

4. kapitula

BLOKE SEKUENTZIALAK

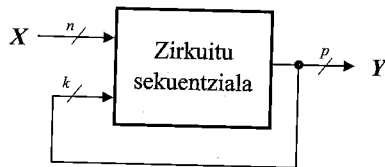
Aurreko kapituluan, bloke konbinazional behinenak aztertu ditugu: multiplexoreak, konparagailuak, deskodegailuak, batugailuak, unitate aritmetiko/logikoak, eta abar. Bloke horiek erabiliz, zirkuitu konbinazional konplexuagoak eraiki daitezke. Kapitulu honetan, beste zirkuitu mota bat aztertuko dugu: zirkuitu sekuentzialak. Lehen zatian, oinarriko osagaiak azalduko ditugu: JK eta D biegenkorrak; bigarrenen, erregistroak: erregistro soilak, desplazamendu-erregistroak eta kontagailuak.

4.1. ZIRKUITU SEKUENTZIALAK

Aurreko kapituluetan aztertu ditugun zirkuitu guztiak —ate logikoak, multiplexoreak, batugailuak...— zirkuitu konbinazionalak dira. Esan dugun bezala, zirkuitu konbinazional baten erantzuna soilik uneko sarreraren mende dago. Hau da, ez dira kontuan hartzen aurreko erantzunak: sistemak ez du memoriarik. Esaterako, batugaiak 2 eta 3 badira, batugailuaren emaitza beti 5 izango da.

Sistema digital gehienetan, ordea, erantzunak ez du izan behar bakarrik uneko sarreraren funtzioa: **sistemaren egoera** ere hartu behar da kontuan erantzun egokia emateko. Hala, sarrera jakin batzuetarako sistemak emango duen erantzuna ez da beti bera izango. Adibide sinple asko eman daitezke. Esaterako, demagun semaforo bat eraiki nahi dugula. Semaforoak kolore-sekuentzia jakin bati jarraitzen dio: berdea, anbarra, gorria, berdea... Beraz, aldaketa bat gertatzen denean, semaforoaren hurrengo kolorea uneko kolorearen arabera da. Gauza bera gertatzen da, adibidez, kontagailu batean (geroago aztertuko ditugu kontagailuak). Kontatzeko agindua hartzen duenean, 1 gehituko dio gordeta duen balioari; hau da, 3tik 4ra igaroko da, edo 6tik 7ra. Emaitza (irteera), ondorioz, desberdina da, kontagailuaren uneko balioaren arabera da, hain zuzen ere.

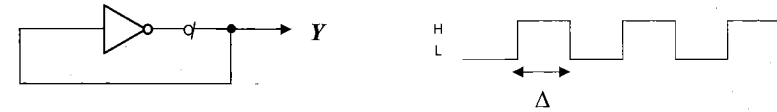
Zirkuitu sekuentzial baten erantzuna emateko, beraz, sarreraren balioez gain, sistemaren egoera ere prozesatu behar da. 4.1. irudian, zirkuitu sekuentzial baten egitura orokorra ageri da.



4.1. irudia. Zirkuitu sekuentzial baten eskema orokorra.

Irudian ageri denez, zirkuituaren irteera batzuk **berrelikatu egiten** dira sarreran. Izan ere, zirkuitu sekuentzialen memoria (aurreko balioak kontuan hartzeko ahalmena) berrelikaduraren ondorioa da.

Hainbat zirkuitu diseina daitezke irteerak sarreran berrelikatuz. Askok ez dute portaera berezirik erakusten. Batzuek, aldiz, bai. Esaterako, 4.2. irudian, adibide bat ageri da.



4.2. irudia. Osziladore baten oinarrizko eskema.

4.2. irudiko NOT atearen irteera sarreran konektatuta dago. Beraz, irteera 0 denean, sarreran 0koa izango dugu, eta, ondorioz, irteera 1 izatera igaroko da. Gero, sarreran 1 izango dugunez, irteera 0 izango da eta abar: zirkuituaren irteerak oszilatzen egiten du. Oszilazio horren periodoa atearen erantzun-denboraren arabera da, Δ . (Adibide soil bat da aurrekoa, ez baitira horrela egiten oszilatzen duten zirkuituak, funtsean antzeko egiturak erabiltzen badira ere.)

4.1.1. Zirkuitu sekuentzialen sailkapena

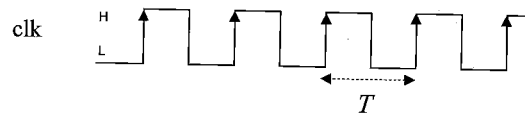
Oro har, bi motatako zirkuitu sekuentzialak ditugu:

- **Asinkronoak:** zirkuituak une oro prozesatzen ditu sarrera-seinaleak, dagokien irteera sortzeko.
- **Sinkronoak:** kontrol-seinale batek, **erlojuak**, adierazten dio zirkuituari noiz prozesatu behar diren sarrerak irteera berriak sortzeko. Bitartean, zirkuituaren erantzuna ez da aldatuko, sarreraren balioak aldatu arren.

Zirkuitu sekuentzial sinkronoak diseinatzea errazagoa da, haien funtzionamendua erloju-seinalearen bidez kontrola baitaiteke. Seinaleak ez dira prozesatzen edozein unetan, baizik eta erloju-seinaleak adierazten duenean, eskuarki seinaleen balioak egonkorak direnean; hala, sarreretan egon daitezkeen “zaratek” edo *glitch*-ek —seinaleen balio iragankorrek, zirkuituan zehar aldaketak gertatzen ari direlako— ez dute eraginik izango sistemaren erantzunean. Liburu honetan zirkuitu sekuentzial sinkronoak bakarrik aztertuko ditugu.

Aipatu dugun moduan, kontrol-seinale berezi bat erabiltzen da zirkuitu sekuentzial sinkronoetan: erlojua (ingelesez, *clock*, eta, laburtuz, *clk*). Izan ere, erloju-seinalea sistema sinkrono baten kontrol-seinale nagusia da, berak adierazten baitu noiz prozesatu behar diren sarrerak irteerak kalkulatzeko, eta, ondorioz, neurri batean behintzat, sistemaren “abiadura” zehazten du.

Erloju-seinalea 1 eta 0 balioen artean etenik gabe oszilatzen duen seinalea da, 4.3. irudikoa bezala.



4.3. irudia. Erloju-seinalea. Goranzko ertzak markatuta daude, gezi batez.

Oro har, erloju-seinalearen balio-aldaketek edo ertzek adierazten dute noiz prozesatu zirkuituaren sarrerak⁸. Bi aukera ditugu, beraz: erloju-seinalea 0tik 1era igarotzen den unea —L-tik H-ra, goranzko ertza— edo 1etik 0ra aldatzen denean —H-tik L-ra, beheranzko ertza—. Bi aukerak berdintsuak dira, eta, sinplifikatzeko asmoz, bakar bat erabiliko dugu: erloju-seinalearen **goranzko ertzean sinkronizatzen diren zirkuituak**.

Erloju-seinalea, definizioz, behin eta berriz errepikatzen den seinale periodiko bat da. Hori dela eta, seinale horren **periodoa** — T — eta **maiztasuna** — f — defini daitezke. Gogoratu: seinalea errepikatzeko behar den denbora-tartea adierazten du periodoak, segundotan; maiztasuna, periodoaren alderantzizkoa da, eta hertzetan (s^{-1}) neurtzen da.

Esaterako,

$$T = 1 \mu s \text{ (mikrosegundo, } 10^{-6} s) \rightarrow f = 1/T = 10^6 \text{ Hz} = 1 \text{ MHz (megahertz)}$$

$$T = 1 \text{ ns (nanosegundo, } 10^{-9} s) \rightarrow f = 1/T = 10^9 \text{ Hz} = 1 \text{ GHz (gigahertz)}$$

Azter ditzagun, dagoeneko, zirkuitu sekuentzial sinkronoen oinarritzko osagaiak.

4.2. BIEGONKORRAK

Biegonkorrak (ingelesez, *flip-flop*) oinarritzko osagaiak dira edozein zirkuitu sekuentzialetan. **Bit bateko memoria** duten gailuak dira biegonkorrak, eta, beraz, bi egoeretako batean egon daitezke: 1 edo 0 (hortik izena, bi egoera egonkor dituztelako).

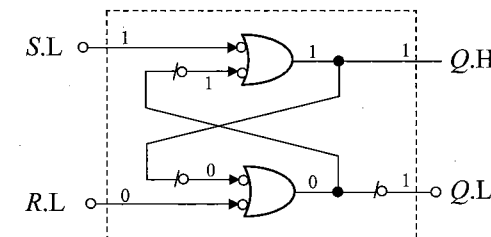
⁸ Zenbait zirkuitutan, une jakin bat erabili beharrean, erloju-seinalearen tarte bat erabiltzen da aldaketak onartzeko; esaterako, erloju-seinaleak 1 balio duen tarte (edo 0koa). Liburu honetan ez ditugu zirkuitu horiek erabiliko.

Gehiago badaude ere, bi dira *flip-flop* nagusiak: JK biegonkorra eta D biegonkorra. Biak zirkuitu sinkronoak dira; hau da, erloju-seinale batek kontrolatzen du noiz gertatuko diren 1 eta 0 egoeren arteko trantsizioak.

JK eta D biegonkorren funtzionamendua azaldu baino lehen, merezi du aztertzea nola lor daitekeen memoria-propietatea ate logiko pare batetik abiatuta. Izan ere, JK eta D biegonkor sinkronoen barruan, oinarritzko egitura bera dago: SR biegonkor asinkronoa. Ikus dezagun zertan den SR biegonkorra.

4.2.1. Bit bateko memoria duen oinarritzko osagaia: SR biegonkor asinkronoa

Kapitulu honen hasieran aipatu dugun bezala, zirkuitu sekuentzialen ezaugarri nagusia berrelikadura da. 4.4. irudian, bi NAND ateen bidez egindako zirkuitu bat ageri da, SR biegonkorra.



4.4. irudia. NAND atekin egindako SR biegonkorren eskema (antzeko egitura bat osa daiteke NOR atekin).

SR biegonkorrak bi sarrera — S , *set*, eta R , *reset*— eta bi irteera — $Q.H$ eta $Q.L$ — ditu. Izan ere, egiaztatuko dugun moduan, bi irteerek seinale bera adierazten dute murriztapen bat onartzen badugu, eta, horregatik, izen bera erabiliko dugu: Q . Irudian ageri denez, zirkuitua berrelikatu dugu, bi irteerak ateen sarreretara eramanez. Azter dezagun zirkuitu horren funtzionamendua.

S (*set*) sarrera aktibatzen denean — $S = 1$ eta $R = 0$ —, $Q.H$ irteerak 1 balioa hartzen du (*or* funtzioaren emaitza). $Q.H$ irteeraren balio berria dela eta, bigarren ate logikoan 0ko bat berrelikatzen da (ezeztatuta dago); ondorioz, bigarren atearen erantzuna 0 izango da, eta, beraz, $Q.L$ irteera ere 1 izango

da (4.4. irudian ageri den adibidea). Beraz, Q irteerak aktibatzen dira (1), haien aurreko balioa edozein izanda.

Bi irteerek balio egonkorra lortu arte, denbora-tarte bat igaro da: biegonkorren erantzun-denbora, hain zuzen ere.

R (reset) sarrera aktibatzen denean $\neg S = 0$ eta $R = 1$, kontrakoa gertatzen da. Beheko atearen irteera 1 izango da, eta, ondorioz, $Q.L = 0$ izango da. Era berean, lehenengo atearen irteera, $Q.H$, 0 izango da, bi sarrerak 0 direlako. Beraz, Q irteerak desaktibatzen dira (0).

Laburbilduz: S aktibatzen denean, zirkuituaren irteera ($Q.H$ eta $Q.L$) 1 izango da, eta R aktibatzen denean, 0.

Zer gertatzen da bi sarrerak 0 direnean? Kasu horretan, zirkuituaren erantzuna uneko irteeraren balioaren araberakoa da. Biegonkorren balioa $Q = 0$ bada, erraz egiaztatzen da bi irteeren hurrengo balioa $Q' = 0$ izango dela; era berean, uneko balioa $Q = 1$ bada, hurrengo irteera $Q' = 1$ izango da⁹.

Hain zuzen ere, hori bera da zirkuitu horren propietate garrantzitsua: aurreko egoeraren memoria gordetzen du. S seinalearen bidez 1 balioa hartu badu, leian geratuko da, S eta R seinaleak 0 diren bitartean. Modu berean, R seinalearen bidez 0 balioa hartu badu, 0ko hori mantenduko du. Beraz, **bit bateko memoria-zirkuitua** da.

4.1. taulan, zirkuituaren portaera ageri da, orain arte analizatu ditugun hiru kasuetarako:

S	R	Q'
0	0	Q
0	1	0
1	0	1

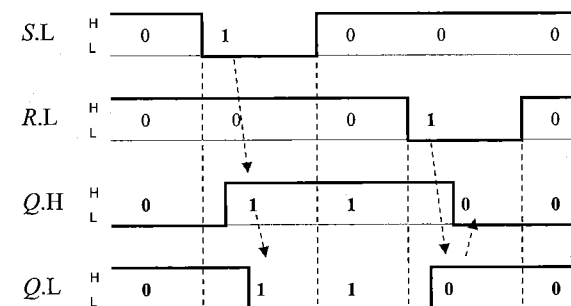
4.1. taula. SR biegonkorren egia-aula.

Laugarren kasu bat falta da analizatzeko, $S = R = 1$. Zoritxarrez, kasu horretan, zirkuituaren erantzuna ezin da interpretatu aurreko kasuetan egin dugun modu berean. Bi irteerek H tentsioa hartuko dute; beraz, $Q.H = 1$ eta

⁹ Q' izena erabiltzen da, askotan, biegonkorrek hartuko duen hurrengo balioa adierazteko.

$Q.L = 0$ izango dira. Ondorioz, ezin dira hartu seinale beratzat bi logiketan¹⁰. Hori dela eta, SR biegonkorra bit bateko memoria gisa erabil daiteke, baldin eta **inoiz ez badira aktibatzen batera bi sarrerak**; kasu hori debekatuta dago.

SR biegonkorren portaera 4.5. irudiko kronograman bildu dugu. Hasieran, biegonkorra 0 egoeran dago, hau da, $Q = 0$ da ($Q.H = L$ tentsioa eta $Q.L = H$ tentsioa). S aktibatzen denean, biegonkorrek 1 balioa hartzen du; $R = 1$ denean, aldiz, 0 balioa.



4.5. irudia. SR biegonkorri dagokion kronograma bat. NAND ateen erantzun-denbora hartu da kontuan, eta, horregatik, $Q.L$ irteera $Q.H$ irteera baino pixka bat geroago aktibatzen da (alderantziz Ora igarotzeko).

SR biegonkorra asinkronoa da: sarrerak aldatzean aldatuko dira irteerak ere. Horregatik, zirkuitu oso sinplea bada ere, ez da erabiltzen, eta horren ordez, JK eta D biegonkor sinkronoak erabiltzen dira.

JK zein D biegonkorrek SR biegonkorren eboluzioa dira. Funtsean, egitura bera dute, baina haien portaera sinkronizatu da erloju-seinale baten bidez (ez dugu ikusiko haien barne-egitura). Aldaketak, beraz, une jakin batzuetan baino ez dira gertatzen: erloju-ertzetan. Erloju-ertzaren aurreko egoera adierazteko, Q edo Q' erabili ohi da, eta erloju-ertza pasa eta gerokoa, Q' edo Q_{t+1} .

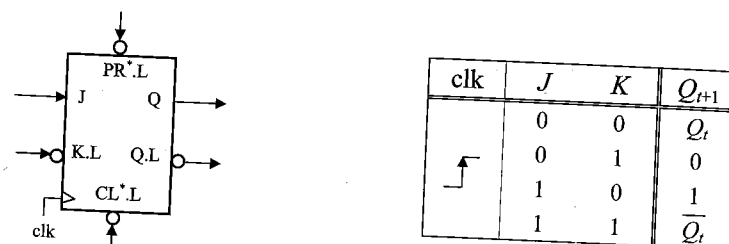
¹⁰ Irakurleak modu sinplean egiaztatu dezake aurrekoa, egia-aula baten bidez. 4.4. irudiko bi irteeren ekuazio logikoak honako hauek dira:

$$(\text{goikoa}) \quad Q.H = S + \bar{R} \cdot Q \quad (\text{behekoa}) \quad Q.L = \bar{R} \cdot (S + Q)$$

Bi funtzioen egia-aulak berdinak dira, kasu batean salbu: $S = R = 1$ denean. Gainerako kasuetarako, interpreta daiteke bi irteerak, $Q.L$ eta $Q.H$, aldagai bakar bat adierazten dutela.

4.2.2. JK biegonkorra

Bit bateko memoria duen biegonkor sinkronoa da JK biegonkorra. 4.6. irudian, ohiko JK biegonkor baten eskema logikoa eta egia-taula ageri dira¹¹.



4.6. irudia. JK biegonkorra eta haren oinarrizko egia-taula.

Biegonkorra sinkronoa denez, erloju-seinaleak (irudian, clk) adieraziko du noiz prozesatu behar diren bi sarrera sinkronoak, J eta K , zirkuituaren erantzuna (irteera) kalkulatzeko. Egia-taulan, Q_t izenak biegonkorren uneko balioa adierazten du, eta Q_{t+1} izenak biegonkorren balio berria, erloju-ertza iristean hartuko duena.

Egia-taulan ageri den moduan, J eta K sarrerak 0 diren bitartean, biegonkorren balioa Q_t mantentzen da. J aktibatuta eta K desaktibatuta badaude erloju-ertza gertatzen denean, biegonkorra 1 egoerara igaroko da; K aktibatuta eta J desaktibatuta badaude, berriz, 0ra igaroko da. Azkenik, bi sarrerak aktibatzen direnean, biegonkorren balioa ezeztatuko da, hau da, Q 0tik 1era edo 1etik 0ra igaroko da.

Zirkuitu sinkronoa bada ere, hau da, haren aldaketak erloju-seinaleak kontrolatzen baditu ere, ohikoa da seinale asinkrono pare bat izatea biegonkorretan: *clear* (CL^*) eta *preset* (PR^*). Izartxoak adierazten du sarreraren portaera asinkronoa. Edozein unetan aktibatzen direla, CL^* seinaleak 0ko bat jartzen du biegonkorrean eta PR^* seinaleak 1eko bat. Debekatuta dago biak batera aktibatzea, biegonkorren portaera ez baitago guztiz zehaztuta kasu horretan (izan ere, CL^* eta PR^* seinaleak JK biegonkorren oinarrian dagoen SR biegonkorren R eta S seinaleak baino ez dira).

¹¹ Biegonkorren J eta K sarrerak edozein logikatan egon daitezke. Ohikoa da, irudian ageri den moduan, bietako bat .H izatea eta bestea .L, baina baita ere biak .H izatea. Eskuarki, irteera bi logiketan ematen da: $Q.H$ eta $Q.L$. Baldin badaude, PR^* eta CL^* sarrerak logika negatiboan izan ohi dira.

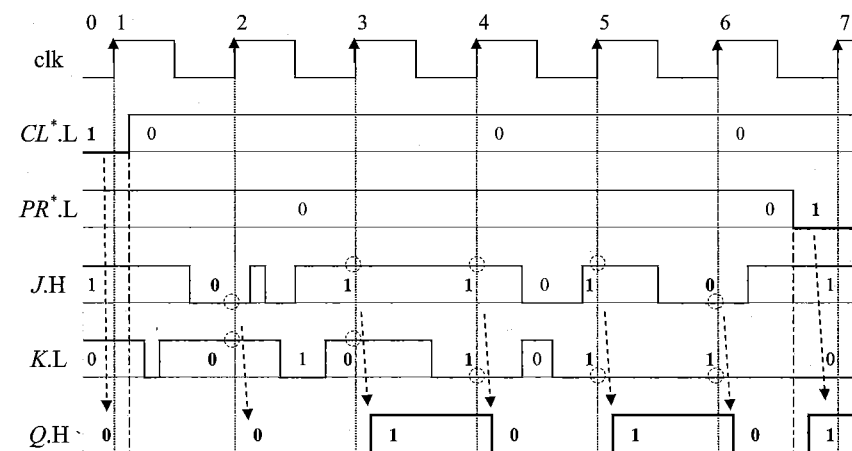
Oro har, bi seinale asinkrono horiek biegonkorra hasieratzeko erabiltzen dira bakarrik; hots, hasierako balio bat kargatzeko. Hortik aurrera, seinale sinkronoak erabiliko ditugu biegonkorra kontrolatzeko.

4.2. taulan, JK biegonkorren egia-taula osoa ageri da.

CL^*	PR^*	clk	J	K	Q_{t+1}
1	0	—	—	—	0
0	1	—	—	—	1
0	0	↑	0	0	Q_t
			0	1	0
			1	0	1
			1	1	$\overline{Q_t}$

4.2. taula. JK biegonkorren egia-taula osoa.

Azter dezagun, kronograma batean, JK biegonkorren funtzionamendua (4.7. irudia):



4.7. irudia. JK biegonkor baten funtzionamenduaren kronograma.

0/1 Hasieran, CL^* (*clear*) seinale asinkronoa aktibaturik dagoenez, J eta K seinaleak ez dute eraginik, eta biegonkorren irteera desaktibatuta egongo da: $Q = 0$. CL^* seinalea desaktibatzen da 1. erloju-ertzaren ondoren; hortik aurrera, eta CL^* edo PR^* seinaleak aktibatzen ez

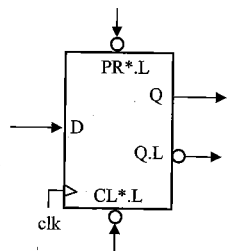
diren bitartean, biegonkorarren portaera sinkronoa izango da, hots, haren balio-aldaketak erloju-ertzekin batera gertatuko dira.

2. erloju-ertza gertatzen denean, $J = K = 0$ dira: beraz, biegonkorra egoera berean mantenduko da: 0. Erloju-ertz hori baino lehen, aldaketa bat (agian, *glitch* bat) egon da K seinalean; berdin da, biegonkorrak ez ditu sarrerak prozesatzeko ertza gertatu arte.
3. erloju-ertzean, $J = 1$ eta $K = 0$ dira. Beraz, biegonkorrak 1 balioa hartuko du. Berritoki ere, 2. eta 3. erloju-ertzen artean J eta K seinaleetan izandako aldaketek ez dute eraginik izan, ez baitira prozesatu.
- 4/5 Erloju-ertz horietan, $J = K = 1$ dira; beraz, biegonkorarren balioa aldatzen da: 1etik 0ra eta 0tik 1era, hurrenez hurren.
6. Une horretan, $J = 0$ eta $K = 1$ dira: ondorioz, biegonkorrak 0 balioa hartuko du.
7. Azkenik, 7. erloju-ertza baino lehen, PR^* (*preset*) seinale asinkronoa aktibatzen da. Beraz, une horretan bertan, erloju-seinalea kontuan hartu gabe, biegonkorrak 1 balioa hartuko du.

4.2.3. D biegonkorra

Esan bezala, bit bateko memoriak dira biegonkorrak, eta memoria horien gauzatze bat baino gehiago dago. Dagoeneko JK biegonkorra aztertu dugu, baina erabiliena, seguru asko, D biegonkorra da.

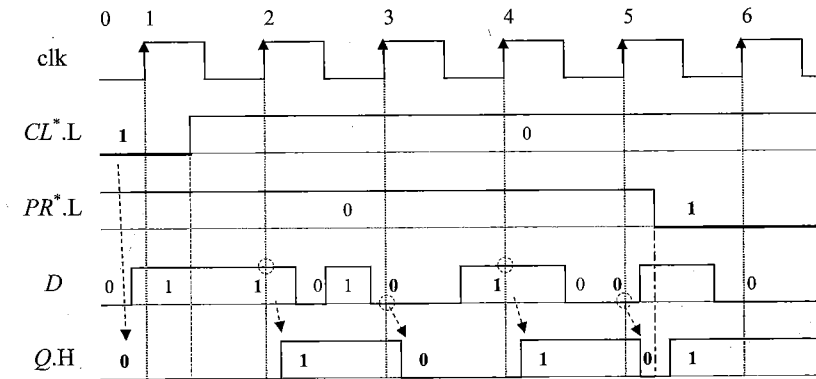
Egitura antzekoa da, baina datu-sarrerara sinkrono bakarra du: D. Horrez gain, ohiko bi sarrerara asinkronoak ere baditu: *clear* eta *preset*. 4.8. irudian, D biegonkorarren ohiko eskema logikoa eta egia-taula ageri dira.



CL^*	PR^*	clk	D	Q_{t+1}
1	0	—	—	0
0	1	—	—	1
0	0	0	0	0
		1	1	1

4.8. irudia. D biegonkor baten eskema eta egia-taula.

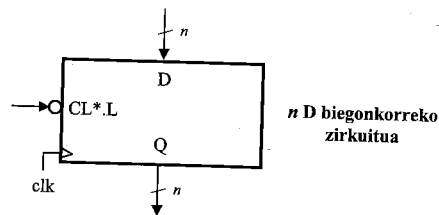
Egia-taulan ageri den moduan, D biegonkorarren portaera oso sinplea da. Erloju-ertzetan, D sarrerako balioa irteerara igarotzen da. Ertzetik ertzera, balioa mantentzen du. Ikus dezagun portaera hori kronograma batean, 4.9. irudian.



4.9. irudia. D biegonkor baten kronograma.

- 0/1 Hasieran, CL^* aktibatuta dagoenez, biegonkorarren irteera 0 izango da, beste seinaleen balioak direnak direla. 1. erloju-ertzean, beraz, ez da D sarrera prozesatzen.
2. 2. erloju-ertza baino lehen, CL^* seinalea desaktibatzen da. Beraz, biegonkorrak portaera sinkronoa izango du hortik aurrera. Erloju-ertzean $D = 1$ denez, biegonkorrak 1 balioa hartuko du.
3. 2. eta 3. erloju-ertzen artean, D sarreraren balioa aldatzen da behin eta berriz, baina aldaketa horiek ez dute eraginik biegonkorarren balioan. 3. erloju-ertzean, berriz, $D = 0$ da; beraz, biegonkorarren irteera $Q = 0$ izango da, eta hala mantenduko da ziklo osoan zehar.
- 4/5 4. erloju-ertzean, D sarrera 1 da; beraz, irteera $Q = 1$ izango da hortik aurrera. Era berean, 5. erloju-ertzean $D = 0$ da, eta biegonkorra 0ra joango da.
6. 6. erloju-zikloa baino lehen, PR^* seinale asinkronoa aktibatuta da; derrigorrean, biegonkorrak 1 balioa hartuko du, erloju-seinaleari kasu egin gabe; hori mantenduko da PR^* seinalea aktibatuta dagoen bitartean.

Ohikoa da erabili behar izatea funtzio bererako D biegonkor bat baino gehiago. Kasu horietarako, n D biegonkor dituen zirkuitu bat erabil daiteke, eskuarki biegonkor guztietarako CL seinale komun bat erabiltzen duena:



4.2.4. JK eta D biegonkorren erabilera

Bi biegonkor erabilienak aztertu ditugu: JK eta D. Izan ere, nahikoa litzateke biegonkor bakar batekin, biek funtzio bera betetzen baitute: bit bateko memoria gordetzea. Hala ere, ohikoa da biak erabiltzea sistema digitaletan, erabilera berezitua dute eta.

JK biegonkorrak erabiltzen dira, batik bat, adierazleak sortzeko; hau da, "pizteko" eta "itxaltzeko" kontrolatu behar den bit bateko informazioa. Adibidez, eragiketa batean gainezkatzeko gertatu den ala ez adierazteko; erabiltzaileari "abisuak" emateko, eta abar. J eta K sarrerak kontrol-seinale gisa erabili ohi dira, adierazlea (biegonkorra) aktibatzeke (J) eta desaktibatzeke (K). 6. kapituluan, adibide batzuk landuko ditugu.

D biegonkorrak, aldiz, informazioa gordetzeko erabiltzen dira batik bat. Izan ere, hurrengo atalean ikusiko dugun bezala, n D biegonkorrekin (eta multiplexoreekin) n biteko erregistro bat eraiki daiteke. Erregistro batean n biteko datu bat gorde daiteke (esaterako, 64 biteko datu bat), eta, hainbat erregistro erabiliz, erregistro-multzo bat osa daiteke, eta, azkenik, memoria bat ere. Erregistro-multzoak eta memoriak 5. kapituluan aztertuko ditugu. Bestalde, 6. kapituluan ikusiko dugun moduan, D biegonkorrak erabiltzen dira kanpo-seinaleak sinkronizatzeko ere; hots, haien aldaketak sistemako erloju-seinalearekin bat etortzeko.

Horrez gain, sistema digitalen kontrol-unitateen egoera gordetzeko erabiltzen dira D biegonkorrak (ikus 6. kapituluua). Sistema digital baten kontrol-unitatea automata bat da, hainbat egoeraren artean eboluzionatzen duena sarrera-seinale batzuen arabera; hau da, aurreikusitako sekuentzia bati jarraitzen dio. Sekuentziadore horiek nahiko modu sinplean eraiki daitezke D biegonkorren bidez. Kapitulu honetako ariketetan eta, batik bat, 6. kapituluan, sekuentziadoreak eta kontrol-unitateen diseinua landuko ditugu.

4.3. BLOKE SEKUENTZIALAK: ERREGISTROAK

Bit bateko memoria duten gailuak aztertu ditugu aurreko atalean, biegonkorrak. Jakina, bit bateko baino memoria gehiagoko gailuak behar dira, oro har, sistema digital guztietan. Gailu horiek erregistroak eta memoriak dira.

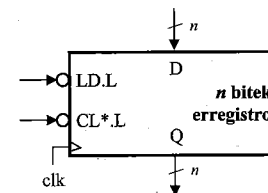
Erregistroa n biteko datu bat gordetzeko gauza den gailua da, eta eskuarki, D biegonkorren bidez eraikitzen da. Erregistro mota asko dago; atal honetan hiru hauek aztertuko ditugu:

- **Erregistro** soila. n biteko datu bat gordetzen du.
- **Desplazamendu-erregistroa**. n biteko datu bat gordetzeaz gain, edukiarekin eragiketa jakin bat exekuta dezake: biten desplazamendua, ezkerreko zein eskuinera.
- **Kontagailua**. n biteko datu bat gordetzeaz gain, eragiketa aritmetiko sinpleak exekuta ditzake: edukiari 1 gehitu ($K + 1$) edo 1 kendu ($K - 1$).

Kasu guztietan, erregistro sinkronoak analizatuko ditugu.

4.3.1. Erregistroak (registers)

n biteko erregistro bat gauza da n biteko datu bat kargatzeko eta gordetzeko, behar diren ziklo guztietan, beste datu bat kargatu nahi den arte. 4.10. irudian, erregistro orokor baten eskema logikoa eta funtzionamenduaren egia-aula ageri dira.



CL*	clk	LD	Q_{t+1}
1	—	—	0
0	↑	0	Q_t
0	↓	1	D

edukia ezabatu
edukia mantendu
sarrerako datua kargatu

4.10. irudia. n biteko erregistro baten ikurra eta egia-aula.

Aukera asko badago ere, ohikoak dira sarrera eta irteera hauek erregistroetan:

▪ Datu-sarrera

D: n biteko datu-sarrera. Hor kokatu behar da erregistroan kargatu nahi den informazioa.

▪ Datu-irteera

Q: n biteko datu-irteera, erregistroaren edukia hain zuzen ere.

▪ Kontrol-seinaleak

LD: Kontrol-seinale nagusia, datu bat kargatu behar dela adierazteko (*load*, kargatu).

CL*: Ohiko hasieratze-seinalea (*clear*), erregistroaren edukia ezabatzeko (0ko bat sartzeko), eskuarki seinale asinkronoa.

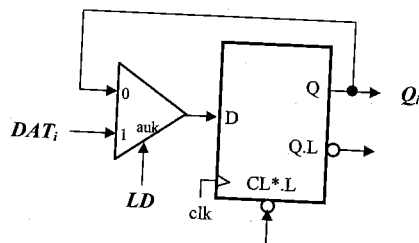
Erregistroaren funtzionamendua 4.10. irudiko egia-taulan ageri da: datuak kargatzen dira, erloju-seinaleak adierazten duenean, baldin eta LD seinalea aktibatuta badago (eta *clear* desaktibatuta); bestela, erregistroaren edukia mantentzen da.

Erregistroak egiteko, D biegonkorak erabiltzen dira. Baina D biegonkor soilak arazo bat dauka erregistroarena egiteko. Izan ere, ziklo bakar batean mantentzen du informazioa, hurrengo zikloan beste datu bat, sarreran dagoena, kargatuko baitu (ikus 4.8. irudiko egia-taula).

Nola mantendu biegonkorren informazioa denboran zehar? Soluzioa ez da zaila: nahikoa da D sarrerara eramatea biegonkorren edukia, Q irteera, berriro kargatzeko. Beraz, bi aukera izango ditugu D sarreran: biegonkorren balioa edo kargatu nahi dugun berria. Hori gauzatzeko, multiplexore bat erabili beharko dugu, bata edo bestea aukeratu ahal izateko.

4.11. irudian, erregistro baten i bitaren eskema logikoa ageri da.

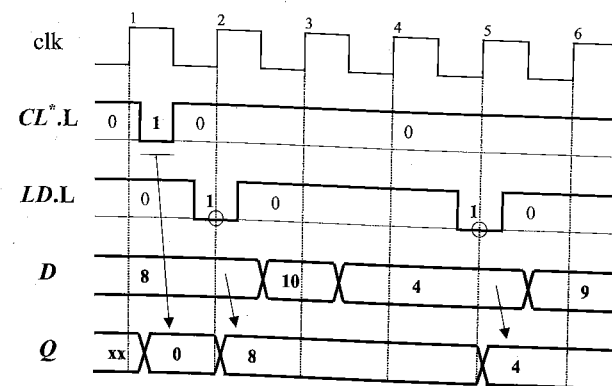
Erloju-ertz bakoitzean, bit bat kargatzen da D biegonkorrean: LD = 0 bada, jadanik kargatuta zegoena, hori baita multiplexorean aukeratu dena; bestela, LD = 1 bada, sarrerako datua kargatuko da. Laburbilduz: datu bat kargatu edo zegoena mantendu.



4.11. irudia. Erregistroen egitura (i bita).

Hala, 4.11. irudiko egitura n aldiz errepikatuz, n biteko erregistroa osatzen da, non LD eta CL* seinaleak komunak baitira bit guztietarako.

4.12. irudian, erregistro baten portaera deskribatzen da, kronograma baten bidez. CL* seinalea aktibatzen denean, erregistroak 0 balioa hartzen du. Gero, 2. erloju-ertzean, LD seinalea aktibatuta dagoenez, sarrerako datua kargatzen da erregistroan, 8ko bat. Balio hori mantentzen da erregistroan, 5. erloju-ertza arte. Une horretan LD seinalea berriro aktibatuta dagoenez, beste balio bat kargatuko da, 4ko bat hain zuzen ere.

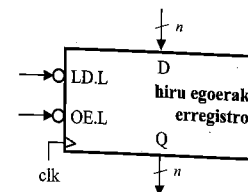


4.12. irudia. n biteko erregistro baten funtzionamenduaren kronograma. n biteko datuak adierazi behar direnean kronogrametan, ohikoa da denak biltzea lerro bakar batean, D eta Q datuekin egin dugun moduan, eta hartzen duten balioa bertan adieraztea (xx = balio ezezaguna).

4.3.1.1. Hiru egoerako erregistroak

Aurreko kapituluan azaldu dugun moduan (ikus 3.2.1. atala), badaude, bi egoera —0 eta 1— izan beharrean, hiru egoera —0, 1 eta Z— dituzten gailuak. Hirugarren egoerak irteera deskonektatuta dagoela adierazten du. Hori dela eta, hainbat irteera puntu komun batera —bus batera— konekta daitezke, baldin eta gehienez irteera bakar bat aktibatuta badago.

Erregistro arruntez gain, badaude hiru egoerako erregistroak ere, 4.13. irudikoa bezalakoak.



4.13. irudia. n biteko hiru egoerako erregistro baten ikurra.

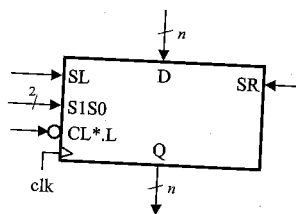
Gailu horiek, ohiko kontrol-seinaleez gain (LD , CL ...), badute beste bat: OE (*output enable*, irteera aktibatuta). OE sarrera aktibatuta dagoenean, erregistroaren edukia irteeran ageri da. Desaktibatuta dagoenean, berriz, irteera Z egoeran egongo da: deskonektatuta.

OE kontrol-seinaleak erregistroaren irteeraren egoera baino ez du kontrolatzen. Beraz, irteera desaktibatuta izanda ere, erregistroaren funtzionamendua ohikoa izango da: datu bat kargatu LD seinalea aktibatuta badago erloju-ertzetan, edukia ezabatu eta 0 bat jarri CL seinalea aktibatzen denean, eta abar.

4.3.2. Desplazamendu-erregistroak (*shift registers*)

Desplazamendu-erregistroak, izenak dioen moduan, erregistroak dira, baina, datuak gordetzeaz gain, eragiketa logiko bat egin dezakete gordeta dituzten datuekin: bit bateko desplazamendua, ezkerrera zein eskuinera.

Desplazamendu-erregistro asko dago, onartzen dituzten sarreraren eta eragiketen arabera: esaterako, noranzko bakarreko desplazamenduak onartzen dituztenak, datuak seriez (bitez bit) bakarrik kargatzen dituztenak, eta abar. Guk desplazamendu-erregistro orokorrean aztertuko dugu, 4.14. irudikoa.



4.14. irudia. Desplazamendu-erregistro orokor baten eskema logikoa.

Hauek dira ohiko sarrerak eta irteerak desplazamendu-erregistro batean:

• Datu-sarrerak

D: n biteko datu-sarrera, desplazamendu-erregistroan, kargatu nahi den n biteko datua.

SL: Bit bateko datu-sarrera (*left*, “ezkerrekoa”). Erregistroko pisu handieneko posizioan kargatzen den bita, edukia bit bat eskuinera desplazatzen denean.

SR: Bit bateko datu-sarrera (*right*, “eskuinekoa”). Erregistroko pisu txikieneko posizioan kargatzen den bita, edukia bit bat ezkerrera desplazatzen denean.

• Datu-irteera

Q: n biteko datua, erregistroaren edukia hain zuzen ere.

• Kontrol-seinaleak

S1S0: Bi biteko kontrol-seinalea, eragiketa adierazteko. Datua mantentzeaz gain, datu bat karga daiteke, edo edukia desplazatu bit bat ezkerrera edo eskuinera. Guztira, beraz, lau eragiketa.

CL*: Hasieratze-seinalea (*clear*), erregistroaren edukia ezabatze (0ko bat sartzeko), eskuarki seinale asinkronoa.

Lau eragiketa egin daitezkeenez, 2 kontrol-bit behar dira — $S1$ eta $S0$ —, eragiketa adierazteko. Aukera asko dago, baina, testu honetan, honela kodetuko ditugu lau eragiketak:

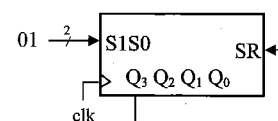
clk	S1	S0	Q ($Q_{n-1} \dots Q_0$)	
	0	0	$Q_{n-1} \dots Q_0$	Edukia mantendu
↑	0	1	$Q_{n-2} \dots Q_0$ SR	Edukia bit bat desplazatu ezkerrera (\leftarrow); eskuinean, SR sarrerako bita kargatu
	1	0	SL $Q_{n-1} \dots Q_1$	Edukia bit bat desplazatu eskuinera (\rightarrow); ezkerrean, SL sarrerako bita kargatu
	1	1	$D_{n-1} \dots D_0$	n biteko datu bat kargatu

Adi! “Ezkerra” edo “eskuina” hitzak anbiguoak izan daitezke testuinguru honetan. Honela ulertu behar dira okerrak saihesteko: ezkerrera, pisu handiagoko bitetarantz; eskuinera, pisu txikiagoko bitetarantz.

Ikus dezagun adibide bat. 4 biteko desplazamendu-erregistro baten edukia 1110 bada, honela geratuko da bit bateko desplazamendua egin ondoren:

hasieran:	1110	
S1S0 = 01 (\leftarrow , ezkerrera):	110-0	SR = 0 izanik
	110-1	SR = 1 izanik
S1S0 = 10 (\rightarrow , eskuinera):	0-111	SL = 0 izanik
	1-111	SL = 1 izanik

Bidenabar, desplazamendu-erregistroen bidez, edukiaren bitez biteko biraketak edo errotazioak egin daitezke. Nahikoa da, horretarako, desplazamendu bat egiten denean, berrelikatzea SL edo SR sarreretan Q_0 edo Q_{n-1} bita, hurrenez hurren, biraketaren noranzkoaren arabera. Adibidez,



edukia hasieran	1100
desplazatu bit bat ezkerrera:	100-1
desplazatu bit bat ezkerrera:	001-1
desplazatu bit bat ezkerrera:	011-0
desplazatu bit bat ezkerrera:	110-0

Aipatu dugun moduan, asko erabiltzen dira kontagailuak sistema digitaletan. Adibidez, begiztetan eman beharreko iterazio kopurua kontrolatzeko, ziklo kopuru jakin bat kontatzeko, denbora-tarte jakin bat itxaroteko, eta abar. Oro har, sor daitezkeen sekuentziak oso erabilgarriak dira kontrol-unitateak eraikitzeko.

Laburpena. Kapitulu honetan zirkuitu sekuentzial sinkrono nagusiak aztertu ditugu. Zirkuitu sinkronoek memoria dute, hau da, haien irteera ez da bakarrik uneko sarreren funtzioa, sistemaren aurreko erantzunak ere kontuan hartzen baitira. Hori lortzeko, zirkuituen irteerak berrelikatzen dira sarreran.

Bit bateko memoria duten zirkuituak aztertu ditugu: JK eta D biegonkorrak. Oinarritzko gailu sinkronoak dira biegonkorrak, eta, horien bidez, asko erabiltzen diren bloke sekuentzial nagusiak sortzen dira: erregistroak.

Erregistro erabilienak hauek dira: erregistro soilak, desplazamendu-erregistroak, eta kontagailuak. Hirurak gauza dira n biteko hitz bat gordetzeko, baina, horrez gain, desplazamendu-erregistroekin edukiaren bit bateko desplazamenduak egin daitezke (ezkerrera zein eskuinera) eta kontagailuekin kontatze-sekuentziak egin daitezke (gorantz edo beherantz): 6, 7, 8, 9... 12, 11, 10... Erregistroak osatzeko, D biegonkorrak eta, kasuan kasu, multiplexoreak, batugailuak eta abar erabili behar dira.

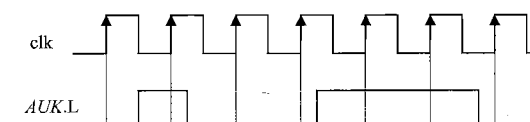
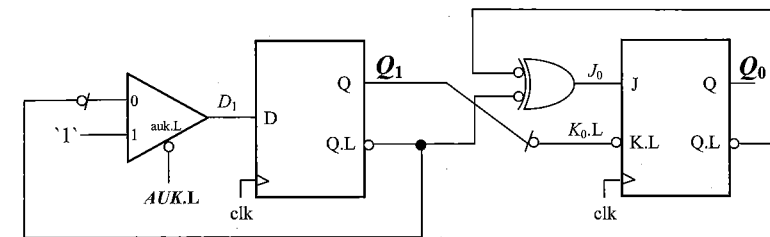
Hurrengo atalean, zirkuitu sekuentzialen portaera analizatuko dugu hainbat ariketaren bidez.

4.4. ARIKETA EBATZIAK

Kapitulua bukatzeko, ariketa batzuk ebatziko ditugu. Lehenengo bi ariketetan, zirkuitu sekuentzial sinpleenen, biegonkorren, funtzionamendua landuko dugu. Horretarako, biegonkorrez eta ateez osatutako zirkuituen kronogramak egingo ditugu. 3., 4. eta 5. ariketetan, biegonkorretan oinarritutako sekuentziadoreen eraikuntza aztertuko dugu: kanpo-seinaleen mende dauden sekuentziak zein kanpo-seinalerik gabekoak. Beste zirkuitu sekuentzial batzuen funtzionamendua aztertuko dugu 6. eta 7. ariketetan. 6.ean, desplazamendu-erregistro baten funtzionamendua aztertuko dugu, eta 7.ean, berriz, kontagailu batena. Oro har, zirkuitu konbinazionalak eta sekuentzialak (sinkronoak) batera ageri dira sistema digitaletan. Horregatik, garrantzitsua da ezagutzea elkarlanean nola diharduten. Hori da, hain zuzen ere, 8. eta 9. ariketetan landuko duguna.

>> 4.1. Ariketa

Hartu kontuan irudiko zirkuitua eta adieraz ezazu biegonkorren D eta J, K sarrerei dagozkien funtzio logikoak. Gero, marraztu ezazu, tentsio-diagrama edo kronograma batean, sistemaren portaera: D_1 eta Q_1 , eta J_0 , K_0 eta Q_0 seinaleen aldaketak denboran zehar. AUK seinalearen portaera irudikoa da, eta Q_1 eta Q_0 biegonkorren hasiera-balioa 0 da.



Oro har, biegonkorrek eta multiplexoreek irudian ageri diren kontrol-seinaleak baino seinale gehiago dituzte. Zirkuitu bati dagokion seinaleren bat agertzen ez bada eskema batean, funtzionamendu egokirako behar duen balioa izango du seinale horrek. Hala gertatzen da ariketa honetan; multiplexorearen ohiko gaikuntza-seinalea aktibatuta ($G = 1$) egongo da, eta biegonkorren hasieratze-seinale asinkronoak desaktibatuta ($CL^* = PR^* = 0$).

Kronograman bi biegonkor aztertu behar direnez, gogora dezagun haien funtzionamendua. Bit bat gordetzen dute erloju-ertza aktibatzen ez den bitartean. Erloju-ertza heltzean, aldiz, biegonkorren sarrerak prozesatzen dira irteera-balioa kalkulatzeko:

clk	D	Q_{t+1}
↗	0	0
↘	1	1

clk	J	K	Q_{t+1}
↗	0	0	Q_t
↘	0	1	0
	1	0	1
	1	1	$\overline{Q_t}$

Gogoratu, biegonkor batek bit bateko "aldagai" bat gordetzen du, eta, eskuarki, bi logiketan ematen du aldagaia, Q.H eta Q.L irteeretan.

Kronograma bete ahal izateko, biegonkorren sarrerak zehazten dituzten funtzioak kalkulatu behar ditugu. Hauek dira adibide honetako biegonkorren sarrera-seinaleen ekuazioak (ikus zirkuituaren irudia):

- D biegonkorren sarrera multiplexorearen irteera da; beraz:

$$\text{balidin } (AUK = 0) \text{ orduan } D_1 = \overline{Q_1} \\ \text{bestela } D_1 = 1$$

- JK biegonkorren sarrerak:

$$J_0 = Q_0 \oplus Q_1 \quad \text{eta} \quad K_0 = \overline{Q_1}$$

Kronograma urratsez urrats betetzen hasteko prest gaude. Kasu honetan, ezinezkoa da seinale baten eboluzioa ezagutzea besteena kontuan hartu gabe eta, beraz, kronograma betetzeko prozesua bertikalean, erloju-ertzez erloju-ertz, egin beharko dugu.

Aurreko kapituluetan esan dugun moduan, edozein zirkuitu fisikok erantzun-denbora (edo atzerapen) jakin bat behar du emaitza emateko, hau da, erantzuna ez da berehalakoa. Ondorioz, zirkuitu konbinazionalen irteerak ez dira aldatuko sarrerak aldatu bezain laster, eta zirkuitu sekuentzialen irteerak ez dira erloju-ertzarekin batera aldatuko, pixka bat geroago baizik.

Kronograma sinplifikatzeko, zirkuitu sekuentzialen atzerapenak soilik irudikatuko ditugu. Has gaitezen kronograma egiten, urratsez urrats.

■ Hasierako balioak

Q_1 eta Q_0 seinaleen hasiera-balioak 0 dira, eta ez dira lehenengo erloju-ertzearaino aldatuko. Seinale horietan oinarrituz, D_1 , J_0 eta K_0 sarreraren balioak kalkula ditzakegu.

Multiplexorearen irteerak emango digu D_1 -en balioa; AUK seinalea 1 denez, 1 sarrerako datua emango du irteeran multiplexoreak: lekoa, alegia. J_0 -ren balioa ezagutzeko, $Q_0 \oplus Q_1$ kalkulatu behar dugu: $0 \oplus 0 = 0$.

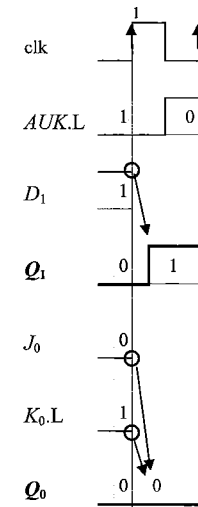
Eta azkenik, K_0 , Q_1 -en osagarria, 1 izango da.

■ 1. erloju-ertza

Erloju-ertza gertatutakoan, biegonkorren irteerak (Q_1 eta Q_0) aldatu egingo dira, sarreretan dauden balioen eta funtzionamendu-taulen arabera:

$$D_1 = 1 \quad \Rightarrow \quad Q_1 = 1 \\ J_0 = 0, K_0 = 1 \quad \Rightarrow \quad Q_0 = 0$$

Balio horiek hurrengo erloju-ertza iritsi arte mantenduko dira.



■ 1. erloju-zikloa

Q_1 , Q_0 , eta AUK seinaleen arabera, D_1 , J_0 eta K_0 seinaleen balioak kalkula ditzakegu.

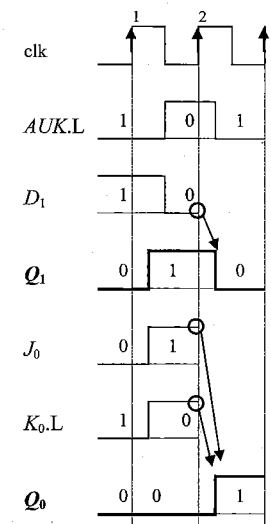
1. erloju-zikloan, AUK seinalearen balioa aldatu egiten da, eta, ondorioz, D_1 ere aldatuko da. Zikloaren hasieran, $AUK = 1$ da eta, beraz, multiplexorearen irteera 1 da; gero, $AUK = 0$ izatera pasatuko da eta multiplexorearen 0 sarrera aukeratu da irteeran; hau da, Q_1 -en osagarria, 0koa, hain zuzen ere.

J_0 -ren balioa $Q_0 \oplus Q_1 = 0 \oplus 1 = 1$ izango da, eta, azkenik, $K_0 = 0$ izango da, Q_1 -en osagarria.

■ 2. erloju-ertza

Erloju-ertza iristean, biegonkorren irteerak (Q_1 eta Q_0) aldatu egingo dira, sarreretan dauden balioen arabera.

$$D_1 = 0 \quad \Rightarrow \quad Q_1 = 0 \\ J_0 = 1, K_0 = 0 \quad \Rightarrow \quad Q_0 = 1$$



■ 2. erloju-zikloa

Berriz ere, D_1 , J_0 eta K_0 seinaleak dituzten balioak kalkulatu behar ditugu.

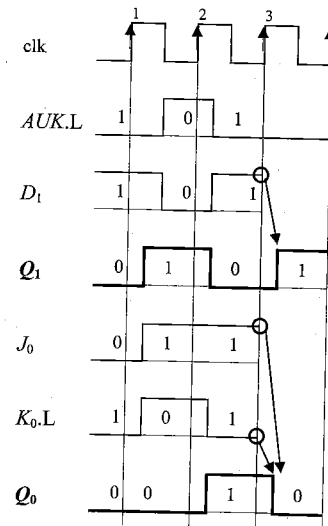
AUK seinalea 1 denez, D_1 ere 1 izango da. J_0 -ren balioa $Q_0 \oplus Q_1 = 1 \oplus 0 = 1$ izango da, eta, azkenik, K_0 , Q_1 -en osagarria, 1 izango da.

■ 3. erloju-ertza

Erloju-ertza gertatzean, biegonkorren irteerak (Q_1 eta Q_0) aldatu egingo dira, sarreretan dauden balioen arabera.

$$D_1 = 1 \Rightarrow Q_1 = 1$$

$$J_0 = 1, K_0 = 1 \Rightarrow Q_0 = 0$$



■ 3. erloju-zikloa

AUK seinaleak 1eko balioa mantentzen du 3. erloju-zikloan zehar, eta beraz, $D_1 = 1$ izango da.

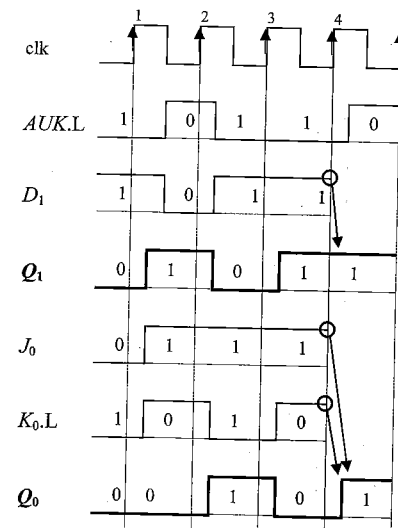
Era berean, $J_0 = Q_0 \oplus Q_1 = 0 \oplus 1 = 1$ izango da, eta, azkenik, K_0 , Q_1 -en osagarria, 0 izango da.

■ 4. erloju-ertza

Erloju-ertzarekin batera, biegonkorren irteerak (Q_1 eta Q_0) aldatu egingo dira, sarreretan dauden balioen arabera.

$$D_1 = 1 \Rightarrow Q_1 = 1$$

$$J_0 = 1, K_0 = 0 \Rightarrow Q_0 = 1$$



■ 4. erloju-zikloa

AUK seinaleak 0ko balioa hartzen du ziklo honetan; beraz, D_1 -ek (Q_1 -en osagarria) 0koa hartuko du.

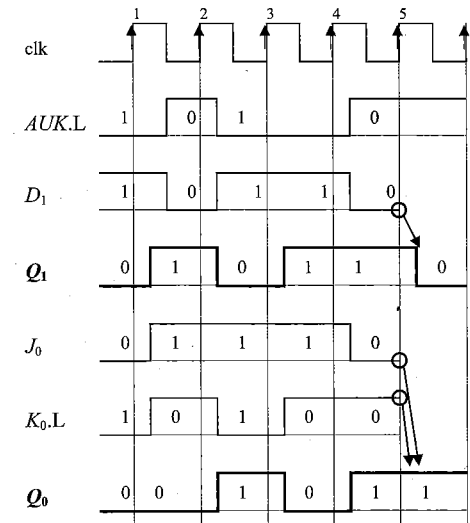
$J_0 = Q_0 \oplus Q_1 = 1 \oplus 1 = 0$ da. Azkenik, K_0 -k, Q_1 -en osagarria denez, 0ko balioa hartuko du.

■ 5. erloju-ertza

Erloju-ertzarekin batera, biegonkorren irteerak (Q_1 eta Q_0) aldatu egingo dira, sarreretan dauden balioen arabera.

$$D_1 = 0 \Rightarrow Q_1 = 0$$

$$J_0 = 0, K_0 = 0 \Rightarrow Q_0 = 1$$



■ 5. erloju-zikloa

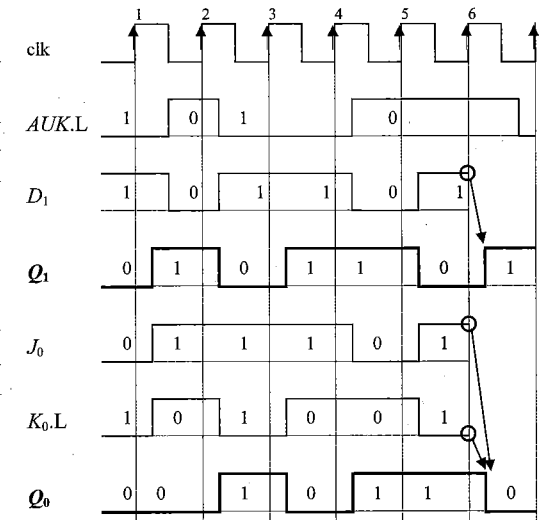
AUK seinaleak 0ko balioa hartzen du ziklo honetan ere eta, beraz, $D_1 = 1$ izango da. J_0 -ren balioa, $Q_0 \oplus Q_1 = 1 \oplus 0 = 1$ da eta, azkenik, K_0 ere, Q_1 -en osagarria, 1 izango da.

■ 6. erloju-ertza

Erloju-ertzarekin batera, biegonkorren irteerak (Q_1 eta Q_0) aldatu egingo dira, sarreretan dauden balioen arabera.

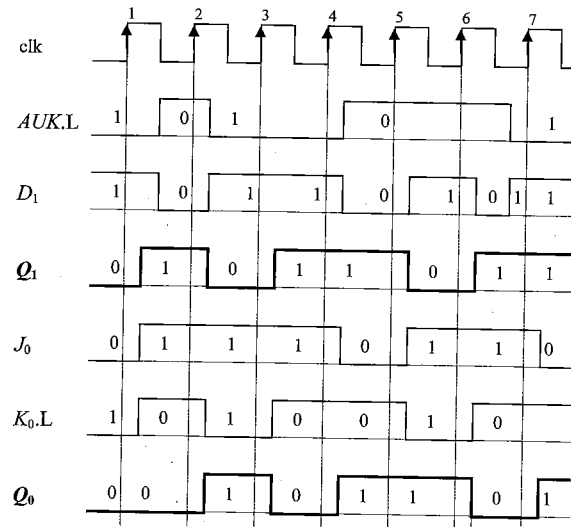
$$D_1 = 1 \Rightarrow Q_1 = 1$$

$$J_0 = 1, K_0 = 1 \Rightarrow Q_0 = 0$$



Azken zikloa besterik ez zaigu geratzen kronograma amaitzeko. Gainerakoetan egin dugun moduan, D_1 , J_0 eta K_0 -ren balioak kalkulatu eta

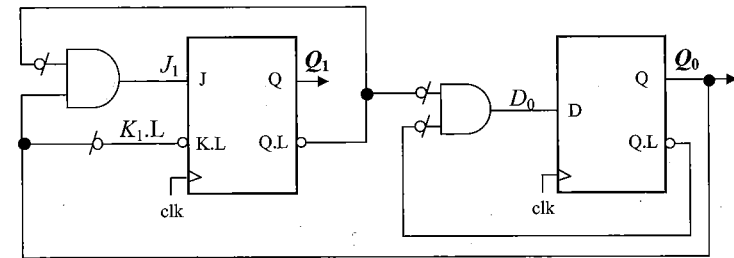
horien arabera, ezagutu ahal izango dugu erloju-ertza eta gero Q_1 -ek eta Q_0 -k edukiko duten balioa. Hona hemen kronograma osoa:



» 4.2. Ariketa

Irudiko zirkuitua bi biteko sekuentziadorea da; biegonkorren balioek sekuentzia jakin bati jarraitzen diote. Q_1 -ek eta Q_0 -k sekuentziadorearen egoera adierazten dute, sekuentziaren balio jakin bat, alegia.

Adieraz itzazu biegonkorren J , K eta D sarrerari dagozkien funtzio logikoak. Gero, bete ezazu egoeren arteko trantsizio-taula eta esan zein segidari jarraitzen dion zirkuituak. Azkenik, osa ezazu sistemaren funtzionamendua islatzen duen kronograma (hasieran, $Q_1Q_0 = 00$).



Zirkuituaren irudian ageri denez, bi biegonkorren sarrerak — J_1 , K_1 eta D_0 — sistemaren egoeraren funtzioak dira; hots, Q_1 -en eta Q_0 -ren funtzioak.

Zirkuitua analizatu behar dugu biegonkorren sarreraren ekuazioak eskuratzeko. Analisi horren arabera, hauek dira JK biegonkorren bi sarrerak, zirkuituaren eskeman ageri diren moduan:

$$J_1 = \overline{Q_1} \cdot Q_0$$

$$K_1 = \overline{Q_0}$$

Era berean, hau da D biegonkorren sarreraren ekuazio logikoa:

$$D_0 = \overline{Q_1} \cdot \overline{Q_0}$$

Funtzio horiek erabili behar ditugu sistemaren portaera taula batean (egoeren arteko trantsizio-taulan) bildu ahal izateko. Egin dezagun, bada, taula hori.

Egoeren arteko trantsizio-taula betetzeko, Q_1 -en eta Q_0 -ren balio posible guztiak aztertu behar dira. Balio horiek sistemaren egoera —uneko egoera (UE)— adierazten dute. Kasu bakoitzerako, J_1 , K_1 eta D_0 sarreraren balioak kalkulatu behar ditugu, eta horien arabera, zein izango den sistemaren hurrengo egoera (HE) erloju-ertza aktibatzean. Hurrengo egoera adierazteko, Q' letrak erabiliko ditugu.

	UE $Q_1 Q_0$	J_1 K_1 D_0	HE $Q_1' Q_0'$
E0	0 0		
E1	0 1		
E2	1 0		
E3	1 1		

E0 egoeran — $Q_1 Q_0 = 00$ —, J_1 , K_1 eta D_0 seinaleek honako balio hauek hartuko dituzte:

$$J_1 = \overline{Q_1} \cdot Q_0 = \overline{0} \cdot 0 = 1 \cdot 0 = 0$$

$$K_1 = \overline{Q_0} = \overline{0} = 1$$

$$D_0 = \overline{Q_1} \cdot \overline{Q_0} = \overline{0} \cdot \overline{0} = 1 \cdot 1 = 1$$

Erloju-ertza gertatzen den unean, sarrerak prozesatuko dira. Beraz, $J_1 = 0$ eta $K_1 = 1$ direnez, JK biegonkorren balio berria $Q_1' = 0$ izango da. Era berean, $D_0 = 1$ denez erloju-ertza aktibatzen denean, D biegonkorren irteera, erloju-ertza eta gero, $Q_0' = 1$ izango da.

Analiza dezagun orain E1 egoera — $Q_1 Q_0 = 01$ —. Kasu honetan, hauek izango dira biegonkorren sarrera-seinaleen balioak:

$$J_1 = \overline{Q_1} \cdot Q_0 = \overline{0} \cdot 1 = 1$$

$$K_1 = \overline{Q_0} = \overline{1} = 0$$

$$D_0 = \overline{Q_1} \cdot \overline{Q_0} = \overline{0} \cdot \overline{1} = 0$$

Erloju-ertzarekin batera, balio horiek prozesatuko dira biegonkorretan. Hala, $Q_1' = 1$ izango da, une horretan J_1 sarrera aktibatuta dagoelako, eta, era berean, $Q_0' = 0$ izango da, $D_0 = 0$ delako.

Prozedura bera erabili behar da taulako gainerako errenkadak betetzeko. Hona hemen taula osoa beteta.

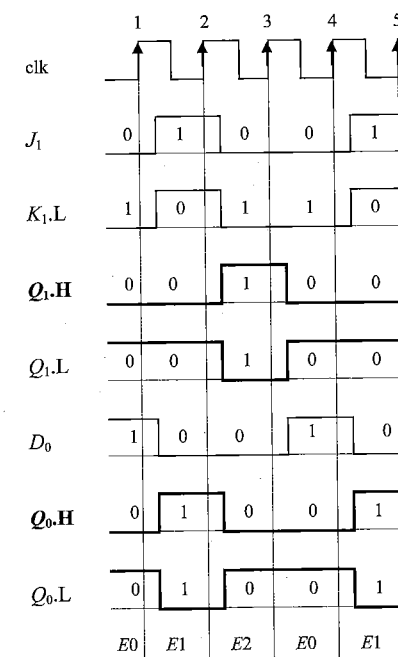
	UE $Q_1 Q_0$	J_1 K_1 D_0	HE $Q_1' Q_0'$
E0	0 0	0 1 1	0 1
E1	0 1	1 0 0	1 0
E2	1 0	0 1 0	0 0
E3	1 1	0 0 0	1 0

Taula horretatik ondoriozta daiteke zirkuituak sortzen duen segida. Hasieran, $Q_1 Q_0 = 00$ da; beraz, sekuentziadorearen hurrengo egoera, erloju-ertza eta gero, $Q_1 Q_0 = 01$ izango da. 01 (E1) egoeratik, 10 (E2) egoerara doa; eta E2 egoeratik, berriro $Q_1 Q_0 = 00$ egoerara doa, hau da, E0-ra. Une horretatik aurrera, sekuentzia osoa errepikatuko da, behin eta berriz.

Beraz, hau da sekuentziadorearen segida osoa, E0 egoeratik abiatuta: 00 – 01 – 10 – 00 – 01 – 10 – 00 – ...

Argi geratzen da sekuentziadorea ez dela berez 11 egoerara joango; hala ere, hasieratzen bada egoera horretan (esaterako, biegonkorren *Preset* hasieratze-seinaleak erabiliz), bere kabuz sartuko da sekuentzian: 11 – 10 – 00 – 01 – ...

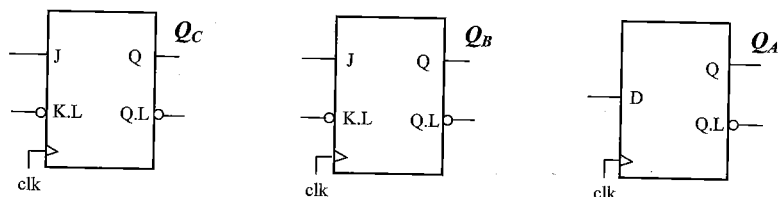
Sekuentziadorearen egoera-aula kronograma batean isla daiteke.



Erloju-ziklo bakoitza taulako errenkada batekin erlaziona dezakegu. Lehenengo zikloan, adibidez, $Q_1 Q_0 = 00$ dira, taulako E0 errenkadako egoera, hain zuzen ere. Beraz, $J_1 = 0$, $K_1 = 1$ eta $D_0 = 1$ izango ditugu, eta, ondorioz, hurrengo egoera $Q_1 Q_0 = 01$ izango da. Modu berean interpreta daitezke kronogramaren gainerako zikloak.

» 4.3. Ariketa

Erabil itzazu bi JK biegonkor eta D biegonkor bat 0-2-4-5-3-7-0-2-4-... sekuentziari jarraitzen dion zirkuitua egiteko.



Sortu behar den sekuentzia hiru bitekoa denez (4, 5 eta 7 egoerak adierazteko, 3 bit behar dira), 3 biegonkor erabili behar ditugu sekuentziadore eraikitzeke. Adibide honetan, bi JK biegonkor pisu handieneko bitetarako, eta D biegonkor bat pisu txikieneko biterako. Sekuentziadore eraikitzeke, biegonkorrek kontrolatzen dituzten sarreraren ekuazioak lortu behar ditugu; hau da, J_C , K_C , J_B , K_B eta D_A .

Sekuentziadorearen portaera taula batean bil daiteke, non adierazten den sekuentziaren bit bakoitza —biegonkor bakoitzaren irteera— nola aldatzen den erloju-ertz bakoitzean, egoera guztietarako. Esaterako, eraiki behar dugun sekuentzian, 0aren atzetik datorren balioa 2a da. Zer adierazten du horrek? Erloju-ziklo batean $Q_C Q_B Q_A$ hirukoteak 000 balioa hartzen badu, erloju-ertz eta gero 010 balioa hartu beharko duela, hain zuzen ere. Modu berean interpretatu behar ditugu sekuentziaren gainerako balioak.

Hona hemen sekuentziari dagokion taula osoa:

Uneko Egoera	Hurrengo Egoera
$Q_C Q_B Q_A$	$Q_C' Q_B' Q_A'$
0	2
1	-
2	4
3	7
4	5
5	3
6	-
7	0

Hiru bitekin adieraz daitezkeen 8 egoeretatik 2 ez dira erabiltzen adibide honetako sekuentzian: 1a eta 6a. Beraz, egoera horiek inoiz agertuko ez direnez, haien hurrengo egoerak zehaztu gabeko gaitzat har daitezke.

Aurreko taulan adierazitako trantsizioak sortzeko, biegonkorren sarrerak kontrolatu behar dira, eta hori egiteko, biegonkorren funtzionamendua hartu beharko dugu kontuan.

D biegonkorren kasuan, erraza da lan hori: D sarrerari jarri behar dugu irteeran lortu nahi dugun balioa; hala, erloju-ertz iristean, balio hori kargatuko da biegonkorrean.

$$Q' = 0 \rightarrow D = 0$$

$$Q' = 1 \rightarrow D = 1$$

JK biegonkorren kasua, berriz, ez da hain sinplea; biegonkorrek aurretik zuen balioak badu lotura erloju-ertz eta gero edukiko duenarekin. Garrantzitsua da JK biegonkorren taula kontuan edukitzea eragin hori nolakoa izango den jakiteko.

Ikus dezagun adibide bat: $Q = 0$ izanik $Q' = 0$ izatea nahi denekoa. Gainerako kasu guztietan gertatuko den moduan, trantsizio hori lortzeko bi aukera daude:

$J = 0, K = 0$ izatea: biegonkorrek aurretik zuen balioa mantenduko du.

$J = 0, K = 1$ izatea: biegonkorren irteeran 0koa ezarriko da.

Bi aukerak aztertuz, hau nabarmen dezakegu: $J = 0$ izan behar da eta K -k edozein balio har dezake, 0 edo 1 (zehaztu gabeko gaia, “-” ikurra taulan).

Modu berean azter daitezke gainerako kasuak. Honako taula honek laburtzen ditu aukera guztiak:

Q	Q'	J	K	J	K
0	0	0	0	0	-
0	1	1	1	1	-
1	0	1	0	-	1
1	1	0	0	-	0

Prest gaude biegonkorren sarrerak definitzeko, sekuentziaren egoera-aldaketa bakoitzarako: JK biegonkorren $J_C K_C$ eta $J_B K_B$ sarrerak, eta D biegonkorren D_A sarrera.

Taula hauetan ageri dira sarrera horien balioak egoera guztietarako:

UE	HE	$J_C K_C$		UE	HE	$J_B K_B$		UE	HE	D_A
Q_C	Q_C'			Q_B	Q_B'			Q_A	Q_A'	
0	0	0	-	0	1	1	-	0	0	0
0	-	-	-	0	-	-	-	1	-	-
0	1	1	-	1	0	-	1	0	0	0
0	1	1	-	1	1	-	0	1	1	1
1	1	-	0	0	0	0	-	0	1	1
1	0	-	1	0	1	1	-	1	1	1
1	-	-	-	1	-	-	-	0	-	-
1	0	-	1	1	0	-	1	1	0	0

Ohikoagoa da taula hori beste modu honetan ematea:

U. Egoera	H. Egoera	$J_C K_C$		$J_B K_B$		D_A
$Q_C Q_B Q_A$	$Q_C' Q_B' Q_A'$					
0 0 0	0 1 0	0	-	1	-	0
0 0 1	- - -	-	-	-	-	-
0 1 0	1 0 0	1	-	-	1	0
0 1 1	1 1 1	1	-	-	0	1
1 0 0	1 0 1	-	0	0	-	1
1 0 1	0 1 1	-	1	1	-	1
1 1 0	- - -	-	-	-	-	-
1 1 1	0 0 0	-	1	-	1	0

Azken pausoa, J , K eta D funtzio minimoak sortzea eta eraikitzea da. Funtzioak minimizatzeko, K-mapak erabiliko ditugu.

J_C

$Q_C Q_A$	00	01	11	10
0	0	-	1	1
1	-	-	-	-

$J_C = Q_B$

K_C

$Q_C Q_A$	00	01	11	10
0	-	-	-	-
1	0	1	1	-

$K_C = Q_A$

J_B

$Q_C Q_A$	00	01	11	10
0	1	-	-	-
1	0	1	-	-

$J_B = Q_A + \overline{Q_C}$

K_B

$Q_C Q_A$	00	01	11	10
0	-	-	0	1
1	-	-	1	-

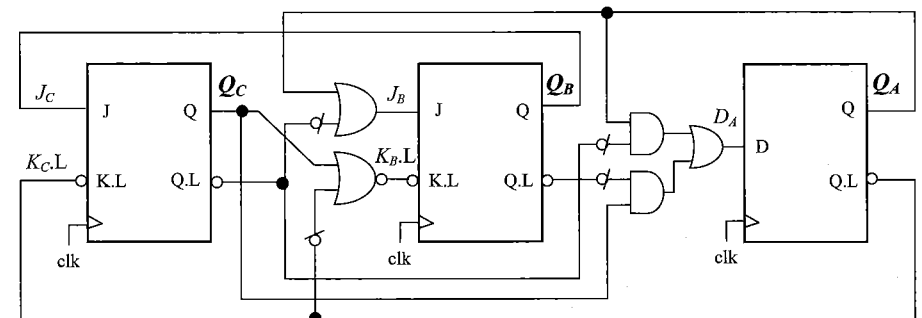
$K_B = \overline{Q_A} + Q_C$

D_A

$Q_C Q_A$	00	01	11	10
0	0	-	1	0
1	1	1	0	-

$D_A = \overline{Q_C} Q_A + Q_C \overline{Q_B}$

J_C , K_C , J_B , K_B eta D_A seinaleen espresio minimoak lortu ditugu. Ariketa amaitzeko, zirkuitua irudikatzea besterik ez zaigu falta. Hau izan daiteke aukera bat.



Azken ohar bat. Sekuentziadore honek ez ditu 1 eta 6 egoerak erabiltzen ohiko funtzionamenduan. Izan ere, zehaztu gabeko gaitzat hartu ditugu bi egoera horiek, eta J , K eta D funtzioak minimizatzeko erabili ditugu. Aztertzen baditugu hartutako erabakiak, hau aurkituko dugu: sekuentziadorea 1 edo 6 egoeretan sartzen bada (esaterako, seinale

aşinkronoren bat, *clear* edo *preset*, aktibatu delako), 3 edo 4 egoeretara joango da, hurrenez hurren.

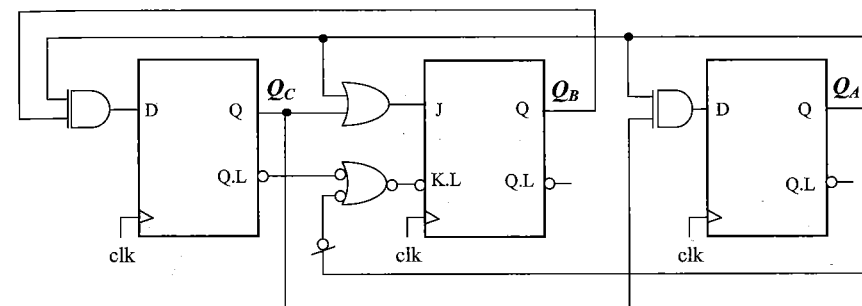
Izan ere, ezohiko funtzionamendua adierazten dute bi egoera horiek, akats bat. Segurtasuna dela eta, erabiltzen ez diren egoeretatik egoera jakin batera (hasierako egoerara, eskuarki 0 egoerara) eraman ohi dira sekuentziadoreak; hala, akats bat gertatzen bada sisteman, bere kabuz sartuko da sekuentzian, hasiera-puntuan.

Ariketa gisa, sekuentziadoreak berriz diseinatzea proposatzen dugu: espero ez diren egoeretatik (1 eta 6) 0 egoerara joan behar da.

«

» 4.4. Ariketa

Diseinatzaile bati $0 - 5 - 3 - 6 - 0 - 5 - \dots$ sekuentziari darraion sekuentziadoreak diseinatzea eskatu zaio, eta honako eskema hau proposatu du:



Egiazta ezazu proposaturiko sekuentziadoreak eskatutako segidari jarraitzen diola; horretarako, idatz itzazu aldagaien ekuazio logikoak eta bete ezazu egoeren arteko trantsizio-taula osoa.

UE							HE		
Q_C	Q_B	Q_A	D_C	J_B	K_B	D_A	Q_C'	Q_B'	Q_A'
0	0	0							
0	0	1							
0	1	0							
0	1	1							
1	0	0							
1	0	1							
1	1	0							
1	1	1							

Gero, egin ezazu zirkuituaren funtzionamendua islatzen duen kronograma. Azkenik, eraiki ezazu sekuentzia berari darraion beste sekuentziadore bat ahal den ate logiko kopururik txikiena erabiliz, hots, funtzioak minimizatuz (biegonkor berak erabiliz).

■

Lehenengo zeregina zirkuituaren analisia da; hots, seinale guztien ekuazioak zehaztea. Hauek dira irudiko zirkuituko seinale nagusien ekuazio logikoak:

$$D_C = Q_B \otimes Q_A$$

$$J_B = Q_C + Q_A$$

$$K_B = Q_C + \overline{Q_A}$$

$$D_A = Q_A \otimes Q_C$$

Funtzio horiek erabiliz, bete dezakegu biegonkorren trantsizio-taula: biegonkorren sarrera-seinaleen balioak eta sekuentziadorearen hurrengo egoera, uneko egoera guztietarako.

Adibidez, taulako lehenengo errenkadan, biegonkorren irteerak, $Q_C Q_B Q_A$, 0 dira; ondorioz, $D_C = 0 \otimes 0 = 1$, $J_B = 0 + 0 = 0$, $K_B = 0 + \overline{0} = 0 + 1 = 1$ eta $D_A = 0 \otimes 0 = 1$ dira. Balio horiek direla eta, erloju-ertza gertatzen denean, $Q_C' = 1$, $Q_B' = 0$ eta $Q_A' = 1$ izango dira.

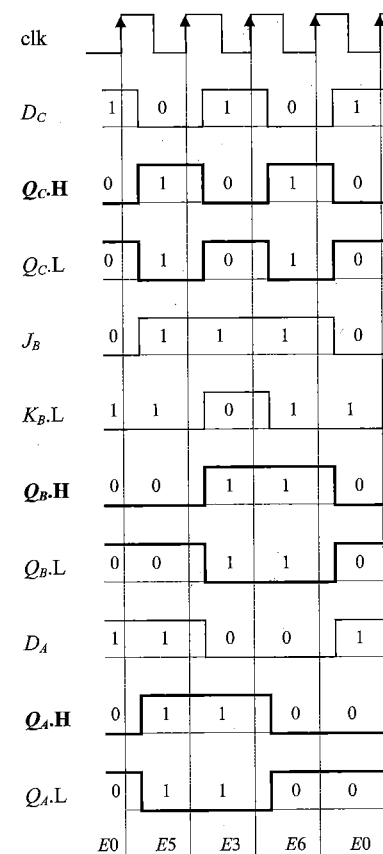
Prozesu berari jarraitu behar zaio taulako gainerako errenkadak betetzeko.

	UE			D_C	J_B	K_B	D_A	HE		
	Q_C	Q_B	Q_A					Q_C'	Q_B'	Q_A'
E0	0	0	0	1	0	1	1	1	0	1
E1	0	0	1	0	1	0	0	0	1	0
E2	0	1	0	0	0	1	1	0	0	1
E3	0	1	1	1	1	0	0	1	1	0
E4	1	0	0	1	1	1	0	1	1	0
E5	1	0	1	0	1	1	1	0	1	1
E6	1	1	0	0	1	1	0	0	0	0
E7	1	1	1	1	1	1	1	1	0	1

Taula hori erabiliz, egiaztatu ahal izango dugu ea zirkuituak jarraitutako sekuentzia diseinatzaileak lortu nahi zuena den ala ez; bertan topa baitezakegu, urratsez urrats, zein den zenbaki baten atzetik datorrena. Hala, lehenengo errenkada aztertuz, 0aren hurrengoa 5koa dela ondoriozta dezakegu; gero, 5aren atzetik, 3a dator; 3aren ondoren, 6a; eta, ondoren, berriro 0a.

Beraz, sekuentzia 0 - 5 - 3 - 6 - 0 - ... dela egiaztatu dugu, diseinatzaileak lortu nahi zuena, hain zuzen ere.

Hurrengo lana kronograma betetzea da. Aurretik betetako taula lagungarri izango dugu zeregin horretan. Izan ere, egoeren arteko trantsizio-taula baino ez da kronograma, modu grafikoan adierazita. Hau da sekuentziadoreari dagokion kronograma:



Hasieran, Q_C , Q_B eta Q_A 0 dira; ondorioz, $D_C = 1$, $J_B = 0$, $K_B = 1$ eta $D_A = 1$. Balio horiek ezagutuz, erloju-ertza gertatutakoan biegonkorren irteeretan edukiko ditugun balioak kalkula ditzakegu. $D_C = 1$ izanik, hurrengo zikloan $Q_C = 1$ izango da; $J_B = 0$ eta $K_B = 1$ direnez, erloju-ertza eta gero, $Q_B = 0$ izango da, eta D_A erloju-ertzaren aurretik 1 denez, erloju-ertza gertatu eta gero, $Q_A = 1$ izango da. Informazio hori guztia egoeren arteko trantsizio-taulako lehenengo lerroan ageri da, eta gauza bera gertatuko da gainerako zikloekin; alegia, taulako lerro jakin batekin bat etorriko direla. Beraz,

kalkulu guztiak errepikatu orde, zuzenean taulako informazioa erabil dezakegu kronograma betetzeko.

Azkenik, sekuentzia bera egiten duen zirkuitu minimoa eraiki behar dugu. Horretarako, JK eta D biegonkorren sarreraren funtzio logiko minimoak sortu behar ditugu.

Abiapuntu gisa, sekuentziadorearen egoeren arteko trantsizio guztiak taula batean bilduko ditugu. 3 biteko aukera guztiak ageri ez direnez, batzuk zehaztu gabeko gai moduan erabil daitezke:

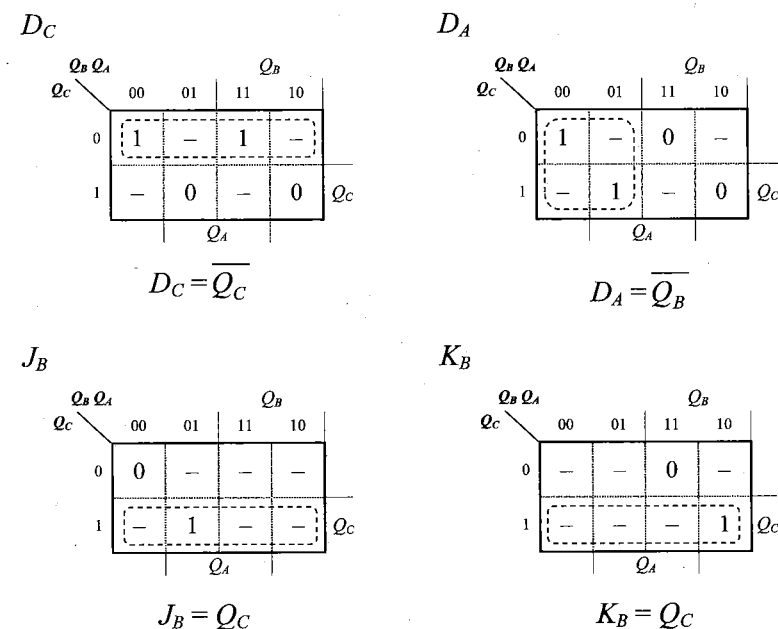
Uneko Egoera $Q_C Q_B Q_A$	Hurrengo Egoera $Q_C' Q_B' Q_A'$
0 0 0	5 1 0 1
0 0 1	- - -
0 1 0	- - -
0 1 1	6 1 1 0
1 0 0	- - -
1 0 1	3 0 1 1
1 1 0	0 0 0 0
1 1 1	- - -

Taula hori oinarri gisa erabiliz, biegonkorren sarreretako balioak erabakiko ditugu. Gogoratu: D biegonkorren kasuan, irteeran nahi den balioa jarri behar da sarreraren; JK biegonkorren kasuan, berriz, biegonkorren balioa hartu behar da kontuan J eta K sarreraren balio egokiak erabakitzeke.

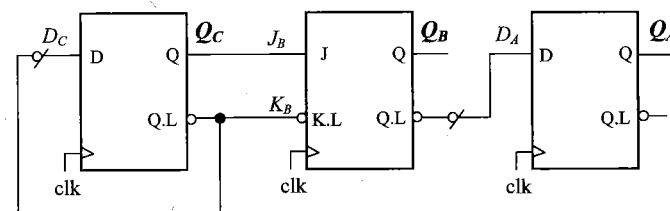
Hona hemen biegonkorren sarreraren balioak kasu bakoitzean:

	U. Egoera $Q_C Q_B Q_A$	H. Egoera $Q_C' Q_B' Q_A'$	D_C	J_B	K_B	D_A
E0	0 0 0	1 0 1	1	0	-	1
E1	0 0 1	- - -	-	-	-	-
E2	0 1 0	- - -	-	-	-	-
E3	0 1 1	1 1 0	1	-	0	0
E4	1 0 0	- - -	-	-	-	-
E5	1 0 1	0 1 1	0	1	-	1
E6	1 1 0	0 0 0	0	-	1	0
E7	1 1 1	- - -	-	-	-	-

Taulako balioak erabiliz, biegonkorren sarreraren adierazpen minimoak lortu behar ditugu. Horretarako, K-mapak erabiliko ditugu.

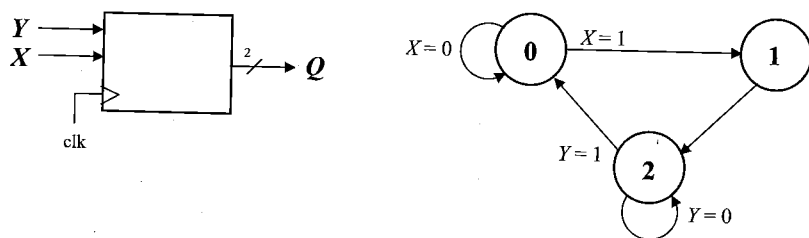


Biegonkorren sarreretarako lortu ditugun funtzioak aintzat izanik, nabarmena da aurreko soluzioa ez dela eraginkorra, funtzio horiek erabiliz eraikitako zirkuitua aurrekoa baino askoz sinpleagoa baita.



>> 4.5. Ariketa

Garatu ezazu X eta Y kanpo-seinaleen mende dagoen sekuentziadore bat, irudiko grafoak adierazten duen sekuentzia sortzen duena.



Ariketa honetan ere sekuentziadore bat eraiki behar da. Sekuentzian, 0, 1 eta 2 egoerak agertzen dira eta, beraz, nahikoak dira 2 bit (2 biegonkor) sekuentzia adierazteko. Enuntziatuan ez da zehazten biegonkorren mota; adibide honetan, esaterako, D biegonkorrak erabiliko ditugu. Adi! sekuentziadore honetan, sistemaren egoeraz gain, X eta Y kanpo-seinaleek parte hartzen dute.

Aurreko bi ariketak ebatzi ditugun modu berean ebatziko dugu hau ere. Taula bat egingo dugu, non adieraziko baitugu sistemaren egoera, erloju-ertza eta gero, sekuentziaren gainean eragina duten aldagai guztien konbinazio posible guztietarako: Q_B , Q_A , X eta Y .

Zirkuitua eraikitzeko D biegonkorrak erabiliko ditugunez, D sarreretan kokatu beharko dira erloju-ertzaren ostean biegonkorren irteeretan izatea nahi ditugun balioak. Beraz, ez da beharrezkoa balio horiek taulan adieraztea, Q' balioak direlako.

UE Q_B Q_A	Kontrol-seinaleak X Y	HE Q_B' Q_A'
0 0	0 -	0 0
	1 -	0 1
0 1	- -	1 0
1 0	- 0	1 0
	- 1	0 0
1 1	- -	- -

Taula horretatik abiatuz, K-mapen bidez, biegonkorren D_B eta D_A sarreretarako funtzio minimoak lortuko ditugu. X eta Y kanpo-seinaleak eta Q_B eta Q_A (sistemaren egoera) dira funtzioen aldagaiak.

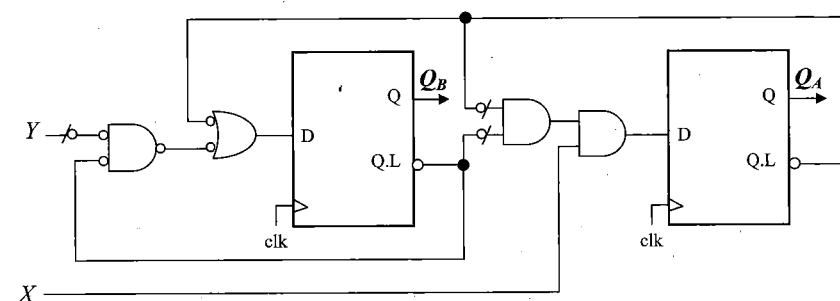
$Q_B Q_A$	XY		X	
	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	-	-	-	-
10	1	0	0	1

$$D_B = Q_A + Q_B \cdot \bar{Y}$$

$Q_B Q_A$	XY		X	
	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	-	-	-	-
10	0	0	0	0

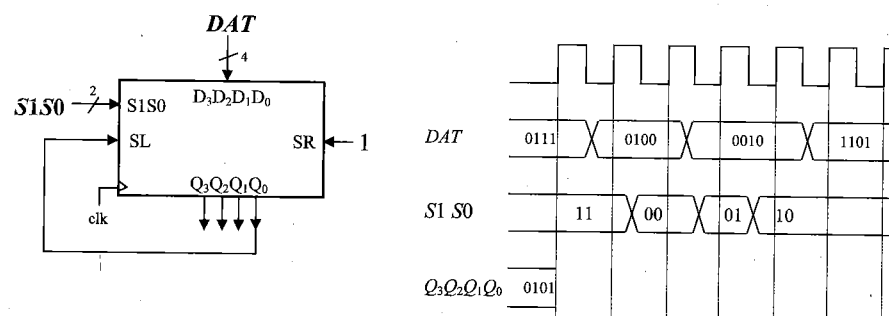
$$D_A = \bar{Q}_B \cdot \bar{Q}_A \cdot X$$

Funtzio horiek eraikitzeko aukera asko dago. Esaterako, hau izan daiteke horietako bat:



» 4.6. Ariketa

4 biteko desplazamendu-erregistroa ageri da irudian, 4.3.2. atalean azaldu duguna. Egin ezazu haren funtzionamendua islatzen duen kronograma.



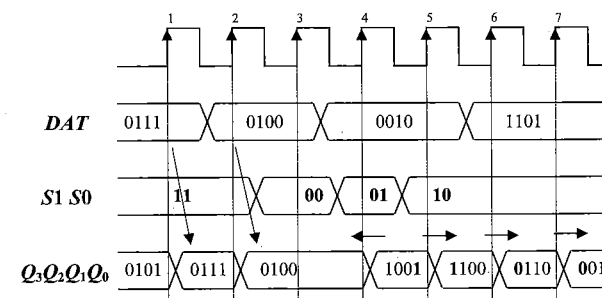
Bit bateko memoriak —biegonkorak— landu ditugu aurreko ariketetan. Ariketa honetan, desplazamendu-erregistro baten portaera islatzen duen kronograma egin behar dugu. Desplazamendu-erregistro mota asko dago, baina erabilienak 4 eragiketa egiteko aukera ematen du: datuak kargatu eta mantendu (erregistro guztiak bezala), eta metatutako datua desplazatu, ezkerrerantz zein eskuinerantz (hortik datorkio izena). 4 eragiketa horiek kodetzeko, bi biteko eragiketa-kodea erabili ohi da —S1 eta S0—, taula honetan ageri den moduan:

S1	S0	Eragiketa
0	0	datua mantendu
0	1	← bit bateko desplazamendua ezkerrera
1	0	→ bit bateko desplazamendua eskuinera
1	1	n biteko datu bat kargatu

Desplazamendu-erregistroa zirkuitu sinkronoa da, eta beraz, edozein eragiketa egiteko, beharrezkoa da erloju-seinalea: erloju-ertza aktibatzean prozesatuko dira S1 eta S0 kontrol-seinaleak.

Bit bateko desplazamenduak egiten direnean hutsik geratuko da posizio bat, ezkerrean (desplazamendua eskuinera denean) edo eskuinean (desplazamendua ezkerrera egiten denean), eta hor kanpoko bit bat kargatuko da: SR sarrerakoa eskuinekoan, edo SL sarrerakoa ezkerrekoan.

Azalpen horiekin, nahikoa informazio badugu kronograma betetzeko.



Lehenengo bi erloju-ertzetan, S1S0 kontrol-seinaleek 11 balioa dute eta, ondorioz, erregistroaren sarreran dagoen datua kargatuko da: lehenengo erloju-ertzarekin batera, 0111 datua, eta, bigarrenetan, 0100 datua.

Hirugarren erloju-ertzean, S1S0 kontrol-seinaleek 00 balioa dute. Kasu horretan, beraz, erregistroaren edukia mantendu egingo da: 0100.

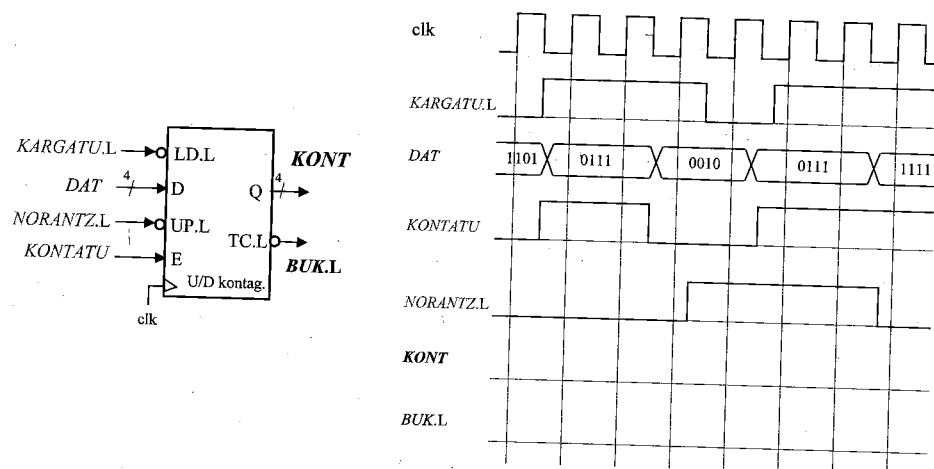
Laugarren erloju-ertzean, S1S0 kontrol-seinaleek bit bateko desplazamendua ezkerrerantz agintzen dute: 01. Erregistroaren edukia posizio bat desplazatuko da ezkerrera; gainera, eskuineko bitean (pisu txikieneko posizioan), desplazamendu-erregistroaren SR sarreran dagoen bita kargatuko da; adibide honetan, 1 konstantea. Hala, erregistroaren edukia 0100 izatetik 1001 izatera pasatuko da.

Hurrengo erloju-ertzetan, S1S0 kontrol-seinaleek bit bateko desplazamendua eskuinera agintzen dute: 10. Hala, erregistroaren edukia posizio bat desplazatuko da eskuinera, eta, horrekin batera, ezkerreko (pisu handieneko) posizioan, SL sarreran dagoen bita kargatuko da; adibide honetan, erregistroaren Q0 bita eraman da SL sarrerara, hau da, edukiaren bit bateko biraketa egiten da. Horregatik, erregistroaren edukia 1001etik 1100ra pasatuko da erloju-ertz horretan; eta 1100tik 0110ra eta 0110tik 0011era hurrengoetan.

(Adi! datu-sarrera aldatzen bada ere 3. ziklotik aurrera, aldaketa horiek ez dute eraginik, desplazamenduak egin behar direlako eta ez datu berrien karga.)

>> 4.7. Ariketa

4 biteko kontagailu bat ageri da irudian, gorantz eta beherantz konta dezakeena (up/down), 4.3.3. atalean azaldu duguna. Egin ezazu haren funtzionamendua islatzen duen kronograma.



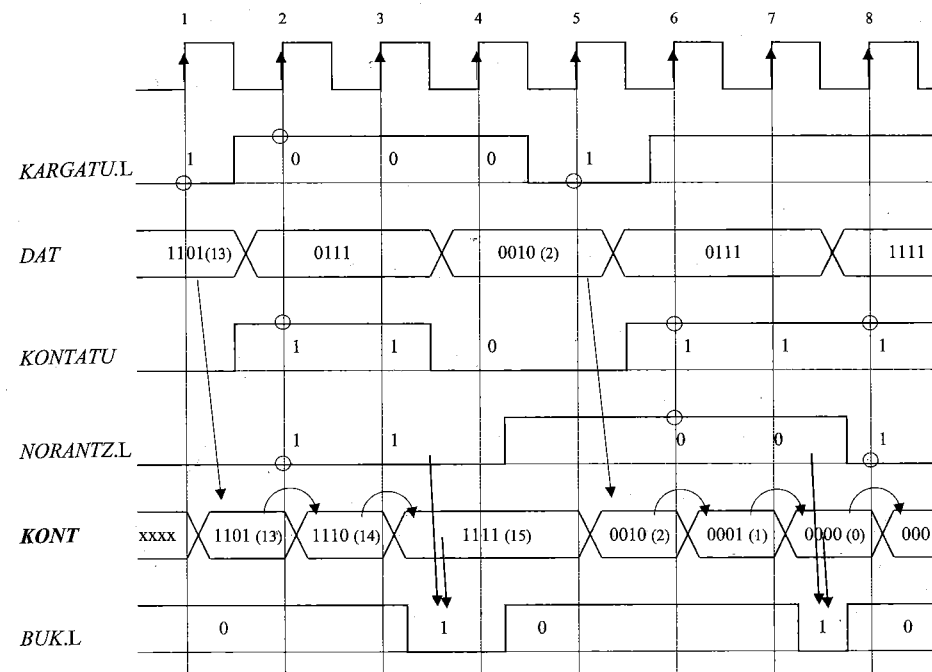
Aurreko ariketan desplazamendu-erregistroaren funtzionamendua aztertu dugu eta, honetan, kontagailuarena landuko dugu.

Ariketa honetako kontagailuak honako kontrol-seinale sinkrono hauek ditu:

- LD (load): datu-sarreran dagoen datua kontagailuan kargatzeko.
- E (enable): edukiari unitate bat gehitzeko edo kentzeko, hau da, kontatzeko.
- UP: kontaktaren noranzkoa adierazteko (E seinalarekin batera): 1 bada, +1 eragiketa beteko da, eta 0 bada, -1 eragiketa.

LD eta E seinaleak batera aktibatuta badaude, karga-eragiketa beteko da; hau da, LD seinaleak lehentasuna du.

Bi irteera ditu irudiko kontagailuak: kontagailuaren edukia (4 bit adibide honetan), eta TC seinalea. TC seinaleak kontaktaren amaiera adierazten du, kontaktaren noranzkoaren arabera, hau da: $TC = (Q = 15) \cdot UP + (Q = 0) \cdot \overline{UP}$. Beraz, adi! TC seinalea irteera baten (Q) eta sarrera baten (UP) mende dago.



1. erloju-ertza iristen denean, KARGATU.L seinala aktibatuta dago eta, beraz, kargatu egingo da kontagailuan datu-sarrerako balioa: 1101, hain zuzen ere.

2. erloju-ertzean, berriz, KARGATU.L seinala desaktibatuta dago eta KONTATU aktibatuta; NORANTZ.L seinala 1 denez, kontagailuaren edukia "inkrementatu" egingo da: 1101etik 1110ra.

3. erloju-ertzean, eragiketa bera errepikatuko da, kontrol-seinaleen balioak berdinak direlako. Beraz, kontagailua 1110tik 1111 izatera pasatuko da.

4. erloju-ertzean, LD (kargatu) zein E (kontatu) kontrol-seinaleak desaktibatuta daude; ondorioz, ez da aldatuko kontagailuaren balioa.

5. erloju-ertza heltzean, berriro ere KARGATU.L seinala aktibatuta dago eta, beraz, datu-sarrerako balioa kargatuko da kontagailuan: 0010, hain zuzen ere. Une honetatik aurrera, desaktibatuta mantenduko da KARGATU.L seinala kronograma osoan zehar; beraz, ez da berriro datu-sarrerako balioa kontagailuan gordeko.

6. eta 7. erloju-ertzetan, eragiketa bera beteko da; KONTATU = 1 eta NORANTZ.L = 0 direnez, kontagailuaren edukia "dekrementatu" egingo da: 0010tik 0001era eta 0001etik 0000ra.

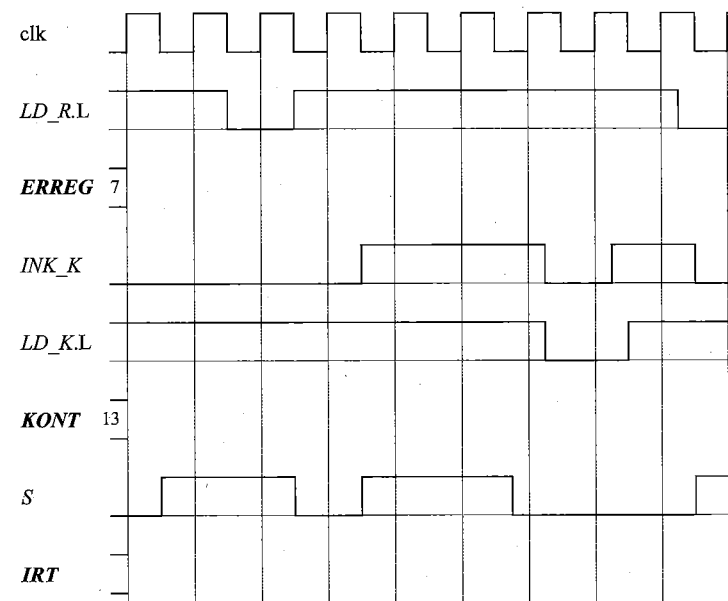
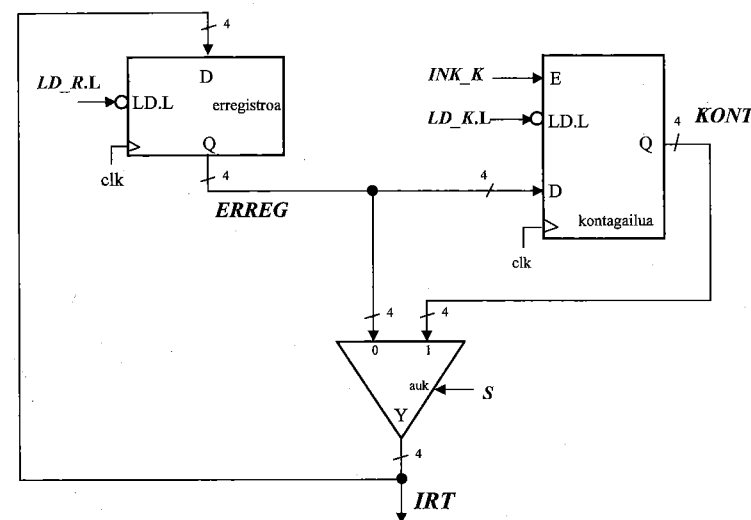
8. erloju-ertzean, kontaktarekin jarraituko da, baina orain gorantz: 0000tik 0001era.

Lehen esan dugun moduan, TC (edo RCO) irteerak kontaktaren mugak adierazten ditu: 4 biteko kontagailuetan, $Q = 1111$, UP sarrera 1 denean, eta $Q = 0000$, UP sarrera 0 denean. Kronograman ageri den moduan, bi bider aktibatuko da *BUK* seinalea (TC irteera): kontagailuaren edukia 15 denean eta *NORANTZ.L* seinalea 1 denean (3. zikloa) eta edukia 0 eta *NORANTZ.L* 0 direnean (7. zikloa).

<<

>> 4.8. Ariketa

4 biteko erregistro bat, kontagailu bat eta multiplexore bat erabiltzen ditu irudiko zirkuituak. Bete ezazu haren portaera islatzen duen kronograma.

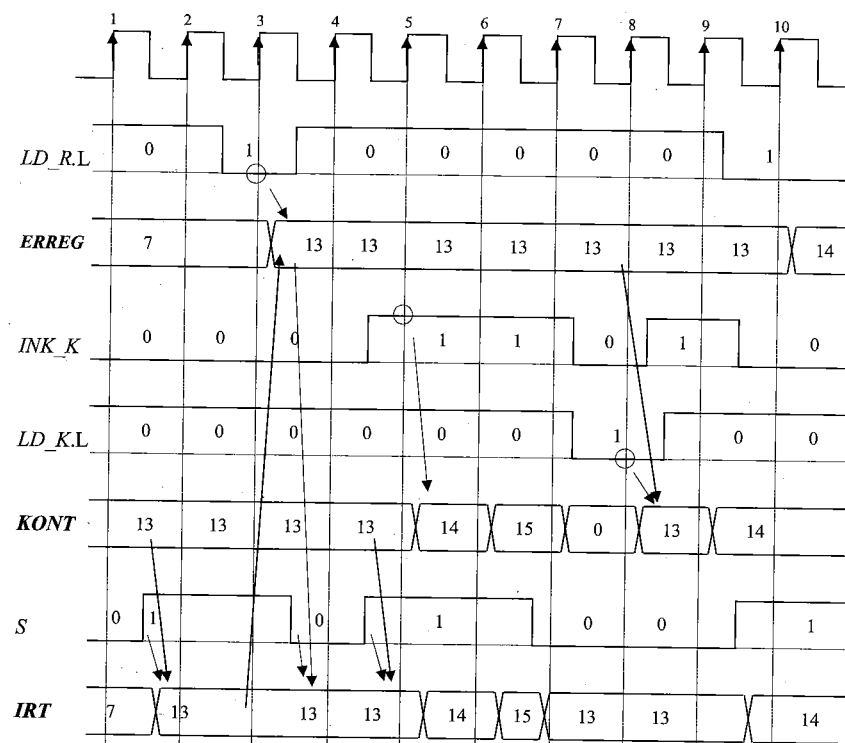


Bi motatako gailuak erabiltzen dira ariketa honetako zirkuituan: sekuentzialak —erregistroa eta kontagailua— eta konbinazionalak —multiplexorea—. Kronograma egiterakoan, zirkuitu bakoitzaren izaera kontuan eduki eta bi portaera nagusi bereizi beharko ditugu: erregistroan eta kontagailuan, zirkuitu sinkronoetan, kontrol-seinaleak erloju-ertzak gertatzen denean soilik prozesatuko dira, baina, multiplexorean, zirkuitu konbinazionalan, sarrerak aldatzen diren unean gertatuko dira aldaketak irteeran. Kasu batean zein bestean, ezin dugu ahaztu zirkuituek denboratarteko bat behar dutela erantzuteko, eta kronograman islatuko dugu.

Honela funtzionatzen dute zirkuitu honetako gailuek:

- Erregistroa: datu bat kargatzen du LD_R seinalea aktibatuta badago (erloju-ertzean).
- Kontagailua: datu bat kargatzen du (LD_K) edo edukia inkrementatzen du (INK_K) erloju-ertza heldzean.
- Multiplexorea: sarreretak datu bat aukeratu eta irteerara eramaten du, S seinalearen arabera.

Ikus dezagun nola egin kronograma.



Zikloz ziklo gertatzen dena aztertzen hasi aurretik, kronogramako kontrol-seinaleei begirada bat ematea lagungarria da. Erregistroaren edukia 3. eta 10. erloju-ertzak gertatu eta gero aldatuko da soilik, LD_R bi erloju-ertz horietan bakarrik baitago aktibatuta. Kontagailuaren edukia, berriz, 5., 6., 7., 8. eta 9. erloju-ertzak gertatu eta gero aldatuko da; 8.ean, datu-sarrerako balioa kargatuko da, eta, gainerakoetan, inkrementatu egingo da. Multiplexorearen irteera, aldiz, gehiagotan aldatuko da, S eta erregistroaren eta kontagailuaren edukien arabera (adi! erlojuak ez du eraginik multiplexorean).

Azter dezagun kronograma, zikloz ziklo.

1. erloju-ertzean, erregistroaren karga-seinalea eta kontagailuaren kargatzeko eta kontatzeko seinaleak 0 dira. Beraz, bien edukia mantenduko da. Multiplexoreko aukeratze-seinalea 0 denez, erregistroaren edukia izango da haren irteeran, 7ko bat, hain zuzen ere.

2. erloju-ertza gertatu baino lehen, S kontrol-seinalea 0tik 1era pasatzen da eta, beraz, multiplexorearen irteera ere (IRT) aldatuko da, orain kontagailuan dagoen balioa aukeratuko baita: 13a, kasu konkretu honetan.

3. erloju-ertza iristen denean, LD_R seinalea aktibatuta dago, eta, ondorioz, datu bat kargatuko da erregistroan: multiplexorearen irteeran dagoena, 13, hain zuzen ere. Hori dela eta, 4. erloju-ertza baino lehen S seinalea aldatu egiten den arren, multiplexorearen irteerako balioa ez da aldatuko, une horretan berdinak baitira erregistroaren eta kontagailuaren edukiak.

5. erloju-ertzean, INK_K seinalea aktibatuta dago, eta, ondorioz, kontagailua inkrementatu egingo da, 13tik 14ra. Aldaketa hori dela eta, multiplexorearen irteera ere aldatuko da, kontagailuaren edukia aukeratuta baitago irteeran ($S = 1$ baita).

6. erloju-ertzean, aurreko eragiketa errepikatzen da; ondorioz, kontagailua eta multiplexorearen irteera 14tik 15era aldatuko dira. Zikloaren amaieran, S kontrol-seinalea 1etik 0ra pasatzen da, eta, ondorioz, IRT , multiplexorearen emaitza, ere aldatu egingo da: erregistroan dagoen balioa (13) hartuko du.

7. erloju-ertza gertatzen denean, berriro ere, kontatzeko seinalea aktibatuta dago eta, beraz, kontagailua inkrementatu egingo da. Une horretan, kontagailuaren edukia 15 denez, 4 bitekin adieraz daitekeen zenbaki handiena, 15etik 0ra pasatuko da: $1111 + 1 = (1) 0000$.

8. erloju-ertzean, LD_K seinalea aktibatuta dago, eta, beraz, datu bat kargatuko da kontagailuan, datu-sarrerakoa: erregistroaren edukia (13).

9. erloju-ertzean, $INK_K = 1$ da, eta, beraz, kontagailuaren edukia gehituko da (13tik 14ra). 9. zikloan zehar, S seinalea 0tik 1era pasatzen da eta, beraz,

multiplexorearen irteera ere aldatu egingo da, kontagailuaren irteerako balioa hartuko du, 14, hain zuzen ere.

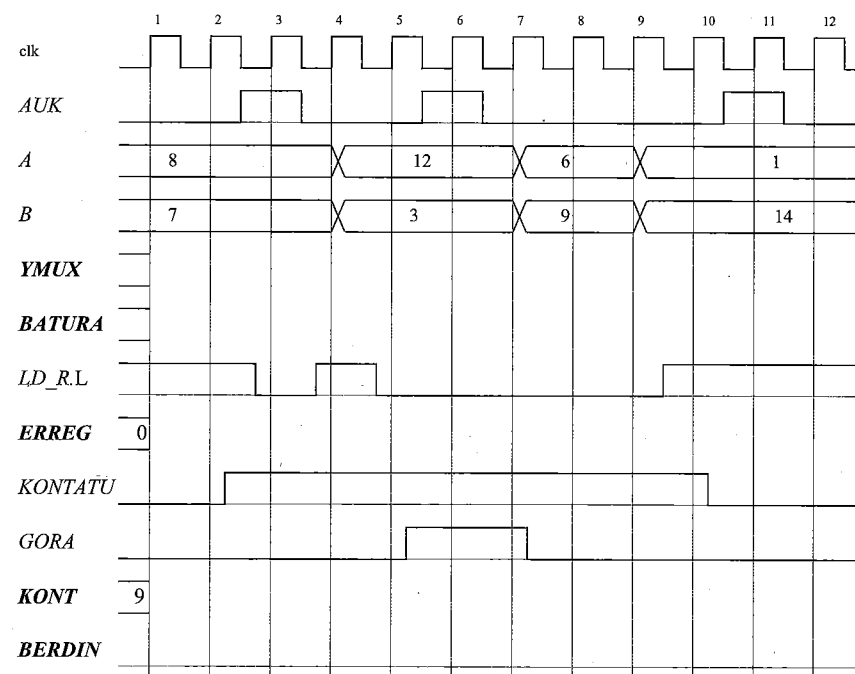
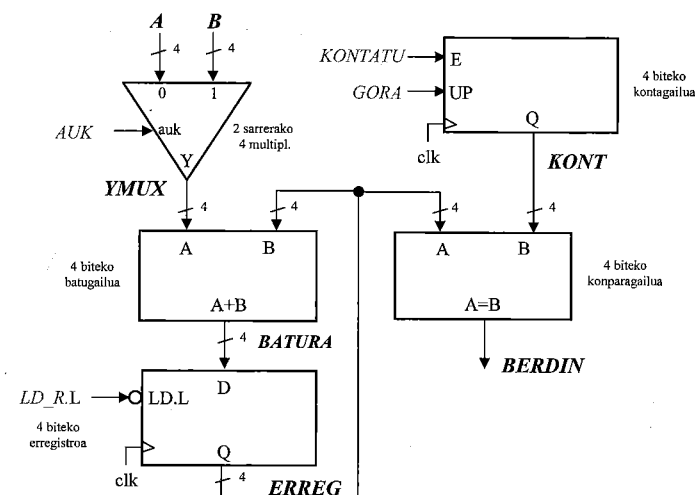
Azkenik, 10. erloju-ertzean, LD_R seinalea aktibatuta dago: multiplexorearen itteerako balioa kargatuko da erregistroan (14).

Adibide honetan, kronograma egitean, zirkuitu sinkrono zein konbinazionalen atzerapenak islatu ditugu. Aldaketak erloju-ertza gertatu baino pixka bat geroago gertatu dira zirkuitu sinkronoetan, eta sarrerek aldatu baino pixka bat geroago zirkuitu konbinazionalak.



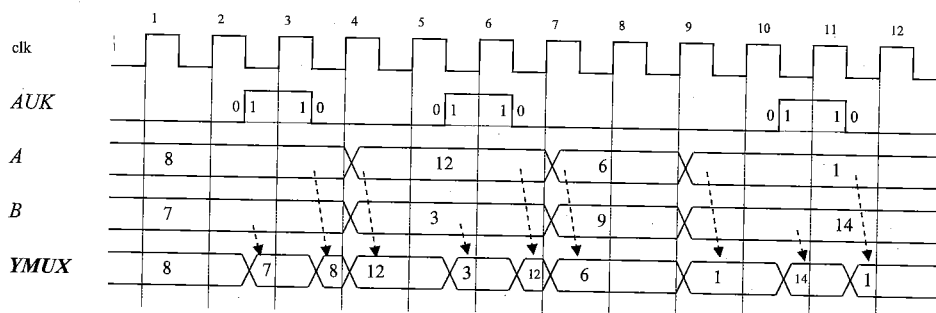
» 4.9. Ariketa

4 biteko erregistro, kontagailu, multiplexore, batugailu eta konparagailu bana erabiltzen ditu irudiko zirkuituak. Bete ezazu haren portaera islatzen duen kronograma (hasierako balioak: ERREG = 0, KONT = 9).



Kronogramari ekin baino lehen, ohartxo bat. Kronograma sinplifikatzearen, ez ditugu batuko zirkuituen erantzun-denborak, bata bestearen atzetik datozenean.

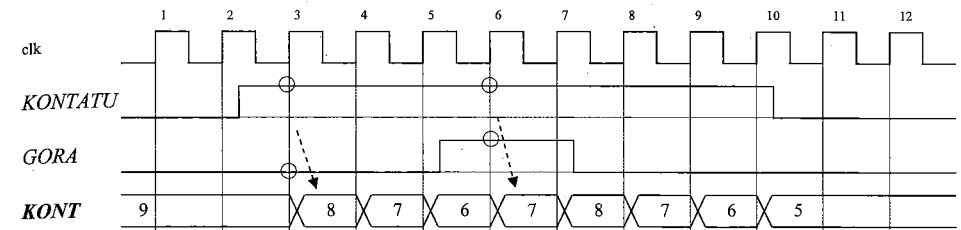
Kronograma betetzen hasi aurretik interesgarria da zirkuitua aztertzea eta atal "independentek" identifikatzea, zirkuituaren analisia sinplifikatzeko. Adibide honetan, esaterako, multiplexorearen irteerako datua AUK hautatze-lerroaren eta A eta B sarrera-datuen mende dago soilik. $AUK = 0$ denean, A sarrerako datua izango da irteeran, eta $AUK = 1$ denean, berriz, B sarrerakoa. Portaera hori 4.18. irudian ageri da.



4.18. irudia. 4.9. ariketako multiplexorearen kronograma.

2. zikloan, AUK 0tik 1era aldatzen denean, $YMUX$ 8tik 7ra pasatuko da (B sarrera). 3. zikloan, AUK 1etik 0ra aldatzen denean, $YMUX$ irteeran berriro ere A sarrera, 8koa, izango dugu. Modu berean analiza daitezke $YMUX$ irteeraren aldaketak denboran zehar.

Kontagailuaren eboluzioa ere bere kasa analiza daiteke, kontatze-seinalearen (eta noranzkoaren) mende baitago soilik. Hasieran, $KONT = 9$ da, eta kontagailuaren edukia aldatuko da $KONTATU = 1$ den ziklo guztietan: gorantz 6. eta 7. erloju-ertzetan ($GORA = 1$ delako), eta beherantz 3., 4., 5., 8., 9., eta 10. erloju-ertzetan ($GORA = 0$ delako). Portaera hori 4.19. irudian ageri da.



4.19. irudia. 4.9. ariketako kontagailuaren kronograma.

Bete ditzagun orain, zikloz ziklo, kronogramako gainerako seinaleak (ikus 4.20. irudia).

Batugailuaren eta konparagailuaren irteerak sarrerak aldatzen diren uneetan aldatuko dira; batugailuaren kasuan, multiplexorearen irteera edo erregistroaren irteera aldatzean (haien edukiak batzen baititu) eta, konparagailuaren kasuan, erregistroaren irteera edo kontagailuaren irteera aldatzen direnean. Erregistroaren irteera, berriz, erloju-ertza iritsi eta gero aldatuko da, LD_R seinaleak hala adierazten badu, jakina. LD_R seinalea aktibatuta badago erloju-ertzean, batugailuaren edukia gordeko da erregistroan.

Lehenengo bi erloju-zikloetan, erregistroaren edukia mantendu egiten da, 0 balioarekin. Hori dela eta, batugailuaren emaitza multiplexorearen irteera aldatzen denean soilik aldatuko da eta, gainera, hartuko duen balioa multiplexorearenarekin bat etorriko da.

3. erloju-ertzean, LD_R seinalea aktibatuta dagoenez, batugailuaren irteerako balioa erregistroan gordeko da, 7, hain zuzen ere. Erregistroaren edukia aldatzearekin batera, batugailuarena aldatuko da: $7 + 7 = 14$ izatera pasatuko da. 3. erloju-zikloan zehar, multiplexorearen irteera aldatu egiten da 7tik 8ra, eta, beraz, batugailuarena ere aldatuko da, $7 + 8 = 15$. Gainerako erloju-zikloetan, portaera bera errepikatuko da batugailuan zein erregistroan (adi! 4 biteko batuketak egiten dira; esaterako, 4. zikloan, $12 + 7 = 3$ da, hau da, $1100 + 0111 = (1) 0011$).

Azkenik, $BERDIN$ irteeraren portaera aipatzea falta da. Kontagailuaren eta erregistroaren edukiak konparatzearen emaitza adieraziko digu honek. Bi balioak berdinak direnean, leko balioa hartuko du, eta bestela 0koa.

