

CPU: Kontrol-Unitatea

Konputagailuen Arkitektura I

1. gaia

Nola eraiki CPUa?

sistema digitala

Oinarrizko blokeak

Konbinazionalak:

ateak

multiplexoreak

konparagailuak

deskodegailuak

batugailuak

unitate aritmetiko eta logikoak

memoriak

Sekuentzial sinkronoak (clk):

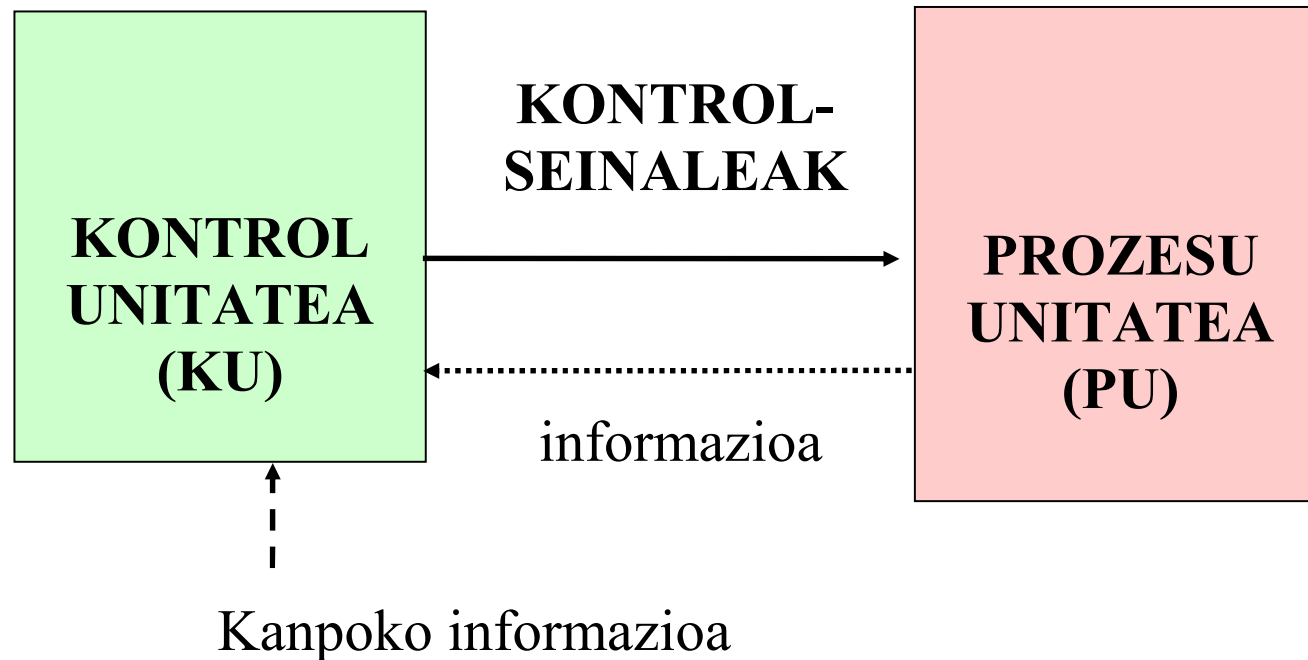
biegonkorrak

erregistroak

kontagailuak

CPU-ren antolaketa

Sistema digital baten egitura generikoa



Sistema digital baten egitura generikoa

Kontrol-unitatea: 1. gaia

egitura estandarra

funtzionamendu espezifikoa kontrol-algoritmoak aginduta

Prozesu-unitatea:

1) erabilpen espezifikoko sistema

erabilpen zehatzaren menpeko egitura

2) helburu orokorreko sistema: prozesadorea

egitura: datu-bidea (*data path*)

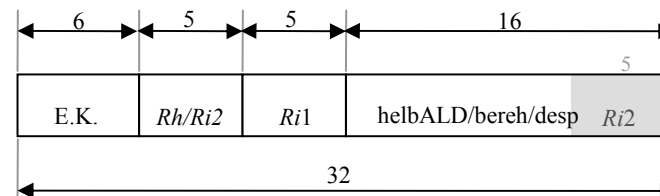
erregistroak + unitate aritmetiko logikoa: 5.gايا

Zirkuitu aritmetikoak (algoritmoak)

Prozesadore sinple bat: BIRD

- 32 biteko 42 agindu
 - eragiketa-kodea: 6 bitekoa
 - 4 helbideratze-modu:
 - berehalakoa
 - datuen tamaina: 16 bitekoak
 - absolutua
 - memoriaren tamaina: 64K x 16 bit → helbideak: 16 bitekoak
 - erregistro bidezko zuzenekoa
 - 32 erregistroko erregistro-multzoa: 5 bit erregistroak helbideratzeko
 - indexatua
 - oinarri-helbidea: 16 bit; indizea: erregistro baten edukia

Aginduen formatu orokorra:



BIRD-en agindu batzuk (1)

Agindu aritmetikoak eta logikoak:

Formatua	Mihiztadura-lengoaia	Eragiketa										
<p>add</p> <table border="1"><tr><td>001001</td><td>Rh</td><td>Ri1</td><td></td><td>Ri2</td></tr><tr><td>6</td><td>5</td><td>5</td><td></td><td>5</td></tr></table>	001001	Rh	Ri1		Ri2	6	5	5		5	<pre>add rh,ri1,ri2</pre>	<pre>rh:= ri1 + ri2</pre> <p>(antzeko aginduak: sub, mul, div, or, and, xor)</p>
001001	Rh	Ri1		Ri2								
6	5	5		5								
<p>addi</p> <table border="1"><tr><td>001010</td><td>Rh</td><td>Ri</td><td colspan="2">berehalakoa</td></tr><tr><td>6</td><td>5</td><td>5</td><td colspan="2">16</td></tr></table>	001010	Rh	Ri	berehalakoa		6	5	5	16		<pre>addi rh,ri,#berek</pre>	<pre>rh:= ri + berek</pre> <p>(antzeko aginduak: subi, muli, divi, ori, andi, xori)</p>
001010	Rh	Ri	berehalakoa									
6	5	5	16									
<p>mov</p> <table border="1"><tr><td>000111</td><td>Rh</td><td>Ri</td><td colspan="2"></td></tr><tr><td>6</td><td>5</td><td>5</td><td colspan="2">16</td></tr></table>	000111	Rh	Ri			6	5	5	16		<pre>mov rh,ri</pre>	<pre>rh:= ri</pre>
000111	Rh	Ri										
6	5	5	16									
<p>movi</p> <table border="1"><tr><td>001000</td><td>Rh</td><td></td><td colspan="2">berehalakoa</td></tr><tr><td>6</td><td>5</td><td></td><td colspan="2">16</td></tr></table>	001000	Rh		berehalakoa		6	5		16		<pre>movi rh,#berek</pre>	<pre>rh:= berek</pre>
001000	Rh		berehalakoa									
6	5		16									

BIRD-en agindu batzuk (2)

Memoriarekin lan egiten duten aginduak:

Formatua	Mihizadura-lengoaia	Eragiketa								
<p>ld</p> <table border="1"><tr><td>000000</td><td>Rh</td><td></td><td>helbidea</td></tr><tr><td>6</td><td>5</td><td></td><td>16</td></tr></table>	000000	Rh		helbidea	6	5		16	<code>ld rh,ALDAGAIA</code>	<code>rh:= MEM[helbidea]</code>
000000	Rh		helbidea							
6	5		16							
<p>st</p> <table border="1"><tr><td>000011</td><td>Ri2</td><td></td><td>helbidea</td></tr><tr><td>6</td><td>5</td><td></td><td>16</td></tr></table>	000011	Ri2		helbidea	6	5		16	<code>st ri2,ALDAGAIA</code>	<code>MEM[helbidea]:= ri2</code>
000011	Ri2		helbidea							
6	5		16							
<p>ldx</p> <table border="1"><tr><td>000010</td><td>Rh</td><td>Ri</td><td>helbidea</td></tr><tr><td>6</td><td>5</td><td>5</td><td>16</td></tr></table>	000010	Rh	Ri	helbidea	6	5	5	16	<code>ldx rh,ALD[ri]</code>	<code>rh:= MEM[helbidea+ri]</code>
000010	Rh	Ri	helbidea							
6	5	5	16							
<p>stx</p> <table border="1"><tr><td>000101</td><td>Ri2</td><td>Ri1</td><td>helbidea</td></tr><tr><td>6</td><td>5</td><td>5</td><td>16</td></tr></table>	000101	Ri2	Ri1	helbidea	6	5	5	16	<code>stx ri2,ALD[ri1]</code>	<code>MEM[helbidea+ri1]:= ri2</code>
000101	Ri2	Ri1	helbidea							
6	5	5	16							

BIRD-en agindu batzuk (3)

Agindu-fluxua kontrolatzeko aginduak (jauziak):

Formatua	Mihizadura-lengoaia	Eragiketa												
<table border="1"><tr><td colspan="3">beq</td></tr><tr><td>011010</td><td style="background-color: #cccccc;"></td><td>Ri</td></tr><tr><td>6</td><td>5</td><td>16</td></tr><tr><td colspan="3">desplazamendua</td></tr></table>	beq			011010		Ri	6	5	16	desplazamendua			<code>beq ri,etiketa</code>	<code>if (ri=0) then PC:= PC_{beq} + desplazamendua</code>
beq														
011010		Ri												
6	5	16												
desplazamendua														

(desplazamendua: beq aginduaren eta etiketak adierazitako aginduaren arteko distantzia)

Adibide-programa bat

Batu bi bektore, A eta B, eta utzi emaitza beste bektore batean, C. Hirurak 16 osagaikoak dira.

Programa BIRDerako:

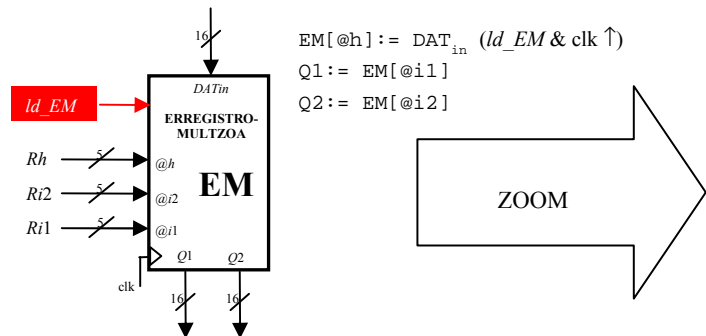
...

```

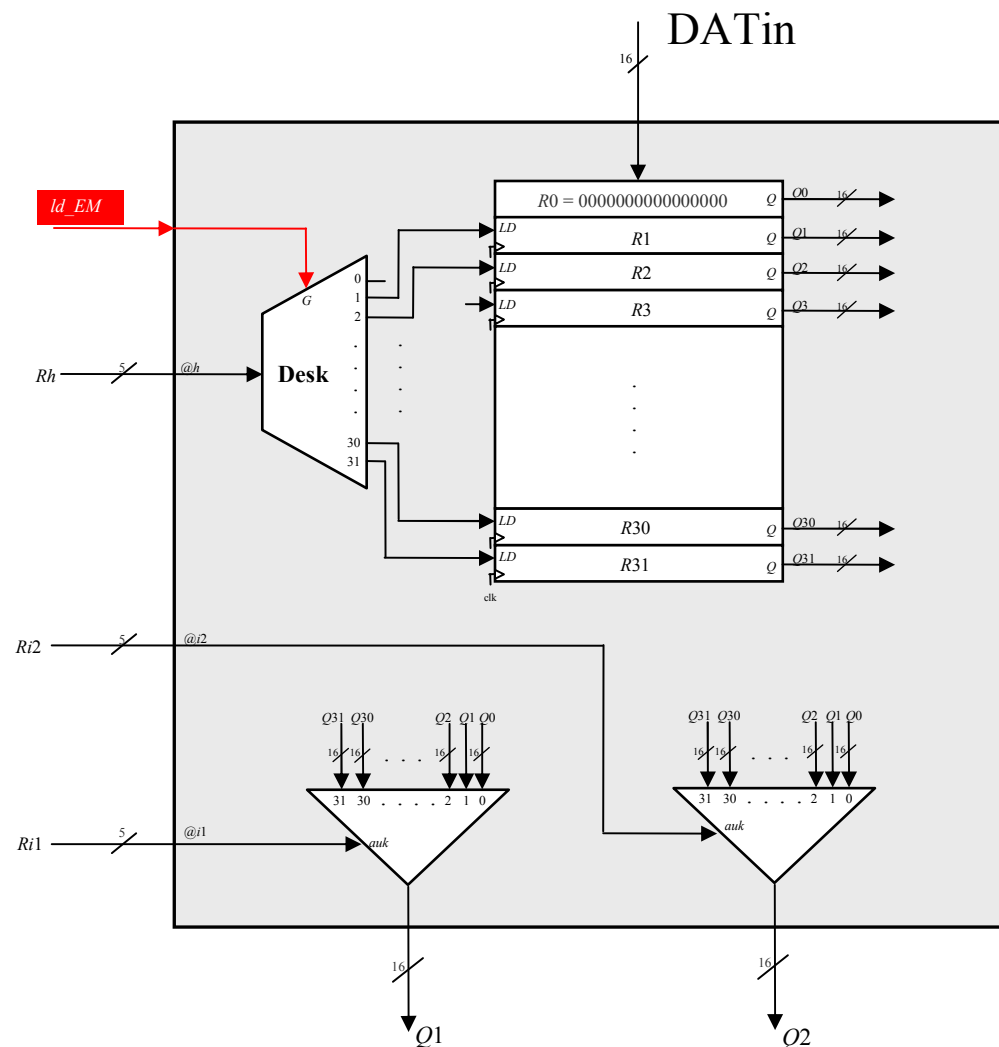
                                movi r1, #0           ; hasieratu indize-erregistroa
                                movi r2, #16          ; 16 osagai bektore bakoitzak
segi:                          ldx  r3, A[r1]        ; irakurri A bektorearen osagai bat
                                ldx  r4, B[r1]        ; irakurri B bektorearen osagai bat
                                add  r5, r4, r3       ; batu bi osagaiak (emaitza, r5-ean)
                                stx  r5, C[r1]        ; gorde emaitza C bektoreko osagaian
                                addi r1, r1, #1       ; inkrementatu r1, bektoreen hurrengo osagaiaren indizea lortzeko
                                subi r2, r2, #1       ; osagai bat gutxiago geratzen da
                                beq  r2, buka         ; r2 = 0 bada, eragiketa bukatu da (jauzi buka-ra)
                                beq  r0, segi         ; bestela, segi batuketa egiten (r0 = 0 → jauzi segi-ra)
buka:                          ...
```

BIRD-en PUaren osagaiak

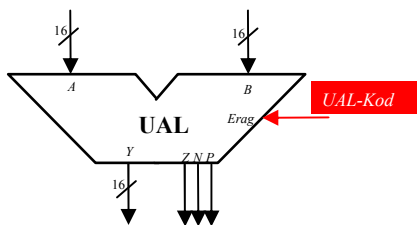
ERREGISTRO-MULTZOA



ZOOM



UNITATE ARITMETIKO-LOGIKOA

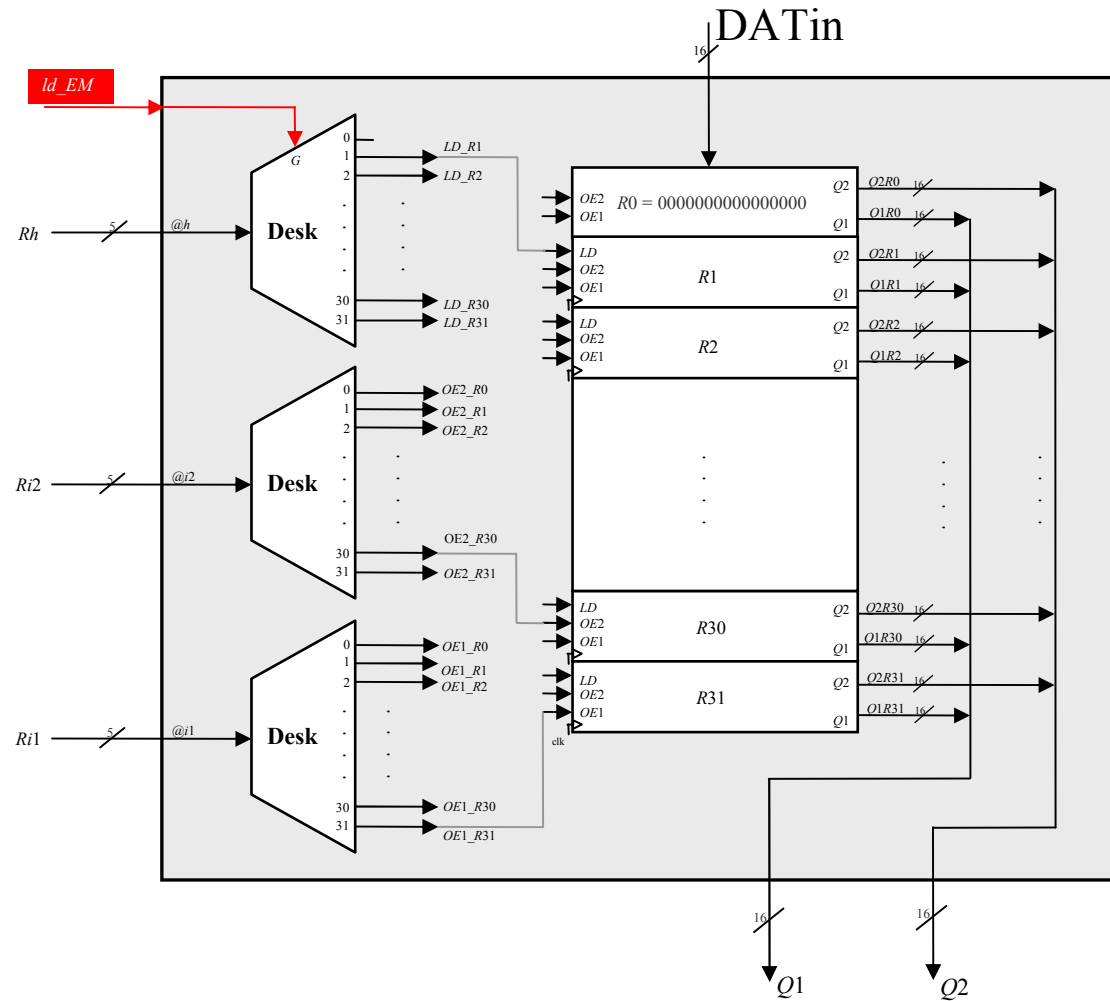


16 eragiketa burutzeko UALa:

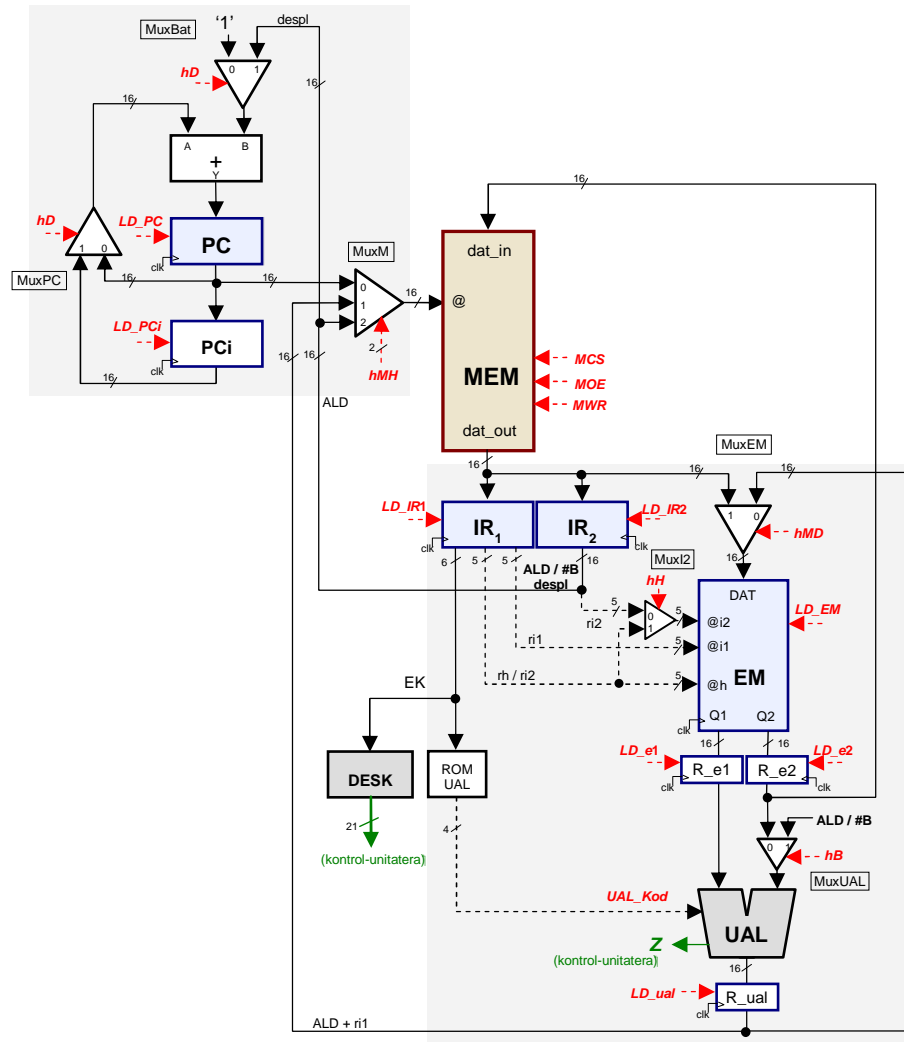
UAL Kod	0000	0001	0010	0011	0100	0101	0110
Y	A + B	A - B	A × B	A/B	A	B	A + 1
UAL Kod	0111	1000	1001	1010	1011	1100	...
Y	A - 1	A or B	A and B	A xor B	← A	→ A	...

BIRD-en PUaren osagaiak

ERREGISTRO-MULTZOA: hiruegoerako erregistroak

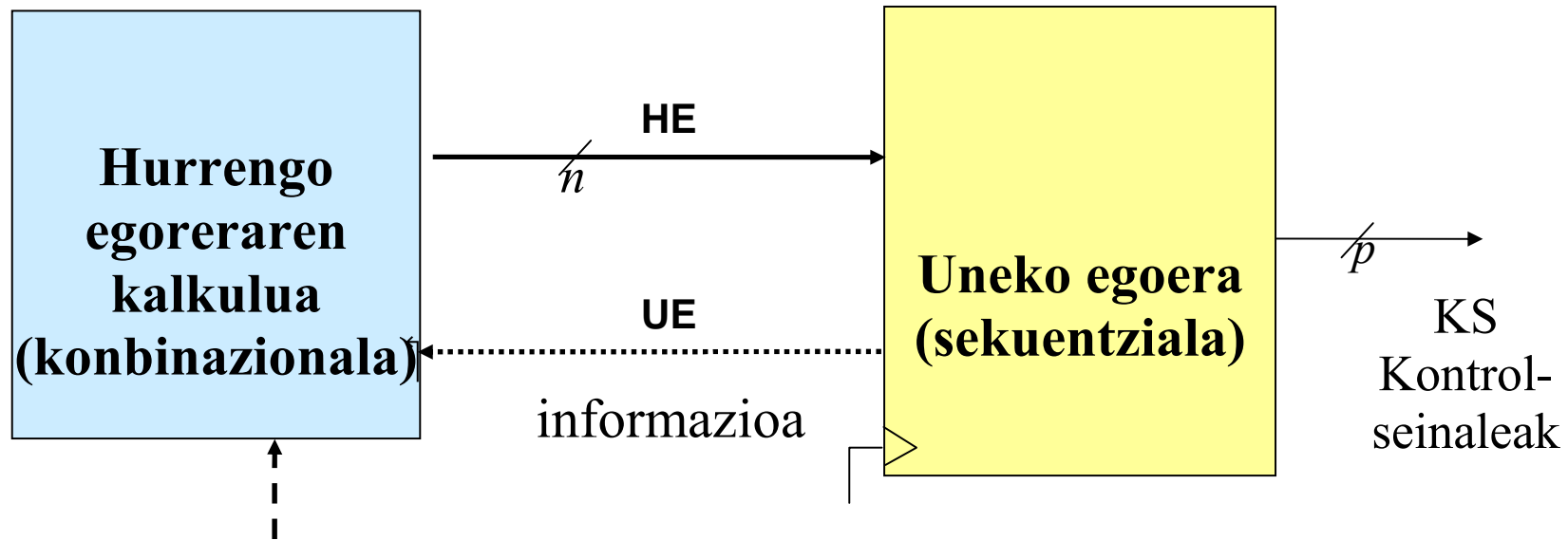


BIRD-en PUA



KONTROL-UNITATEA

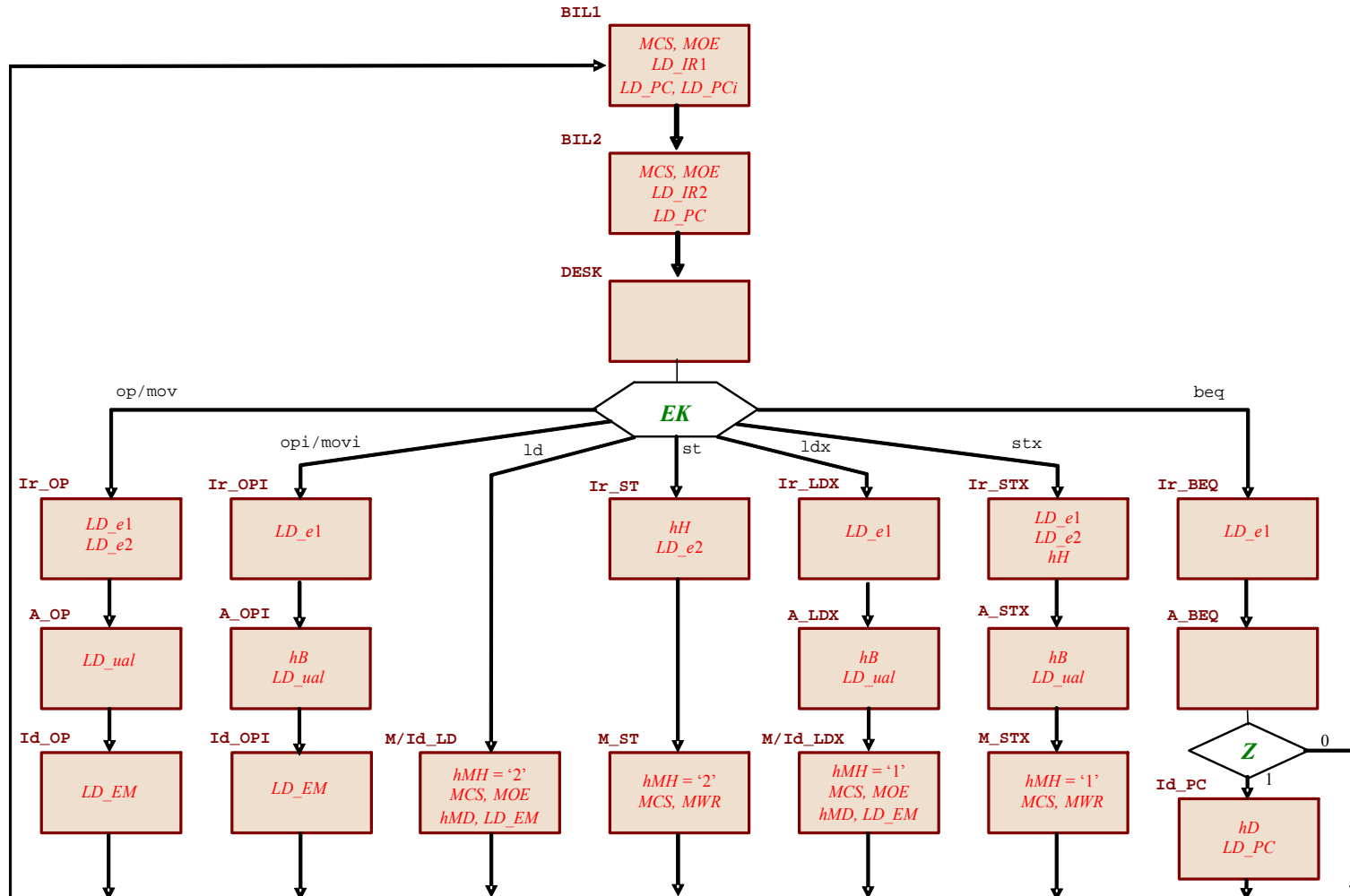
Egitura generikoa: egoeren sekuentziatzea



informazioa (PUtik edo kanpotik)

Funtzionamenduaren zehaztapena: kontrol algoritmoa

BIRD-en kontrol-algoritmoa



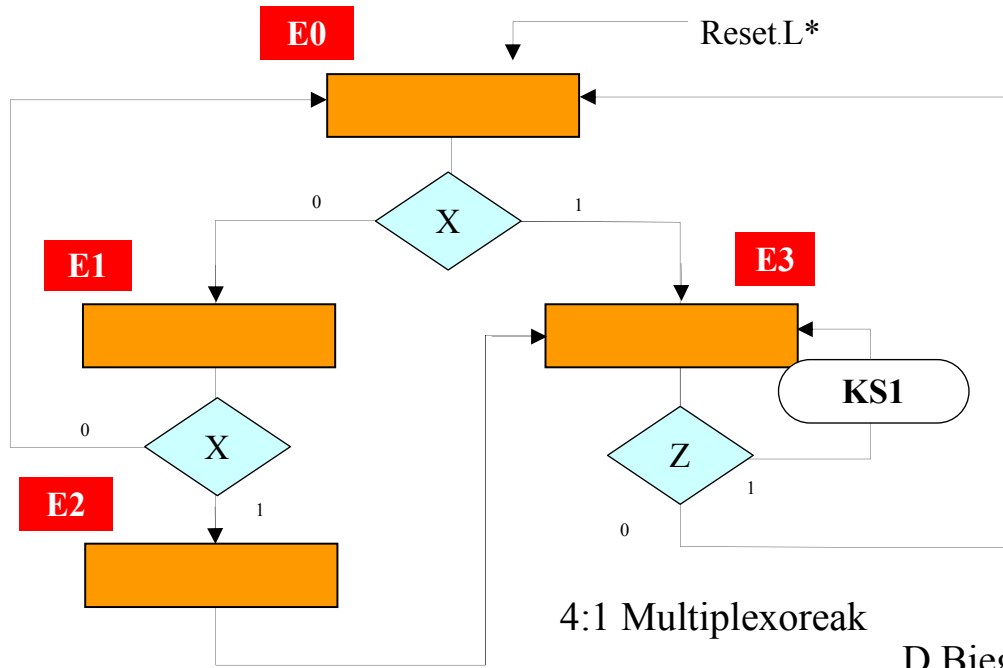
Algoritmoa eraikitzeko metodoa

➤ Multiplexoreen metodoa

- D biegonkorrak egoera gordetzeko
- Multiplexoreak (eta ateak) hurrengo egoera sortzeko
- Deskodegailua (eta ateak) kontrol-seinaleak sortzeko

➤ Metodo kableatuta (ez bakarra)

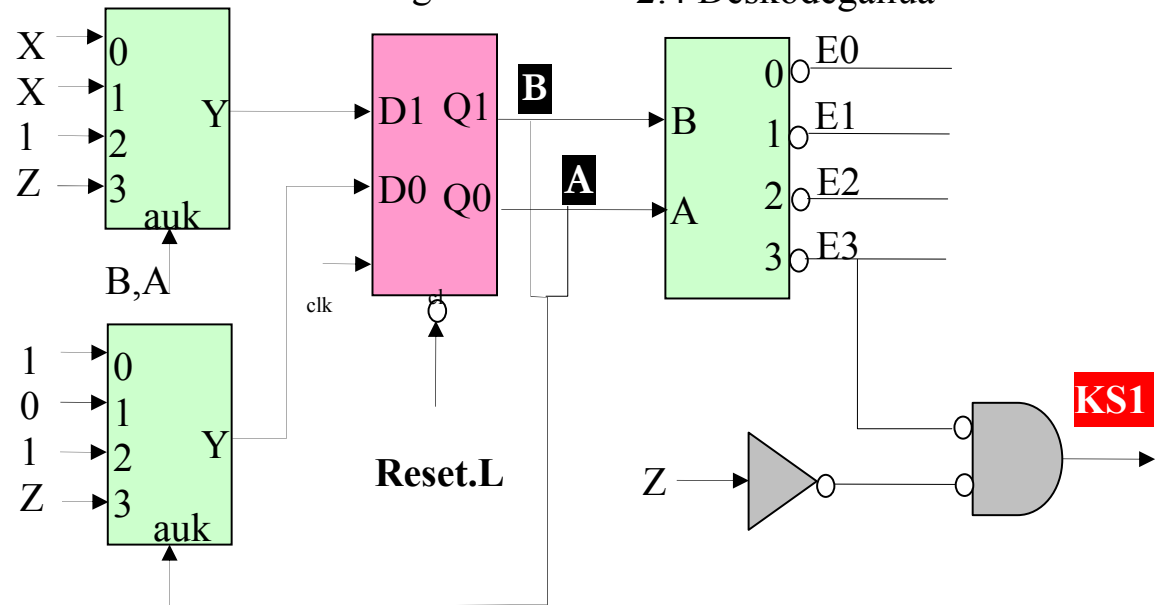
ADIBIDEA



4:1 Multiplexoreak

D Biegonkorrak

2:4 Deskodegailua



Kableatutako diseinuaren abantailak eta desabantailak

- Exekuzio azkarra
- Sekuentziatze eta exekuzio logika konplexua
- Diseinatzeko eta funtzionamendua egiaztatzekeo zaila
- Diseinu ez malgu, ez moldakorra
- Agindu berriak gehitzeko zailtasunak

Gehien bat, RISC prozesadoreetan erabilia

KONTROL UNITATEA

mikroprogramatua

Egoera → **kontagailu** bat

Ertzean: kontatu $-E_i \rightarrow E_{i+1}$ - edo *kargatu (jauzi)* $-E_i \rightarrow E_j$

- Egoerekin erlazionatutako informazioa: **ROM MEMORIA** batean egoeren arteko trantsizio taula + kontrol seinaleak

- Funtzionamendua: egoeren sekuentziatzea eta egoera bakoitzean eragiketa batzuen exekuzioa

- Analogiaz: mikroprograma (firmware)
mikroaginduen sekuentzia

- Mikroagindu bat = kontrol-algoritmoko egoera bat ez da onartzen baldintzapeko irteerarik egoera bakoitzeko baldintza bat bakarrik onartzen da

Aukera posible bat (1)

***KS bakoitzeko bit bat (0/1 balioak)**

***UEn aztertzen den baldintzaren edo kontrol-aldagaiaren “kodea”**

***Jauzia eragiten duen baldintzaren balioa: jauzi-baldintza**

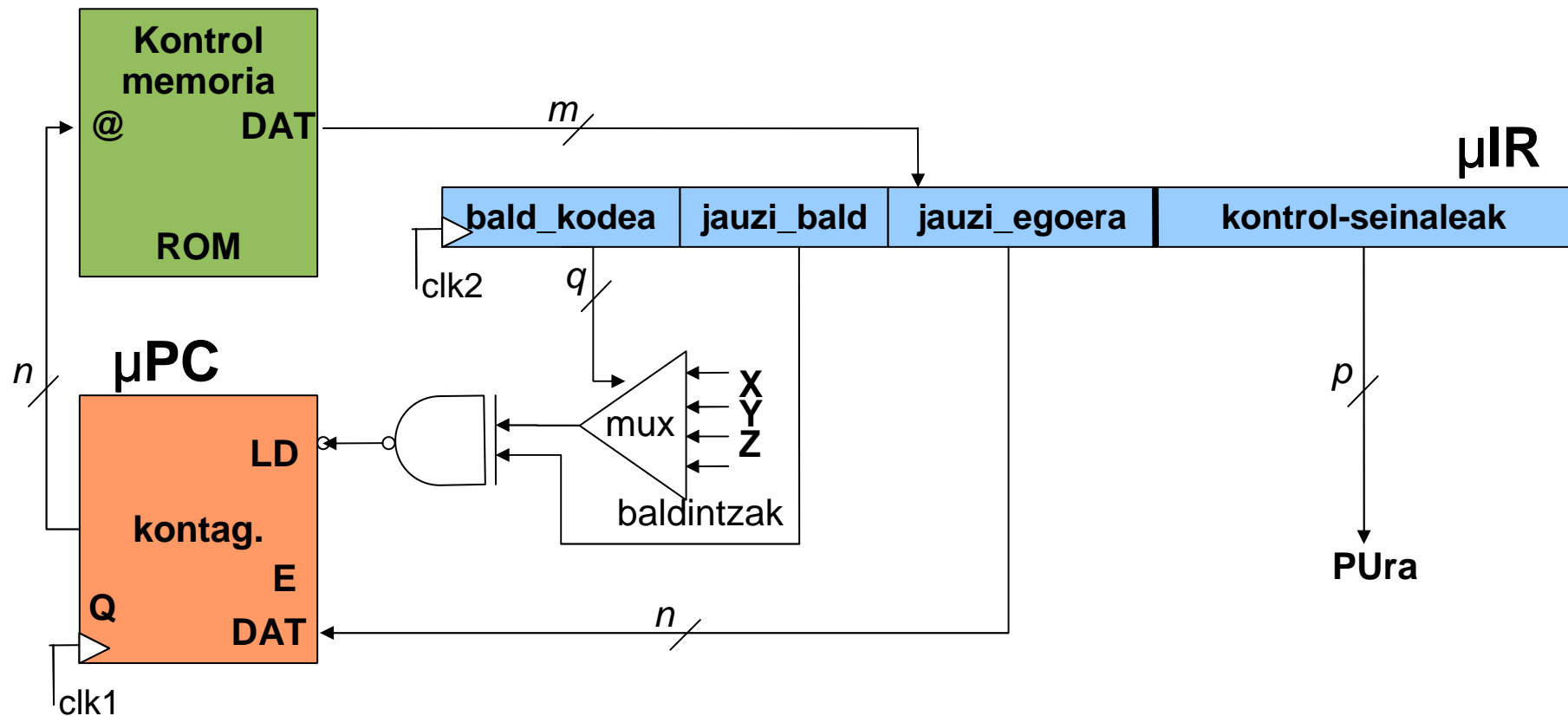
***Jauzia zein egoeratarara (bakarra)**

Aukera posible bat (1)

mikroaginduen formatua (1)

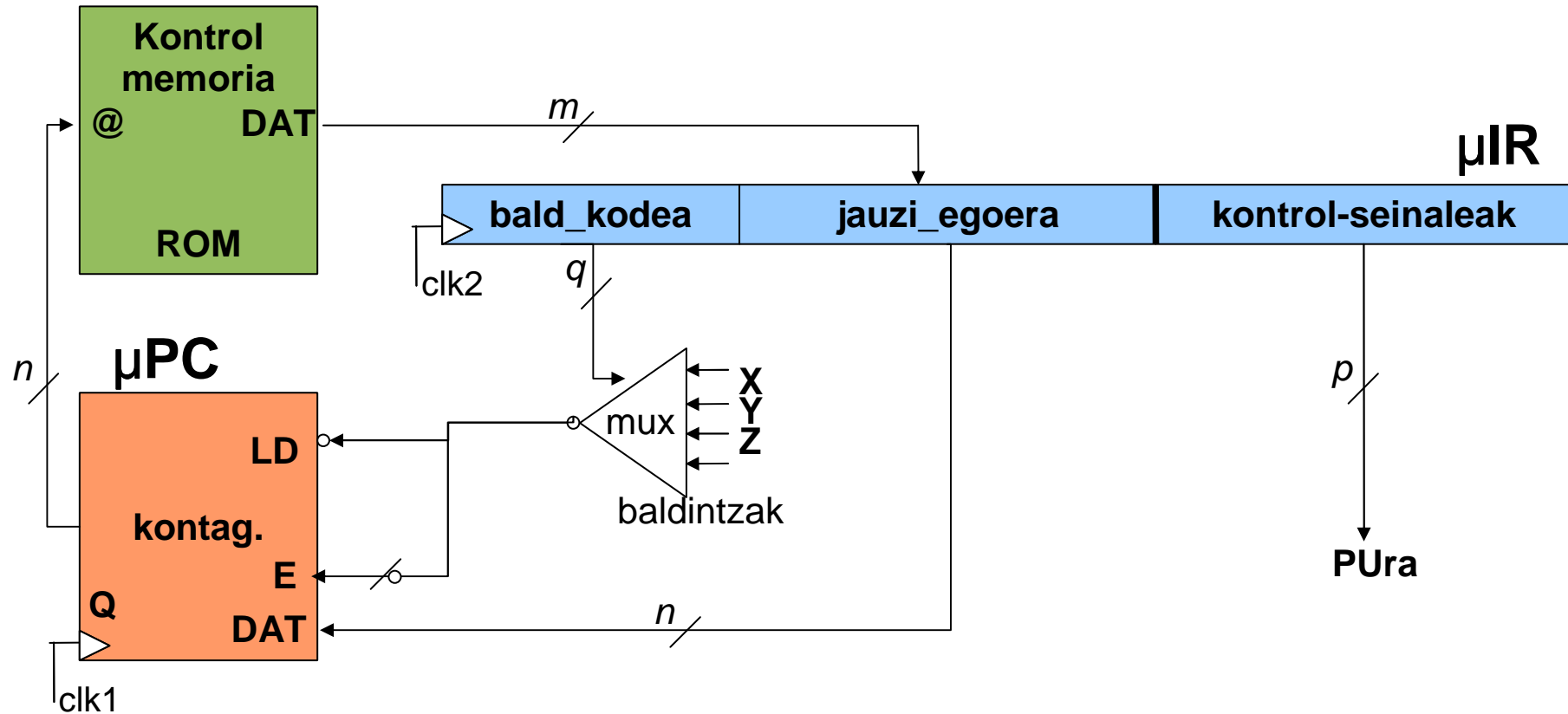


mikroprogramatutako KUaren egitura (1)



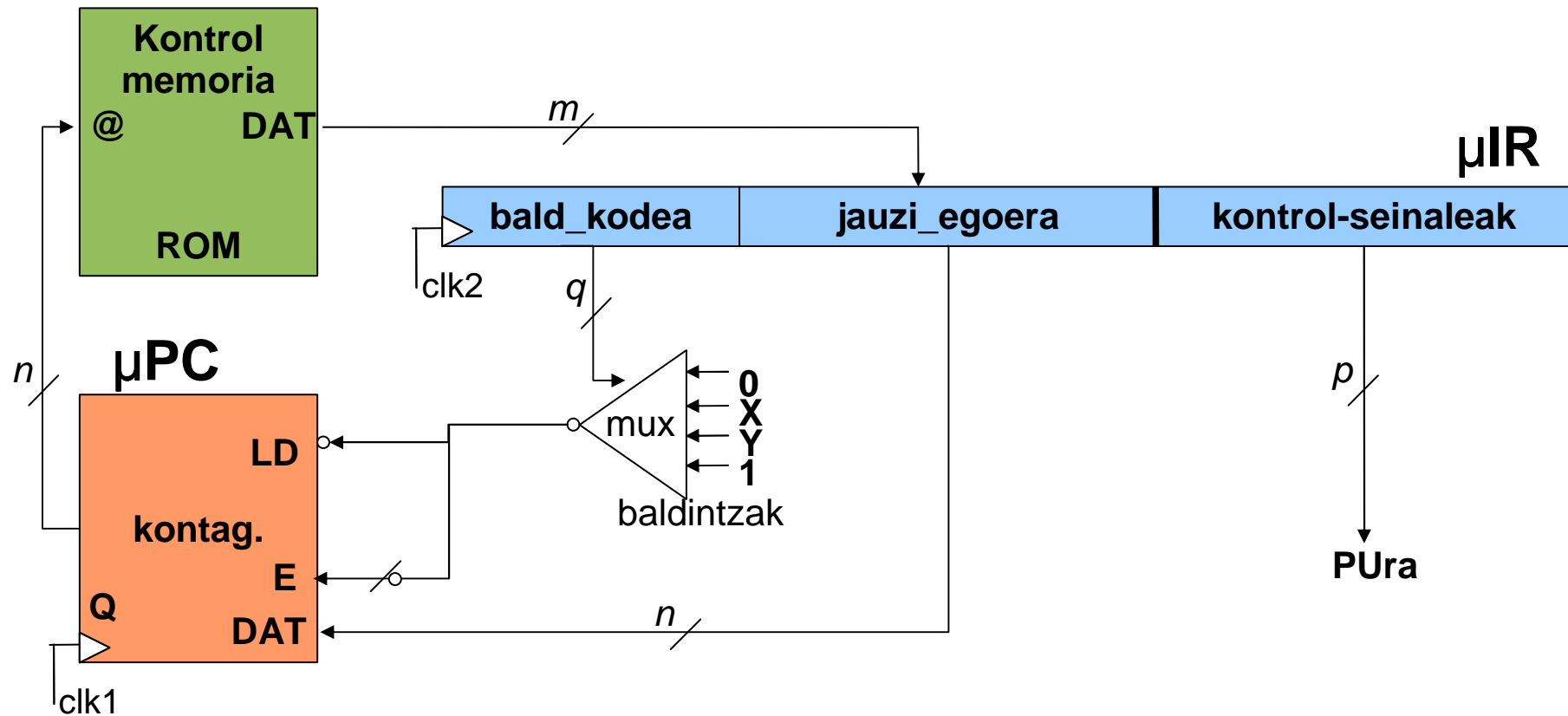
Beste aukera bat (2)

KU mikroprogramatuaren egitura (2)



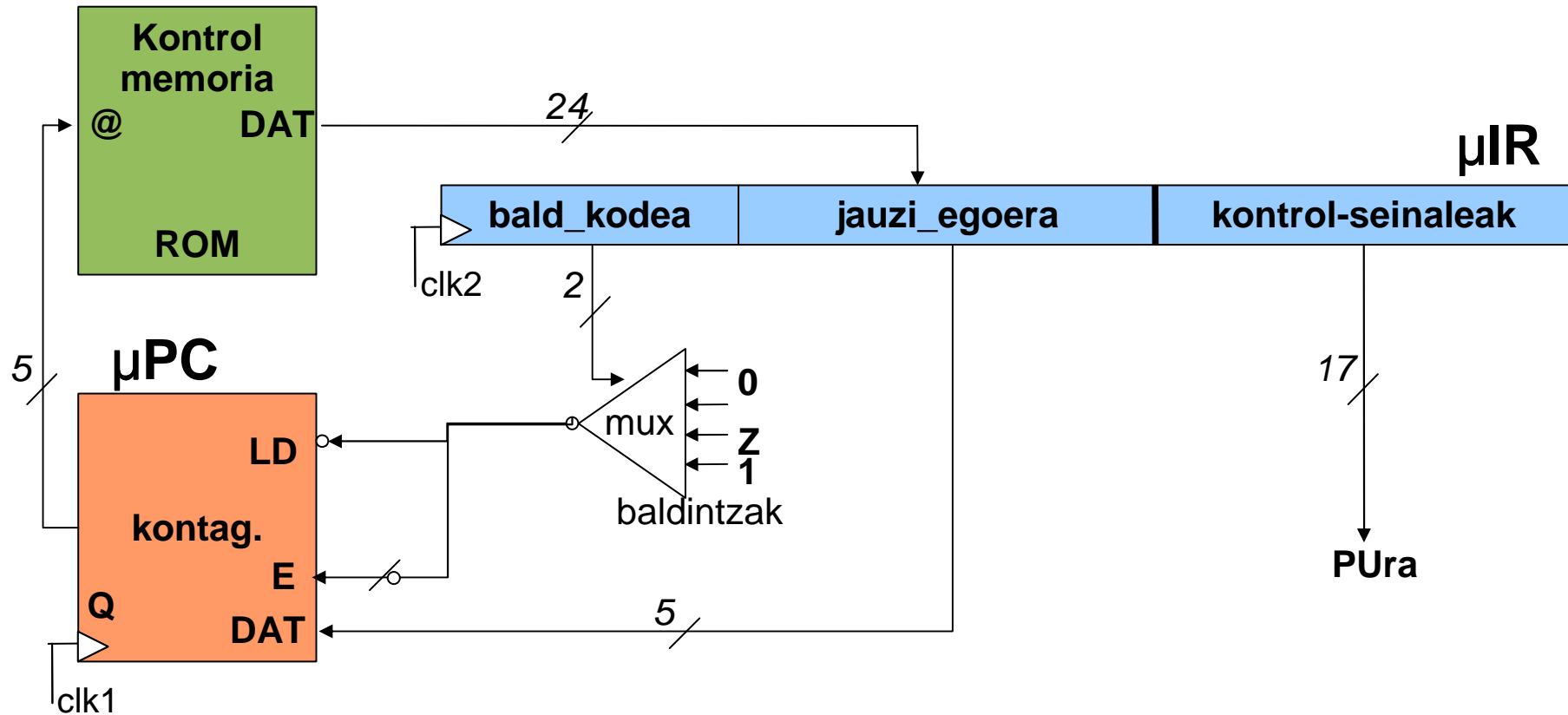
Oraindik beste aukera bat (3)

KU mikroprogramatuaren egitura (3)

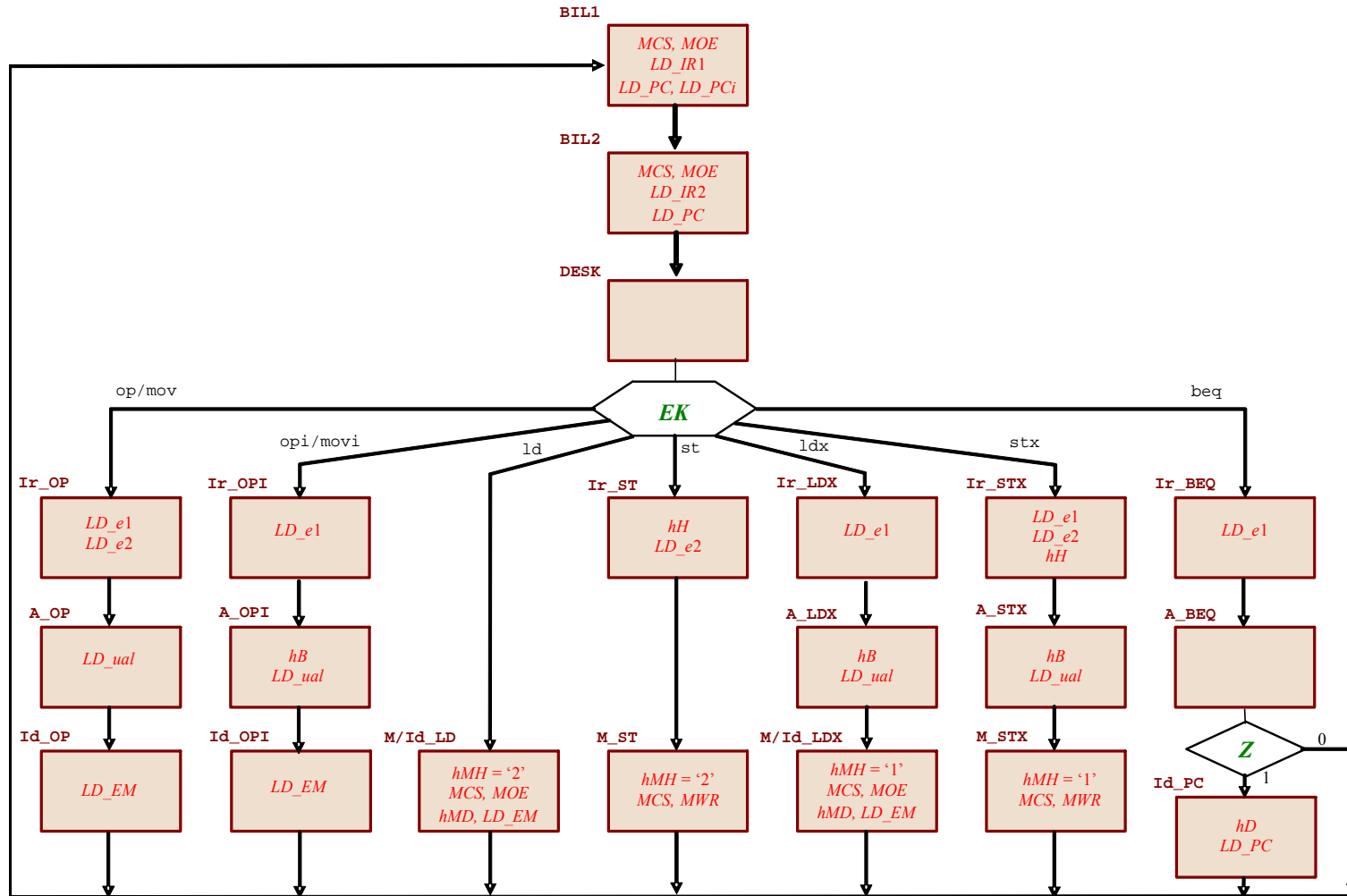


Gure aukera (4)

BIRDen KU mikroprogramatuaren egitura (4)



BIRD-en kontrol-algoritmoa



Mikroprograma

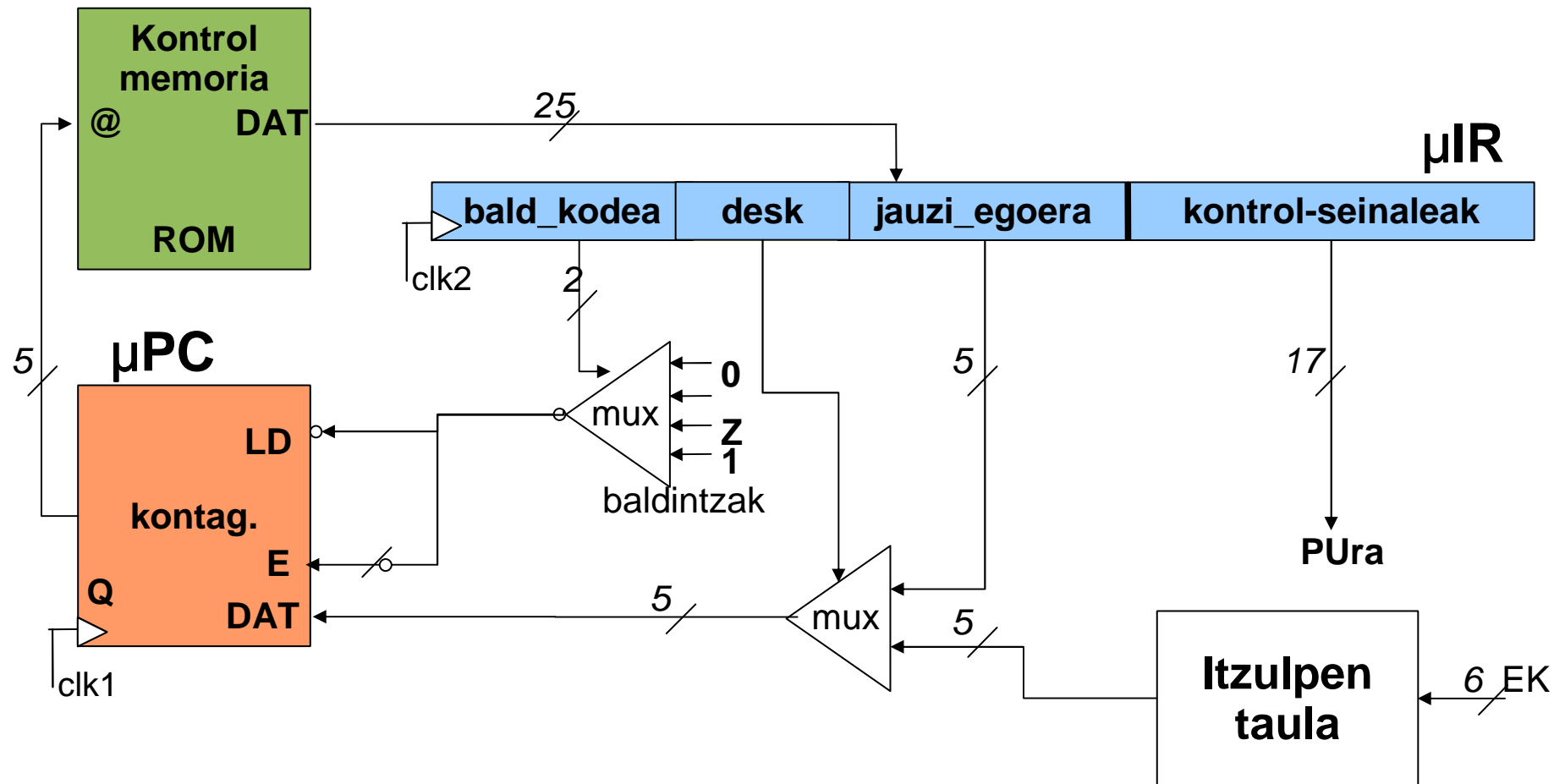
- Mikroagindu bakoitzean adierazi behar dena:
 - Kontrol-seinaleen balioak eta hurrengo egoera ondoz ondoko egoerara jauzi edo beste batera
 - ➡ Bitarrez: aspergarria
 - ➡ Mikromihiztadura lengoaia

Deskodetze egoera berezia da. Jauzi egoera ez da bakarra. Hori lortzeko, kontrol unitatean hardware gehiago:

itzulpen taula bat

EK → zein egoeratan hasten den bere exekuzioa.

Deskodetze arazoa konponduta



BIRDen MIKROPROGRAMA

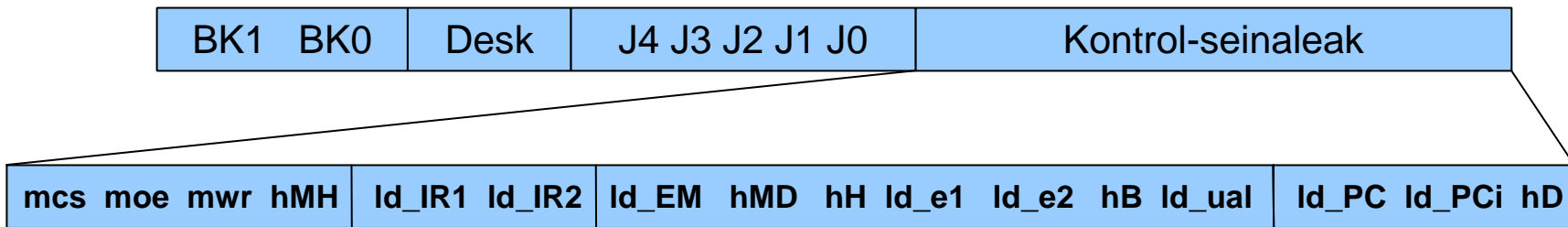
```
0: ir1:=mem(pc); pc:=pc+1; pci:=pc;          /*bil1*/
1: ir2:=mem(pc); pc:=pc+1;                  /*bil2*/
2: goto Eragiketa_kodea;                    /*desk*/
3: R_e1:=EM(Ri1); R_e2:=EM(Ri2);            /*op, mov*/
4: R_ual:=ual_kod(R_e1,R_e2);               /*ual-kod: ROM_UAL*/
5: EM(Rh):=R_ual; goto 0;
```

ITZULPEN-TAULA

Eragiketa-kodea → aginduaren exekuzio fasea hasten den kontrol-memoriako helbidea

E.K.	@hasiera	E.K.	@hasiera	E.K.	@hasiera
000000	01001(ld)	001010	00110(aggi)	010011	00011(and)
000010	01100(ldx)	001011	00011(sub)	010100	00110(andi)
000011	01010(st)	001100	00110(subi)	010101	00011(or)
000101	01111(stx)	001101	00011(mul)	010110	00110(ori)
000111	00011(mov)	001110	00110(muli)	010111	00011(xor)
001000	00110(movi)	001111	00011(div)	011000	00110(xori)
001001	00011(add)	010000	00110(divi)	011010	10010(beq)

BIRDen mikroaginduen formatua



BIRDen mikroprograma bitarrez

memoriako helbidea

edukia

(bil1)	00000	00 0	xxxxxx	1 1 0 00	1 0	0 x x 0 0 x 0	1 1 0
(bil2)	00001	00 0	xxxxxx	1 1 0 00	0 1	0 x x 0 0 x 0	1 0 0
(desk)	00010	11 1	xxxxxx	0 0 0 xx	0 0	0 x x 0 0 x 0	0 0 x
(op/mov)	00011	00 0	xxxxxx	0 0 0 xx	0 0	0 x 0 1 1 x 0	0 0 x
	00100	00 0	xxxxxx	0 0 0 xx	0 0	0 x x 0 0 0 1	0 0 x
	00101	11 0	00000	0 0 0 xx	0 0	1 0 x 0 0 x 0	0 0 x
(opi/movi)	00110	00 0	xxxxxx	0 0 0 xx	0 0	0 x x 1 0 x 0	0 0 x
...

Mikroprogramatutako diseinuaren abantailak eta desabantailak

- KUaren diseinua errazten du:
 - merkeagoa
 - erroreak izateko joera gutxiago
 - sistematikoagoa
 - erraz aldatzen da
- Agindu multzo desberdinak emulatzeko aukera ematen du
- Exekuzio motelagoa

Gehien bat CISC prozesadoreetan erabilia

Mikroprogramazio metodo alternatiboak

mikroprogramazio horizontala: mikroagindu bakoitzak, paraleloan exekutatu behar den mikroeragiketa asko adierazten ditu

17

Kontrol-seinaleak

mikroprogramazio bertikala: mikroagindu bakoitzak mikroeragiketa bakarra edo gutxi batzuk adierazten ditu kontrol seinale guztiek ez dute aldi berean eragiten: kodetu

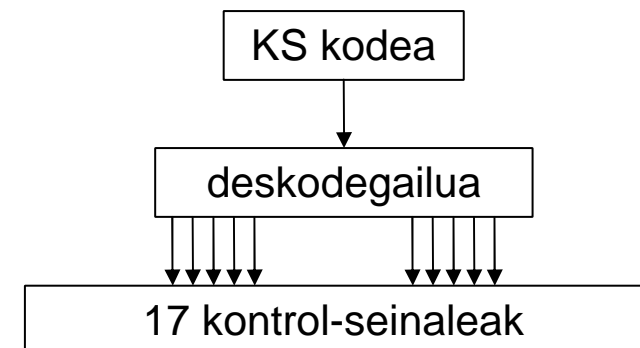
mikroprograma luzatzen da
paralelismoa ezinezkoa

5

KS kodea

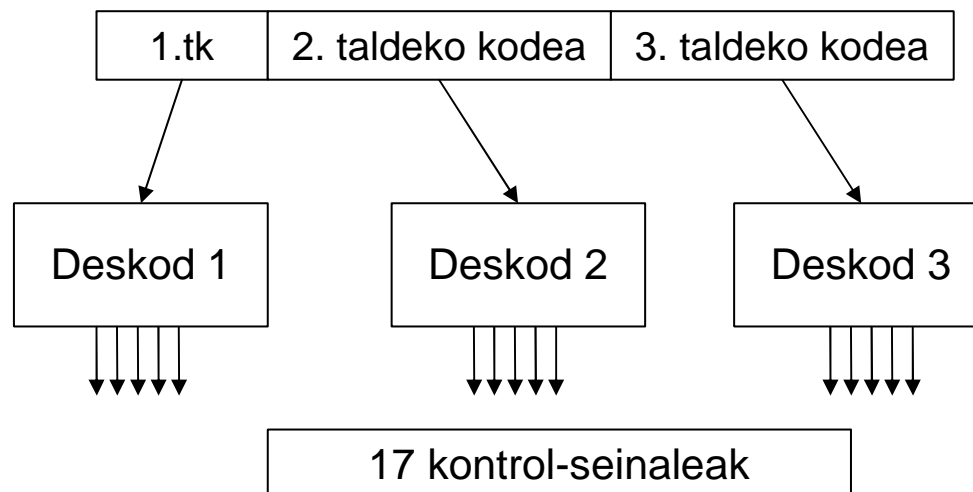
deskodetua

17 kontrol-seinaleak



tarteko soluzioa: mikroagindu bakoitzak mikroeragiketa bat baino behiago adierazten du

kontrol seinaleak multzo disjuntutan kodetzen dira



Nanoprogramazioa

nanoprogramazioa: mikroagindu bakoitza pausu segida baten bidez exekutatzen da: nanoagindua

⇒ nanoprograma

abantaila: kontrol memoriaren tamaina osoaren aurrezkoa

desabantaila: konplexutasun handiagoa eragiketa motelagoa