

## 7. gaia. Oinarri zilto prozesadore baten diseinua

Sistema digital garrantzitsuenak, konputagailuak dira, eta ez dira erabilizten eginikizun jalkin bat egitello, ondorioz, helburu orokorreko sistema digitalak dira.

Aginduak selventzietan antolatuak, eta datuak batera, programak sortzen dira, eta aginduak eta datuak aldatuz, programak desberdinak sortu daitezke, eta horrela gauza desberdinak egitea lortu.

Konputagailuak bi osagai garrantzitsuz daude osatuta:

- Osagai fisikoa: hardwarea
- Programak: softwarea.

Prozesadore baten hardwarean, procesu-unitateen zein kontrol-unitatean oiharriztu da.

### Oinarriko Konzeptuak

Konputagailuak sistema digital oso kompleksoak direnez, arpisistemak banatzen dira, von Neumann-en arkitektura erabiliz.

Hiru arpitalede bereizten dira:

- Prozesadorea: "CPU": Programatzeko aginduak eta eragigaiak irakurri eta prozesatzeko aktibitateak.
- Memoria: Bertan hartzeko prozesatu beharreko agindu eta datuak, eta emaitza bertan idazten diren.
- Sarsegia/irteerako arpisistema: Konputagailua eta koppa-adolako informazioa elkarbundtzeari arduratzeko diren.

- Busotz: Komputagailen osagai garrantzitsua kontzientziaz arduratzen dira.

### Prozesadoreen Osagai haugusikoa

eta Prozesadorea diseinatzeko orduan, bi bantek, Kontrol-unitatea eta prozesu-unitateak.

### Prozesu-unitatearen gaiaak:

- Programa-montagailua: "PC" Exekutatu beharreko aginduen gordetzen du.
- Agindu-erregistroa: "IR" Exekutatzera ari den agindua gordetzen du.
- Erregistro multzoa: "EM" Erregistroak memoria triliko osagaiak dira, beira oso adarrak, beraez bortean gordetzen dira aldi baterako eragigaiak, emaitza...
- Unitate aritmetiko-logikoa: "UAL": Aintzirako eragileko aritmetiko, zein logikako egiteak arduratzen da.

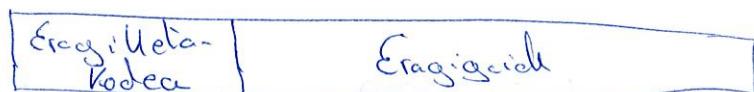
### Prozesadorearen aginduak

Komputagailu bakoitzak, dituen agindu-unitateen bidez, makina-lengoaia definitibo du. Aginduak exekutatzeko eszikizun den hardwareari "ISA" deitu ohi zaio.

Komputagailu oso agindu simpleak exekutatu dituzte sailik beraez, programatzekoan oso zailak sortzen lituztelako korreterako deinde programazio lengoaiaak, hauen bidez erredagoa de ~~ta~~ programatzera, etc ondoren, beraien pasetako dute programazio lengoaietako makina-lengoaiaita, "komplikatu"-bidez.

## Aginduen formak

Mallina - lehgoa iko agindu batetik eragileta simple bat eta eragileek definitzen ditu, horretarako formatu-jokien bat erabiltzen da, gure klasuen; disertatibo dugun prozesadoreak, txozera eintza da. Bitxkopurue biten zelitako da.



Eragileta-kodeak definitu aginduak bete beharreko eragileak:  
 add : (Bilketak) sub (Kentaka) move (datu bat kopiatu), ld (memoria irakurri).  
 Agindu - multzoak, K eragilek baditu logikoa, bitxello eragileta kodea, ikango da.

Eragileak adierazteko hainbat orduera; helbideratze-modultua:

- Berehaldean: eragileak aginduen dago:

move r2, #8 ; r2 := 8;

→ Erregistro bidezko erreferentzia: Eragileak erregistro batetik dago aginduen, eragileak duen erregistroa adierazten da.

add, r1, r3, r4 ; r2 := r3 + r4

- Absolutua: eragigaiaren memoria dago, aginduen haren helbidea adierazi.

ld, r2, ALD ; r2 := MeM[ALD]

- Indextua: Eragigaiaren memoria, haren helbideak erakusteko, aginduen adierazten diren aukerri-helbideak eta indize-helbideak batu:

stra, ALD[r1] ; MeM[ALD+r1] := r2

## Agiindult exekutatzeko Algoritmoak

1. Agiinduaren bilaldeia: Memoriaren dorden agiindult haratu eta prozesadorearen eranmen behar dira.

PC-ean daude exekutatu beharreko agiindu, memoria helbideak, eta IR, agiindu-erregistroa, ~~lotea~~ utzi behar dira:

$$IR := \text{Mem}[PC];$$

2. Agiinduaren desklodetza: Agiindua IR-erregistroaren hurrengo ondoren eragilete-kodea auriztu behar da, Desklodegezuraren bidez.

Lohengoa bi urratsik beti dira berdinak:

3. Erregigaien irakurketa; Agiindult behar dituen erregigaiak eskludo behar dira. Erregistro-multzoa erregistroak irakurri eta eduki "lantza" erregistroaren utzi. (Eragigaiak agiinduak bedearaz, ea da ezer egia behar).

4. Agiinduaren Exclusioa: Eragigaiak lortu ondoren, etekintza behar dira, OAI, Memoria do bestea gailuen bidez.

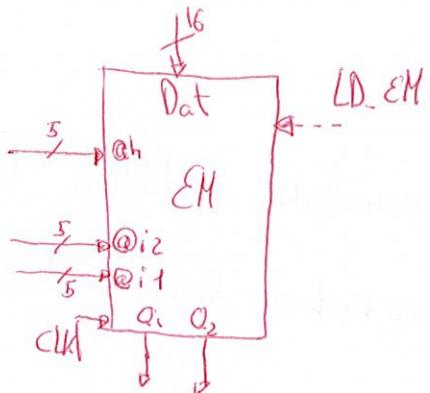
5. Emaitzaaren ideaketa: Eragiketaren emaitza norberak gertu behar da.

## Bird prozesadorearen diseinua

Bird prozesadoren 16 bit-ekoa izango da, beraez, prozesu-unitateko erregistroak zein memoria-posizioko 16 bitekoa izango dira.

Hurreb, PC erregistroa 16 bit-ekoa dela eszerikoa dugun, eta Memoria beraez,  $64K \times 16$  bitekoa da.

Erregistro multzoa osetzekoa 16 bit-ekoa 38 erregistro erabili dituenet erregistroren helbideak 5 bitekoa izango dira. ( $2^5 = 32$ )



Ziklo berean posible da, bi erregistro irakurtea, beti baitaude irakurteaz.

Hiru helbide sarekoak dode: @h, @i2, @i1

Eta to erregistroa irakurri daiteke, edo hiz beti o izango da, baina oin daiteke horren gainera idatzia.

Bird moldearen UAL-a honakoak izango da. Bi 16 bitello bi erregistroi kartuz, 16 bitello emaitza sortzen du. Eta 4 bitello kodetako beharko du.

Y @eta UAL Kodetako					
A+B	0000	A	0100	AorB	1000
A-B	0001	B	0101	AandB	1001
A×B	0010	A&1	0110	AxorB	1010
A/B	0011	A-1	0111	Desp-estA	1011
				Desp-estA	1100

Hoskoraz, gain 3 adierazte ditu, Z( $y=0$ ); N( $y<0$ ); P( $y>0$ ) badira.

Aginduak eta formatuak

Agindu Aritmetiko eta logikoa

OP, rh, ri1, ri2  $\Rightarrow$  rh = ri1 op ri2

Bi erregistrotik, beste erregistro bat sortzea.

mov rh, rit  $\Rightarrow$  rh := rit

Datuak erregistro betetik bestera erematzeko erabiltzen da.

Opi rh, rit, #berak  $\Rightarrow$  rh := rit op berek  $\Rightarrow$  div rh, rit, #1 = 0

$\Rightarrow$  rh := r1 / 1

mov rh, #berak  $\Rightarrow$  rh := berak  $\Rightarrow$  mov rh, 5  $\Rightarrow$  rh := 5

Memoriarekin lan egiten duten aginduak

Memoriarekin bi eragileta egin daitzeke irakurketa eta  
ideazketa.

ld rh, Ald  $\Rightarrow$  rh := Mem[Ald]

st rh, Ald  $\Rightarrow$  Mem[Ald] := ri2

Load aginduarekin memoria posizio bat irakurten ari gara, odea  
Store aginduarekin memoria idatza ari gara.

ldx rh, Ald[rit]  $\Rightarrow$  rh := Mem[Ald + rit]

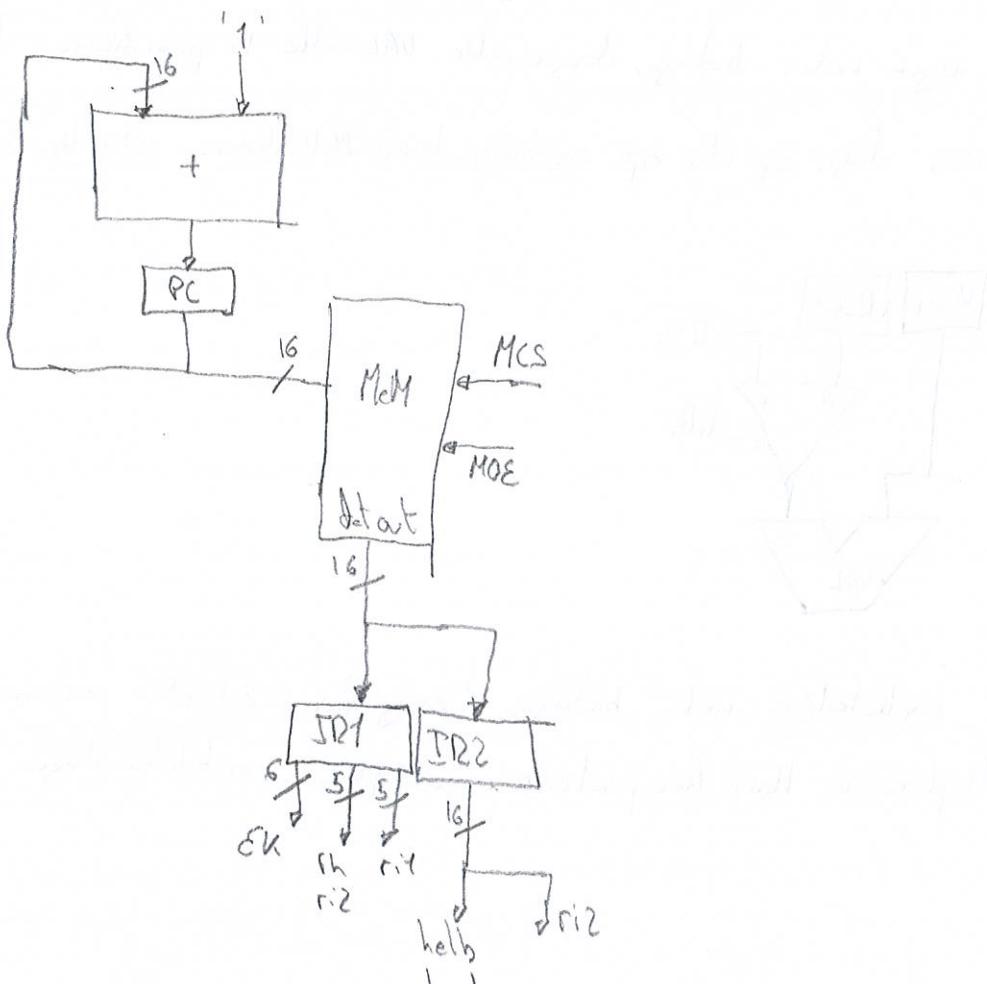
stx rh, Ald[rit]  $\Rightarrow$  Mem[Ald + rit] := ri2

Agindu fluxuaren Kontrola

bezit rit, etiketa if (rit=0) pc := pc + desp

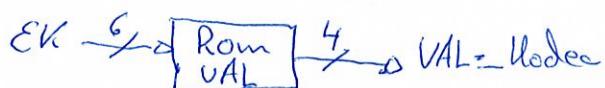
Bird prozesadorearen prozesu-unitatea

Aginduak biletzko PC-a erabiltzen da, eta bi agindu  
prozesatuko ditu aldi berean, Andorioz, PC-ari batuagile bat  
ezarrirako dio gure:

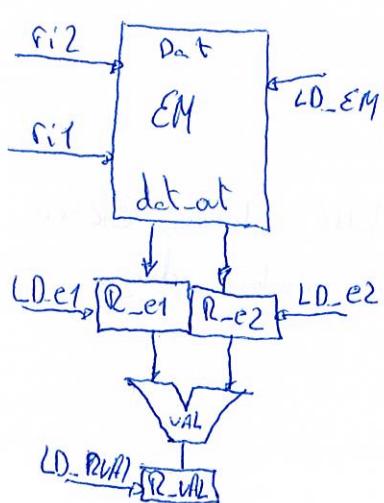


Aurrez illusi dugun bedala agindu desberdinek prozesatze batera ditugu, gure prozesadoreak. Gagindua hauetako prozesatikoa da:

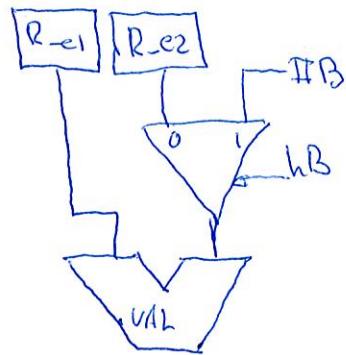
EK Rom-VAL era eramanago dugu.



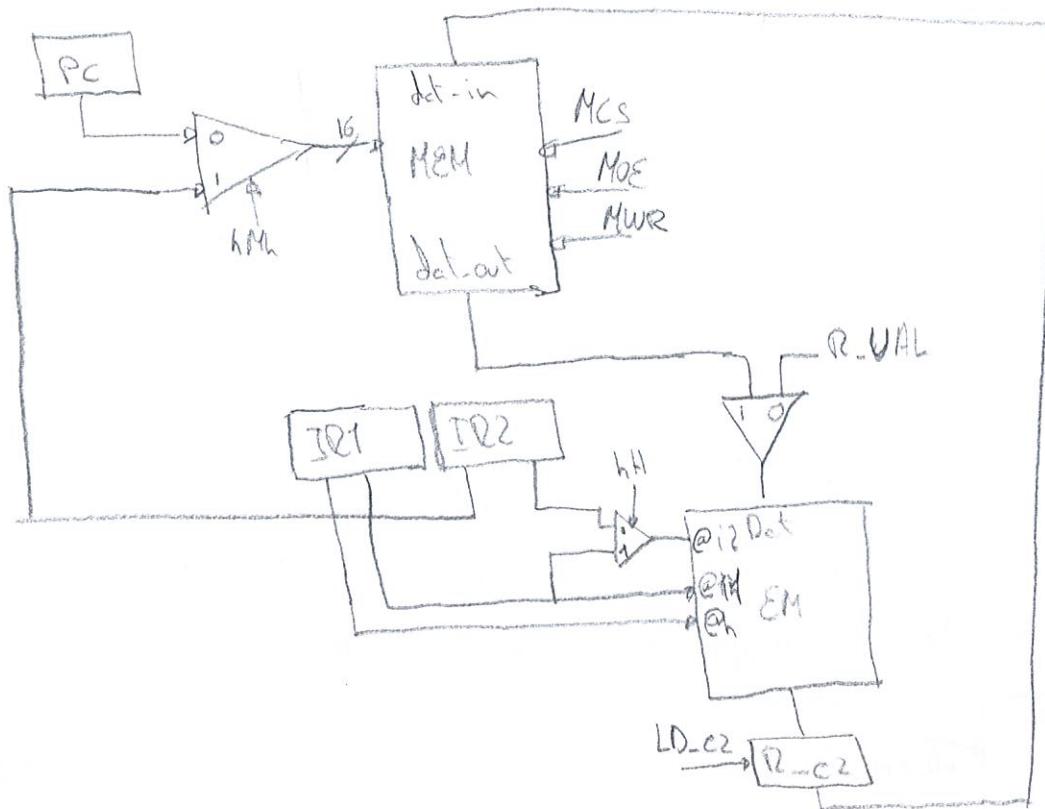
Ondoren r1 eta r12 VAL-era eraman behar dira, baina horretarako EK-ra eta bi erregistrotara eraman beharko ditugu.



Opi eragileta egia nahi badugu, da gureko VAL-eko B posizioaren bi eragigai joatea behar duzu, op eta op egitello, beraoz, Multiplexorean edariko duzu:



LD agindua exekutatzea nahi badugu, 2. agindu ez berdin eraman nahi duzu multiplexorean. Hori konpontzello multiplexorean erabiliko duzu.



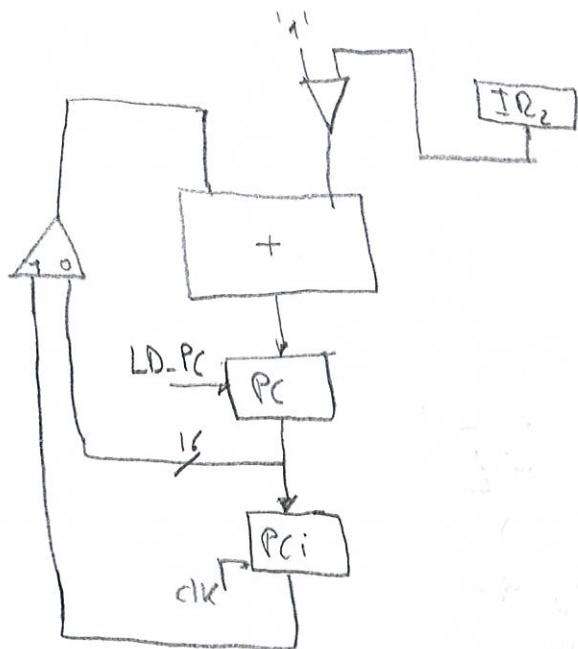
St agindua egitello, (id-8bita), IR1 eko 5 bitello isteera, @i2-ra konektatu beharko da, eta horretarako multiplexorean erabili duzu.

bez agindua osatzello, PC-a dagoen lehenik beste batera salto egia behar du, bera, PC-a gorde egia behartlo dugu.

bez eragileta egia behar dugun edo ea jaliitello, lehenik ilusio behar dugu ea r28 @zero den edo ez, eragileta soilik osatzen de r28ko bidea.

Bera, r28 VAL-aren sarrerako eraman behar da eta honela, eragileta egia ondoren 2 itzeren zero ematen badugu, bez eragilete prozesatu behar dugu.

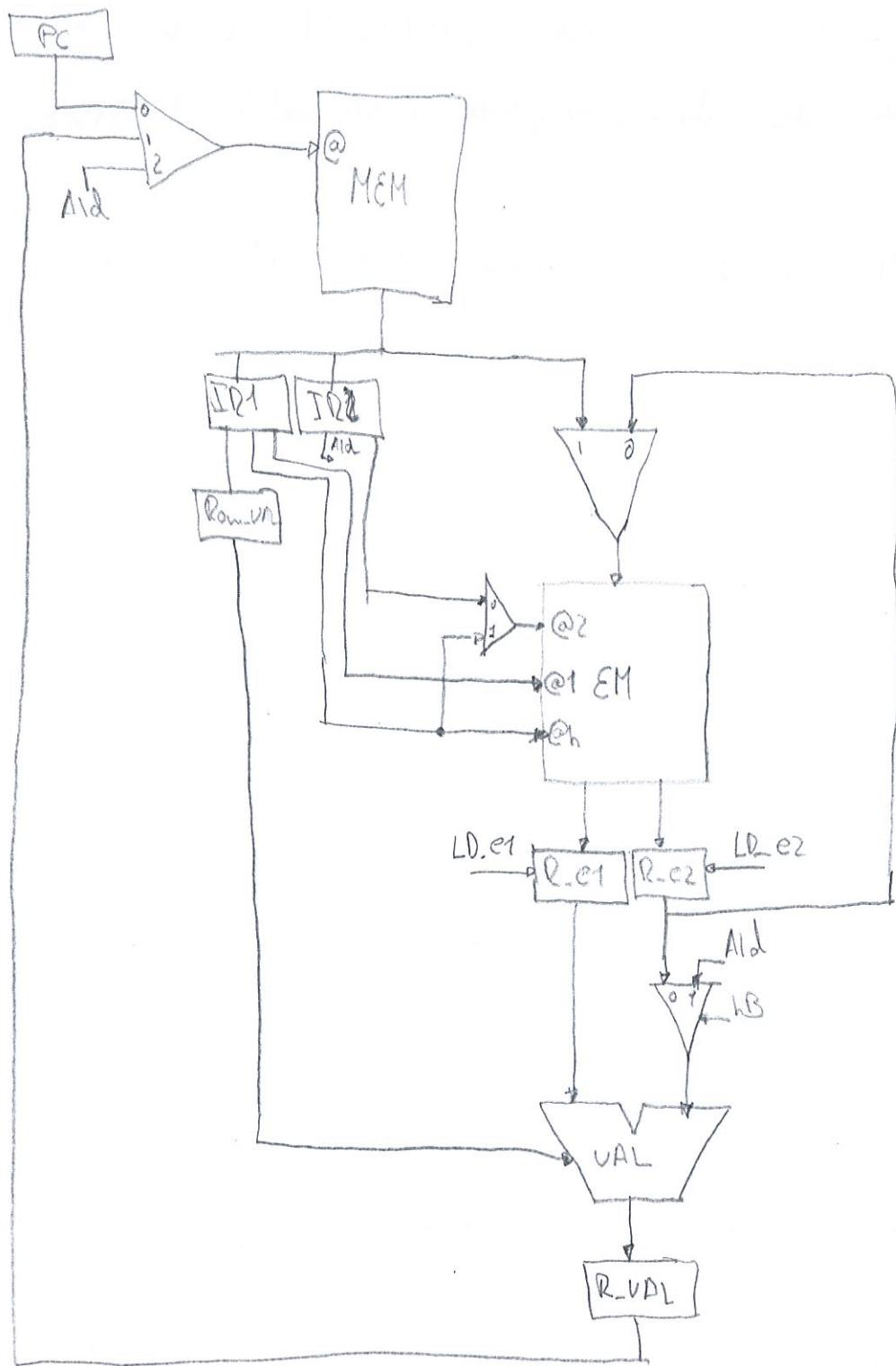
Baina, eragilete hori egiten badugu,  $PC := PC + 1$  era pasatzen da bera, gorde egia behartlo dugu.



$PC_i$ -k exekutatzeko ari garen  $PC$ -an izango dugu eta  $PC$ -n hurrengoa.

Irx agindua ( $Idx, r13, AL[re]$ ) osatzello, lehenik eta behin helbidea kalkulatu behar da, horretarekin,  $r13 + r2$  eragileta egia behartu da, dagoenello eragileta hori eginda dego op motatik aginduak prozesatzello.

Berez, A helbide hori erabili ahal izatello multiplexore handiago bat ezarri beharreko dugu Memoriaren sarreran.



Stx aginduak ere helbidea kalkulatu behar du eta gero st operazio egin, baita degoeneko noraietako elementu ditugu, agindu hau baprozesadoko.

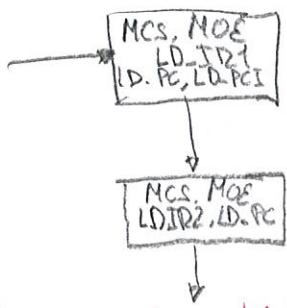
## Kontrol-unitatea

### Aginduen bilaketa

Agindu bat osatzen bi aldiz, irakurri behar da, hor da, PC eta PC+1, 32 bit behar baititugu eta gure soiliuk 16 lortzen ditugu irakurketa bultzear.

Hori egin ahal izatello lehenik memoria tikkirri behar da, MCS eta MOE aktibatua. Ondoren, IR1-ean gorde behar da irakurda, LD eta PC-a aktibatu behar dugu, keter, beraz, LD-IR1 aktibatuko da. Gainera PC-a aktibatu behar dugu, horretarako, batugailura doan multiplexoreen balioa "0" izan behar da. Izaan edo PCi-ko balio hori behar da baititugu.

Hurrengo zikloan, LD-IR2-a aktibatua, IR2-an ordetako dugu, Memoria tikkirri lortzen dugun balioa. Horretarako, MOE eta MCS aktibatu behar izan dira. Eta bitartean PC-a hurrengo helbidea prozesatzeko hasten da, LD-PC aktibatzen dugutelako.



Lehenengo bi zikloak, beti errepetitzen dira

### Aginduen desplazamendua

EK Sarraia bat erakilliko dugu eta honellaz aderizago ditu, ader bezalitzear, prozesatzeko antzekoak diren aginduak elkar-tulko ditugu.

## Op/mou aginduaaren exeluzioa

Lehenik op agindua irallurri beharttu da, beroz, LD-ek eta  
LD-ek altibatu beharttu dira, gero, RON-VAL-aren bidez adierazitxo  
Zai o VAL-ari ze eragileta egia behar duen, eta eragileta hori  
R-VAL-ean gordetxo dugu, LD-VAL altibatu, eta azkenik eragileta  
hori Errazkistro-Nutzoan gordetxo dugu, LD-CM altibatu.

## G7.2. arketaren taula

egoera	PCI	PC	@MEM	MEM erag	MEM dat_out	M[1000H]	IR <sub>1</sub>	IR <sub>2</sub>	EM[r5]	EM[r6]	R_e1	R_e2	R_ual
b111	?	3810H	3810H	"	0001H	00015H	7	?	?	?	?	?	?
b112	3810H	3811H	3811H	RD	0110H	0110H	01105H	R1	11	11	11	11	11
DesK	11	3812H	3812H	-	2	"	"	0110A	R1	"	11	11	11
H1D-LD	11	0110H	0110H	LD	0002H	"	"	0002H	-	-	-	-	-
S-1	11	11	3812H	RD	beq R5	"	"	beq S5	"	"	11	11	11
B112	3812H	3813H	3813H	RD	0000H	"	beq S5	"	"	11	11	11	11
DesK	11	3814H	3814H	C2	2	"	"	0006H	"	"	11	11	11
T-REG	11	11	11	C8	2	"	"	11	"	"	11	11	11
A-Beq	"	"	"	"	"	"	"	"	"	"	11	11	11
D111	"	"	"	RD	ldx r6	"	"	"	"	"	0002H	W	W
G112	3814H	3815H	3815H	RD	0100H	W1	11	11	11	11	11	11	11
DesK	11	3816H	3816H	-	2	"	"	1000H	"	"	11	11	11
T-DX	11	"	"	"	"	"	"	"	"	"	W	W	W
A-LDX	11	"	"	"	"	"	"	"	"	"	11	11	11
HALD	"	1000H	1000H	RD	C0FFH	"	"	H	H	"	0002H	W	W
B111	"	3816H	3816H	RD	beq R0	"	"	11	11	11	11	11	11
B112	3817H	3817H	3817H	RD	0004H	"	beq S0	"	"	11	11	11	11
DesK	11	3818H	3818H	-	2	"	"	0004H	W1	11	11	11	11
T-CST	11	"	"	-	2	"	"	"	"	"	11	11	11
A-BEQ	11	"	"	"	"	"	"	"	"	"	000CH	W	W

### G7.1: ariketaren taula

## G7.2. ARIKETA

Bedi ondoko programa zatia:

	@mem	EDUKIA
	...	
Pos:	0110H	0000 0000 0000 0010
	...	
Taula:	1000H	0000 0000 0000 0001
	1001H	1111 1111 1111 0001
	1002H	0000 0000 1111 1111
	...	
	380FH	...
ld r5, Pos	3810H	000000 00101 00000
beq r5, zero	3811H	0000 0001 0001 0000
	3812H	011010 00000 00101
ldx r6, Taula[r5]	3813H	0000 0000 0000 0110
	3814H	000010 00110 00101
beq r0, buk	3815H	0001 0000 0000 0000
	3816H	011010 00000 00000
zero: movi r6, -1	3817H	0000 0000 0000 0100
	3818H	001000 00110 00000
buk: stx r6, Taula[r0]	3819H	1111 1111 1111 1111
	381AH	000101 00110 00000
	381BH	0001 0000 0000 0000
	381CH	

Agindu-sekuentzia kontuan hartuta, erantzun ondoko galderei:

- Zenbat ziklo behar dira aurreko programa zatia exekutatzeko ?
- Memoriako posizioetan aginduen kode bitarra ageri da. Informazio horretan oinarrituta esan, zein den agindu bakoitzari dagokion eragiketa-kodea.
- Bird makinaren prozesu-unitatea kontuan hartuta, zein da memoriaren helbide-sarreran dagoen multiplexorearen funtzioa?

Irudiko memoriaren edukia eta aurrean dituzun kontrol- eta prozesu-unitateak kontuan izanik, bete ondorengo taula. Horretarako, adierazi egoeraz egoera lehenengo lerroan idatzita dauden erregistro, memoria eta abarren balioak.

## G7.1. ARIKETA

Izan bedi honako programa zati hau:

	@mem	EDUKIA
	...	
A:	1110H	0000 0000 0001 1000
B:	1111H	0000 0000 0001 0011
ema :	1112H	
	1113H	
	.....	.....
ld r7, A	4100H	000000 00111 00000
	4101H	0001 0001 0001 0000
subi r9,r7,#2	4102H	001100 01001 00111
	4103H	000000000000 00010
beq r9, bi	4104H	011010 00000 01001
	4105H	0000 0000 0000 0100
beq r0, buk	4106H	011010 00000 00000
	4107H	0000 0000 0000 0110
bi: addi r10,r0,#1	4108H	001010 00100 00001
	4109H	0000 0000 0000 0001
stx r7,ema[r10]	410AH	000101 00111 01010
buk: stx r9,ema[r1]	410BH	0001 0001 0001 0010
	410CH	000101 01001 00001
	410DH	0001 0001 0001 0010
	410EH	

Agindu sekuentzia hau kontuan hartuta erantzun ondoko galderei. Horretarako, kontuan izan r1 erregistroan FFFFH datua dagoela.

- Zenbat ziklo behar dira aurreko sekuentzia exekutatzeko?
- addi aginduaren eragiketa-kodea 10 dela jakinik, 4108H eta 4109H helbideetan dagoen informazioa zuzena al da? Hala ez bada, zer dago gaizki?  
Gaizki dago, rh etc oit ez dade bider datzitsa.
- Erregistro multzoko zenbat erregistrotan idatz daiteke aldi berean? Eta irakurri?
- Programa horretan, zein agindutan idazten da memorian? Zein da gordetzen den balioa? Memoriako zein posiziota idatziko da?
- Bete ezazu ondorengo taula, non, egoeraz egoera, erregistro, memoria eta erregistro-mutzoko hainbat osagaien balioak erakusten diren.