

SISTEMA DIGITALAK DISEINATZEKO OINARRIAK

OINARRIZKO PROZESADORE BATEN DISEINUA

7. GAIA

Bird, oinarrizko prozesadore bat

- Zer da **prozesadore** bat?
- Konputagailuaren atal bat da. Ingelesez CPU deritzo: *Central Processing Unit*.
- Programen (aginduen) exekuzioaz arduratzen da prozesadorea. Programak memorian metatuta egoten dira.
- Konputagailuaren beste atal batzuk: Memoria, Sarrera/Irteerako azpiatala (teklatua, pantaila, diskoa, sagua...)

Bird, oinarrizko prozesadore bat

- Prozesadore bat **diseinatzeko**, bi erabaki:
 - 1) **Agindu-multzoa** (*IS, Instruction Set*).
Prozesadore bakoitzak bere agindu-multzo propioa du, bere “lengoaia”. Prozesadore ezberdin asko eta fabrikatzaiete asko historian zehar: Intel, Motorola, DEC, IBM
 - 2) Prozesadorea osatuko duten **osagaiak** eta **haien arteko konexioak**
- Bi erabakiak lotuta daude: batak bestea baldintzatzen du.

Bird, oinarrizko prozesadore bat

- **Agindu-multzoa**

- Bird prozesadorearen agindu-multzoa dugu definituko dugu, azpimultzo bat, zehazki.
- Prozesadorearen agindu-multzoari “Makina Lengoaia” esaten zaio. Lengoaia bitarra da.
- Oro har, programak ez dira idazten Makina Lengoaiian, goi-mailako lengoaietan baizik: Ada, Java, C++...
- Programa horiek prozesadorearen makina lengoaiara itzuli behar dira:
konpiladoreak.

Bird, oinarrizko prozesadore bat

- **Agindu-multzoa**

- Makina lengoaia bitarrez definitzea oso astuna litzateke.
- Ordez, “**mihiztadura lengoaia**” erabiltzen da, mihiztadura lengoaiatik makina lengoaiara itzultzea oso simplea baita.
- Adibideak:

```
add r2,r3,r4
```

```
sub r15,r2,r1
```

```
ld r3,A
```

- **Agindu aritmetiko-logikoak**

- 7 agindu aritmetiko-logiko, 3 eragingaikoak.
Eragingai guztiak erregistro-multzoan (EM-an):
helburu-erregistro bat eta 2 iturburu-erregistro.
- Formatua: ***op rh,ri1,ri2***. Adibideak:

add r2,r3,r4

sub r15,r2,r2

mul r19,r2,r1

div r15,r25,r11

and r4,r4,r4

or r3,r29,r29

xor r31,r20,r13

- **Erregistro-multzoari buruz (EM):**
 - 32 erregistro: r0tik r31ra arte.
 - 16 biteko erregistroak.
 - **r0** erregistroa **berezia** da: 0 datu konstantea du eta ezin da aldatu (ezin da, beraz, helburu-erregistro gisa erabili).

- **Agindu aritmetiko-logiko gehiago**
 - Beste 7 agindu aritmetiko-logiko, 3 eragingaikoak. Eragiketa berberak baina eragingai bat **berehalakoa** (beste biak EM-an).
 - Formatua: ***opi rh,ri1,dat***. Adibideak:

addi r2,r3,#8	subi r15,r2,#-4
muli r19,r2,#-2	divi r7,r25,#12538
andi r4,r4,#B100000000000000	
ori r3,r29,#XF7D0	xori r31,r20,#xFFFF

- **Berehalako datuak**

- 16 biteko konstanteak
- Zenbaki gisa interpretatuz gero, tartea:
[- 2^{15} , + 2^{15} -1] (2rako osagarria)
- Balioa hamartarrez, bitarrez edo hamaseitarrez adieraz daiteke (ikus aurreko adibideak).

- **Memoriako aginduak I**
 - 2 agindu memorian **irakurtzeko: Id eta Idx**
 - 1) Formatua: *Id rh,aldaia*. Adibidea:
`ld r3,A`
 - 2) Formatua: *Idx rh,aldaia[ri]*. Adibidea:
`ldx r15,A[r1]`

- **Memoriako aginduak II**
 - 2 agindu memorian **idazteko**: **st** eta **stx**
 - 1) Formatua: *st ri,aldaia*. Adibidea:
st r7,BATURA
 - 2) Formatua: *stx ri1,aldaia[ri2]*. Adibidea:
stx r15,A[r1]

- **Datuak mugitzeko aginduak**
 - 2 agindu: **mov** eta **movi**.
 - Formatua: *mov rh, ri; movi rh,#dat*
 - Adibideak:
`mov r3,r8`
`movi r11,#XA810`
- **Jauzi-aginduak** (bakarra ikusiko dugu)
 - Formatua: **beq ir,etiketa**
`beq r15,baldin`

Programa baten adibidea:

```
{bi zenbakiren arteko handiena kalkulatzen duen programa}

.title adibide1

X:      .value 15;
Y:      .value 16;
HAND:   .word 1;

.proc main

    ld      r2,X
    ld      r3,Y
    sub    r1,r2,r3

    andi   r1,r1,#B10000000000000000000000
    beq    r1,LEH
    mov    r10,r3
    beq    r0,BUK
    mov    r10,r2
    BUK:   st     r10,HAND

    retm
.endp main

.end|
```

Beste adibide bat:

```
.title FAKTURA

PREZIO: .word 1;
BEZ: .value 21;
EURO: .word 1

.proc main

    outs "sakatu salneurria:"
    in
    st r1, PREZIO

    ld r1,BEZ
    ld r2,PREZIO

    mul r3,r1,r2
    divi r4,r3,#100
    add r4,r2,r4
    st r4,EURO

    outs "Azken prezioa hau da: "
    mov r1,r4
    out

    retm
.endp main

.end
```

SISTEMA DIGITALAK DISEINATZEKO OINARRIAK

OINARRIZKO PROZESADORE BATEN DISEINUA

7. GAIA

BIRD, oinarrizko prozesadore bat

- **Aginduen kodeketa bitarra**
 - 21 agindu ezberdin definitu ditugu orain arte. Gehiago ditu Bird-ek: 36.
 - Bitarrez, nolakoa da bakoitzak?
 - Nola “berezituko” ditugu aginduak haien artean?
 - Agindu bakoitzak identifikadore bat izango du, “ERAGIKETA-KODE” ezberdin bat, hasierako 6 biten bitartez adieraziko dena.

BIRD, oinarrizko prozesadore bat

- Aginduen eragiketa-kodea

Agindua	Eragiketa kodea (hamartarrez)	Eragiketa kodea (bitarrez)	Agindua	Eragiketa kodea (hamartarrez)	Eragiketa kodea (bitarrez)
ld	0	000000	ldx	2	000010
st	3	000011	stx	5	000101
mov	7	000111	movi	8	001000
add	9	001001	addi	10	001010
sub	11	001011	subi	12	001100
mul	13	001101	muli	14	001110
div	15	001111	divi	16	010000
and	19	010011	andi	20	010100
or	21	010101	ori	22	010110
xor	23	010111	xori	24	011000
			beq	26	011010

BIRD, oinarrizko prozesadore bat

- **Aginduen gainontzeko informazioa**
 - Erregistroak identifikatzeko: 5 bit
 - ***op*** motako aginduak:
 - EK: 6 bit, rh: 5 bit, ri1: 5 bit, ri2: 5 bit
 - Guztira: 21 bit
 - ***opi*** motako aginduak:
 - EK: 6 bit, rh: 5 bit, ri1: 5 bit, berehalako datua: 16 bit
 - Guztira: 32 bit

BIRD, oinarrizko prozesadore bat

- **Aginduen gainontzeko informazioa**
 - **mov** agindua:
 - EK: 6 bit, rh: 5 bit, ri1: 5 bit.
Guztira: 16 bit
 - **movi** agindua:
 - EK: 6 bit, rh: 5 bit,
berehalako datua: 16 bit. Guztira: 27 bit
 - **ld/st** aginduak:
 - EK: 6 bit, rh/ri: 5 bit, aldagaiaren helbidea:
16 bit. Guztira: 27 bit

BIRD, oinarrizko prozesadore bat

- **Aginduen gainontzeko informazioa**
 - **Idx/stx** aginduak:
 - EK: 6 bit, rh/ri1: 5 bit, aldagaiaren helbidea: 16 bit, ri/ri2: 5 bit.
Guztira: 32 bit
 - **beq** agindua:
 - EK: 6 bit, ri: 5 bit, helburuko agindua (etiketa duena): 16 bit.
Guztira: 27 bit
 - Simplifikatzeko: **agindu guztiak 32 bitekoak**

BIRD, oinarrizko prozesadore bat

op rh, ri1, ri2

EK	rh	ri1	ri2
6	5	5	11

opi rh, ri1, #bereh

EK	rh	ri1	berehalakoa
6	5	5	16

mov rh, ri1

EK	rh	ri1	
6	5	5	16

movi rh, #bereh

EK	rh		berehalakoa
6	5	5	16

ld rh, ALD

EK	rh		helbidea
6	5	5	16

ldx rh, ALD [ri1]

EK	rh	ri1	helbidea
6	5	5	16

st ri2, ALD

EK	ri2		helbidea
6	5	5	16

stx ri2, ALD[ri1]

EK	ri2	ri1	helbidea
6	5	5	16

beq ri1, etiketa

EK		ri1	desplazamendua
6	5	5	16

SISTEMA DIGITALAK DISEINATZEKO OINARRIAK

OINARRIZKO PROZESADORE BATEN DISEINUA

7. GAIA

BIRD, osagaiak

► Oinarrizko osagaiak

- Erregistro-multzoa.
- Unitate aritmetiko-logikoa.
- Memoria. Ez da prozesadorearen zati bat, baina memoria atzitu behar duenez, memoriaren hainbat ezaugarri finkatu beharra dago.

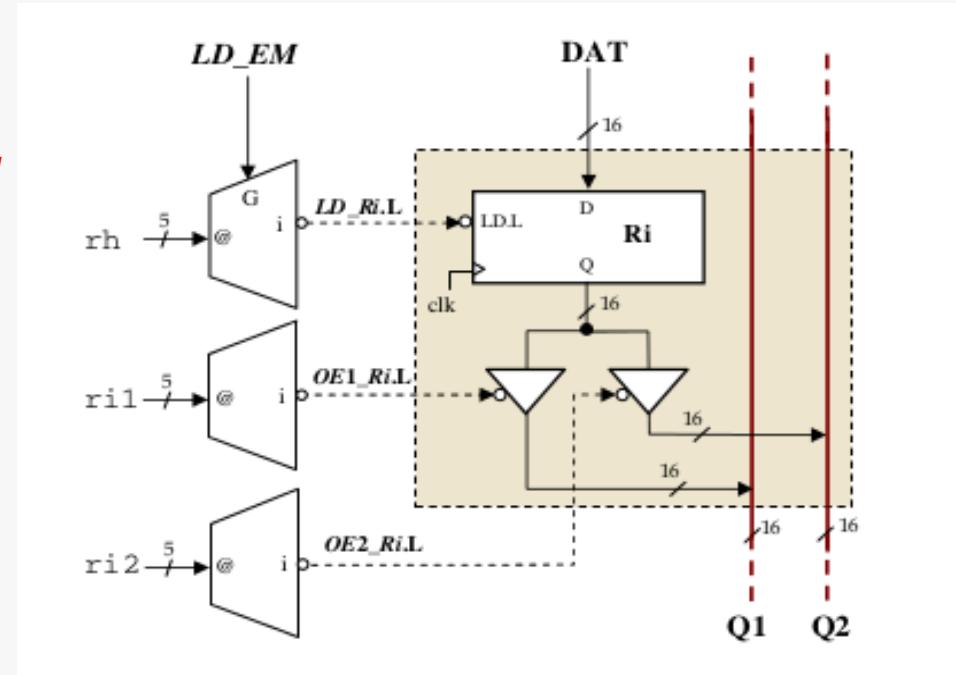
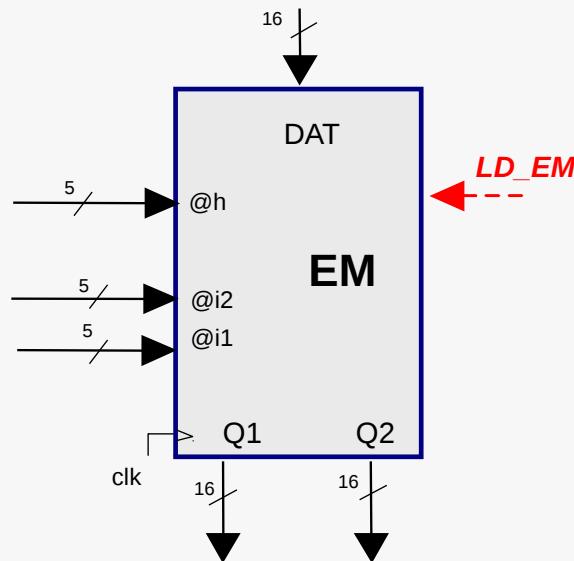
BIRD, osagaiak

► Erregistro-multzoa (EM)

- 16 biteko 32 erregistro.
- r0 erregistroak 0 datua du, beti.
- Erregistro bat helbideratzeko: 5 bit.
- Eragiketa bat baino gehiago egin daitezke aldi berean:
 - * Bi erregistro irakurri. Beti irakurtzen.
 - * Erregistro batean idatzi. Idazteko: LD_EM seinalea.
- Hiru helbide-sarrera: @h, @i1, @i2.

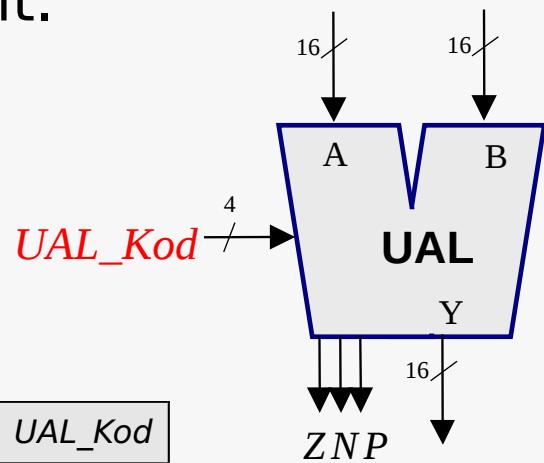
BIRD, osagaiak

► Erregistro-multzoa (EM)



► Unitate Aritmetiko Logikoa (UAL)

- Eragigaiak eta emaitzak: 16 bit.
- Eragiketa-kodea: 4 bit.
- Hiru adierazle: Z, N eta P.

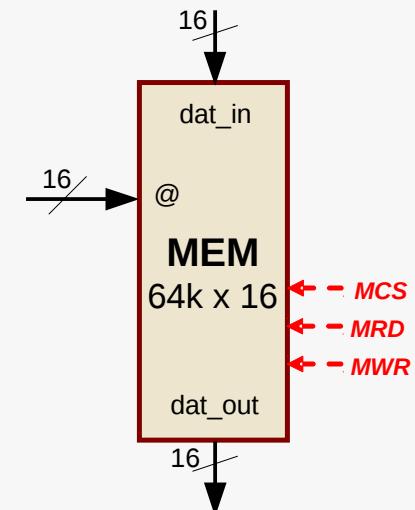


Y	UAL_Kod	Y	UAL_Kod	Y	UAL_Kod
$A + B$	0000	A	0100	$A \text{ or } B$	1000
$A - B$	0001	B	0101	$A \text{ and } B$	1001
$A \times B$	0010	$A + 1$	0110	$A \text{ xor } B$	1010
A / B	0011	$A - 1$	0111	Desp_ezkA	1011
				Desp_esk A	1100

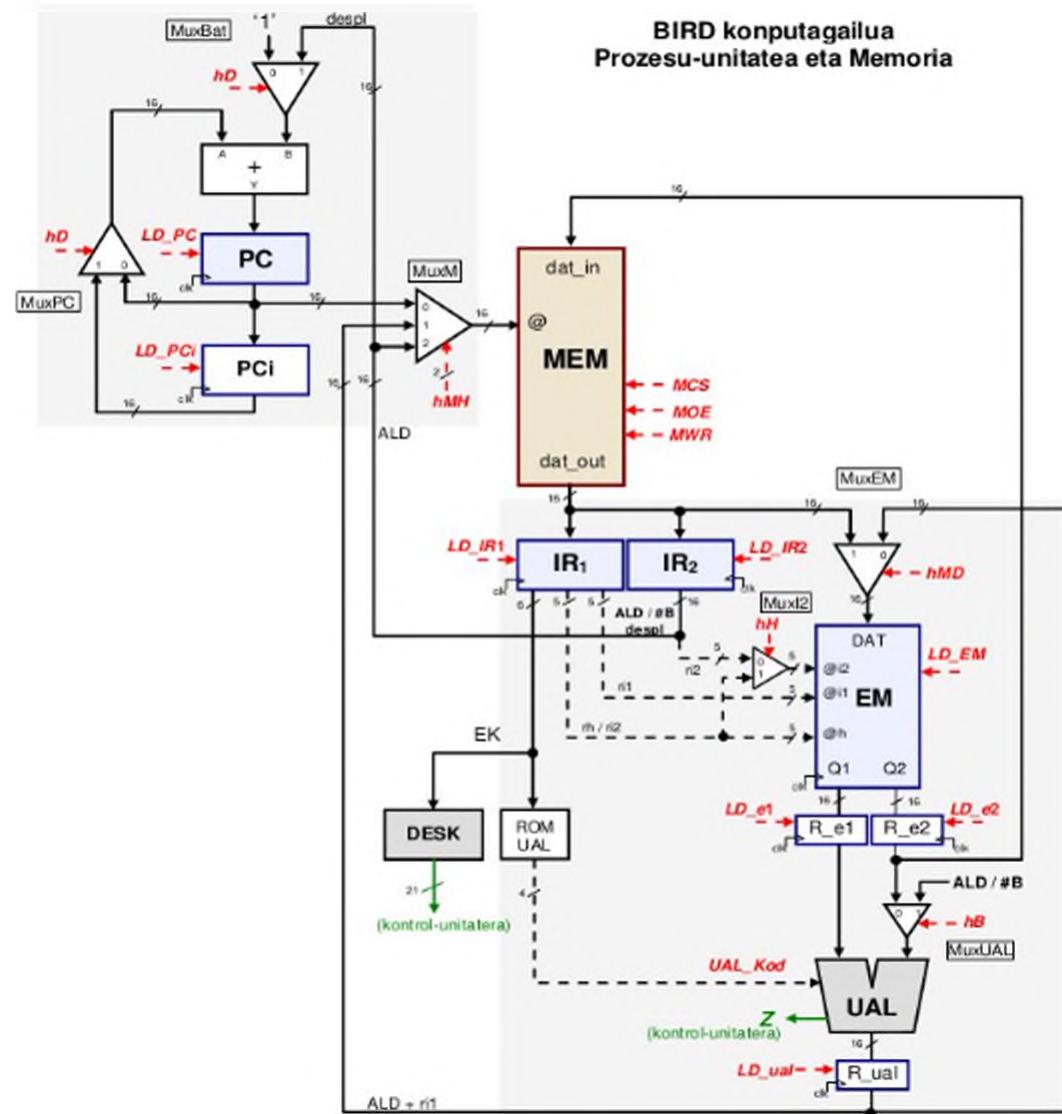
BIRD, osagaiak

► Memoria nagusia (RAM)

- Edukiera: 64 k x 16.
- Helbidea: 16 bit.
- Datua (hitza) in/out: 16 bit.
- Datu-irteera (dat_out): hiru-egoerakoa



BIRD konputagailua Prozesu-unitatea eta Memoria



SISTEMA DIGITALAK DISEINATZEKO OINARRIAK

BIRD PROZESADOREAREN KONTROL-UNITATEA

7 GAIA

BIRD prozesadorearen KU

- Aginduak exekutatzeko, prozesadoreak behin eta berriro errepikatzen ditu aginduen exekuzio-faseak: bilaketa, deskodeketa, gainontzeko urratsak.
- ASM algoritmoak exekuzio-fase horiek kontrolatu behar ditu:
 - * Bilaketa (agindu guztiekin)
 - * Deskodeketa (agindu guztiekin)
 - * Gainontzeko urratsak (aginduaren araberakoak)

BIRD prozesadorearen KU

► Aginduaren bilaketa (*fetch*)

- Egin beharrekoak:
 - * Agindua *irakurri*: 2 hitz ($PC=PC+1$)
 - * Agindua *gorde*: IRn (IR1 eta IR2)
- Automatak: bi egoera, 2 hitz direlako
- Bestelakoak: aginduaren hasierako helbidea gorde PCI erregistroan (jauzi bat bada beharko da agian)

BIRD prozesadorearen KU

Bilaketa fasea: 1. egoera

- **BIL1** egoera:

$IR1 := MEM[PC]; PC := PC+1; PCI := PC;$

BIL1

MEM $\rightarrow MCS, MOE$	$/* hMH = '0' \rightarrow MEM[PC] */$
$IR_1 \rightarrow LD_IR1$	
PC $\rightarrow LD_PC$	$/* hD = 0 \rightarrow PC + 1 */$
$PCI \rightarrow LD_PCI$	

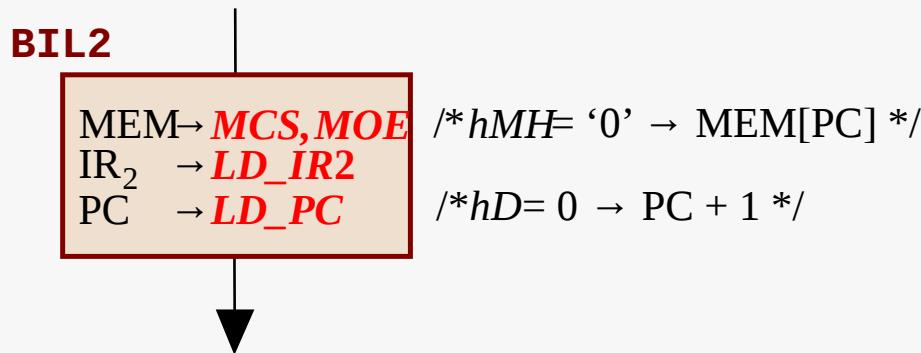


BIRD prozesadorearen KU

Bilaketa fasea: 2.egoera

- **BIL2** egoera:

$IR2 := MEM[PC]; PC := PC+1;$

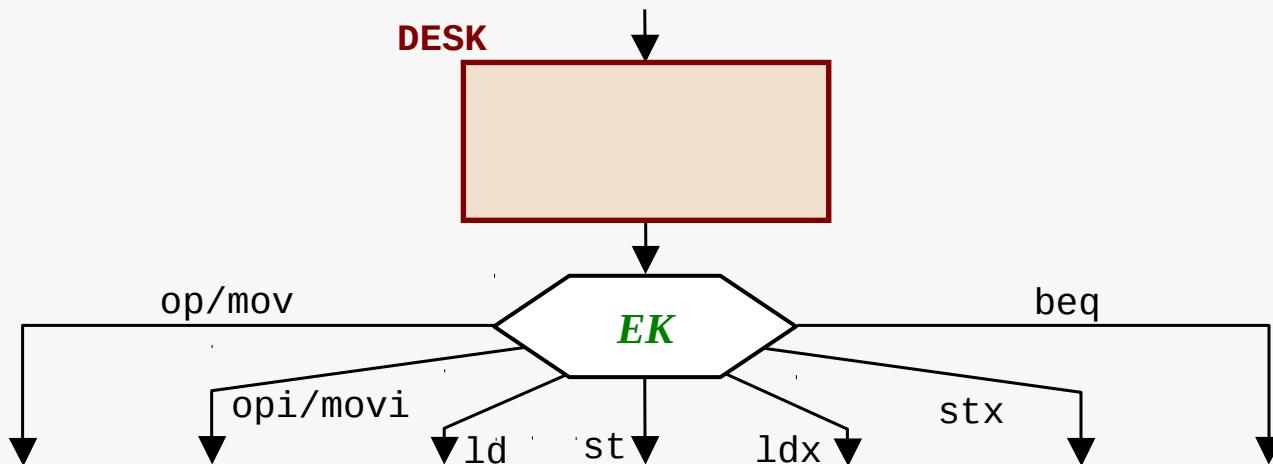


BIRD prozesadorearen KU

Deskodeketa: 3.egoera

► Deskodeketa fasea

- PUko deskodegailuaren arabera, hainbat adar algoritmoan



BIRD prozesadorearen KU **Exekuzioaren kontrola: op/mov**

3 egoera:

Ir_OP

- * Irakurri eragingaiak (ri1/ri2)
- * Gorde R_e1 eta R_e2 erregistroetan

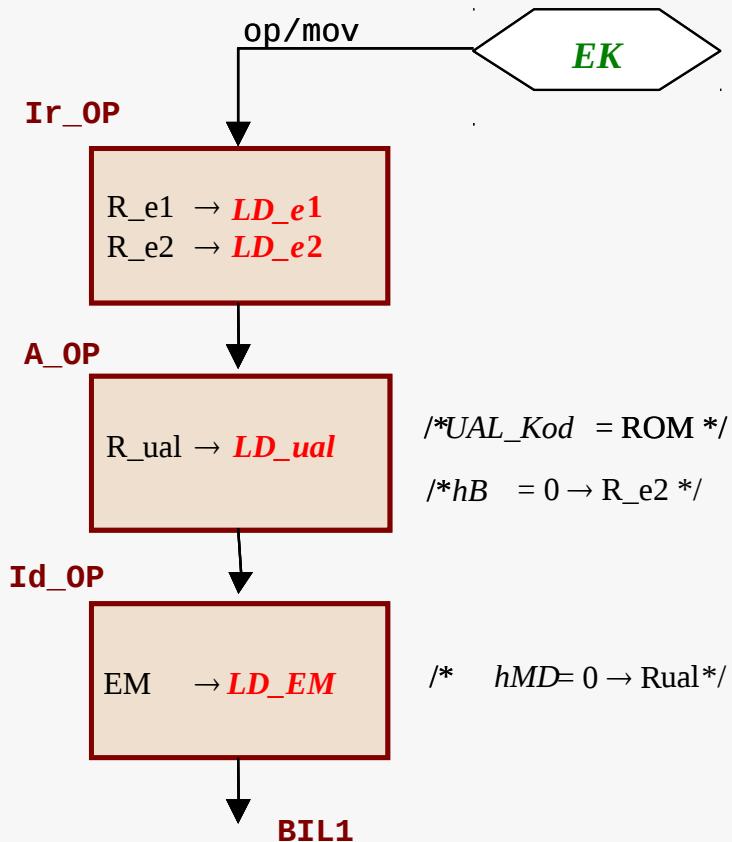
A_OP

- * Eragiketa UALean; emaitza gorde: R_ual

Id_OP

- * Gorde emaitza EMko rh erregistroan

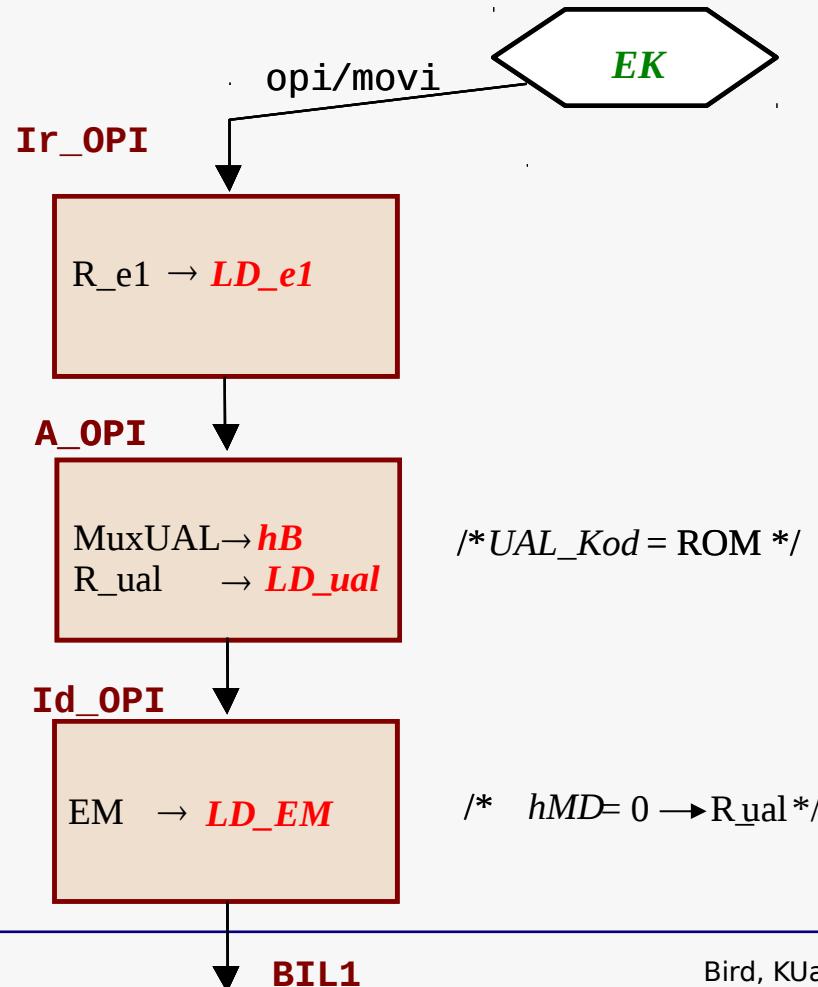
BIRD prozesadorearen KU Exekuzioaren kontrola: op/mov



BIRD prozesadorearen KU Exekuzioaren kontrola: opi/movi

opí/movi:

2. eragigaia IR2n

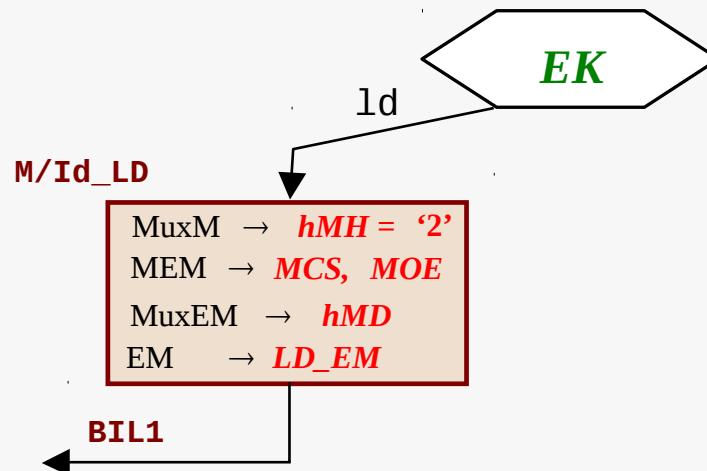


BIRD prozesadorearen KU Exekuzioaren kontrola: Id

Egoera bakar batean:

M/Id_LD

- * Irakurri memorian IR2k adierazitako helbidean eta irakurritakoa rh-n gorde



BIRD prozesadorearen KU Exekuzioaren kontrola: st

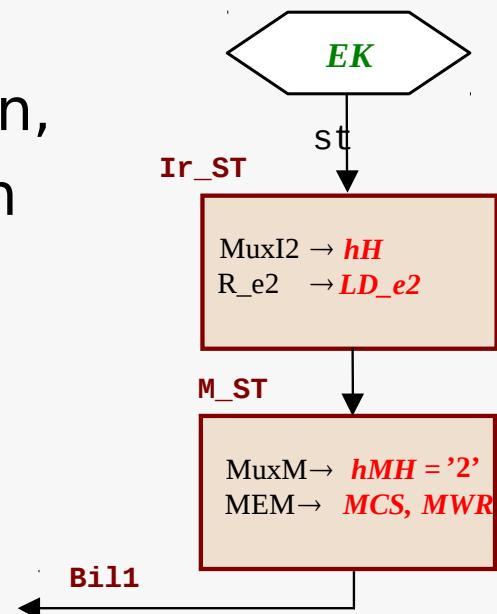
Bi egoera:

Ir_ST

- * ri2 erregistroa irakurri eta R_e2-n gorde

M_ST

- * Idatzi R_e2ko edukia memorian,
IR2k adierazten duen helbidean



BIRD prozesadorearen KU

Exekuzioaren kontrola: Idx

Hiru egoera:

Ir_LDX

- * ri1 irakurri eta gorde R_e1 erregistroan

A_LDX

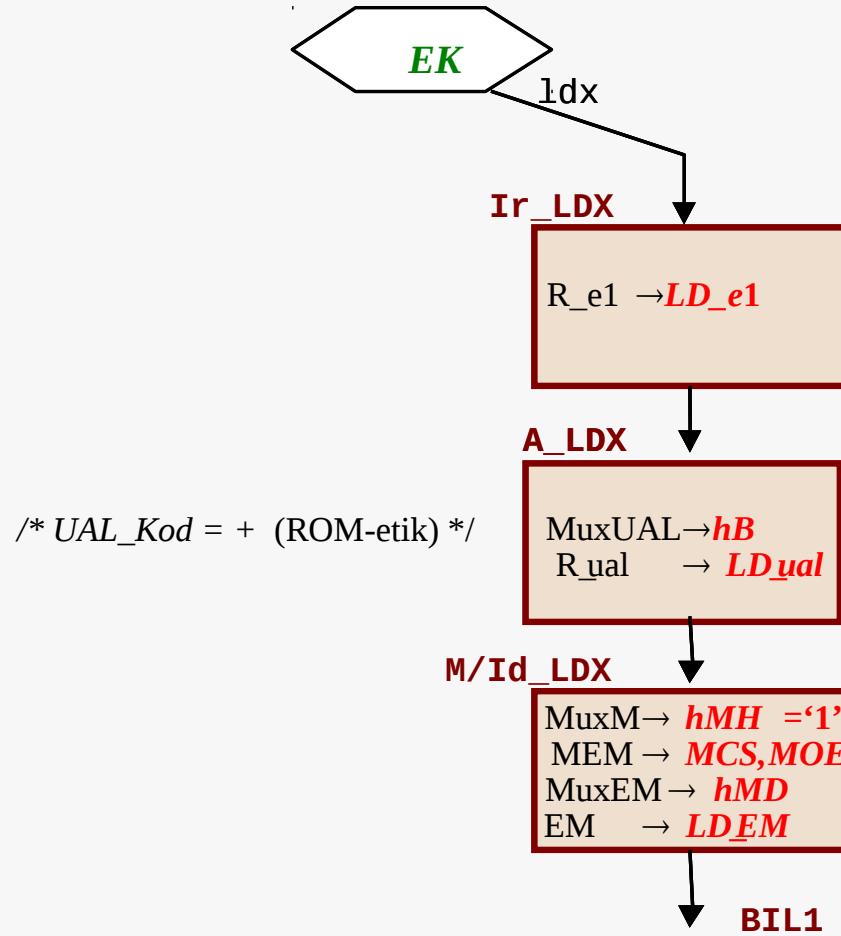
- * UALean batuketa egin helbidea lortzeko
(R_e1+IR2)

- * Helbidea lortuta, gorde R_ual erregistroan

Id/M_LDX

- * Irakurri memorian R_ual-ek adierazten duen
helbidean eta irakurritakoa gorde rh-n

BIRD prozesadorearen KU Exekuzioaren kontrola: idx



BIRD prozesadorearen KU **Exekuzioaren kontrola: stx**

Hiru egoera

Ir_STX

- * ri1 irakurri eta gorde R_e1-en
- * ri2 irakurri eta gorde R_e2-n

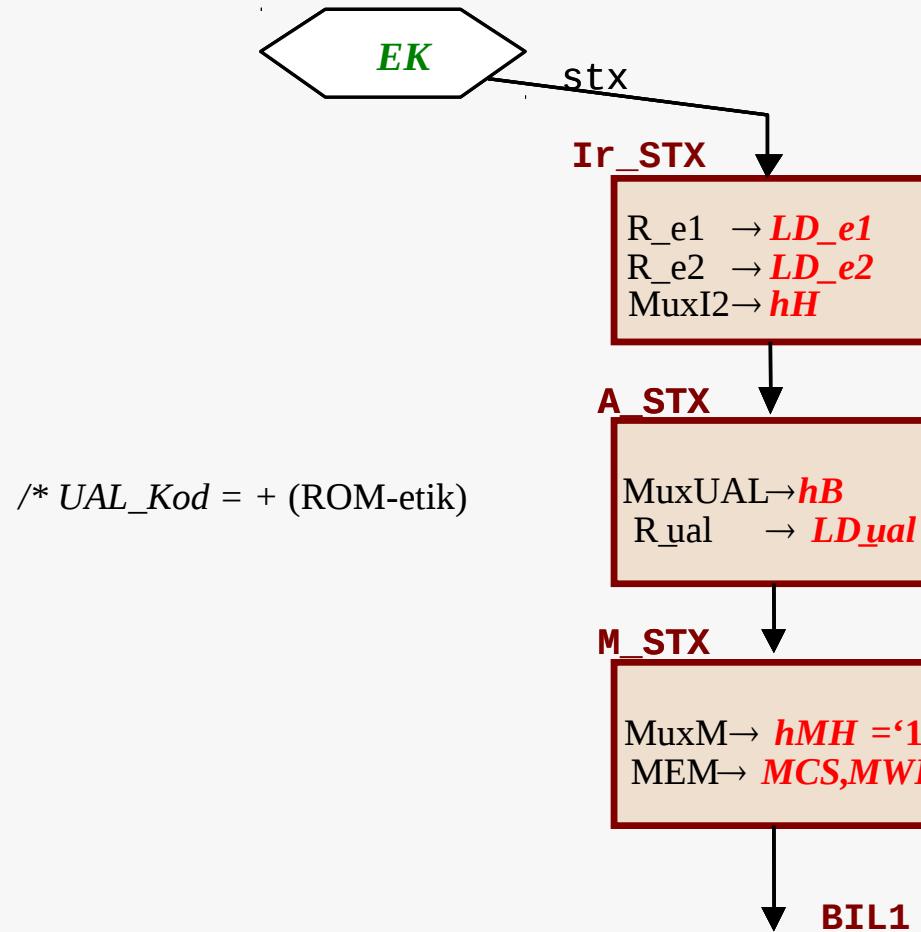
A_STX

- * UALean batuketa egin helbidea lortzeko
(R_e1+IR2) eta gorde R_ual-ean

Id/M_STX

- * Idatzi memorian R_e2ko datua R_ual-ek
adierazitako helbidean

BIRD prozesadorearen KU Exekuzioaren kontrola: stx



BIRD prozesadorearen KU **Exekuzioaren kontrola: beq**

Bi edo hiru egoera (Z seinalearen arabera)

Ir_BEQ

- * ri1 irakurri eta gorde R_e1-en

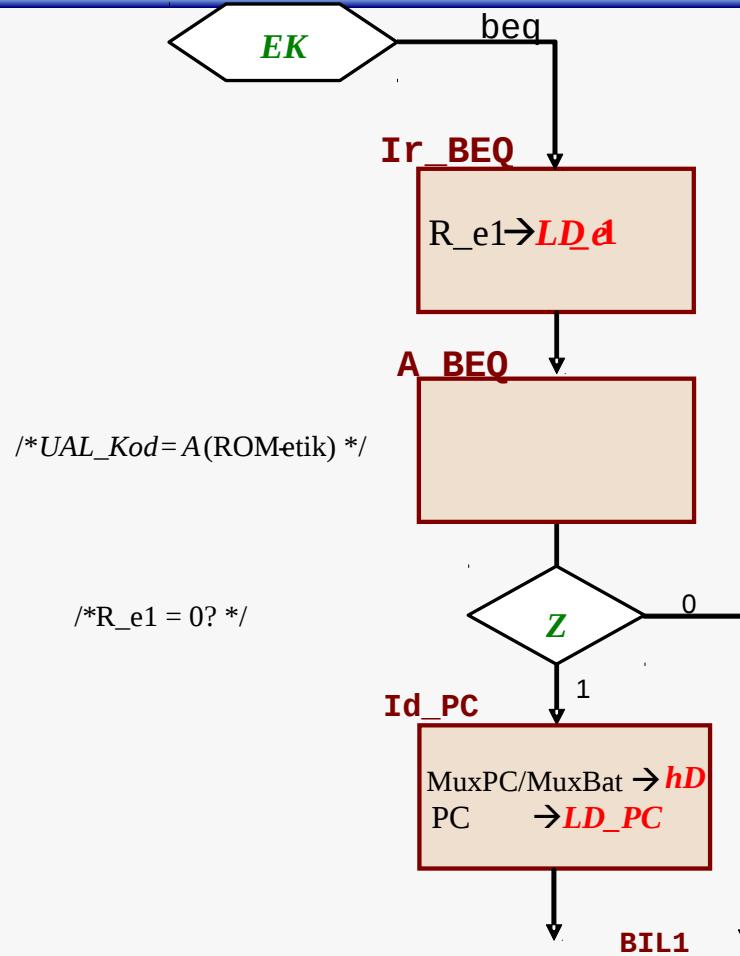
A_BEQ

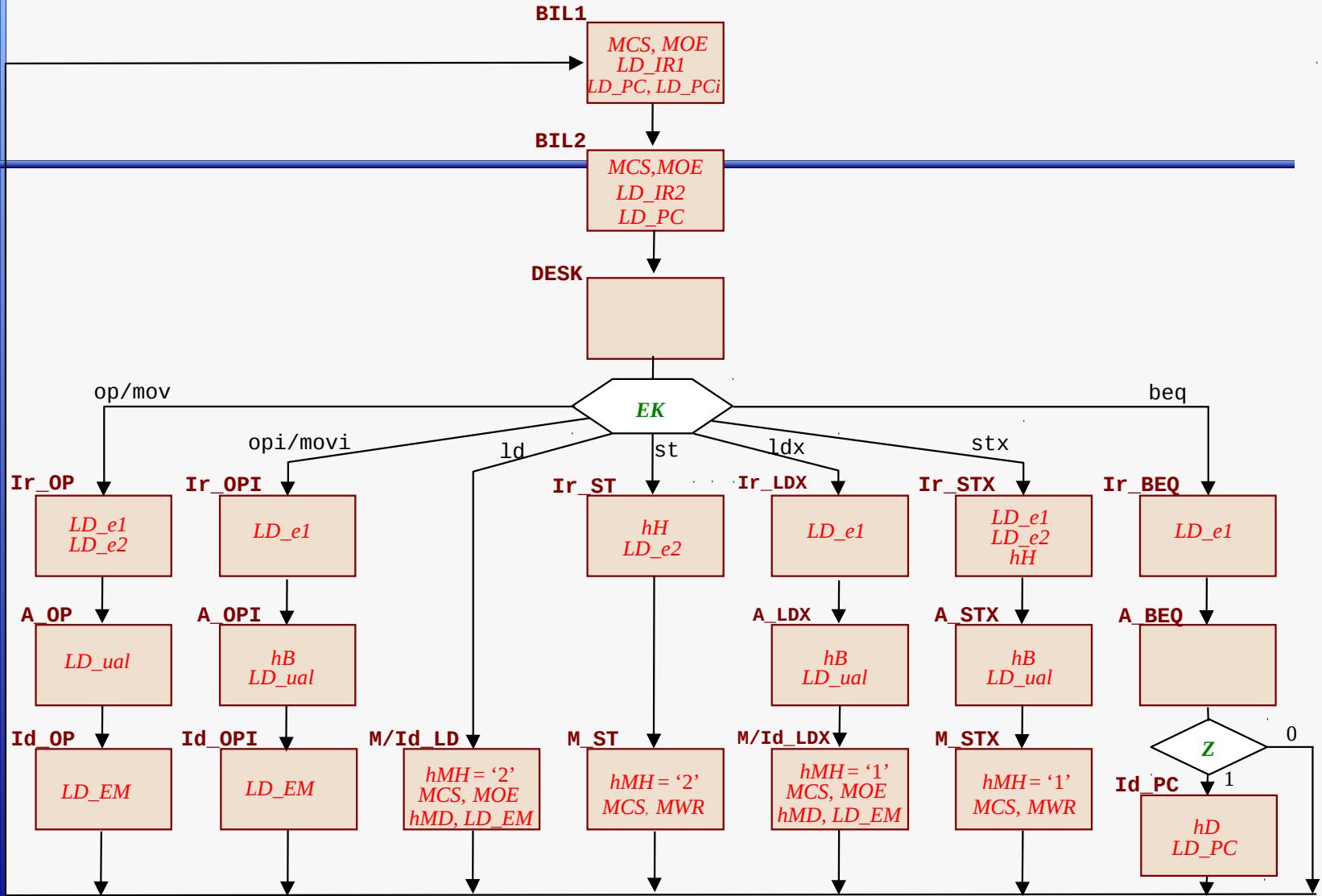
- * UALean “konparaketa” (erregistroa 0a den)
- * Aztertu Z adierazlea hurrengo egoera erabakitzeko

Id_PC (Z = 1 denean soilik)

- * Batu desplazamendua (IR2) eta PCi eta gorde emaitza PC-n jauzia burutu dadin

BIRD prozesadorearen KU Exekuzioaren kontrola: beq





BIRD prozesadorearen KU **Zenbait iruzkin**

- Etengabe exekutatzen da algoritmoa
- Hasieraketa eta bukaera
- Bird prozesadoreak agindu gehiago ditu
- Programa baten exekuzio-denbora

BIRD prozesadorearen KU Zenbait iruzkin

Agindu-mota bakoitzaren ziklo-kopurua:

op/mov/opi/movi	→	B₁	B₂	D	Ir	A	Id	6 ziklo
Id	→	B₁	B₂	D	-	-	MId	4 ziklo
st	→	B₁	B₂	D	Ir	-	M	5 ziklo
Idx/stx	→	B₁	B₂	D	Ir	A	MId	6 ziklo
beq	→	B₁	B₂	D	Ir	A	(Id)	5/6 ziklo

BIRD prozesadorearen KU Zenbait iruzkin

Kontrol-unitatearen eraikuntza

