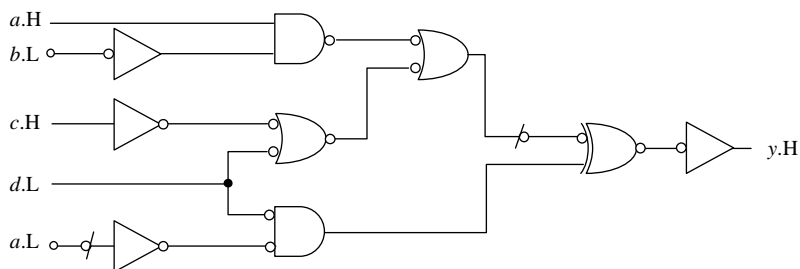


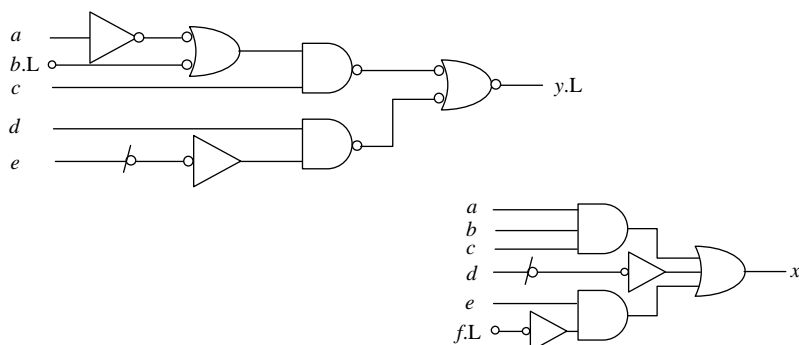
2 GAIA: ARIKETAK

Liburuan proposatutako ariketak

- 2.1.** Analiza ezazu honako zirkuitu hau, eta adierazi gauzatzen duen funtzio logikoa. Sarrerako balio logikoak $a = 0$, $b = 1$, $c = 0$ eta $d = 1$ izanik, adierazi, zirkuituan zehar, ateen irteeretan zein sarreretan sortzen diren balio logikoak eta fisikoak.



- 2.2.** Adierazi irudiko zirkuituek gauzatzen dituzten funtzio logikoak. Kasu bakoitzean, eman erabiltzen diren ate logikoen izenak, eta kalkulatu zirkuituaren erantzun-denbora maximoa (ateen erantzun-denbora = 2 ns). Eman erantzun-denbora maximoen adibide bana.



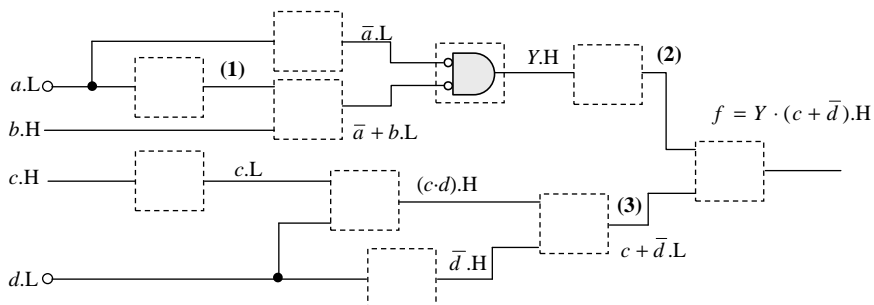
- 2.3.** Eraiki itzazu $f = \bar{a}b + a\bar{b}c + \bar{a}c\bar{d}$ eta $g = d(a + \bar{b}c) + b\bar{a}$ funtzioak:

- nahi dituzun atek erabiliz (kopuru minimoa)
- NAND atek erabiliz
- NOR atek erabiliz
- edozein ate erabiliz, baina zirkuituaren erantzun-denbora minimizatuz

Sarrerako aldagaiak logika negatiboan daude, eta funtzioak bi logiketan eman behar dira. Kalkulatu zirkuitu bakoitzaren erantzun-denbora, ateen erantzun-denbora 2 ns bada.

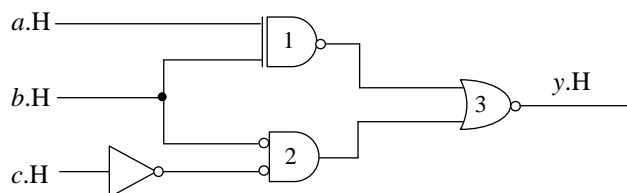
- 2.4.** $f = Y(c + \bar{d})$ funtzioa eraiki nahi dugu, soilik NOR atek erabiliz. Irudian, funtzioa eraikitzeko eskema dugu, tartean datu batzuk ematen direlarik. Bete itzazu falta diren ateen irudiak, adierazten den f funtzioa lortzeko.

Zirkuituaren tarteko emaitza gisa, Y funtzioa sortzen da; eman funtzio horren adierazpen sinpleena. Era berean, zer funtzio betetzen dira zirkuituko (1) eta (2) puntuetan? zuzena da (3) puntuan emandako adierazpena?



2.5. Eraiki ezazu $f = ((a \oplus b) + (a \otimes c)) \cdot d$ funtzioa XOR eta NAND ateak erabiliz. Aldagai guztiak eta funtzioa logika positiboan daude.

2.6. Zirkuituen logika menderatzen ez duen diseinatzaile batek irudiko zirkuitua proposatu du $y = a \oplus b + bc$ funtzio logikoa gauzatzeko:

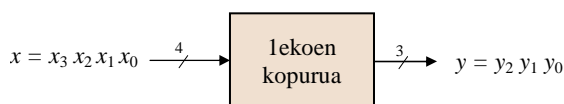


Erantzun honako galdera hauei:

- Egokia da zirkuitua? Zer funtzio exekutatzen du zirkuituak? Minimiza ezazu haren adierazpena.
- Zein osagai gehitu edo kendu behar zaio zirkuituari $y = a \oplus b + bc$ funtzioa bete dezan, 1, 2 eta 3 zenbakidun ateak ukitu gabe?

2.7. Egin itzazu lehen kapituluan proposatu diren 1.7., 1.8. eta 1.9. ariketetako funtzioei dagozkien zirkuituak. Erabil ezazu ahalik eta ate kopururik txikiena. Kalkulatu zirkuituen erantzun-denbora, ateen erantzun-denboraren funtzio gisa.

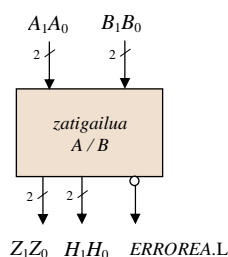
2.8. Zenbakien lekoen kopurua kalkulatzeko duen zirkuitua diseinatu behar da. 4 biteko zenbakiak ($x = x_3 x_2 x_1 x_0$) prozesatuko ditu sarrera gisa, eta, irteeran, sarrerako zenbakiaren lekoen kopurua adieraziko du, 3 bitetan ($y_2 y_1 y_0$). Esaterako, sarrerako zenbakia 1011 bada, lekoen kopurua 3 da; beraz, irteeran, bitarrez, 3 zenbakia lortu beharko da: $y = y_2 y_1 y_0 = 011$.



2.9. 4 biteko zenbakiak ($x = x_3 x_2 x_1 x_0$) prozesatzen ditu zirkuitu batek. Haren irteera aktibatzen da sarrerako zenbakian gutxienez bi leko jarraian baldin badaude. Esaterako, $x = 0101$ bada, $y = 0$ izango da; eta $x = 1110$ bada, $y = 1$.

Egin ezazu funtzio horren egia-taula, sortu adierazpen minimoa, eta eraiki zirkuitua (nahi dituzun ateak erabiliz).

2.10. $A(A_1, A_0)$ eta $B(B_1, B_0)$ bi biteko zenbaki arrunten zatiketa egiten duen zirkuitua diseinatu behar da. Irteera gisa, zatidura eta hondarra eskainiko ditu: $Z(Z_1, Z_0)$ eta $H(H_1, H_0)$.



“Zati 0” eragiketa egiten bada, *ERROREA* seinalea aktibatuko du. Kasu horretan, zatidura eta hondarra ez daude definituta.

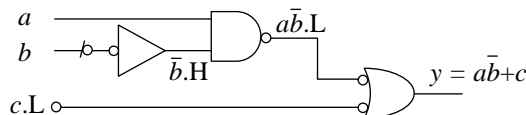
Egin itzazu funtzio horien egia-etaulak, sortu adierazpen minimoak, eta eraiki zirkuitu osoa (nahi dituzun ateak erabiliz).

Zein da zatigailuaren erantzun-denbora?

Liburuan ebatzitako ariketak

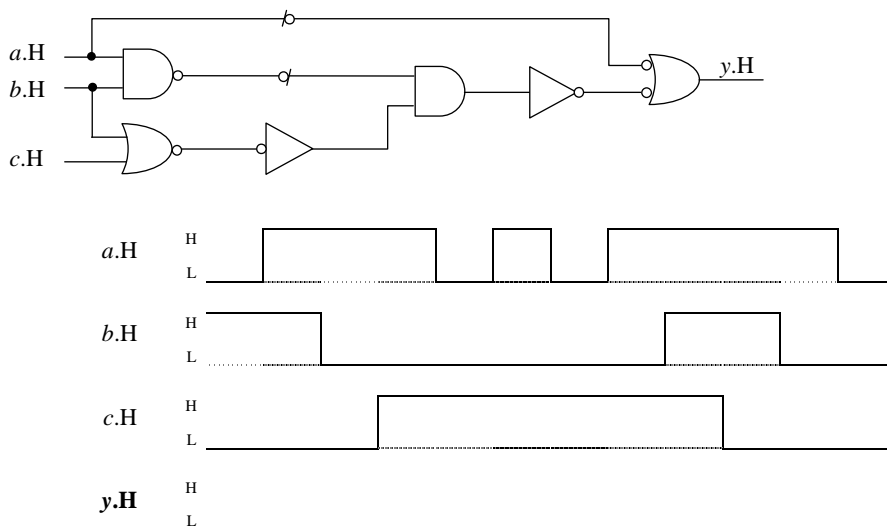
- 2.1.** Irudiko zirkuituak $y = a\bar{b} + c$ funtzioa gauzatzen du, bi NAND ate eta NOT ate bat erabiliz. a eta b sarrerak logika positiboan datoz, eta c sarrera logika negatiboan.

Hiru ateen erantzun-denbora berdina da: Δ . Gogoratu: zirkuitu fisiko guztiek bezala, ateen ere denbora (txikia bada ere) behar dute sarrerak prozesatzeko eta emaitza sortzeko.



Une jakin batean, $a = b = 1$ eta $c = 0$ sarrera-balioak prozesatzen ari dira, eta, ondorioz, $y = 0$ da. Egoera egonkorra dela eta, b aldagaiaren balioa aldatzen da: 1etik 0ra. Egin ezazu kronograma bat agerian uzteko noiz aldatuko den y irteeraren balioa 0tik 1era.

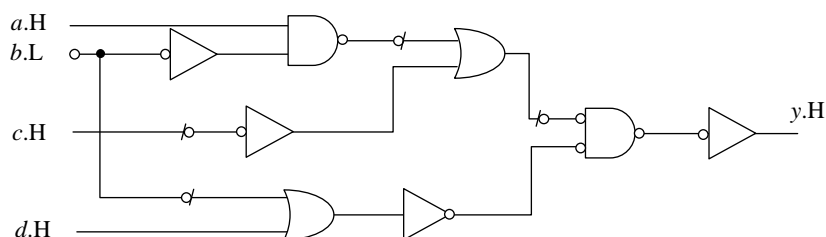
- 2.2.** (a) Analizatu irudiko zirkuitua, eta adierazi gauzatzen duen funtzio logikoa.
 (b) Une batean, sarrerako aldagaien balio logikoak $a = 1$, $b = 0$ eta $c = 1$ dira. Adierazi, zirkuituan zehar, ateen irteeretan eta sarreretan sortzen edo prozesatzen diren balio logiko zein fisikoak.
 (c) Azkenik, bete ezazu irudiko kronograma.



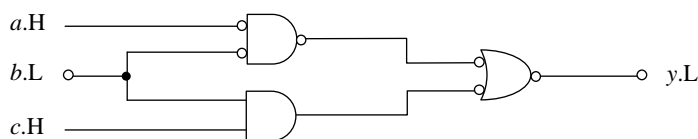
- 2.3.** $y = b\bar{a} + \bar{d}(a\bar{b} + c)$ funtzio logikoa egin behar da; lau sarrerak — a , b , c eta d — logika positiboan daude, eta irteera — y — bi logiketan sortu behar da. Eraiki ezazu funtzio hori:

- (a) Nahi dituzun atek erabiliz (NOT atek eta bi sarrerako AND, NAND, OR, eta NOR), baina ate kopurua minimoa izanik. Adieraz itzazu erabili dituzun ateen izenak.
 (b) Bakarrik NAND atek erabiliz.
 (c) Bakarrik NOR atek erabiliz.
 (d) Nahi dituzun atek erabiliz, baina zirkuituaren erantzun-denbora minimizatuz.
 Kalkulatu aurreko zirkuituen erantzun-denbora, ateen atzerapena Δ izanik.

- 2.4.** Adieraz ezazu irudiko zirkuituak egiten duen funtzioa, minimizatu lortutako adierazpena, eta egin funtzio minimoari dagokion zirkuitu berria.



- 2.5.** Zirkuituen logika menderatzen ez duen diseinatzaile batek irudiko zirkuitua proposatu du $y = b(a+c)$ funtzio logikoa gauzatzeko:



Erantzun honako galdera hauei:

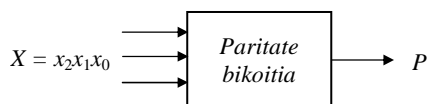
- (a) Egokia da zirkuitua? Zer funtzio exekutatzen du benetan zirkuituak? Minimiza ezazu funtzio horren adierazpena. Zirkuitua egokia ez bada, zuzendu ezazu, baina haren “egitura” aldatu gabe.
- (b) Eraiki $y = b(a + c)$ funtzioa, baina soilik NAND ateak erabiliz.
- 2.6.** Datuak gorde edo transmititu behar direnean, ohikoa da erroreak izatea, datuen hainbat bit alda edo gal daitezkeelako. Estrategia asko dago arazo horri aurre egiteko; hau da sinpleena: datu bakoitzari bit bat gehitzea, erroreak detektatu ahal izateko.

Bit hori **paritate-bita** deitzen da, eta kodeen lekoen kopurua kontrolatzeko erabiltzen da, bikoitia (paritate bikoitia) edo bakoitia (paritate bakoitia) izan dadin. Hala, detektatzen bada kode batean paritatearekin bat ez datorren lekoen kopurua, erroretzat hartuko da.

Har dezagun, adibidez, paritate bikoitiaren kasua eta hiru biteko kode hau: 010. Kode horri gehitu behar zaion paritate-bita 1 da, lekoen kopurua bikoitia izan dadin (bi). Beraz, paritate-bita gehituta (esaterako, ezkerrean), lau biteko kode hau izango dugu: **1-010**. Bit bat aldatzen bada kode berrian —esaterako: 1-010 \rightarrow 1-011—, erraz detektatuko da errorea, lekoen kopurua bakoitia (hiru) izango delako, bikoitia izan beharrean.

Erroreak detektatzeko metodo hori sinplea da, baina ez da oso eraginkorra; esaterako, ezin dira detektatu 2 biteko erroreak (oro har, errore kopuru bikoitia). Halaber, ezin da jakin zein bitetan gertatu den errorea; hots, ezin da jatorrizko kodea berreskuratu. Hala ere, asko erabiltzen da.

Ariketa honetan, 3 biteko zenbakien paritate-bita (bikoitia) sortzen duen zirkuitua diseinatu behar da. Zirkuituaren sarrera $X = x_2x_1x_0$ zenbakia da, eta irteera P , paritate-bita.



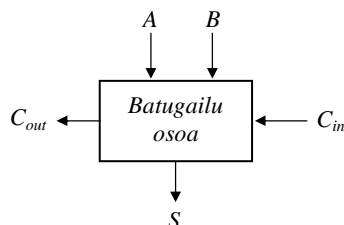
- 2.7.** Prozesadore baten osagai nagusietako bat unitate aritmetiko/logikoa da; eta, hor, batugailua dugu elementurik garrantzitsuenak: eragiketa aritmetiko gehienak batuketan bidez gauzatzen dira.

Ariketa honetan, bit bateko bi zenbakien batuketa egiten duen zirkuitua sortu behar da (3. kapituluaren ikusiko den legez, erraza da erabiltzea zirkuitu hori n biteko zenbakiak batzeko). Egin behar dugun zirkuituak 3 bit batu behar ditu: bi batugaiak gehi aurreko eragiketa batetik datorren bururakoa (carry). Emaitza gisa, 2 bit eman behar ditu: batura zein bururakoa. Zirkuitu horri batugailu osoa (full adder) deritzo.

Ikus dezagun batuketaren adibide bat:

Batuketa hamartarra	n biteko zenbakien batuketa bitarra	Bit bateko zenbakien batuketa bitarra
$\begin{array}{r} 1\ 0\ 1\ 0 \\ \uparrow\ \uparrow\ \uparrow\ \uparrow \\ \mathbf{1\ 3\ 4\ 7} \\ \mathbf{2\ 8\ 2\ 4} \\ \hline 4\ 1\ 7\ 1 \end{array}$	$\begin{array}{r} 1\ 1\ 1\ 0 \\ \uparrow\ \uparrow\ \uparrow\ \uparrow \\ \mathbf{0\ 1\ 0\ 1} \quad (5) \\ \mathbf{0\ 1\ 1\ 1} \quad (7) \\ \hline 1\ 1\ 0\ 0 \quad (12) \end{array}$	$\begin{array}{r} \downarrow 0 \quad (C_{in}) \\ \mathbf{1} \quad (A) \\ \downarrow \mathbf{1} \quad (B) \\ (C_{out})\ 1\ 0 \quad (S) \end{array}$

Beraz, zirkuituaren sarrerak honako hauek izango dira: bit bateko bi zenbaki (A eta B) eta sarrerako bururakoa (C_{in}). Eta irteerak, batura (S) eta irteerako bururakoa (C_{out}). Zirkuituaren egitura honako hau izango da:



Zirkuitua diseinatu ondoren, kalkula ezazu haren erantzun-denbora (bit bateko batuketa bat egiteko denbora), ate guztien atzerapena Δ izanik.

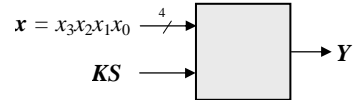
- 2.8.** Zirkuitu aritmetiko jakin batek bi funtzio egiten ditu, KS izeneko kontrol-seinalearen arabera. Zirkuituak 4 biteko BCD zenbakiak ($x = x_3x_2x_1x_0$, 0tik 9ra) prozesatzen ditu¹, eta honako emaitza hau sortzen du:

KS = 0

baldin ($x \geq 4$) orduan $Y := 1$
bestela $Y := 0$

KS = 1

baldin ($x = 3$ ren multiploa) orduan $Y := 1$
bestela $Y := 0$



Zirkuitu hori diseinatu behar da, nahi diren ateak erabiliz.

Gero, kalkulatu egindako zirkuituaren erantzun-denbora, ateen atzerapena Δ izanik.