Euskal Herriko Unibertsitatea
Informatika Fakultatea

eman ta zabal zazu

Konputagailuen Arkitektura eta Teknologia Saila

Konputagailuen Arkitektura I

Memoria-sistema 5 (ebazpena): Orriztaketa + bankuak + atzipen-denbora guztira

Konputagailu baten memoria-sistemak honako ezaugarri hauek ditu:

- Helbideratze-unitatea bytea da, eta hitzak 8 bytekoak dira.
- **Alegiazko memoria:** 1 MB-ko memoria orriztatua. Orriak 1 kB-koak dira eta helbideen itzulpenerako TLBa erabiltzen da. TLBan asmatzean itzulpen-denbora 1 ziklokoa izango da eta hutsegitean 20 ziklokoa. Hasieran, TLBa hutsik dago.
- **Memoria nagusia:** 16 kB-koa da eta 2 banku ditu, bakoitza 4 modulu tartekaturekin. Bere atzipen denbora 10 ziklokoa da (1 ziklo tartekatze-bufferretik).

Hauxe eskatzen da:

a) Helbide logikoaren zein fisikoaren eskema, eremuak bitetan zeintzuk diren adieraziz. Zenbat orri izan ditzake programa batek gehienez? Zenbat orri ditu memoria nagusiak?

b) Memoria nagusiari dagokionez, zein da banku baten tamaina bytetan? Eta moduluaren tamaina bytetan?

c) Konputagailu honetan programa hau exekutatzen da:

```
for (i=0;i<200;i++)
 B[0]=B[0]+A[i]*A[i+1];
```

```
        movi r1,#0
        movi r2,#199
        load r5,B[r1]
begizta: load r3,A[r1]
        load r4,A[r1+8]
        mul r4,r3,r4
        add r5,r4,r5
        addi r1,r1,#8
        subi r2,r2,#1
        bge r2,begizta
        store r5,B
```

Programa 0 helbide logikotik aurrera dago metatua. A bektorea 8192 helbide logikoan hasten da, eta B bektorea 10000 helbide logikoan. Bai aginduak eta bai bektoreen osagaiak hitz batekoak dira.

Kalkulatu memoria atzitzeko behar den denbora (itzulpen-denbora eta atzipen-denbora) begiztaren lehenengo pasaldian sortzen diren helbide logikoetarako. Horretarako, zehaztu itzazu, argi eta garbi, taula baten bidez, atzipen denborak kalkulatzeko egiten dituzun urratsak. Orri-taularen edukia honako hau da:

Orri logikoa:	0	14	8	9	....
Orri fisikoa:	2	6	10	11	....

d) Kalkulatu memoria atzitzeko behar den denbora (itzulpen-denbora eta atzipen-denbora) programaren exekuzioan sortzen diren helbide logiko guztietarako.

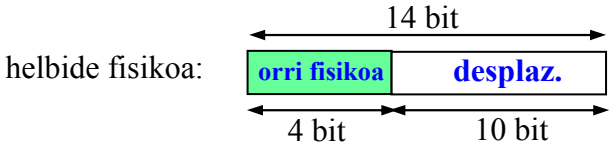
Ebazpena

Lehenik, memoria-sistemaren ezaugarriak ongi aztertu behar dira enuntziatuan, behar den informazio guztia ondorioztatzeko.
Hala, kontuan hartu beharko dugu memoria-sistema honek **8 byteko hitzak** erabiltzen dituela, eta **helbideratze-unitatea bytea** dela. Azter dezagun orain atal bakoitzean eskatzen digutena.

(a) Helbide logikoaren zein fisikoaren eskema, eremuak bitetan zeintzuk diren adieraziz. Zenbat orri izan ditzake programa batek gehienez? Zenbat orri ditu memoria nagusiak?

Alegiazko memoriaren ezaugarriak hauek dira: 1 MB-koa (2^{20}) da, eta orriztatua dago, orrien tamaina 1kB-koa izanik. Informazio horretatik gauza asko ondorioztatu ditzakegu **helbide logiko**en egiturari buruz. Alde batetik, byteen helbide logikoak emateko 20 bit behar direla, 1 MB = $1 \times 2^{20} = 2^{20}$ delako. Beste aldetik, alegiazko memoria orriztatua dagoenez, helbide logikoak bi eremu izango ditu: **orri logikoa** adierazten duten bitak, eta atzitu nahi dugun byteak orri logikoaren barruan duen **desplazamendua**. Eta orri logikoak 1kB-koak direnez, 10 bit behar dira byteen desplazamendua adierazteko ($1024 = 2^{10}$). Ondorioz, 20 – 10 = 10 bit izango ditugu orri logikoa adierazteko (hortaz, **programa** batek 2^{10} = **1024 orri** izan ditzake). Honelaxe geratuko da helbide logikoaren eskema:

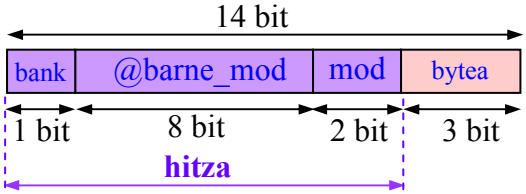
Memoria nagusiari dagokionez, jakin badakigu 16 kB-koa dela. Hala, **helbide fisikoa** adierazteko 14 bit behar dira (16 kB = $2^4 \times 2^{10} = 2^{14}$). Hortaz, helbide logikoa helbide fisiko bihurtzeko prozesuan, 20 bit 14 bihurtuko dira. Desplazamendua berdina denez, orri fisikoa adierazteko eremua 4 bitekoa (14 – 10) izango da (hala, **memoria nagusiak** 2^4 = **16 orri fisiko** izango ditu).



(b) Memoria nagusiari dagokionez, zein da banku baten tamaina bytetan? Eta moduluaren tamaina bytetan?

Memoria nagusiaren egiturari buruzko informazioa kontuan hartuz, ondorioztatzen dugu 14 biteko helbide fisikoko pisu txikieneko 3 bitek bytea adierazten dutela, eta, ondorioz, beste 11 bitek adieraziko dutela hitza. Hitza adierazten duen eremuan beste hiru eremu bereizi behar ditugu: alde batetik, memoria nagusia ondoz-ondoko 2 bankuk osatzen dutenez, pisu handieneko bitak bankua adieraziko du; horrez gain, banku bakoitzean 4 modulu tartekatu daudenez, bilatzen ari garen hitza zein modulutan dagoen adierazten duen eremua beharko dugu (**mod**, 2 bitekoa, $2^2 = 4$ delako, 4 izanik modulu kopurua, hain zuzen), eta, beste aldetik, modulu horren barruan hitza zein helbide zehatzetan dagoen ere (**@barne_mod**, 10 – 2 = 8 bitekoa). Honelaxe geratuko da helbide fisikoaren egitura:

helbide fisikoa:



Beraz, badakigu memoria nagusia 2 bankuk osatzen dutela. Hortaz, banku bakoitzaren tamaina memoria osoaren erdia izango da: 8 kB, alegia.

$$\frac{16\text{ kB}}{2\text{ banku}} = \frac{2^{14}\text{ byte}}{2\text{ banku}} = 2^{13}\text{ byte/banku} = 8\text{ kB/banku}$$

Hala berean, banku bakoitzean 4 modulu daudenez, modulu bakoitza 2 kB-koa da:

$$\frac{8\text{ kB/banku}}{4\text{ modulu/banku}} = 2\text{ kB/modulu}$$

(c) Ematen diguten programa exekutatzean, begiztaren lehenengo pasaldian honako helbide logiko hauek sortzen ditu prozesadoreak:

	Agindua / datua	Helbide logikoa
	`movi r1,#0`	0
	`movi r2,#199`	8
	`load r5,B[r1]`	16 / 10000
`begizta:`	`load r3,A[r1]`	24 / 8192
	`load r4,A[r1+8]`	32 / 8200
	`mul r4,r3,r4`	40
	`add r5,r4,r5`	48
	`addi r1,r1,#8`	56
	`subi r2,r2,#1`	64
	`bge r2,begizta`	72

Kontuan izan programaren azken agindua (`store r5,B`) ez dela begiztaren barruan exekutatzen, behin ere ez, bakar-bakarrik exekutatuko da behin begizta amaitzean.

Gogora ditzagun taulan bete beharreko balioak kalkulatu ahal izateko erabili behar ditugun ekuazioak.

Alde batetik, helbide logikotik abiatuta, orri logikoa eta desplazamendua kalkulatzeko:

orri logikoa = @logikoa `div` orriaren tamaina bytetan = @logikoa `div` 1024

desplazamendua = @logikoa `mod` orriaren tamaina bytetan = @logikoa `mod` 1024

Orri-taulan bilatuko dugu zein den orri logiko bakoitzari dagokion orri fisikoa, eta hori eta desplazamendua kontuan hartuz, helbide fisikoa kalkulatuko dugu, honelaxe:

@fisikoa = orri fisikoa × orriaren tamaina bytetan + desplazamendua = $o.f. \times 1024 + d$

Byterako helbide fisikoari dagokion hitza kalkulatuko dugu ondoren:

hitza = @fisikoa `div` hitzaren tamaina bytetan = @fisikoa `div` 8

Beste aldetik, ondoz-ondoko bankuak direnez, hitzaren eremuak honelaxe kalkulatuko ditugu:

bankua = hitza `div` bankuaren tamaina hitzetan = hitza `div` 2^{10} = hitza `div` 1024

hondarra = hitza `mod` bankuaren tamaina hitzetan = hitza `mod` 2^{10} = hitza `mod` 1024

Aurreko zatiketaren hondarrean, oraindik bi eremu daude, bereizi beharrekoak, eta balio horretatik ateratzen dira modulua eta barne-helbidea:

@barne_mod = hondarra `div` modulu-kopurua bankuan = hondarra `div` 4

mod = hondarra `mod` modulu-kopurua bankuan = hondarra `mod` 4

Azken lau ekuazioak orokortu daitezke beste modu honetan ere, bankua kalkulatzean lortu dugun hondarraren beharra saihesteko:

bankua = (hitza `div` modulu-kopurua bankuan) `div` moduluen tamaina hitzetan =

= (hitza `div` 2^2) `div` 2^8 = (hitza `div` 4) `div` 256

@barne_mod = (hitza `div` modulu-kopurua bankuan) `mod` moduluen tamaina hitzetan =

= (hitza `div` 2^2) `mod` 2^8 = (hitza `div` 4) `mod` 256

mod = hitza `mod` modulu-kopurua bankuan = hitza `mod` 2^2 = hitza `mod` 4

Ekuazio horiek guztiak kontuan harturik, hau da ateratzen den taula:

@l	$o.l.$	d	$t_{itzulp.}$ (1)	$o.f.$	d	@f	hitza	bankua	hondarra	@barne_mod	mod	$t_{atzip.}$ (2)
0	0	0	20	2	0	2048	256	0	256	64	0	$10^{(a)}$
8	0	8	1	2	8	2056	257	0	257	64	1	$1^{(a)}$
16	0	16	1	2	16	2064	258	0	258	64	2	$1^{(a)}$
10000	9	784	20	11	784	12048	1506	1	482	120	2	$10^{(b)}$
24	0	24	1	2	24	2072	259	0	259	64	3	$1^{(c)}$
8192	8	0	20	10	0	10240	1280	1	256	64	0	$10^{(d)}$
32	0	32	1	2	32	2080	260	0	260	65	0	$10^{(e)}$
8200	8	8	1	10	8	10248	1281	1	257	64	1	$1^{(f)}$
40	0	40	1	2	40	2088	261	0	261	65	1	$1^{(g)}$
48	0	48	1	2	48	2096	262	0	262	65	2	$1^{(g)}$
56	0	56	1	2	56	2104	263	0	263	65	3	$1^{(g)}$
64	0	64	1	2	64	2112	264	0	264	66	0	$10^{(h)}$
72	0	72	1	2	72	2120	265	0	265	66	1	$1^{(h)}$

Azalpenak:

(1) Itzulpen-denbora 20 ziklokoa da orri logiko bat lehenengo aldiz atzitzen denean, TLBan ez baitago orri horri buruzko erreferentziarik, eta hutsegitea izango da bilaketaren emaitza; hortaz, orri-taulan bilatu beharko du sistemak bilatzen ari garen orri logikoaren zenbakia, eta TLBan idatzi dagokion orri fisikoaren zenbakia. Hortik aurrera, hurrengo erreferentzietan, ziklo bakarra beharko da

itzulpena egiteko, jadanik TLBan egongo delako orri logiko horren erreferentzia, eta asmatzea izango da bilaketaren emaitza. Taulan ikusten denez, hutsegitea (20 zikloko itzulpen-denbora) gertatzen da hiru kasutan, 0, 9 eta 8 orri logikoak lehenengo aldiz ageri direnean. Beste erreferentzia guztiak asmatzeak dira, eta ziklo bakarra behar da horiek itzultzeko.

(2) Atzipen-denbora 1 ziklokoa da atzitu behar den hitzari dagokion barne-helbidea jadanik irakurri baldin bada eta tartekatze-bufferretan baldin badaude barne-helbide bera duten hitzak. Kontrako kasuan, berriz, atzipen-denbora 10 ziklokoa da, memoria nagusian irakurri behar delako.

- **(a)** Programaren lehenengo agindua atzitzeko 10 ziklo behar dira, memoria nagusian lehenengo aldiz irakurri behar delako. Baina lehenengo aginduarekin batera, memoria nagusian lau modulu tartekatu daudenez, hurrengo hiruak ere kargatuko dira tartekatze-bufferretan, barne-helbide berean daudelako (0 bankuko 64 barne-helbidean, 0, 1, 2 eta 3 moduluetan, hain zuzen); horregatik, hurrengo hiru aginduak (bigarrena, hirugarrena eta laugarrena) atzitzeko ziklo bana behar da, tartekatze-bufferretatik eskuratzen direlako.
- **(b)** Hirugarren aginduaren ondoren, B[0] osagaiaren helbidea dator. Taulan ikusten den bezala, B[0] osagaiari 1 bankuko 120 barne-helbidea dagokio. Hori denez 1 bankuan irakurtzen den lehenengo helbidea, B[0] osagaia atzitzeko 10 ziklo behar dira.
- **(c)** Aurreko atzipenean 1 bankuko tartekatze-bufferretan 120 barne-helbideen edukiak kargatu diren arren, 0 bankuko tartekatze-bufferren edukiak ez dira aldatu: oraindik ere 0 bankuko 64 barne-helbideen edukiak daude kargatuta bertan. Horregatik, B[0] osagaiaren helbidearen ondoren datorren laugarren aginduaren helbidea atzitzeko, ziklo bat behar da bakarrik, laugarren agindua 0 bankuko tartekatze-buffer batean dagoelako oraindik (0 bankuko 64 barne-helbidean, 3 moduluan, baitago 4. agindua).
- **(d)** Laugarren aginduaren ondoren, A[0] osagaiaren helbidea dator. Taulan ikusten den bezala, bai laugarren agindua eta baita A[0] osagaia ere 64 barne-helbidean daude, baina, 0 bankuko tartekatze-bufferretan oraindik ere 64 barne-helbideen edukiak dauden arren, A[0] osagaia atzitzeko 10 ziklo behar direla jarri dugu, hain zuzen ere A bektorea 1 bankuan dagoelako, eta 1 bankuko tartekatze-bufferretan ez daude 64 barne-helbideen edukiak, B[0] osagaia irakurtzean kargatu diren 120 barne-helbideei dagozkienak baizik. Horregatik, taula betetzean kontu handiz begiratu behar da ea aurreko barne-helbidea banku berekoa den ala ez.
- **(e)** Bosgarren agindua eskuratzeko 10 ziklo behar dira, berriro irakurri behar delako memoria nagusian: 0 bankuan egon arren, aurreko lauen beste barne-helbide batean dagoelako. Horregatik, tartekatze-bufferretan dauden hitzak ez dira erabilgarri, 64 barne-helbidekoak baitira, eta bosgarren agindua 65 barne-helbidean dago.
- **(f)** A[1] osagaia atzitzeko 1 ziklo behar da, 1 bankuko tartekatze-bufferretan 1 bankuko 64 barne-helbideak daudelako oraindik.

(g) Hurrengo hiru aginduak eskuratzeko ziklo bana behar da, bosgarren aginduarekin batera kargatu direlako 0 bankuko tartekatze-bufferretan, 0 bankuko 65 barne-helbideetan baitaude.

(h) Azkenik, 9. aginduaren kasuan (0 bankuko 66 barne-helbidean dagoenez), berriz irakurri behar da memoria nagusian, eta 10 ziklo behar dira, baina azken bi aginduak atzitzeko ziklo bana beharko da bakarrik, 9.arekin batera kargatzen direlako tartekatze-bufferretan (taulan begiztako azken agindua bakarrik agertzen den arren, hemen programaren azken agindua ere kontuan hartu dugu).

Laburbilduz, programak begiztaren lehenengo pasaldian sortzen dituen helbideak itzultzeko 70 ziklo behar dira (3 hutsegite × 20 ziklo + 10 asmatze × 1 ziklo), eta behar den informazioa memoria nagusian eskuratzeko, 58 ziklo behar dira (5 irakurketa memorian × 10 ziklo + 8 irakurketa tartekatze-bufferretan × 1 ziklo). Beraz, guztira, **128 ziklo** behar dira begiztaren lehenengo pasaldiko helbide guztiak itzultzeko eta memoria nagusia atzitzeko.

(d) Aurreko atalean lehenengo pasaldian behar den denbora kalkulatu dugu. Oraingo honetan, berriz, programa osoa exekutatzean prozesadoreak sortzen dituen helbide logiko guztiak itzultzeko eta memoria nagusia atzitzeko behar den denbora kalkulatu behar dugu.

Taula osoa betetzea luzeegia litzateke. Horregatik, pentsatzeari ekingo diogu, eta ondorio orokorrak ateraz gero, denbora kalkulatu ahal izango dugu.

Hasteko, lehenbizi helbide logikoak itzultzeko behar den denbora kalkulatuko dugu, eta ondoren memoria nagusia atzitzeko behar dena.

Helbide logikoak itzultzeko behar den denborari buruz nahikoa dugu aurreko atalean idatzi dugun ekuazioa orokortzea:

$$t_{itzulpena}\,(\text{ziklotan}) = n \text{ hutsegite} \times 20 \text{ ziklo} + m \text{ asmatze} \times 1 \text{ ziklo}$$

Hau da, jakin behar dugu guztira zenbat hutsegite gertatzen diren TLBan, eta zenbat asmatze. Esan dugun legez, hutsegiteak gertatzen dira orri logiko bat lehenengo aldiz atzitzen denean (suposatuko dugu orri logikoak memoria nagusian kargatu ondoren bertan jarraituko dutela programaren exekuzioa amaitu arte, hau da, memoria nagusian nahikoa tokia dagoela programa honetako orri logiko guztiak sartzeko, eta sistemak ez dituela beste batzuengatik ordezkatuko toki faltagatik).

Hala, nahikoa da jakitea zenbat orri logiko betetzen dituen programak (aginduek eta datuek). Azter ditzagun banan-banan.

Taulan ikusi dugun legez, agindu guztiak 0 orri logikoan daude. Hortaz, hutsegite bakarra sortuko da aginduen helbideen ondorioz.

A bektoreari dagokionez, taulan ikusi dugu lehenengo osagaiak (A[0] eta A[1] ageri dira taula horretan) 8 orri logikoan daudela. Begiztaren adierazpena aztertzen badugu, ikusiko dugu azken pasaldian (`i` = 199 denean) A[199] eta A[200] osagaiak atzituko direla. Beraz, A bektoreak 201 osagai ditu, A[0] eta A[200] artekoak. Osagai bakoitzak 8 byte okupatzen dituenez, A bektoreak guztira 1608 byte okupatuko ditu. Orri bakoitzean 1024 byte sartzen direnez, ondorioztatzen da A bektoreak 2 orri beteko

dituela. Edozein kasutan, egiaztatu behar da hori zehatza den ala ez, gerta daitekeelako 3 orri okupatzea, baldin eta A[0] osagaia ez badago orri baten hasieran. Ez da hau kasua, jadanik ikusi dugulako A[0] osagaia 8 orri logikoan dagoela, 0 desplazamendua duela. Baina beti merezi du egiaztatzea. Horretarako, azken osagaiaren helbide logikoa kalkulatuko dugu, eta gero hortik ondorioztatu zein orri logikotan dagoen. Azken osagaiaren helbide logikoa kalkulatzeko kontuan hartuko dugu zein den bektorearen hasierako helbidea, zenbat osagai dituen, eta zenbat posizio okupatzen dituen osagai bakoitzak. Honelaxe:

$$@A[200] = 8192 + 200 \times 8 = 9792$$

Eta helbide horri dagozkion orri logikoa eta desplazamendua:

$$@\text{logikoa} = 9792 \rightarrow \quad \text{orri logikoa} = 9792 \texttt{ div } 2^{10} = 3068 \texttt{ div } 1024 = 9$$

$$\text{desplazamendua} = 9792 \texttt{ mod } 2^{10} = 9792 \texttt{ mod } 1024 = 576$$

Hau da, A bektoreak okupatzen dituen orri logikoak hauek dira: 8, 9.

B bektoreari dagokionez, B[0] osagaia da atzitzen den bakarra eta, taulan ikusi dugunez, 9 orri logikoan dago eta 784 desplazamendua dagokio.

Laburbilduz, aginduen eta bektoreen artean, hauek dira betetzen dituzten orri logikoak: 0, 8, 9. Hau da, 3 orri logiko, 3 hutsegite.

Orain asmatze-kopurua zein den jakiteko, atzipen-kopuru osoa kalkulatu behar dugu. Horretarako, programa aztertuko dugu.

Begiztatik kanpo, lau agindu daude, hiru begiztaren aurretik, eta bat begiztaren ondoren, eta hauek behin bakarrik atzitzen dira, programaren exekuzioa hastean eta bukatzean, hain zuzen. Horrez gain, begiztaren aurretik zein ondoren B[0] osagaia atzitzen da.

Begiztaren barruan, berriz, 7 agindu daude eta A bektorearen 2 osagai atzitzen dira pasaldi bakoitzean Hau da, 9 atzipen pasaldi bakoitzean; begizta 200 aldiz errepikatzen denez, atzipen-kopurua, guztira, hau da:

$$\text{atzipen-kopurua} = (3 + 1) + (7 + 2) \times 200 + (1 + 1) = 1806$$

Atzipen horietatik 3 hutsegite dira. Ondorioz (1806 – 3) = 1803 asmatze izango dira. Hala, itzulpen-denbora osoa:

$$t_{itzulp.} = 3 \text{ hutsegite} \times 20 \text{ ziklo} + 1806 \text{ asmatze} \times 1 \text{ ziklo} = 1866 \text{ ziklo}$$

Atzipen-denborari dagokionez, berriz, nahikoa dugu taula begiratzea, bertan atzipenen portaera islatzen baita. Taulan ikusi dugu datuak 1 bankuan daudela eta aginduak, berriz, 0 bankuan. Hori dela eta, nahiz eta datuen eta aginduen helbideak tartekatuta ageri, batzuek ez dute eraginik besteen atzipen-denboran, banku desberdinetan egiten direlako atzipenak. Azter ditzagun kasu posible guztiak.

Hasteko, begiztaren aurretik dauden 3 aginduak eta datua —B[0] osagaia— atzitzeko, taulan ageri den ziklo-kopurua behar da:

$$\text{Begiztaren aurretik: } 10 + 1 + 1 + 10 = 22 \text{ ziklo}$$

Begiztaren barruan, berriz, aginduak eta datuak atzitzeko denborak independenteki kalkulatuko ditugu, banku desberdinetan daudenez, ez baitute batzuek besteen eraginik nozitzen.

Aginduei dagokienez, lehenengo pasaldia eta hurrengo pasaldi guztiak bereizi behar ditugu, begiztaren lehenengo aginduaren atzipen-denbora desberdina delako lehenengo pasaldian eta hurrengo guztietan. Izan ere, begiztaren lehenengo pasaldian, taulan islatzen den moduan, 1 ziklo behar da agindu hori atzitzeko, tartekatze-bufferrean dagoelako: 259 hitza izanik, 0 bankuko 64 barne-helbidean dago, eta lehenengo agindua (256 hitza) irakurri denean, tartekatze-buferretan kargatu dira 256, 257, 258 eta 259 hitzak, 0 bankuko 64 barne-helbidean, 0, 1, 2 eta 3 moduluetan. Baina bigarren pasalditik aurrera gauzak aldatzen dira, begiztako lehenengo agindu horren aurretik begiztako azkena exekutatu delako, 265 hitza hain zuzen ere, eta hori 66 barne-helbidean dago, 1 moduluan. Hala, tartekatze-bufferretan 264, 265, 266 eta 267 hitzak daude. Hortaz, berriro 259 hitza behar denean (begiztaren lehenengo agindua, alegia), ez dago tartekatze-bufferrean jada, eta 10 ziklo beharko dira, beste barne-helbidean izanik, berriro irakurri behar delako memoria nagusian.

Hala, aginduen atzipen-denbora kalkulatzeko begiztaren lehenengo pasaldia bereizi behar dugu hurrengo pasaldi guztietatik.

Aginduen atzipen-denbora:

begiztaren lehenengo pasaldian: $2 \times 10 + 5 \times 1 = 25$ ziklo

begiztaren hurrengo pasaldietan: $3 \times 10 + 4 \times 1 = 34$ ziklo

Begizta, guztira, 200 aldiz errepikatzen denez, lehenengo pasaldia kenduta 199 pasaldi gehiago geratzen dira. Hala, aginduen atzipen-denbora, guztira:

Begiztaren barruan, aginduak: $25 + 34 \times 199 = 6791$ ziklo

Datuei dagokienez, berriz, begiztaren barruan `A[i]` eta `A[i+1]` osagaiak atzitzen dira. A bektorearen osagai guztiak 1 bankuan daudenez, nahiz eta prozesadoreak haien helbideak aginduen helbideen artean sortu, aginduen helbideek ez dituzte 1 bankuko tartekatze-bufferretako balioak aldatzen, aginduak 0 bankuan daudelako. Esate baterako, begiztaren lehenengo pasaldian (`i=0`) A bektorearen `A[0]` eta `A[1]` osagaiak (1280 eta 1281 hitzak, hurrenez hurren) atzitu behar dira bakarrik, baina, 4 modulu daudenez, bi osagai horiekin batera beste bi osagai ere irakurri dira eta tartekatze-bufferretan kargatu; zehazki, begiztaren lehenengo pasaldian `A[0]`, `A[1]`, `A[2]` eta `A[3]` osagaiak batera irakurri dira, lauak 1 bankuko 64 barne-helbidean daudelako, 0, 1, 2 eta 3 moduluetan hain zuzen. Hala, begiztaren bigarren pasaldian (`i=1`), `A[1]` eta `A[2]` osagaiak atzitu behar direnean, 1281 eta 1282 hitzak, hain zuzen, jadanik daude tartekatze-bufferretan eta ziklo bana bakarrik behar da horiek atzitzeko. Begiztaren hirugarren pasaldian (`i=2`), berriz, A[2] eta A[3] osagaiak atzitu behar direnean, 1282 eta 1283 hitzak, hain zuzen, jadanik daude tartekatze-bufferretan eta orain ere ziklo bana bakarrik behar da horiek atzitzeko. Begiztaren laugarren pasaldian (`i=3`), berriz, A[3] eta A[4] osagaiak atzitu behar dira, 1283 eta 1284 hitzak, hain zuzen; lehenengoa jadanik dago tartekatze-bufferrean eta orain ere ziklo bat beharko da hori atzitzeko, baina bigarrena beste barne-helbide batean dagoenez (1 bankuko 65 barne-helbidean), memoria nagusia atzitu behar da berriro, eta 10 ziklo

behar dira hori atzitzeko. Eskema batean islatu dezakegu A bektorea atzitzeko helbideen segida, ea portaera errepikakorra ondorioztatzen dugun.

begiztaren pasaldia	Tartekatze-bufferren edukia **(atzitu beharreko osagaiak)** modulua M0	M1	M2	M3	ziklo-kopurua
i=0	**A[0]**	**A[1]**	A[2]	A[3]	10 + 1
i=1	A[0]	**A[1]**	**A[2]**	A[3]	1 + 1
i=2	A[0]	A[1]	**A[2]**	**A[3]**	1 + 1
i=3	A[0] **A[4]**	A[1] A[5]	A[2] A[6]	**A[3]** A[7]	1 + **10**
i=4	**A[4]**	**A[5]**	A[6]	A[7]	1 + 1
i=5	A[4]	**A[5]**	**A[6]**	A[7]	1 + 1
i=6	A[4]	A[5]	**A[6]**	**A[7]**	1 + 1
i=7	A[4] **A[8]**	A[5] A[9]	A[6] A[10]	**A[7]** A[11]	1 + **10**
i=8	**A[8]**	**A[9]**	A[10]	A[11]	1 + 1
i=9	A[8]	**A[9]**	**A[10]**	A[11]	1 + 1
i=10	A[8]	A[9]	**A[10]**	**A[11]**	1 + 1
i=11	A[8] **A[12]**	A[9] A[13]	A[10] A[14]	**A[11]** A[15]	1 + **10**
i=12	**A[12]**	**A[13]**	A[14]	A[15]	1 + 1

Agerikoa da, beraz, bosgarren pasalditik aurrera (`i=4`), begiztaren pasaldiak launaka multzokatu ditzakegula: lauetatik, azkenekoan (1 + 10) ziklo behar dira osagaiak atzitzeko, barne-helbide desberdinetan daudelako; beste hiruretan, berriz, (1 + 1) ziklo bakarrik behar dira, osagaiak tartekatze-bufferretan daudelako jadanik. Begiztaren lehenengo lau pasaldiak salbuespena dira: lehenengoan (10 + 1) ziklo behar dira, hori baita A bektorearen osagaiak atzituko diren lehenengo aldia; hurrengo atzipenak, berriz, besteen berdinak dira: bigarren eta hirugarren pasaldietan (1 + 1) ziklo behar dira, eta laugarren pasaldian, (1 + 10) ziklo.

Hau da, begiztaren pasaldi-kopurua (200) launaka multzokatzen badugu (zati lau egiten badugu, alegia: 50), orduan lortzen dugu zenbat aldiz errepikatzen den aurreko atzipen-segida eta guztira behar den ziklo-kopurua kalkulatu ahal izango dugu: hala, 49 multzotan [(1 + 1) × 3 + (1 + 10)] = 17 ziklo behar dira A bektorearen osagaiak atzitzeko, eta lehenengo multzoan [(10 + 1) + (1 + 1) × 2 + (1 + 10)] = 26 ziklo.

Laburbilduz, begiztaren barruan A bektorearen osagaiak atzitzeko behar den ziklo-kopurua:

Begiztaren barruan, A bektorea: 26 + 17 × 49 = 859 ziklo

Azkenik, begiztaren ondoren dauden agindua (`store`) eta datua —B[0] osagaia— atzitzeko, (1 + 10) = 11 ziklo behar dira. Izan ere, azken agindua 80 helbide logikoan dago; taulan ageri ez den arren, agindu honi dagozkion balio guztiak kalkulatu ditzakegu, eta ikusiko dugu azken agindua ere 0 orri logikoan dagoela, eta 80 desplazamendua dagokiola. Hala, 2 orri fisikoan egongo da, eta 2128 helbide fisikoan (266 hitza, alegia); 0 bankuan egongo da, 66 barne-helbidean eta 2 moduluan. Horregatik, agindu hori aurreko biekin batera irakurriko da tartekatze-bufferretan, eta behar den unean 1 ziklo bakarrik beharko da, tartekatze-bufferrean atzituko delako. Datuari dagokionez, berriz, 10 ziklo behar dira, begiztaren barruan, azken pasaldian, A bektorearen `A[200]` osagaia atzitu delako (9792 helbide logikoa, 11840 helbide fisikoa, 1480 hitza, 1 bankuko 114 barne-helbidea, 0 modulua) eta azken aginduan, berriz, B[0] osagaia atzitu behar da (10000 helbide logikoa, 12048 helbide fisikoa, 1506 hitza, 1 bankuko 120 barne-helbidea, 2 modulua), eta bi osagai horiek barne-helbide berean ez daudenez, tartekatze-bufferretan dauden balioak ez dira baliagarri, eta memoria nagusian idatzi behar da zuzenean B[0] osagaiaren helbidean. Hala:

Begiztaren ondoren: 1 + 10 = 11 ziklo

Memoria nagusia atzitzeko, beraz, guztira, 892 ziklo behar dira:

$t_{atzip.}$ = 22 ziklo + 6791 ziklo + 859 ziklo + 11 ziklo = 7683 ziklo

Eta itzulpen-denbora eta atzipen-denbora batuz:

1866 ziklo + 7683 ziklo = 9549 ziklo

Hau da, programa osoa exekutatzean sortzen diren helbide guztiak itzultzeko eta memoria nagusia atzitzeko, 9549 ziklo behar dira.