten dituzten funtzio logikoak beste metodo batzuen bidez eskuratzen ikasiko dugu.

- **Logika positiboa eta negatiboa**

Ateari buruzko adibidea aipatu berri dugu. Bertan, 1 balio logikoa (egia) 5-Veko tentsioarekin adierazita zegoen, eta 0 balio logikoa (faltsua), 0 V-eko tentsioarekin adierazten zen. Harreman hori arbitrarioa da, arazo hori konpontzeko kontrako esleipena ere egin baitaiteke; hau da, 1 balioa 0 V-en bidez adieraziz; eta 0a 5 V-en bitartez. Era horretan, zirkuitua desberdina izan arren, kontrol-funtzio bera beteko luke. Beraz, irizpidea aukeratzea diseinatzailearen esku dago.

Elektronika digitalean, beti bi tentsio-mailekin egiten da lan, 0 eta 1 balio logikoak adierazi ahal izateko. Tentsio horien balio zehatza kasu bakoitzean erabilitako teknologiaren araberakoa izango da. Diseinatzaileak, kasu bakoitzean dituen baldintzen arabera, balio logiko eta tentsio-mailen arteko harreman jakin bat zehaztuko du. Egoera horren ondorioz, bi logika mota bereizten dira elektronika digitalean:

- ✓ Logika positiboa: 1 balioa maila handieneko tentsioarekin adierazten da, eta 0 balioa, berriz, maila txikieneko tentsioarekin.
- ✓ Logika negatiboa: 0 balioa maila handieneko tentsioarekin adierazten da, eta 1 balioa, berriz, maila txikieneko tentsioarekin.

1.3 Autoebaluazioa

1.1 ariketa

Pasa hamartarretara zenbaki hamaseitar hauek: E8, D1A, FE0, CACA, CAFE, DE5E0.

Ebazpena

- ✓ E8H=232_{10}=11101000_2
- ✓ D1AH=3354_{10}=110100011010_2
- ✓ FE0H=4064_{10}= 111111100000_2
- ✓ CACAH=51914_{10}= 1100101011001010_2
- ✓ CAFEH=51966_{10}= 1100101011111110_2
- ✓ DE5E0H=910816_{10}= 11011110010111100000_2

1.2 ariketa

Pasa hamaseitarretara zenbaki hamartar hauek: 256, 3687, 12584, 698574, 12234842.

Ebazpena

- ✓ 256_{10}=100H
- ✓ 3687_{10}=E67H
- ✓ 12584_{10}=3128H
- ✓ 698574_{10}=AA8CEH
- ✓ 12234842_{10}=BAB05AH

1.3 ariketa

Pasa hamaseitarretara zenbaki bitar hauek: 1100110, 1001111101, 110101010101, 111100000001110, 111110001101010100011110.

Ebazpena

- ✓ 1100110_2=66H
- ✓ 1001111101_2=27DH
- ✓ 110101010101_2=D55H
- ✓ 111100000001110_2=780EH
- ✓ $111110001101010100011110_2$=F8D51EH

1.4 ariketa

Pasa hamartarretara edo bitarretara zenbaki zortzitar hauek: 2654, 5752, 25467, 2101445.

Ebazpena

- ✓ 2654_8=1452_{10}=10110101100_2
- ✓ 5752_8=3050_{10}=101111101010_2
- ✓ 25467_8=11063_{10}=10101100110111_2
- ✓ 2101445_8=557861_{10}=10001000001100100101_2

1.5 ariketa

Pasa zortzitarretara zenbaki hamartar hauek: 256, 3687, 12584, 698574, 12234842.

Ebazpena

- ✓ 256_{10}=411_8
- ✓ 3687_{10}=7147_8
- ✓ 12584_{10}=30450_8
- ✓ 698574_{10}=2524316_8
- ✓ 12234842_{10}=56530133_8

1.6 ariketa

Egin aurreko ariketetako oinarri-aldaketak Windos-eko kalkulagailua erabiliz.

1.7 ariketa

Pasa hamartarretara BCD zenbaki hauek: 38, 568, 785, 1111, 5566.

Ebazpena

- ✓ 38_{10}=111000_{BCD}
- ✓ 568_{10}=10101101000_{BCD}
- ✓ 785_{10}= 11110000101_{BCD}
- ✓ 1111_{10}= 1000100010001_{BCD}
- ✓ 5566_{10}= 101010101100110_{BCD}

1.8 ariketa

Pasa hamartarretara BCD zenbaki hauek: 1100110, 10110010110, 111100110000111, 10000000110000001.

Ebazpena

- ✓ 1100110_{BCD}=66_{10}
- ✓ 10110010110_{BCD}=596_{10}
- ✓ 111100110000111_{BCD}=7987_{10}
- ✓ 10000000110000001_{BCD}=10181_{10}

1.9 ariketa

Sinplifikatu adierazpen hau:

$$(a+b+\overline{c})\cdot(\overline{a}+\overline{b}+c)\cdot(\overline{a}+\overline{b}+\overline{c})\cdot(\overline{\overline{a}+\overline{c}})\cdot(\overline{a}+c)$$

Ebazpena

$$a\cdot\overline{b}\cdot c$$

1.10 ariketa

Identifikatu hurrengo esakunean definitutako funtzioa eta aldagaiak, eta zehaztu dagokion adierazpena ala egia-taula:

“Familia bateko lau kideek —Andres, Beatriz, Carlos eta Diana— bi jatetxe hauetako batera afaltzera joateko bozkatuko dute: Pepe jatetxea edo Juan jatetxea. Gehiengoak aukeratutako jatetxera joango dira afaltzera. Berdinketarik badago, Beatrizek bozkatutako jatetxera joango dira".

Ebazpena

- ✓ Funtzioa:
  - f → Pepe jatetxea aukeratu dute

- Aldagaiak:
  - A → Andresek Pepe jatetxera joatea bozkatu du
  - B → Beatrizez Pepe jatetxera joatea bozkatu du
  - C → Carlosek Pepe jatetxera joatea bozkatu du
  - D → Dianak Pepe jatetxera joatea bozkatu du
- Harreman logikoa:

A	B	C	D	f(A,B,C,D)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Ariketa hau ebazteko, beste aldagai edo funtzio batzuk ere zehaztu daitezke; esaterako, "f → Juan jatetxera joatea aukeratu dute", edo horren baliokidea, "f → aukeratutako jatetxea ez da Pepe jatetxea". Edonola ere, funtzioa aukeratu ondoren, irizpide bera mantendu behar da ebazpen osoan.

1.11 ariketa

Lortu funtzio honen egia-taula, eta adierazi notazio laburtuan:

$$f(a,b) = a \cdot b + \overline{a} \cdot b$$

Ebazpena

- Notazio laburtua:

$$f(a,b) = \prod\nolimits_2 (0,2)$$

- Egia-taula:

a	b	f(a,b)
0	0	0
0	1	1
1	0	0
1	1	1

1.12 ariketa

Lortu funtzio honen adierazpen kanonikoa:

$$f(a,b,c) = a \cdot \left(\overline{b} + \overline{c}\right) + c$$

Ebazpena

$$f(a,b,c) = \sum_3(1,3,4,5,6,7) = \overline{a} \cdot \overline{b} \cdot c + \overline{a} \cdot b \cdot c + a \cdot \overline{b} \cdot \overline{c} + a \cdot \overline{b} \cdot c + a \cdot b \cdot \overline{c} + a \cdot b \cdot c$$

1.13 ariketa

Lortu funtzio honen adierazpen kanonikoa:

$$f(a,b) = a \oplus b$$

Ebazpena

Egia-taulan oinarrituta, hau lortzen da:

$$f(a,b) = \sum_2(1,2) = \overline{a} \cdot b + a \cdot \overline{b} = \prod_2(0,3) = (a+b) \cdot (\overline{a} + \overline{b})$$