

# ■ 2 ■

## 10 gehi 10... 100! Oinarrizko sistema digitalen portaera

---

### SARRERA

Prozesadore baten osagai nagusietako bat unitate aritmetiko/logikoa da, non oinarrizko eragiketa aritmetikoak eta logikoak exekututzen baitira. Saio honetan, eragiketa horietako sinpleenak aztertuko ditugu: **NOT**, **AND** eta **OR eragiketa logikoak**, eta eragiketa aritmetiko sinpleena, zenbaki naturalen **batuketa**.

Zirkuitu integratu programagarri batean gauzatuko ditugu eragiketa horiek, eta haien funtzionamendua egiaztatuko dugu. Horretarako, sistema digitalen prototipoak egiteko fakultateko ikasleek erabiltzen duten "diseinu-plaka" eta industrian asko erabiltzen den softwarea erabiliko ditugu.

### DESKRIBAPENA

Konputagailuetako prozesadoreek informazio digitala prozesatzen dute. Informazioa edo datuak adierazteko, kode bitarrak erabiltzen dira, eta kode horien "digitu" edo "bit" bakoitzak bi balio baino ezin du hartu: 0 edo 1. Bi balio horiei balio logiko deritze, eta, maiz, beste modu honetan ere adierazten dira: egiazkoa (1) eta faltsua (0).

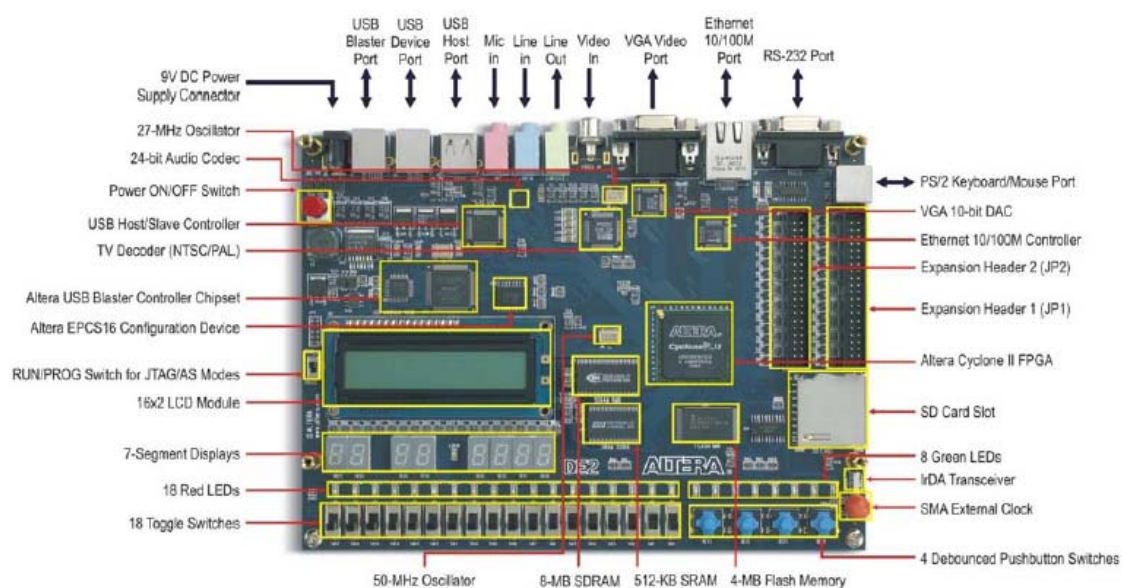
Konputagailuak zirkuitu digitalez osatuta daude, eta zirkuitu horiek kode bitarrak prozesatzen dituzte. Zenbait zirkuitu oso eragiketa sinpleak egiten dituzte, baina, jakina, konplexutasun handiko zirkuituak ere badaude konputagailu baten barruan.

Laborategi-saio honetan, oinarrizko zirkuitu digital batzuen portaera aztertuko dugu:

1. lehenik, oinarrizko hiru **eragiketa logikoak** egiten dituzten zirkuituak: **NOT** (ezeztapen logikoa), **OR** (batuketa logikoa) eta **AND** (biderketa logikoa) izeneko zirkuituak.
2. eta, gero, oinarrizko **eragiketa aritmetikoa**, **batuketa**, alegia, egiten duen zirkuitua; adibide honetarako, 4 biteko bi zenbaki natural batzen dituen oinarrizko batugailua.

Zirkuitu horien portaera logikoa analizatzeko, sistema digitalen diseinurako tresna estandar konplexu bat erabiliko dugu: Altera etxeko Quartus II aplikazioa; software-tresna oso ezaguna eta erabilia da industrian zein unibertsitatean eta ikerkuntza-zentroetan. Tresna horren bidez, zirkuitu horiek definitu eta marraztu, haien sarrerak (prozesatzen duten informazioa) eta irteerak (emaitzak) zehaztu, eta, azkenik, sistema logikoa txip programagarri batean integra daiteke.

Software-tresna horrekin batera, prototipoak sortzeko eta egiaztatzeko oso egokia den proba-txartel bat erabiliko dugu (DE2 txartela). Programatuko dugun txiparekin batera, txartelak hainbat osagai ditu: etengailuak (seinale bitarrak adierazteko: 1/0), LED diodoak (seinale digitalak "ikusteko bonbillatxoak"), LCD pantaila txiki bat, 7 segmentuzko digituak, eta beste hainbat gailu. Hona hemen txartel horren irudi bat. Erdialdean edo, FPGA motako txipa dago (Cyclone II familiako EP2C35F672C6N izena du!); zirkuitu horretan programatuko dugu sistema digital osoa. Beheko aldean, datuak sartzeko etengailuak (ingelesez, *switch*) daude, eta, haien gainean, LED diodoak (seinale bitarrak ikusteko); ezkerrean LCD pantaila eta 7 segmentuzko digituak.



1. irudia. DE2 prototipo-txartela.

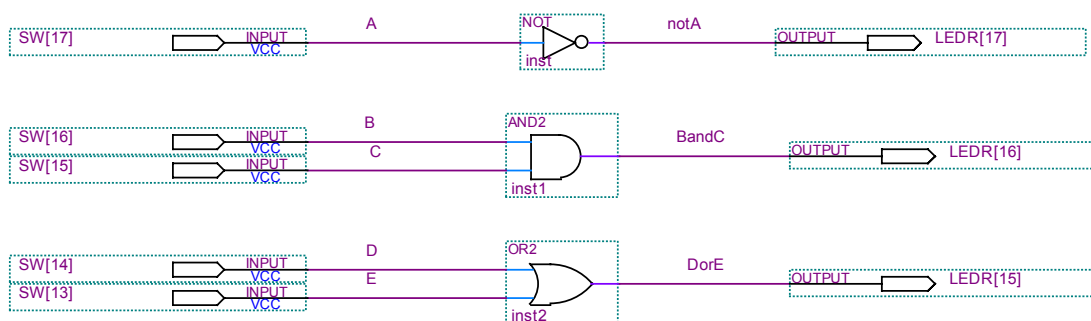
## 1. ARIKETA: ate logikoen portaera.

Proposatzen dizugun lehenbiziko ariketan, oinarrizko hiru zirkuitu logikoen funtzionamendua analizatu behar da. Oro har, zirkuitu horiei *ate logiko* deritze eta, konkretuki, hiru hauek analizatuko dituzu: **NOT** atea (ezeztapen logikoa), **OR** atea (batuketa logikoa) eta **AND** atea (biderketa logikoa). Osagai horiek abiapuntua dira sistema konplexuagoak eraiki ahal izateko. Izan ere, Boole matematikariak egiaztatu zuenez, edozein funtzio logiko, konplexuena ere, oinarrizko zirkuitu horien konbinazio gisa eraiki daiteke.

Ate logikoen sarrerek eta irteerak bitarrak dira, hau da, 0 edo 1 balioak soilik hartzen dituzte. NOT atea sarrera bat eta irteera bat ditu; irteeran, sarrerako kontrako balioa ematen du. OR eta AND ateen bi sarrera eta irteera bat dituzte; irteeran bi sarreren batura edo biderkadura logikoa ematen dituzte, hurrenez hurren.

Egiazta dezagun hiru ate horien funtzionamendua.

1. Ireki **ateak.qpf** izeneko fitxategia, Quartus aplikazioarekin, eta gero, **ateak.bdf** fitxategia. Eskema logiko hau ageriko da pantailan:



2. **irudia.** Oinarrizko ate logikoak: NOT atea, AND atea eta OR atea.

Horiek dira erabiltzen diren sinbolo edo ikurrak hiru funtzio logiko horietarako.

**Oharra:** *or* funtzioa adierazteko + ikurra erabiltzen da ( $D+E$ ); *and* funtziorako  $\cdot$  ikurra ( $B \cdot C$ ), eta *not* funtziorako barra bat aldagaiaren izenaren gainean ( $\overline{A}$ ).

2. Baina, jakina, ez da nahikoa zirkuituak "marraztea". Orain zirkuitu horiek DE2 txarteleko FPGA txipean "programatu" behar dituzu; hori baino lehen, ordea, hainbat informazio gehitu behar zaio marraztu dugun zirkuituari: sistemaren sarrera eta irteera logiko bakoitzari txarteleko sarrera- edo irteera-gailu bat esleitu behar diogu (etengailuak, LED diodoak...).

Probak azkarrago egin ditzazun, gehituta dago informazio hori guztia eskema logikoan. Hala, txarteleko sarrera (etengailu, *switch*) eta irteera (LED diodo) hauek erabiliko ditugu:

- Sarrerak:

- A** → SW17
- B** → SW16, **C** → SW15
- D** → SW14, **E** → SW13

- Irteerak:

- not A** → LEDR17
- B and C** → LEDR16
- D or E** → LEDR15

Prest duzu sistema digitala FPGA txipan programatzeko. Ez da zaila hori egitea; segi irakasleak azalduko dituen urratsak.

**Kito! Sortu duzu dagoeneko zure lehen zirkuitu digitala.** Egiaztatu nahi izango duzu nola funtzionatzen duen, ezta? Bada, horretarako, nahikoa duzu zirkuitu bakoitzaren sarrerei dagozkien etengailuetan 0 edo 1 balioa jartzea eta zirkuituaren erantzuna ikustea, dagokion LED diodoan. Gogoratu: diodoa pizten bada, emaitza 1 da; eta ez bada pizten, emaitza 0 da.

Idatzi lortzen dituzun emaitzak honako taula hauetan. **ADI:** egiazta itzazu sarreretako konbinazio guztiak!

> **NOT ate logikoaren portaera** (sarrera bat eta irteera bat)



A	$\bar{A}$
0	
1	

> **AND ate logikoaren portaera** (bi sarrera eta irteera bat)



B	C	$B \cdot C$

> **OR ate logikoaren portaera** (bi sarrera eta irteera bat)



D	E	$D + E$

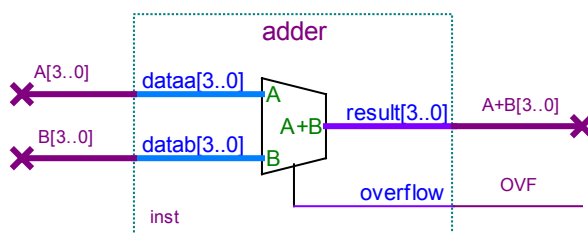
Sortu dituzun taulei **egia-taula** deritze eta oinarrizko zirkuitu digitalen portaera adierazteko erabili ohi dira.

## 2. ARIKETA: 4 biteko bi zenbaki naturalen batuketa aritmetikoa.

Batuketa oinarritzko eragiketa da edozein sistema digitaletan (eta, jakina, edozein prozesadoretan!); hainbeste erabiltzen da, ezen ez da beharrezkoa batugailuak behin eta berriro diseinatu behar, jadanik diseinatuta baitaude. Batugailu sinpleenak sarreretako bi zenbakiak batu egiten du, eta irteeran batura eskaintzen du.

Bigarren ariketa praktiko honetan, batugailu sinple baten funtzionamendua egiaztatuko dugu: zirkuituak **4 biteko bi zenbaki natural** batzen ditu, eta **batura** sortzen du, beste 4 biteko zenbaki natural bat; horrez gain, bit bateko seinale bat sortzen du, **bururakoa** (*overflow* ingelesez), batura zuzena den adierazteko (adi: zenbait kasutan, 4 biteko bi zenbakiaren batura ezin da adierazi bakarrik 4 bit erabiliz).

Irudi honetan, 4 biteko batugailu baten eskema logikoa ageri da.



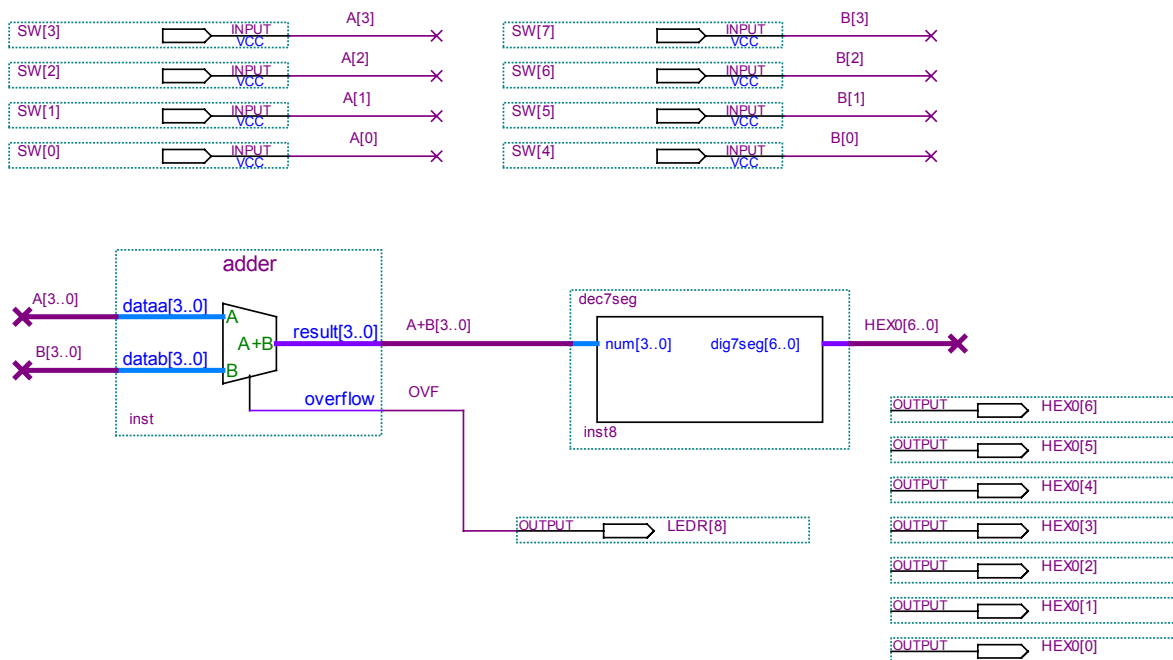
**3. irudia.** 4 biteko bi zenbaki, **A** eta **B**, batzen duen oinarritzko batugailua (ingelesez, *adder*). Emaitza gisa, **A+B** eta **OVF** (bururakoa) seinalea eskaintzen ditu.

1. Batugailuari ekin baino lehen, galdera pare bat. Zenbat zenbaki natural kodetu daitezke 4 bitekin? Zein dira zenbaki horien kode bitarrak? Gogora ezazu: konputagailuek ez dute 10 oinarrian lan egiten, 2 oinarrian baizik, bitarrean!  
Beraz, errepara dezagun 2 oinarria. Bete ezazu hurrengo taula:

Zenbaki naturala	Kode bitarra (4 bit)	Zenbaki naturala	Kode bitarra (4 bit)
0	0 0 0 0		
1	0 0 0 1		
2	0 0 1 0		
3			

Egiazta dezagun orain batugailuaren funtzionamendua.

- Ireki ezazu **batugailua.qpf** fitxategia Quartus programarekin, eta, gero, **batugailua.bdf** fitxategia. Honako eskema ikusiko duzu pantailan:



#### 4. irudia. Batugailua eta 7 segmentu deskodegailua.

Ikus dezakezunez, eskema horretan batugailua eta beste zirkuitu bat ageri da. Batugailuaren emaitza, batura alegia, bitarrez adierazita dago, 4 bitetan, baina emaitza "erosoago" ikusteko, ohikoa da beste kode bat erabiltzea: 7 segmentuzko kodea. Kode horren bidez, zenbakiak adierazteko oso ezagunak diren gailuak kontrolatzen dira: "7 segmentuzko digituak". Kode batetik bestera itzultzen duen zirkuituari deskodegailu deritzo.

- Aurreko ariketan bezala, zirkuituaren sarrerei eta irteerei proba-txartelaren etengailuak, LEDak eta digituak esleitu behar zaizkie. Ariketa honetarako, honelako esleipena erabiliko dugu:

- Sarrerak:

**A** → SW3, SW2, SW1, SW0  
**B** → SW7, SW6, SW5, SW4

- Irteerak:

**A+B** → HEX0 digitua  
**OVF** → LEDG7

4. Azkenik, aurreko ariketan bezala zirkuitua programatu behar da. Gero, egiazta ezazu zirkuituaren funtzionamendua. Horretarako, probak egin behar dira; adibidez, taula honetakoak. bete ezazu taula eta idatzi lortzen dituzun emaitzak.

A		B		A+B		OVF
hamartarrez	bitarrez	hamartarrez	bitarrez	hamartarrez	bitarrez	
0		3				
4		5				
1		6				
8		2				
11		2				
9		8				

Piztu al da *overflow* seinalea kasuren batean? Zergatik? Proposatu bururakoa edo *overflow* sortzen duen beste kasu bat.

**Horixe da gaurko guztia.** Analizatu dituzun zirkuituak seinale bitarrak prozesatzen dituen edozein sistema digitalaren barruan aurki ditzakezu. Esaterako, zenbaki naturalen batugailua prozesadoreen *unitate aritmetiko/logiko* izeneko gailuaren oinarritzko osagaia da. Jakina, azpisistema hori hemen analizatu duguna baino dezente konplexuagoa da, eta hainbat eragiketa — batuketak, kenketak, biderketak, zatiketak...— egiteko eta formatu desberdinetan adierazitako datuak prozesatzeko gauza da. Horiek eta beste hainbat gai ikas ditzakezu gurekin, hala nahi baduzu.

## LABURPENA

### Praktika honetan ikusitako kontzeptuak:

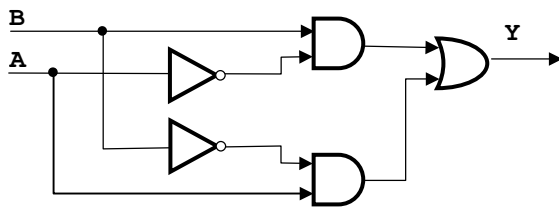
- > **Balio logikoak.** Zirkuitu digitalek soilik bi balio, 1 eta 0, hartzen dituzten seinaleak prozesatzen dituzte.
- > **NOT, AND eta OR ate logikoak.** Gailu logiko sinpleenak dira; seinale logikoak prozesatu eta haien ezeztapena, biderketa edo batuketa egiten dituzte.
- > **Kodeketa bitarra.** Konputagailuek prozesatzen dituzten datuak kodetuta daude, bitarrez.  $n$  bitekin  $2^n$  zenbaki kodetu daitezke.
- > **Batugailuak** *unitate aritmetiko/logikoen* oinarriak dira; unitate horiek exekutatzen dituzte prozesadoreen eragiketak.
- > Hainbat kasutan,  $n$  biteko bi zenbakiren batura ezin da  $n$  bitetan adierazi; beraz, batugailuak seinale berezi bat —**bururakoa (overflow)**— aktibatzen du egoera hori seinalatzeko (zure kalkulagailuan ere gauza bera gertatzen da adieraz daitekeen zenbaki maximoa gainditzen denean).
- > Hainbat **tresna** —**softwarea** eta **hardwarea**— erabiltzen dira sistema digitalak diseinatzeko eta, azkenik, gailu osoa txip batean (adibidez, FPGA motako txip batean) integratzeko.



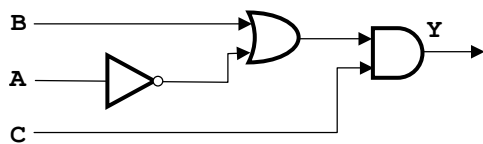
## GEHIAGO PROBATZEKO GOGOA?

Hona hemen ariketa batzuk, egin ditzazun gustua baduzu. Emaitzak ikusi nahi badituzu, jo fakultateko web orrira.

1. Praktika honetan analizatu dituzun oinarrizko hiru ate logikoen portaera ikusita, zein izango da bi zirkuitu hauen erantzuna, sarrera-balioen (1/0) konbinazio guztietarako?



A	B	Y
0	0	



A	B	C	Y
0	0	0	

- 2.** Zein da 32 bitekin kode daitekeen zenbaki natural handiena? Eta 64 bitekin?

Zenbat bit erabili behar dira 777 zenbaki naturala kodetzeko eta zein kode dagokio?

Zer zenbaki natural kodetzen da honela 16 bitekin: 1100110011001100?

- 3.** Bitarrez adierazita dauden batuketa hauek kontuan harturik, zeinetan datuak eta emaitzak 6 bitekoak baitira, zein da batura, bitarrez, kasu bakoitzean? Zein kasutan sortuko da bururakoa (*overflow*)?

**(a)**  $011001 + 110110$

**(b)**  $000001 + 100000$

**(c)**  $101100 + 010011$

## **OHARRAK**



