KONPUTAGAILUAREN OINARRIZKO EGITURA

IÑAKI ALEGRIA

ELHUYAR INFORMATIKA-TALDEA

udako euskal unibertsitatea

ELHUYAR INFORMATIKA-TALDEA

KONPUTAGAILUAREN OINARRIZKO EGITURA

IÑAKI ALEGRIA LOINAZ. Informatikan Lizentziatua.
Donostiako Informatika-Fakultateko irakaslea.

(c) Copiryght : IÑAKI ALEGRIA - U.E.U.
(c) " : ELHUYAR KULTUR ELKARTEA
ISBN : 84-86158-27-3
LEGE GORDAILUA: SS 462-85
INPRIMATEGIA: AURREZKI KUTXA MUNIZIPALA

Azalaren diseinua: Julio Pardo

AURKIBIDEA

	Orr.
1.- SARRERA	5
1.1.- SARRERA	7
1.2.- KALKULU-MAKINEN GARAPENA	11
1.3.- KONPUTAGAILUEN BELAUNALDIAK	15
2.- OINARRIZKO KONTZEPTUAK	19
2.1.- HARDWARE-A ETA SOFTWARE-A	21
2.2.- KONPUTAGAILUAREN OINARRIZKO BARNE-EGITURA	21
2.3.- INFORMAZIO KONPUTAGAILUAN	23
2.4.- KONPUTAGAILUAREN MAILAK	26
2.5.- INFORMATIKARIAK	31
3.- KONPUTAGAILUAREN EGITURA	33
3.1.- SARRERA	35
3.2.- KONPUTAGAILU-MOTAK EGITURAREN ARAUERA	35
3.3.- MEMORIA NAGUSIA	41
3.4.- UNITATE ARITMETIKO-LOGIKOA	51
3.5.- KONTROL-UNITATEA	55
3.6.- MIKROPROGRAMAZIOA	59
3.7.- KONPUTAGAILU BATEN BARRUKO FUNTZIONAMENDUA	61
3.8.- ARIKETA EBATZIAK	66
4.- DATUEN ADIERAZPIDEA	73
4.1.- SARRERA	75
4.2.- DATUEN SAILKAPENA	76
4.3.- ZENBAKIEN ADIERAZPIDEA	78
4.4.- DATU ALFANUMERIKOAN ADIERAZPIDEA	98
4.5.- DATU-EGITURAK	99

5.- AGINDUAK KONPUTAGAILUAN 107

5.1.- MAKINA LENGOAIA/LENGOIA SINBOLIKOAK 109

5.2.- ITZULTZAILEAK 111

5.3.- AGINDU-MOTAK 113

5.4.- AGINDUAK MAKINA-LENGOAIAZ 122

5.5.- AGINDUAK MIHIZTADURA-LENGOAIAZ 133

6.- SARRERA/IRTEERAKO OSAGAIAK 141

6.1.- SARRERA 143

6.2.- PERIFERIKO ETA EUSKARRIEN SAILKAPENA 144

6.3.- SARRERAKO PERIFERIKOEN DESKRIPZIOA 146

6.4.- IRTEERAKO PERIFERIKOEN DESKRIPZIOA 152

6.5.- BITARTEKO MEMORIAK 154

6.6.- UNITATEEN ARTEKO KONPARAKETA 162

6.7.- PERIFERIKO/KONPUTAGAILU KONEXIOA 163

6.8.- ETENAK 167

7.- SISTEMA ERAGILEA 173

7.1.- KONTZEPTUA ETA OSAGAIAK 175

7.2.- HASIERAKO PROGRAMA-KARGATZAILEA 178

7.3.- GAINBEGIRALEA. SISTEMAREN KARAGATZAILEA 178

7.4.- EGINKIZUN-KUDEATZAILEA. KONTROL-SENTENTZIAK 179

7.5.- ITZULTZAILEAK ETA ESTEKATZAILEA 180

7.6.- LIBURUTEGIAK 183

7.7.- SISTEMA ERAGILEAREN FUNTZIONAMENDUA 186

7.8.- PROZESU-MOTAK 188

7.9.- BESTE ZENBAIT KONTZEPTU 198

1.- S A R R E R A

1.1.- SARRERA

1.2.- KALKULU-MAKINEN GARAPENA

1.3.- KONPUTAGAILUEN BELAUNALDIAK

1.1.- SARRERA

INFORMATIKA I izeneko asignaturaren helburu nagusienetariko bat Informatikari dagozkion kontzeptuak aztertzea eta ulertaraztea da.

Baina zer esan nahi dugu Informatika hitzarekin? Informatika informazioaren tratamendu automatikoaren zientzia dela dio definizio hedatuenak beraz, informazioarekin eta makinekin zerikusi handia du.

Beste definizio pragmatikoago bat ondoko hauxe dugu: konputagailuak maneiatzeko eta diseinatzeko tekniken multzoa da. Eta edonork daki Informatika eta konputagailuak edo ordenadoreak guztiz lotuta daudela.

Saia gaitezen ba konputagailu bat zer den definitzen:

- Teknologiari begira tresna digital eta elektronikoa da, hau da, osagai digitalek osatzen dute eta erabil dezakeen informazioak diskretua edo kuantifikagarria izan behar du.

- Erabilpenei begira, helburu orokorreko makina automatikoa dugu, hots, bere erabilpenen muga bakarrak aipatutako ezaugarriei dagozkienak dira, baina horren gain edozein kalkulu-mota edo eginkizun burutu dezake (helburu orokorra) eta, zer eginkizun burutu behar den eta nola esanez gero, konputagailuak inoren laguntzarik gabe (automatikoki) burutuko du. Baina kalkulu hauek oso arin, informazio-kantitate handia erabiliz eta errorerik gabe, burutzen ditu. Beraz, abiadura eta edukiera handia eta fidagarritasuna dira konputagailuen ezaugarri garrantzitsuenak.

Aurreko definizioa aztertuz konputagailuari zeregina zehaztearekin ez dugu aski, zeregin hori nola burutzen den ere zehaztu behar bait da. Zeregina edo eginkizunaren zehaztapena Programa baten bidez egiten da. Horregatik konputagailua programagarria dela esaten da eta programak egiteko moduak eta teknikak aztertzen dituen informatikaren arloari Programazioa deritzo.

1.1.1.- KONPUTAGAILUAREN EZAUGARRIAK

Honaino konputagailu bat definitzeko kontzeptu garrantzitsuenak zehaztu ditugu baina, zein ezaugarrik finkatzen dituzte konputagailuen arteko diferentziak? Hain zuzen ere konputagailuek bereizten dituzten ezaugarriak ondoko hauek dira:

ABIADURA: Ezaugarri honek kalkulatzeko ahalmena adierazten du, hau da, zenbat oinarrizko eragiketa burutzen duen denbora-unitateko. Konputagailu guztiek abiadura ikaragarria daukate es-

kuzko kalkuluarekin edo makina mekanikoz baliaturiko kalkuluarekin konparatuz, segundu batean milioi bat eragiketa baino gehiago burutu bait dezakete. Hala eta guztiz ere, konputagailuen artean arinagoak eta motelagoak daude.

EDUKIERA: Konputagailuek egikaritzen dituzten eragiketek datuak maneiatzen dituzte eta, aipatu dugunez, informazio-kopuru handia tratatzeko gai izatea da konputagailuen ezaugarri garrantzitsu bat. Edukierak konputagailuan zenbat informazio gorde edo biltegiratu daitekeenaren neurria adierazten digu. Ikusiko dugunez, informazio hau Memoria Nagusian edota Kanpo-Memorian gorde daiteke. Memoria Nagusiaren edukiera K-tan edo M (Mega)-tan neurtzen da konputagailuaren arabera eta Kanpo-Memoriaren edukiera M-tan, K batek mila karaktere eta Mega batek milioi bat karaktere edukitzen dutelarik.

SEGURTASUNA ETA FIDAGARRITASUNA: Konputagailuak eragiketa asko denbora laburrean burutzen badu ere, bere errore-marjina eskuzko lanarena adinakoa izango balitz ez litzateke oso erabilgarria izango. Horregatik, konputagailuen beste funtsezko ezaugarri bat segurtasuna da. Gaur egungo konputagailuak oso fidagarriak dira, hau da, oso seguruak. Horrek ez du esan nahi ez dezaketenik okerrik egin, baizik eta erroreren bat gertatzekotan konputagailuaren mekanismoen bidez igertzen eta askotan zuzentzen dela. Gertatzen diren errore gehienak giza akatsen ondorioak dira (programari,datuei edo sistema eragileei dagozkienak).

MOLDAGARRITASUNA: Konputagailuak helburu orokorretarako egindako makinak direla esan dugu, beraz, eragiketa logiko edota aritmetikoen sekuentzia batean deskonposa daitekeen edozein eginkizun egikaritzeko gai dira. Dena dela, konputagailu batzu problema-mota zehatz batera zuzendurik daude, beste problema-mota batzu burutzen motelagoak direlarik.

1.1.2.- KONPUTAGAILUAREN ERABILPENAK

Konputagailuak dagokien helburu orokorra dela medio, programa daitekeen edozein eginkizun egikaritzeko gai direla esan dugu. Hona hemen gaur egun konputagailuak dituen erabilpenik hedatuenak:

- KALKULU ZIENTIFIKOA: Lehenbiziko konputagailuen helburua mota honetako problemak ebaztea zen. Gaur egungo konputagailu handienak kalkulu zientifikoan erabiltzen dira (meteorologia, aerodinamika, petrolioaren prospekzioa, ...).

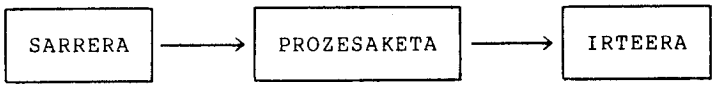
- ADMINISTRAZIO-LANAK: Konputagailuak, hasiera batean, erabilpen-mota honetara zuzendurik ez bazeuden ere, gero eta gehiago erabili ziren arlo honetan eta gaur egun administrazio- edo jestio-lana (kontabilitatea, sal-erosketak, errenta-deklarazioak, banku-kontuak) konputagailuaren erabilpen hedatuentzat jo dezakegu.

- INDUSTRIAN: Azken urteotan garapen handia izan dute mota honetako erabilpenek. Robotak, makinen kontrola eta antzeko erabilpenak gero eta gehiago hedatu dira, batzutan helburu orokorra ez duten mikrokonputagailuen bidez egiten badira ere. Adibidez, kotxeen muntai kateen kontrolagailuak, robotak, berogailuen kontrolak.

- EGUNEROKO BIZITZARI DAGOZKIONAK: Gaur egun oso hedaturik ez badaude ere, nork ez du entzun noizbait MIKRO hitza? Mota honetako erabilpen arruntenak etxeko mikro baten bidez egin daitezke. Adibidez, denda txiki batean fakturak egiteko sistema edo etxeko liburuen fitxategia mota honetako erabilpenak dira.

1.1.3.- KONPUTAGAILUAREN EGITURA

Informazio-sitema guztiek 1.1 irudian azaldutako egitura daukate.

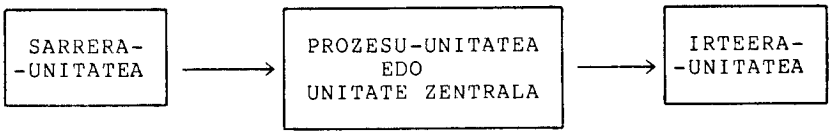


1.1 irudia.- INFORMAZIO-SISTEMEN EGITURA

Sarrerako informazioa datuak dira eta irteeran lortutakoa emaitzak .

Informatika informazioaren tratamendu automatikoa denez gero, konputagailuaren bidezko informazio-sistema batek aurrean ikusiriko egituran oinarriturik dago. Beraz, konputagailu batek egiten duena, sarreran emandako datuak prozesatzea eta burutu nahi diren emaitzak kalkulatzea da. Horretarako, prozesaketa funtsezkoa da. Konputagailuari prozesaketaren zehaztasunak programaren bidez ematen dizkiogu eta beraz, datuak hartu eta emaitza lortzeko egikaritu behar den eragiketa-sekuentzia zehatz-mehatz adierazi behar diogu programan.

Aurrerago zehatzago aztertuko badugu ere, konputagailu baten oinarrizko egitura 1.2 irudian agertzen dena dela esan dezakegu.



1.2 irudia.- KONPUTAGAILUAREN EGITURA

- Sarrera-unitatean datuak ezartzen dira.

- Irteera-unitatean, prozesaketa bukatzean, emaitzak kanporatuko dira.

- Prozesu-unitatean, egikaritu behar den programa gordetzeko unitatea dago eta programak zehazten dituen eragiketak burutzeko kalkulu-unitatea ere bai. Beraz, bi parte daude: bata Informazioa gordetzekoa edo Memoria eta bestea kalkuluak burutzekoa edo Prozesu-Unitate Zentrala (CPU).

1.1.4.- ERAGIN SOZIALAK

Azken urteotan gertaturiko informatikaren garapenak gizartearen aldaketa eragin du. Hona hemen gaur ikus ditzakegun ondorio garrantzitsuenak:

LANGABEZIAREN GAINEKO ERAGINA. Azken urteotan gero eta lan gehiago egiten ari da konputagailuaren bidez. Orain dela gutxi, lantegietan esaterako, jende asko behar zen jestio-lanak aurrera eramateko. Hau aldatzen ari da eta lan aspergarrienak ordenadorea erabiliz egiten dira. Horrela lan gehiago eta azkarrago burutzen da langile-kopuru berarekin, produktibitatea handitzen delarik.

Orain, denbora gehien behar duten lanak makinak egiten baditu, langile asko eta asko lanik gabe gelditzen dira. Ondorio logikoena denek lan gutxiago egitea bada ere, ez da horrela gertatzen: Sobran gelditzen direnak kalera bidaltzen dituzte eta geratzen direnek lehen baino lan gehiago egin behar izaten dute askotan.

Maiz esan ohi da: "Informatikak lan astun, aspergarri eta arriskutsuenak kendu eta beste lan kreatiboago batzuk sortzen ditu". Hori ere ez da guztiz egia: Alde batetik makinak askotan ez ditu benetako lan gogorrak egiten, gizonak baizik; eta beste aldetik informatizazioak sortutako lanpostu berriak ez ditzake edonork bete, horretan eskolatuak baizik. Normalean lanik gabe geratzen direnen portzentaia handi bat adinekoek eta eskola gutxikoek osatu ohi dute. Horrez gain, pertsona horiek lanpostu berriak lortzeko zailtasun handiak aurkitzen dituzte.

BIZITZA PRIBATUAREN GAINEKO ERAGINA. Ordenadoreak izugarrizko informazio-kantitate handiak erabil ditzake, eta oso azkar gainera. Horregatik, ordenadore batean jende guztiaren datuak gorde litezke, zentralizaturik.

Zer datu-mota? Edozein: helbidea, lana, adina, zenbat irabazten duen, bere kontu korrontea, zer erosten duen, non, antezedenteak, eta abar eta abar.

Gu konturatu gabe, guri buruzko makina bat datu sartzen da gaur egun konputagailuetan.

Bizitza pribatuaren defentsa bakarra lege egoki bat da: herri askotan atera dira horretarako legeak. Arazoa hauxe da: erabiliko ote du Estatuak bere datu biltzeko ahalmena gizartea kontrolatzeko? Horrela ba, botere handiko Estatu polizialerantz joko dugu.

INFORMAZIOAREN FIDAGARRITASUNA. Batzutan, gauza bat egia dela frogatzeko, entzun ohi da: "Hala da, konputagailuak esan du eta. Konputagailua ez da erratzen".

Badakigu konputagailuak eragiketa asko denbora laburrean egin dezakeela eta errorerik gabe gainera. Gizakiak, lan errepikakor bat egiten duenean, huts ugari egiten du. Makinak ez, ordea. Makinak ez du pentsatzen, programa betetzera mugatzen da. Zirkuitu elektronikoetan okerren bat gertatuz gero ere, badaude algoritmoak oker hori harrapatu eta zuzentzeko.

Honela bada, ordenadoreak zeharo fidagarriak dira. Ezin dute hutsik egin. Hortik sortzen da mitoa: "Konputagailua ezin erratu da; berak esandakoa egia hutsa da". Oso kontutan eduki behar dugu, alabaina, konputagailuak programa bat betetzen duela, eta ongi gainera. Eta programa hori zuzena ez bada? Emaitzak faltsuak izango dira, noski.

Beraz esan dezakegu ordenadorea erabat fidagarria dela agindua betetzen, baina bere fidagarritasuna, beti ere, programazioaren zuzentasunaren mende dagoela.

1.2.- KALKULU-MAKINEN GARAPENA

Kalkulurako dispositibo azkar eta zehatz baten bilaketatik sortu da ordenadorea. Baina ordenadorearen sorkundea posible egin duten baldintza asko izan dira. Hona hemen batzu: azken mendeko garapen teknologikoa, kontzentrazio finantzarioa, konputazioari buruzko oinarrizko teorien aurrerapena, etab.

1.2.1.- LEHENENGO PAUSOAK

ABAKOA da ezagutzen dugun kalkulurako tresnarik zaharrena. Harriz edota bolaz osaturik zegoen eta zenbakien balioa, haiek okupatzen zituzten posizioen bitartez adierazten zen. Grezian erabilia Cristo-ren aurreko V.mendean, Confucio-ren garaian Txinan eta baita Colon aurreko zibilizazio amerikarretan ere.

Abakoa aintzinako tresna izan arren, gaur egun ere erabiltzen da bere erarik aurreratuenetan ekialdeko herrialde askotan. Matematikari hinduek asmatutako errepresentazio-sistema hamartarra arabiarren bitartez Europara heldu zenean, abakoaren erabilpena baztertu egin zen mendebaldeko gizartean.

John Napier (edota Neper) matematikari eskoziarrak (1550-1617), zenbaki-kalkulua erraztu nahiz, makilatxo zifratuz baliatuz biderkaketak eta zatiketak egiteko prozedura bat asmatu zuen.

1.2.2.- MAKINA MEKANIKOAK

Errenazimentutik aurrera, merkataritzaren, nabigazioaren eta astronomiaren garapenarekin batera, berrikuntza teknologikoaren aldeko giroa sortu zen. Garai honetakoak dira aurreneko erloju mekanikoak. Lehen kalkulagailu mekanikoa 1642. urtean agertu zen, Pascal-ek asmatuta. Makina honen tamaina tabako-kartoai batena adinakoa zen eta engranaje eta gurpilez osaturik zegoen. Zerotik bederatzira zenbakitutako engranaje-sail bat zuen. Gurpil batean bira osoa egin ondoren, bederatzitik zerora pasatzeak gurpil honen ezkerrekoan posizio bateko saltoa eragiten zuen, beraz, bururakoa automatikoki gehitzen zion ezkerreko zifrari. Bere funtzionamendu-printzipioa, egungo autoen kilometro-kontagailuaren printzipio bera zen. Pascal-ek, kenketak egiteko, osagarrien batuketa erabili zuen, beraz, puntu honetan ordenadoreen diseinugileei aurreratu egin zitzaiela azpimarratzekoa da.

Leibnitz-ek (1646-1716), 30 urte geroago, oinarrizko lau eragiketa matematikoak egiten zituen beste makina bat asmatu zuen. Leibnitz-en makinak, Pascal-en makinaren hobekuntza bat suposatu zuen. Pascal eta Leibnitz-en makinek oztopo handiak aurkitu zituzten zabalkuntzarako, oso konplexuak bait ziren eskuz egiteko, garaiko teknologia kaskarrarekin. Ordurako, Iraultza Industriala gertatu gabea zen oraindik.

Makina hauek ez genitzake automatikotzat har, eragiketa guztiak burutzeko, operatzailaren presentzia ezinbestekoa bait da.

Jaquard-ek (1752-1834), XIX. mendearen hasieran, ehundegi automatiko bat asmatu zuen, makina kontrolatzeko programak txartel zulatuen bidez sartzen zirelarik. Jaquard-en ehundegiak arrakasta handia eduki zuen eta ehungintzarako aurrerapen galanta izan zen.

Espezialista askorentzat ordenadorearen historia, Charles Babbage XIX. mendeko matematikari eta asmatzaile ingelesarekin hasi zen. Egungo ordenadoreen oinarriak Babbagek aurresan zituen. Bere garaiko logaritmo-taulak hobetzeko asmoz, 1822. urtean, kenketa-makina asmatu zuen.

Hamar urte geroago beste ideia hobe bat izan zuen, alegia makina analitikoa. Kenketa-makina prozesu bakar baterako tresna bazen ere, Babbagek xede orokorragoko makina bat eraikitzea ere pentsatu zuen.

Makina hori edozein problema matematiko egiteko gai izateaz aparte, aurreko emaitzetan oinarrituz, makina bera bere agindu-se-

kuentzia aldatzeko gai izatea izan zen Babbageren helburua. Beraz, makina automatikoa eta programagarria zen. Diseinuz, makina analitikoak bost unitate zituen:

- Sarrera-unitatea: Datuak eta aginduak sartzeko.
- Memoria: Datuak eta bitarteko emaitzak gordetzeko.
- Kontrol-unitatea: Eragiketen exekuzio-ordena erregulatzeko.
- Unitate Aritmetiko-Logikoa: Eragiketak egiten dituena.
- Irteera-unitatea: Emaitzak kanporatzen dituena.

Babbagek, Jaquarden txartel zulatuen sistema hartu zuen problemaren datuak eta programaren aginduak makinaren barnean sartzeko.

Tamalez, garai hartako teknologia eta gizartearen premiak Babbageren teoriak baino ahulagoak ziren eta makina hau ez zen bukatu. Babbageren ideiak erabat gauzatu ez baziren ere, bere ekarpena funtsezkoa izan zen eta gure ordenadoreak makina analitikoaren antzeko eskema batean oinarritzen dira. Babbageren makina, automatikoa zen, baina kanpo-programaduna. Kalkulagailu mekanikoak eta Babbageren makina dira kalkulu-tresnen historiaurrea.

XIX. mendean, Iraultza Industrialaren ondorioz, informazio-pila handiak tratatu beharra sortzen da. EEBB-etako 1880. urteko, zentsua egin ondoren (7 urte t'erdi behar izan ziren eskuz burutzeko), biztanleriaren hazkundea zela medio, zentsua analizatzeko orduko bideak desegokiak zirela konturatu ziren Erroldaketa-bulegoan.

1879. urtean, Herman Hollerithek asmatutako sistemak txartel zulatuak erabiltzen zituen erroldaketarako. Baiezko erantzunak txartelean zulo batez errepresentatzea pentsatu zuen eta ezezkoak zulorik gabe. Honela sortu zen informazioaren kodeketa bitarra.

Txartel zulatuetan, sexua, arraza, adina, etab., zuloen bidez gordetzen ziren. Hollerithen makinak 50-80 txartel zulatu irakurtzen zituen minutuko, eta zirkuitu elektrikoen bidez, kontagailu bat pizten zuen. Hollerithen sistemaren laguntzarekin 1890. urteko zentsua (80 urtekoan baino 10 milioi biztanle gehiago) bi urte t'erdian bete zen.

Hollerithek enpresa bat eraiki zuen makina hauek saltzeko. Urteak aurrera joan ziren eta enpresa hau beste batzurekin elkartu zen eta elkarte horretatik IBM enpresa jaio zen.

XX. mendean gizarte-baldintzek posible egiten dute konputagailuaren sortzea, eta honekin, kalkulatzeko tresnen historiaurretik,

Aintzinatera iragango gara.

1.2.3.- MAKINA ELEKTRONIKOAK

1937. urtean, Howard Aiken, Harward Unibertsitateko irakaslea, kalkulagailu guztiz automatikoa diseinatzen hasi zen IBMren laguntzarekin. Zazpi urte beranduago, Babbageren ideietan oinarrituz, MARK I kalkulagailua, historiako lehen kalkulu-makina elektronikoa bukatu zuen.

MARK I, Babbageren ideien lehen gauzatzea zen: teknologiari mende bat baino gehiago kostatu zitzaion Babbageren teoriak egia bilakatzeko adina garatzea. MARK I konplexua eta tamaina handikoa zen (16 m-ko luzera eta 2 m-ko altuera). Taulak kontsultatzeko eta oinarrizko eragiketa matematikoak egiteko gai zen. Sistema hamartarra erabiltzen zuen oinarri gisa eta 23 digituko zenbakiak erabiltzeko gauza zen. Elementu elektromekanikoz osatua zegoen eta, beste zenbait arrazoiren artean, bere atal higikorren inertziagatik, oso geldia zen: bi zenbaki batzeko 0,2 s behar zuen eta 8 digituko hiru zenbakiren biderkaketa egiteko 1 s.

Etengailuen ordez huts-balbulak erabiltzeak, lehen kalkulagailu guztiz elektronikoa, ENIAC izenekoa, eman zuen. MARK I baino bi urte geroago bukatu zen eta bere eraiketak aurrerapen handia suposatu zuen teknologiaren aldetik.

ENIACek, eragiketa guztiak, Irteera-Sarrerakoak ezik, 16 motatako huts-balbulez osatutako zirkuitu elektronikoaren bidez egiten zituen 1.600 m2 okupatuz, 30 t-ko pisua zuen eta energia elektriko asko kontsumitzen zuen (martxan jartzen zen bakoitzean Filadelfiako argiek sekulako beheraldia jasaten zuten). ENIACen emaitzak itzelak izan ziren, MARK I baino askoz ere azkarragoa bait zen: MARK I-en astebeteko lana, ENIAC konputagailuak ordubetean burutzen zuen.

Baina ENIACen programazioa ehundaka larako entxufatuz eta etengailu batzu aktibatuz egiten zen. Problema edo lan bakoitzak zirkuituaren konfigurazio berezia behar zuen eta beste problema desberdinetarako programatzeak barne-zirkuituen konfigurazioaren aldaketa eskatzen zuen; lan guzti hau geldia eta astuna gertatzen zen eta horregatik ezin da makina programagarritzat hartu.

1946. urtean Von Neumann-ek, konputagailuen diseinuari buruzko bere teoriak aurkeztu zituen. Von Neumann-en kontzeptu adierazgarriena, programa ere memoria barnean gordetzea izan zen.

Von Neumann-ek agindu-sail bat kableatzea eta agindu hauek kontrol zentral baten menpean exekutatu edo egikaritzea proposatu zuen. Horrez gainera, eragiketak kontrolatu behar zituen eragiketa-kodeak datu bezala kode bitarrean gordetzea proposatu zuen. Honela, Von Neumann-en makinak ez zuen zirkuituen konfigurazioaren

	MAKINA	URTEA	EZAUGARRIAK
ESKUZKOAK	Abakoa		Lehenengo kalkulurako tresna
	Napieren taulak	1600	Adierazpide hamartarra Biderkaketak erraztu
MEKANI-KOAK	Pascalen makina	1642	Bururakoaren automatizazioa Batuketak eta kenketak bakarrik Emaitzaren memoria Adierazpide hamartarra
	Leibnitzen makina	1694	Pascalen makinan oinarriturik Biderkaketak eta zatiketak
	Babbageren kenketa-makin	1822	Logaritmoak metodo errepikakor baten bidez
	Babbageren makina analitikoaren projektua	1840	Automatikoa Gaur egungo konputagailuen egitura Ahalmen handikoa (gehiegi) Ez zen eraiki
	Hollerith-en sistema	1879	Informazio prozesamendua,ez bakarrik kalkulu aritmetikoa
ELEKTRO-MEKANIKOA	MARK I	1944	Makina analitikoaren ezaugarriekin eraikia Helburu orokorra
ELEKTRO-NIKOAK	ENIAC	1946	MARK I baino askoz arinagoa Ez programagarria(ez helburu orokor)
	Von Neumann-en ideiak	1946	Programa memorian Adierazpide bitarra eta kodeketa Programaren aldaketak egikaritzean

1.1 taula.- KALKULU-MAKINEN GARAPENA

aldaketarik behar programa berri bakoitzarentzat eta aginduak datuak bezain laster prozesa zitzakeen. Bestalde, programak bere burua aldarazi zezakeen. Datu bezala gordetako aginduak aritmetikoki erabil bait zitzakeen. Posibilitate honek ematen dio Von Neumann-en makinari bere unibertsaltasuna eta horregatik zenbait autorek 1949. urtea, hau da, Von Neumann-en ideietan oinarritutako lehen makina sortu zen urtea, ematen dute informatikaren jaioturtetzat.

1.3.-KONPUTAGAILUEN BELAUNALDIAK

Von Neuman bere ideietan oinarrituriko makina bat eraikitzen hasi zen EDVAC projektuan parte hartuz, baina, problema teknikoak izan zirela medio, EDVAC ez zen lehenengo konputagailua izan, Ingalaterran garaturiko EDSAC projektua 1949.ean lehenengo konputagailua izan bait zen. EDSAC-ek konputagailuen lehenengo belaunal-

diari hasiera ematen dio. Eraikitako lehen konputagailu komertziala 1951.ean plazaratu zen. UNIVAC I zuen izena eta EEBB-tako Erroldaketa Bulegoan instalatua izan zen. Bateratsu 50.eko hamarkadan, IBMren 600 eta 700 serieak ere aurkeztu ziren.

Hamarkada horretako konputagailuek osatzen dute lehen belaunaldia, beren ezaugarri nagusia zirkuituetako oinarrizko osagai bezala huts-balbulak erabiltzea izanik.

Bigarren belaunaldia 1958. urtean hasten da. Ordurako transistorea asmatuta dago eta huts-balbulen ordez erabiltzen da konputagailuen eraikuntzan. Abantaila handia suposatu zuen aldaketa honek. Transistorea txikiagoa eta merkeagoa delako eta kantitate handitan egin daitekeelako. Honez gainera, energia elektriko gutxiago kontsumitzen du. Ondorioz, konputagailu txikiagoak, merkeagoak, ahaltsuagoak eta azkarragoak lortu ziren. Honela hasten da serieko produkzioa eta konputagailuen erabilera gero eta arlo gehiagotara zabaltzen da. Oso garrantzi handia izan zuen programatzeko mihiztadura-lengoaien sorrerak, programazioa erraztu bait zuen.

1964. urtean IBM 360-aren agerpenarekin, hirugarren belaunaldia hasten da. Transistore bakunak elkarren artean konektatzen ibiltzea baino Si edo Ge-zko pastila batean biltzea interesgarriagoa zela ikusi zen eta honela sortu ziren zirkuitu integratuak edo CHIPak.

Hirugarren belaunaldiko konputagailuak aurrekoak baino txikiagoak, azkarragoak eta fidagarritasun handiagokoak dira. Multiprogramazioa eta teleprozesua agertzen dira eta mikrokonputagailuak gestioen munduaz jabetzen dira. Hauen erabilera errazteko goimailako lengoaiak sortzen dira, programaziorako aurrerapen izugarria izango direlarik.

70.eko hamarkadaren hasieran jaioko litzateke laugarren belaunaldia batzuren ustez, baina informatikako profesional batzuek ez dute hori onartzen eta, 3. belaunaldiaren aldaketa bat besterik ez dela kontsideratzen dute. IBM 370 eta Burroughs 700 konputagailuak har ditzakegu 4. belaunaldiko makina errepresentagarritzat. Ezaugarriak hauexek dira: Ferritazko nukleoaren ordez memoria elektronikoa erabiltzen da, goimailako lengoaia berri asko ateratzen dira eta Sistema Eragile gero eta ahalmentsuagoak dituzte konputagailuek.

Integrazio-teknikak izugarri aurreratu ziren eta honi esker 1970.ean lehen mikroprozesadoreen komertzializazioa lortu zen. Gaur egun mikrokonputagailuek daramaten mikroprozesadore edo zirkuitu integratua da beraien ezaugarri eta atal nagusia.

Mikrokonputagailuak merkeak, kontsumo txikikoak eta kalkulu-abiadura handikoak dira, orain 15 urte pentsa ezin zitekeen moduan hedatu direlarik.

Zenb	OSAGARRIAK			SOFTWAREA	
	ALU oinarri-zikloa	MEMORIA atzipen-denbora	KANPO MEMORIA	LENGOAIA	SIST.ERAGIL
1	BALBULAK 0,1-1 msg	HODI ELEKTROSTATIKOAK ATZERA LERROAK 1 msg	TXARTEL EDO PAPER ZULATUA	MAKINA-LENGOAIA	BAT ERE EZ
2	TRANSISTOREAK 1-10 mikroseg.	FERRITAZKO NUKLEOAK 1-10 mikroseg.	DISKOAK ZINTAK	MIHIZTADURA LENGOAIA	OINARRIZKO SISTEMA ERAGILEA
3	ZIRKUITU INTEGRATUAK 0,1-1 mikroseg	FERRITAZKO NUKLEOAK 0,1-10 mikroseg	2.goak+ F.M.M. *	GOIMAILAKO LENGOAIAK	SISTEMA ERAGILE OSOA
4	ZIRKUITU INTEGRATUAK 0,1-1 mikroseg	ERREGISTRO ERDIEROALEAK 0,1 mikroseg.	3.goak+ arinago	LENGOAIA ALGORITMIKOAK ETA ERAZAGUTZAILEAK	SISTEMA ERAGILE SOFISTIKATU **

* ferritazko memoria masiboak
** editoreak, alegiazko memoria eta paralelismoa

1.2 taula.- KONPUTAGAILUEN GARAPENA

1.3 taulan, lehenengo konputagailu elektromekanikoa, ENIAC, eta gaur egungo mikroordenadore oso famatua (Apple) konparatuko ditugu:

	ENIAC	APPLE
Prezioa	486.000 $	1.200 $
Kontsumoa	100.000 w	100 w
Pisua	30.000 kg	4,8 kg
Erloju-abiadura	100.000 Hz	1.000.000 Hz
Batezbesteko denbora matxura bitartean	12 h	2 urte

1.3 taula.- ENIAC ETA APPLE ARTEKO KONPARAKETA

Kontuan eduki behar da, Apple Mikroordenadorearen ahalmena ENIACek eduki zuen ahalmena baino handiagoa dela.

30 urtetan, ordenadoreak 10 aldiz azkarragoak egin dira, beren tamaina eta energi kontsumoa 1.000 aldiz txikiagoa da eta gaur egun ia edozeinek beretzat eros ditzakeen ordenadore txikiek eskaintzen duten zerbitzua orain dela 20 urte ordenadore handiok

ematen zutenarekin konpara daiteke.

2.- O I N A R R I Z K O K O N T Z E P T U A K

2.1.- HARDWARE-A ETA SOFTWARE-A

2.2.- KONPUTAGAILUAREN OINARRIZKO BARNE-EGITURA

2.3.- INFORMAZIOA KONPUTAGAILUAN

2.4.- KONPUTAGAILUAREN MAILAK

2.5.- INFORMATIKARIAK

Beste gaietan erabili eta garatuko diren zenbait kontzeptu azaltzea da gai honen helburua. Kontzeptu hauek oso arruntak dira informatikaren munduan eta ahalik eta modu errazenean adierazten saiatuko gara.

2.1.- HARDWARE-A ETA SOFTWARE-A

Memorian gordetako programa baten bidez kalkulatzeko makina automatiko, elektroniko eta digitalari konputagailua deritzogu. Baina konputagailu bati dagozkion kontzeptu guztiak bi multzotan bana ditzakegu: alde batetik teknologiari dagozkionak eta bestetik informazioari eta programei dagozkienak. Lehenengo kontzeptuak HARDWARE izenarekin biltzen ditugu, bigarrenak SOFTWARE izenarekin (batzutan TRESNERIA eta PROGRAMERIA hitzak ere erabiltzen dira). Beraz defini ditzagun kontzeptu hauek:

HARDWAREa.- Teknologiari dagozkion edo ikuitzeko moduko osagaien multzoa da. Zirkuituek, transistoreek, motoreek, memoriek, dispositiboek eta abarrek osatzen dute Hardware-a.

SOFTWAREa.- Informazioari dagozkion osagaien multzoan datza. Konputagailuak funtzionatu ahal izateko behar dituen elementu ikusezinek osatzen dute multzo hau, hots, datuek, emaitzek eta batez ere programek.

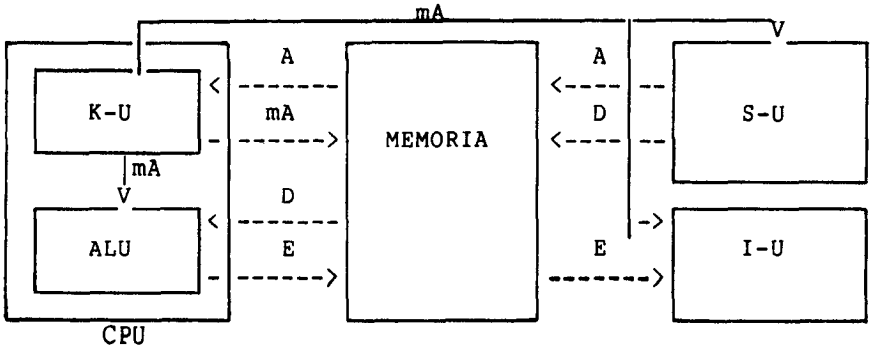
Dena dela, oso kontzeptu kontrajarriak badirudite ere, funtzio bera hardwarearen bidez edo softwarearen bidez egin daiteke. Horregatik Hardwarea eta Softwarea logikoki baliokideak direla esaten da. Horrela biderkatzaile bat konputagilu batzutan zirkuituen bidez egiten da (HARDWAREaz), beste batzuetan berriz, batuketa-segida batez egindako programa baten bidez (SOFTWAREaz). Beste kasu batzutan ez dago oso garbi elementu bat edo gehiago hardware-ari edo software-ari dagozkion eta FIRMWARE kontzeptua sortzen da. Horrela aurrerago aztertuko dugun mikroprogramazioa Hardware eta Sotfware bitartean dago eta FIRMWARE-aren barruan sailkatzen da.

2.2.- KONPUTAGAILUAREN OINARRIZKO BARNE-EGITURA

2.2.1.- OSAGAIAK

Babbagek asmatutako eskemari Von Neumann-en ideiak gehituz gaur egungo konputagailuen egitura lortzen da (ikus 2.1 irudia). Definitu ditzagun osagaiak banan bana:

MEMORIA.- Unitate honetan informazioak bil daitezke geroago eskuratzeko. Memorian, nahiz datuak nahiz programak gordetzen dira. Memoria Nagusia edo Barne-Memoria (Memoria deritzona) eta Kanpo-Memoria bereizten dira. Kanpo-Memoria S/I unitatearekin lotuta dago eta Memoria Nagusia berriz informazioa CPUrekin zuzenean lotuta dago.



D - datuak
E - emaitzak
K-U - Kontrol Unit.
S-U - Sarrera Unit.
A - aginduak
mA- mikroaginduak
ALU - Un. Aritmetiko-Logikoa
I-U - Irteera Unit.

2.1 irudia.- KONPUTAGAILUAREN OINARRIZKO EGITURA

KONTROL-UNITATEA (KU).- KU-k agindu bakoitza memoriatik eskuratu ondoren aztertzen du eta ezagutzen duenean agindu hori burutzeko egikaritu behar diren mikroaginduak sortu eta beste unitateetara bidaltzen ditu. Beraz sistema gidatzen duen osagaia da.

UNITATE ARITMETIKO-LOGIKOA (ALU).- Memorian dauden datuak erabiliz eragiketak burutzen dituen osagaia da. Eragiketak bi motatakoak izan daitezke: aritmetikoak (batuketa, kenketa, biderkaketa ...) eta logikoak (konparaketak, AND, OR ...). ALU-k eta KU-k Prozesurako Unitate Zentrala (CPU) osatzen dute. CPU honek, bere kalkuluak eta kontrolak egiteko, oso edukiera txikiko baina atzipen-abiadura izugarriko memoria antzeko elementu batzu dauzka: erregistroak deitutakoak. Horrexegatik asko erabiltzen diren informazio laburrak erregistrotan gordetzen dira. CPU-a konputagailuaren funtsa da eta beraz konputagailua identifikatzen du.

SARRERA/IRTEERA-UNITATEA.- Erabiltzailearekiko komunikazioa unitate honen bidez egiten da. Bi atal bereizten dira SARRERA-UNITATEA eta IRTEERA-UNITATEA.
Programa (aginduak) eta datuak Sarrera-Unitatea barrena ematen zaizkio konputagiluari. Sartutako informazioa (datuak eta aginduak) Memorian kokatuko dira. Irteera-Unitatean barrena emaitzak kanporatzen dira, erabiltzaielak lor ditzan.
Dena dela Sarrera/Irteera-Unitatean Kanpo-Memoriak daude zeintzuei mota guztietako informazioak (aginduak eta datuak) kanporatuko bait zaizkie, geroago irakurtzeko prest gera daitezen.
Unitateen arteko komunikazioa zirkuitu batzuren bidez egiten da, BUS-ak deitutakoen bidez, hain zuzen.

2.2.2.- KONPUTAGAILU-MOTAK

Hiru konputagilu-mota bereizten dira ahalmen edo potentziaren arabera; Konputagailuak, Minikonputagailuak eta Mikrokonputagailuak haundienetatik txikienetara sailkatuz.

	MEMORIA NAGUSIAREN EDUKIERA	KANPO-MEMORIAREN EDUKIERA
KONPUTAGAILUA	1000 K ->	100 M ->
MINIKONPUTAG.	256 K - 8000 K	5 M - 200 M
MIKROKONPUTAG.	1 K - 1000 K	0,1 M - 10 M

2.1 taula.- KONPUTAGAILU-MOTEN EZAUGARRIAK

Sailkapen hau faktore desberdinak kontutan hartuz egiten da, abiadura eta edukiera nagusiak badira ere. Konputagailu gehienak oso homogeneoak direla esan beharra dago, hots, oso zaila dela edukiera handiko konputagailu batek abiadura txikia edukitzea eta alderantziz.

Abiadura neurtzeko unitateak oso aldakorrak direnez gero, 2.1 taulako sailkapena aurkezterakoan edukiera zehazten dugu.

2.3.- INFORMAZIOA KONPUTAGAILUAN

Konputagailuaren barnean informazioa Memorian gordetzen da, bai Memoria Nagusian, bai Kanpo-Memoria, bigarren kasu honetan informazioak erabili ahal izateko Memoria Nagusitik pasa behar duelarik. Informazioak, esan dugunez, bi motatakoak izan daitezke: aginduak eta datuak eta nola edo hala konputagailuak agindua zer den eta datua zer den jakin beharko du.

Lehenbiziko gaian konputagailuaren ezaugarritako bat digitala izatea zela esan genuen, beraz ezin du informazio jarraia erabili. Hain zuzen ere, Memorian informazioa gordetzen duten osagaiak biegonkorrak dira, BIT-ak deitutakoak, hau da, osagai bakoitza bi egoeratan bakarrik egon daiteke, egoera bati 0 balioa eta besteari 1 balioa asignatzen zaielarik. Bit-a konputagailuaren informazio-unitate txikiena da.

Horregatik, zenbakiak adierazpide bitarrean (2 oinarrian) adierazten dira Memorian, bit bakoitza zifra bat bait da adierazpide horretan. Aginduak eta informazio alfabetikoa bitez kodetuak adieraziko dira.

Fisikoki bit-ak teknologia desberdinez eraikitzen dira, ferritazko nukleoak eta zirkuitu erdieroaleak zabalduenak izanik.

Bit bat zer den aztertu ondoren goazen Memoriarekin zerrikusia daukaten beste osagaiak definitzera:

BYTE.- Karaktere bat adierazteko behar den bit multzoa da eta jeneralean zortzi bit-ek osatzen dute, kasu honetan ZORTZIKOTE ere deitzen delarik. Konputagailu batzutan sei bit-ek osatzen dute eta SEIKOTE deitzen da.

HITZA.- CPUren eta Memoriaren arteko informazio-trukerako unitateari hitza deritzo baina konputagailuaren eta autorearen arabera definizio desberdinak dauzkagu. Halaber, beste definizio oso hedatu batzuk honako hauek ditugu:

- Memoriatik irakurketa batez eskuratzen den bit-multzoak HITZ bat osatzen du
- Zenbaki oso bat adierazteko behar diren bit-ek HITZ BAT osatzen dute.

Gehienetan definizio guztiek bit-kopuru bera adierazten dute. Hitzaren luzerarik arruntena 16 eta 32 bitetakoa da nahiz eta konputagailuen garapena dela eta gero eta luzeagoak izan.

HELBIDEA.- Informazio bat Memoria Nagusitik eskuratzeko, bere helbidea zehaztu behar da. Memoria Nagusiak buzoi multzo baten antza du, non informazioa buzoien edukina eta helbidea buzoiaren zehaztasuna bait dira. Beraz, helbideak informazioa unibokoki bereizten du, zeren eta memoriako bi lekutan edukin berdina egon badaiteke ere, leku hauei beti helbide desberdinak dagozkie.
Memoriako helbideak zenbaki arruntak dira Ø-tik hasita. Helbide batean momentu desberdinetan datu desberdinak koka daitezke. Honen ondorioz, programa batean helbide bat aipatzea eta aldagai baten izena aipatzea baliokideak dira.

GELASKA edo POSIZIOA.- Helbide batek identifikatzen duen bit-kopuruari gelaska edo posizioa deritzo, beraz, gelaska bakoitzari helbide bat dagokio. Konputagailuaren arabera gelaskaren luzera kasu batzutan byte-arena da, beste batzutan berriz hitzarena.
Gestio-lanetarako diseinaturiko konputagailu gehienetan gelaska eta byte-a baliokideak dira (8 bit gehienetan) eta kasu honetan karakterekako konputagailuak direla esaten da.
Problema zientifikotarako bereziki eraikitako konputagailu gehienetan gelaska eta hitza baliokideak dira eta orduan hizkako konputagailuak direla esaten da.

K edo K-BYTE.- 1Ø24 bytek K bat osatzen dute baina gehinetan kalkulua errazteagatik mila bytekotzat hartzen da. Memoriako edukiera neurtzeko unitatea da.

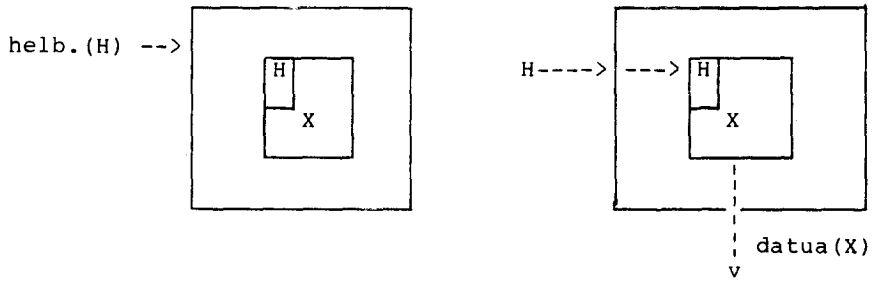
M edo MEGABYTE.- 1Ø24 Kbytek osatzen dute baina aurreko kasuan be-

zala mila K izango balu bezala hartzen da.

Esan dugunez, informazioa Memorian gordetzen da eta zenbait denbora pasatu ondoren informazio hori eskura dezakegu. Beraz Memorian bi eratako eragiketak burutzen dira, bata idaztea (informazioa gordetzea) eta bestea irakurtzea (informazioa eskuratzea).

Azter ditzagun bi eragiketa hauek:

IRAKURRI.- Memorian dagoen gelaska bat eskuratzean datza. Horretarako, gelaskaren helbidea zehaztu beharko zaio Memoriari eta IRAKURRI ondoren gelaskaren edukina erabiltzeko prest geldi-tzen da (ikus 2.2 irudia).



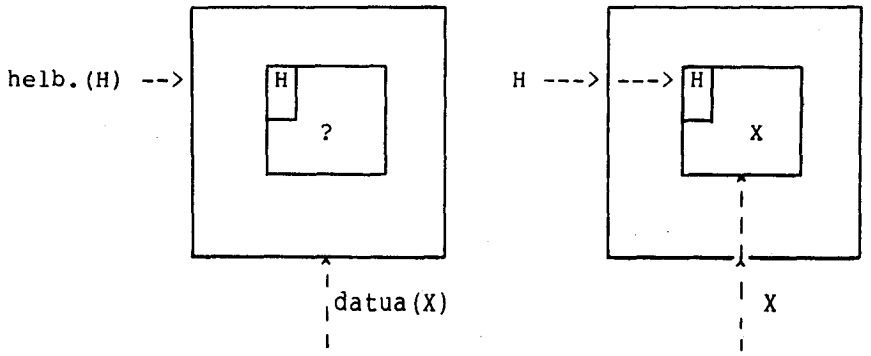
a) irakurri baino lehen b) irakurri eta gero

2.2 irudia.- MEMORIAN IRAKURRI eragiketa

IDATZI.- Memoriako gelaska baten edukinean informazio baten gordetzea da IDATZI eragiketaren ondorioa. Idatzi baino lehen gelaskaren helbidea eta gorde nahi den informazioa zehaztuko zaizkio Memoriari eta, idatzi ondoren, informazioa zehaztutako gelaskan kokatzen da (ikus 2.3 irudia).

Irakurritakoan gelaskaren edukinak irauten du, idatzitakoan berriz zegoen edukina galdu egiten da informazio berri batez ordezkatuz.

Informazio bat eskuratzeko, asmoz Memoriako gelaska batera heltzeko agindua ematen den unetik informazio hori erabiltzeko prest dagoen momentura arte joaten den denbora-epea ATZIPEN-DENBORA da. Atzipen-denbora Memoria Nagusian mikrosegundotan eta nanosegundotan neurtzen da eta konputagailuaren ezaugarri garrantzitsua dugu.



a) idatzi baino lehen b) idatzi eta gero

2.3 irudia.- MEMORIAN IDATZI eragiketa

2.4.- KONPUTAGAILUAREN MAILAK

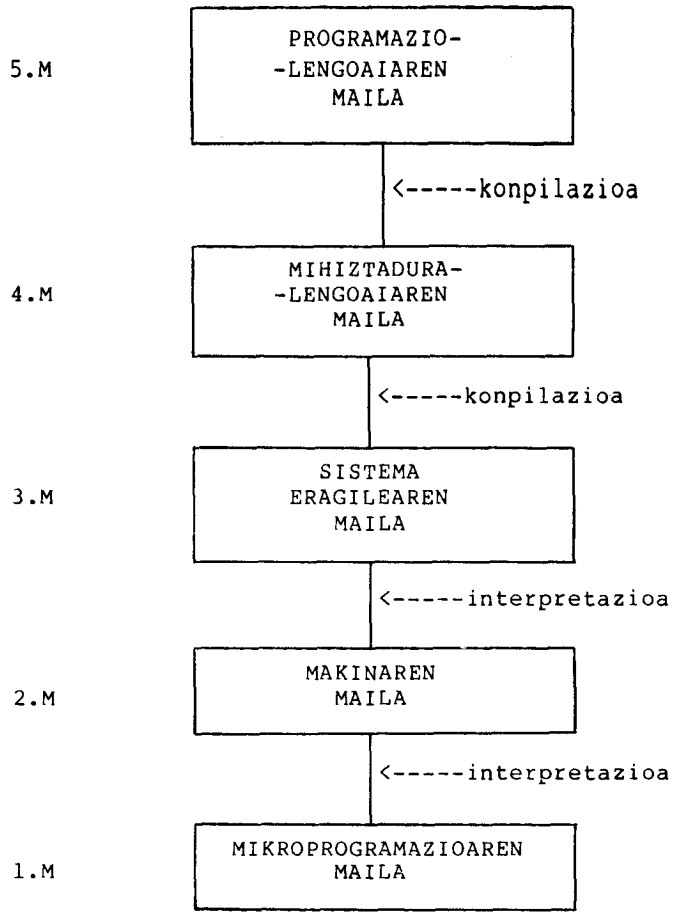
Esan dugunez, konputagailu bat agindu multzo bat burutzeko gai da eta programa bat, azken finean, multzo horretako agindu-segida bat da. Kontrol-Unitateak aztertzen eta burutuerazten duen agindu-multzo honek makina-lengoaia osatzen du.

Dena dela, makina-lengoaia honetan programatzea (makina-lengoaiaren bidez) oso astuna eta aspergarria gertatzen da, akats asko sortzen delarik. Irtenbide baten beharra zegoen eta ebazpena hauxe da: Gure konputagailua agindu ahalmentsuagoz eta problematik hurbilagoz horniturik dagoela pentsa genezake eta horrela alegiazko konputagailu bat definitzen dugu, berari dagozkion agindu-multzoarekin batera. Argi dago makina-mailatik beste maila batera pasa garela. Daukagun problema bakarra maila horretako lengoaian dagoen programa bat makina-lengoaia bihurtzea da eta horretarako itzultzaile bat behar da.

Gaurko konputagailuak maila desberdinek osatzen dituzte, beraz konputagailu bat alegiazko konputagailu desberdinez osatutako multzotzat har ditzakegu. Alegiazko maila bakoitzari bere lengoaia edo agindu-multzoa dagokio, maila batetik beste txikiago batera pasatzeko itzultze-prozesu bat dagoelarik.

Bereizten diren maila arruntenak 2.4 irudian ikus daitezkeen bostak dira.

Adibidez, programazio-lengoaiaren mailan gaudenean hardware-k nola funtzionatzen duen ez dugu ezertarako jakin beharrik (eta programak egiteko jakin behar dugun lengoaiak eta konputagailuaren ezaugarriek mailak horren alegiazko konputagailuak osatzen dute). Maila bakoitzari lengoaia bat dagokio eta maila batean idatzitako programa bat hurrengo mailara pasatzeko itzulpen-prozesu bat egin behar da. Itzulpen hau bi motatakoa izan daiteke:



2.4 irudia.- KONPUTAGAILU GEHIENETAKO MAILAK

KONPILAZIOA.- Programa konpiladore baten bidez maila bati dagokion lengoaia batean idatzitako programa bat beste mailari dagokion lengoaiara itzultzen da. Lortutako programa azken lengoaia honi dagokion Alegiazko Makinan egikari daiteke.

INTERPRETAZIOA.- Interpretatzaile batek maila bateko programa hartzen du eta agindu bakoitza itzuli ondoren egikaritzen da. Beraz, ez da beste programa baliokidea sortzen, baizik eta itzulitako agindu bakoitza momentu berean egikaritu egiten da.

Konpilazioaren eta interpretazioaren arteko diferentzia itzulpen literarioa eta konferentzia baten batpateko itzulpenaren artekoaren antzekoa da, literarioan itzulitako liburua sortzen da, batpatekoan berriz esaldi bakoitza itzuli egiten da baina ez da erregistraturik geratzen.

Azter ditzagun orain aipatutako konputagailuen mailak:

1. MAILA.- MIKROPROGRAMAZIOA

Kontrol-Unitateak sortzen eta bidaltzen dituen mikroaginduek osatzen dute maila honetako lengoaia. Mikroagindu hauek Hardware-k zuzenean egikaritzen dituen eragiketak dira, Memoriatik irakurtzen, Unitate batetik bestera informazio bat bidatzen eta antzekoak.

2. MAILA.- MAKINA KONBENTZIONALA

Kontrol-Unitateak ulertzen dituen aginduek maila honi dagokion lengoaia (makina-lengoaia) osatzen dute. Maila honetan batuketak, kenketak, konparaketak, mugimenduak eta jauziak dira agindu arruntenak. Konputagailuaren arabera, agindu-aukera desberdina dago eta aginduaren formatoa ere desberdina da. Beraz, makina-lengoaia konputagailuaren egituraren menpekoa da, 1.mailako lengoaia heinean.

Kontol-Unitateak maila honetako aginduak interpretatzen ditu mikroprogramazio-mailako mikroaginduak sortuz. Interpretazio hau bi modutan egin daiteke konputagailuaren arabera:
- Zirkuitu finko batzuren bidez (guztiz Hardwareaz).
- Kontrol-Memoria baten bidez, non agindu bakoitza osatzen duten mikroaginduak zehazten bait dira. Kasu honetan, konputagailu baten makina-lengoaia aldakorra da eta konputagailua mikropramagarria dela esaten da. Konputagailu bat mikroprogramatzea makina-agindu berriak definitzean datza. Makina-agindu bat mikroagindu-segida baten bidez definitzen da, mikroagindu-segida hau Kontrol-Memorian kokatuko delarik.

Makina-mailako aginduek bi alderdi dituzte, bata eragiketa-kodea da, zeinek zer eragiketa burutu behar den zehazten bait du eta bestea eragigaia edo eragigaiak, datuak nondik eskuratu behar diren zehaztuz.

3. MAILA.- SISTEMA ERAGILEA

Konputagailuaren baliabideak (Memoria, CPU, Sarrera/Irteera, ...) gestionatzeko makina-lengoaiaz egindako programa batzutan datza maila hau, programa hauek errutinak deitzen direlarik.

Maila hau, errutinez aparte, 2.mailari dagozkion agindu gehienek osatzen dute, maila batean dauden aginduak maila altuagotan ager daitezke eta. Maila honetako aginduak errutinak baldin badira Sistema Eragileak interpretatzen ditu. Arrazoi horrengatik maila hau hibridoa dela esaten da.

Errutina hauen bidez memoriaren antolaketa, bi programen egikaritzapen paraleloa, Sarrera/Irteerarako eragiketak, erabiltzailearekiko komunikazioa eta antzeko gauzak errazten dira.

4. MAILA.- MIHIZTADURA-LENGOAIA

3.mailatik 4.era diferentzia handia dago. Alde batetik lehenengo 3 mailatan programak digitu-segida batez adierazten dira, 4. mailatan, berriz, lengoaia sinboliko batez. Metodo honen bidez programak idaztea ez da horren astuna eta askoz akats gutxiago sortzen da.

Beste alde batetik beheko mailatarako itzulpena ez da interpretazioaren bidez egiten, programa konpilatua baizik. Itzulpena lortzen duen programari mihiztatzailea deritzo.

Mihiztadura-lengoaiako maila beheko mailatako lengoaien adierazpide sinbolikoa da, hau da, 2. edo 3. mailako programak modu erosoago batean idazteko aukera eskaintzen digute. Mihiztadura-lengoaia garai batean oso hedatua izan zen, gaur egun erabilpen txikia badu ere.

Mihiztadura-lengoaiaren agindu bakoitza 3.mailan agindu bakar bat bihurtzen da, hau da, 3. eta 4.mailako aginduak baliokideak dira baina kode desberdinak erabiltzen dituzte.

5. MAILA.- GOI-MAILAKO LENGOAIAK

Maila honek programazio-lengoaia ahalmentsuak eskaintzen dizkigu, programazioa askoz errazago, ulergarriago eta fidagarriago gertatuz. Maila honetako aginduak konplexuagoak, egituratuagoak eta gure pentsamoldeekiko gertuagoak dira, agindu bakoitza beheko mailatako agindu-multzo batekiko baliokidea izanik.

Itzultzen duen programa (interpretatzailea ere izan daiteke) konpiladorea da gehienetan eta itzulpena 4.mailako edo gehienetan 3.mailako lengoaian gelditzen da.

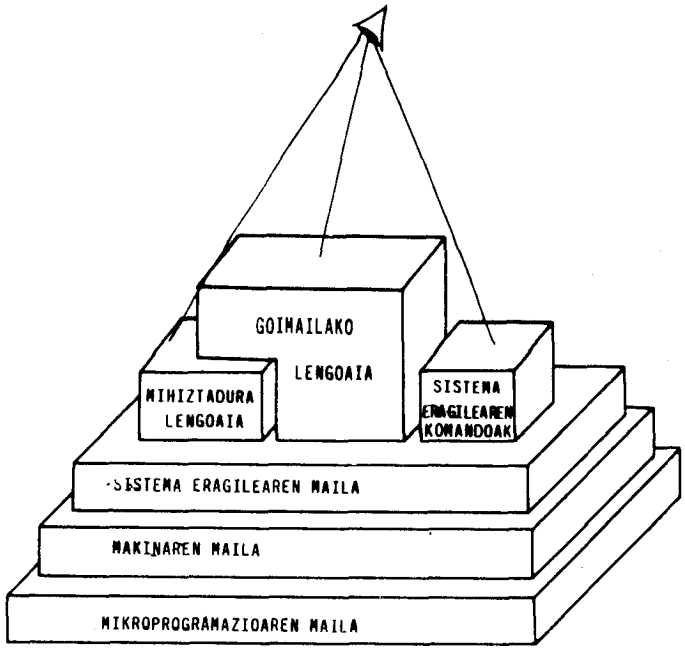
Goimailako programazio-lengoaia hedatuenak honako hauek dira:

- FORTRAN eta ALGOL kalkulu zientifikoetarako.
- COBOL eta RPG gestio-lanetarako.
- PL-1 bietarako (gestio-lanetarako eta erabilpen zientifikoetarako).
- BASIC bietarako, baina Mikrokonputagailuetan.
- PASCAL eta ADA-ri oso metodologia ona datxekie eta unibertsitatetan oso hedatuak dira, gero eta erabiliagoak izanik beste arlotan ere.
- LISP, PROLOG, SIMULA, GPSS, ... erabilpen zehatzetarako.

BASIC lengoaiaren itzulpena, askotan, interpretatzailaren bidez egiten da.

6. MAILA eta goragokoak problema mugatu batzu ebazteko zuzendutako alegiazko makinetan datza. Sistema horiei Sistema Adituak deritze eta gaur egungo ikerkuntzaren helburu nagusietako bat dira.

Sistema horietako programazioa ez da programazio-lengoaia baten bidez egiten, hizkuntza naturalaedo sasinaturalaren bidez baizik.



2.5 irudia.- MAILEN ARTEKO ERLAZIOAK

Laburpen moduan 2.5 irudian mailen arteko erlazioak ikus ditzakegu. Gaineko mailak erabiltzaileak erabiltzen dituenak dira. Alde batetik, goimailako lengoaiak daude programak egiteko, batzutan programa berezitan mihiztadura-lengoaia erabili arren, eta beste alde batetik Sistema Eragilearen komandoak ditugu konputagailuak zer eginkizun eta zer baldintzapenetan burutu behar dituen zehazteko erabiltzen direlarik. Komando hauei Kontrol-sententziak ere deitzen zaie.

Erabiltzaileek ez dituzte azpiko 3 mailetako lengoaiak erabiltzen, baina beste mailen oinarria dira. Konputagiluaren diseinua egiten denean 1. eta 2. maila erabakitzen dira eta Sistema Eragilea burutzerakoan 1. eta 2. maila erabiliz, 3.mailako errutinak egitea da lehenengo eta oinarrizko pausoa.

Asignaturaren helburua 2. eta 4.mailak aztertzea izango da, 1. eta 3. mailak batzutan aipatuko badira ere.

2.5.- INFORMATIKARIAK

Oraingoan, berriz, aplikazio bat ordenadorez gauzatzeko moduan egon dadin zer-nolako pausoak eman behar diren ikusiko dugu.

Bi fase handi berezi behar ditugu: garapen-fasea, non aplikazioa analisatu eta programatu egiten den eta explotazioa, zeinetan gauzatzen bait da aplikazioaren konputagailuratzea.

Has gaitezen ba, garapen-fasearekin.

2.5.1.- GARAPENA: ANALISIA ETA PROGRAMAZIOA

Aplikazio bat proposatzen denean ondo aztertu behar dira zein diren aplikazio horren helburuak. Argi eta garbi zehaztu behar dira helburu horiek, anbiguetaterik egon ez dadin. Problemarekin zerikusirik duten gauza guztiak bildu beharra dago eta, konputagailuratu aurretik, eragiketa guztiak eskuz nola egiten ziren ongi aztertu.

Datu-bilketa egin eta gero, aurkitu diren soluzio posible guztien artean egokiena aukeratuko da. Eta erabaki hori hartu ondoren, problemaren inplementazioa dator.

Inplementazio hori samurtzeko, badaude metodo batzu eta horietako bat beheranzko diseinua delako metodologia dugu. Oso interesgarria, batez ere problema oso konplexua baldin bada. Teknika hori erabiliz gero, problema orokorra beste azpiproblema batzutan zatitzen da eta azpiproblema horietako bakoitza era berean beste batzutan, eta horrela soluzio errazeko problemetara iritsi arte. Eta honekin bukatuko dugu analisia. Analistak, datu-bilketan lortutako informazioa landuz gero, programatzaileari emango dio orri standard batzuren bidez.

Analisia egin ondoren, programazioa dugu hurrengo urratsa. Problemaren zati bakoitza konputagailuak ulertuko duen lengoaian idatzi behar da: programatu egin behar da. Lengoaia horri programazio-lengoaia esango diogu.

Aurrekoa analistek egiten duten bezala, prozesu hau programatzaileei dagokie. Berorien lana ez da, ordea, programa idaztearekin amaitzen: programa ondo dagoela egiaztatu arte (sintaxi eta exekuzio-errorerik gabe alegia) jarraitu behar dute programa probatzen.

2.5.2.- EXPLOTAZIOA: PLANIFIKAZIOA, ERABILPENA ETA TRANSKRIPZIOA

Aurreko fasean garatutako aplikazioa, hau da, analisatu eta programatu duguna, konputagailura eraman behar da orain. Horretan datza explotazioa.

Aurren-aurrena planifikazioa egin beharra dago. Esate baterako hilean zehar zer aplikazio diren ordenadoretik pasa beharrekoak, noiz, maiztasuna, etab. Beste zenbait aplikazio urtean behin pasa beharrekoak izango dira, edota egunerokoak. Horixe da, bada, planifikazioa: ordenadorean egin behar diren lanen segida prestatu, noiz egikaritu behar diren, zer maiztasunekin, zenbat denbora irauten duen bakoitzak, etab.

Planifikatu eta gero erabilpena bera dator. Operatzaileak prestatezen dituzte konputagailuari eman zaizkion datuak, sartu beharrekoak banatuz eta artxiboak kontrolatuz. Berorien eginbeharrak dira fitxak ordenadorean sartzea, zinta eta disko magnetikoak aldatzea, konputagiluaren martxa mantentzeko lanak egitea, emaitzak bildu eta dagozkionari bidaltzea. Gai izan behar dute, horretaz gain, edozein etenalditatik ateratzeko, zenbait problema tekniko konpontzeko etab. Makinaren portaera eta erabilpenari buruzko historiala ere eramango dute normalean.

Eta bukatzeko, aipa dezagun transkripzioa. Datuak ordenadorearentzako euskarri fisiko egokitan jartzean datza transkripzioa: sarrerako datuak fitxa zulatutara pasa, zintetan edo kasetetan grabatu, etab. Lan hau zulatzaileek (jeneralean hala esan ohi zaie) egiten dute.

3.- K O N P U T A G A I L U A R E N E G I T U R A

3.1.- SARRERA

3.2.- KONPUTAGAILU-MOTAK EGITURAREN ARAUERA

3.3.- MEMORIA NAGUSIA

3.4.- UNITATE ARITMETIKO-LOGIKOA

3.5.- KONTROL-UNITATEA

3.6.- MIKROPROGRAMAZIOA

3.7.- KONPUTAGAILU BATEN BARRUKO FUNTZIONAMENDUA

3.8.- ARIKETA EBATZIAK

3.1.- SARRERA

Aurreko kapituluan definitutako unitateen egitura eta antolamendua agertzea da kapitulu honen helburua, beraz, Memoria, Unitate Aritmetiko-Logikoa, Kontrol-Unitatea eta Sarrera/Irteera-Unitatea banan banan aztertuko ditugu, ondoren Unitate hauen arteko informazio-trukea zehaztuko dugularik. Kapitulu honetan, azken finean, aipatutako 2.maila eta 1.mailaren arteko erlazioa aztertuko dugu. Unitateen arteko komunikazioa bus-en bidez egiten dela adierazi dugu baina, lau bus-mota desberdin hauek bereizten dira:

DATU-BUSA.- Bus honetan barrena informazioa, nahiz datuak nahiz aginduak, bideratzen da CPU eta Memoriaren artean.

HELBIDE-BUSA.- Bus honetan barrena CPU-k eskuratu nahi dituen datuen helbideak zehazten dizkio Memoriari.

KONTROL-BUSA.- Bus honetatik agindu, bakoitza burutzeko sortutako mikroaginduak bideratzen dira beste unitatetara.

SARRERA/IRTEERA-BUSA.- Sarrera/Irteera-Unitatearekin lotutako busa da, bere funtzioa informazioa barneratzea edo kanporatzea delarik.

3.2.- KONPUTAGAILU-MOTAK EGITURAREN ARAUERA

Unitateen egitura guztiak ez dira berdinak lehen mailako lengoaiak eta batez ere bigarren mailako lengoaiak baldintzatzen dituzte.

2.mailako lengoaiaren faktore garrantzitsuena aginduaren egitura da eta egitura honen arabera honako lau konputagailu-mota hauek bereizten dira: 3 helbideko aginduak egikaritzen dituztenak, bi helbidekoak, helbide bakarrekoak eta helbide gabekoak.

3.2.1.- HIRU HELBIDEKO AGINDUAK

Ondoko formato hau dagokie agindu hauei:

ERAGIKETA-KODEA/1.ERAGIGAIA/2.ERAGIGAIA/3.ERAGIGAIA

Formato honen barnean eragigai bakoitza edozein helbideratze-moduren bidez adieraz daiteke eta eragigai batek (3.ak jeneralean) emaitzaren helbidea zehazten du, beste biak eragiketaren datuak direlarik. Eragiketa aritmetikoak adierazteko oso egitura egokia da. Adibidez: A=(B-C)/(B+C)*D formula kalkulatzeko programa horrelakoa izango litzateke (kontutan hartu laugarren mailako lengoaia erabiltzen dugula adibidea ulergarria egiteko baina bigarren mailan bitez bit idatzitako agindu baliokideek osatuko zuten programa)

```
KEN   B,C,A   ; a=b-c
GEHI  B,C,Z   ; z=b+c
BIDER Z,D,Z   ; z=z*d=(b+c)*d
ZATI  A,Z,A   ; a=a/z=(b-c)/(b+c)*d
```

OHARRAK: emaitza hirugarren eragigaian adierazten da.
Z bitarteko eremu bat da.

Agindu hauen eragozpenak honako hauek dira. Alde batetik Memoria hiru aldiz atzitu behar dute; bi, datuak irakurtzeko eta emaitza gordetzeko bestea. Horren ondorioz, aginduaren egikaritzapena nahiko motela gertatzen da. Bestalde hiru eragigai izateagatik luze samarra gertatzen denez, 2.mailako aginduak ahalik eta laburrenak diseinatzen saiatu behar da. Eragozpen hauek kontutan hartuta, agindu askotan datu bat eta emaitza helbide berberean daudela ikusita konputagailu askotan 2. mailako aginduak bi eragigaik osatzen dituzte.

3.2.2.- BI HELBIDEKO AGINDUAK

Agindu hauei ondoko formato hau dagokie:

ERAGIKETA-KODEA/1.ERAGIGAIA/2.ERAGIGAIA

Kasu honetan eragigai batek datu baten eta emaitzaren helbidea adierazten du eta beste eragigaiak beste datuaren helbidea zehazten du. Agindu hauek aurrekoak baino arin eta laburragoak dira, baina datu baten edukina aldatzea eragozpena gerta daiteke. Eragozpen hau mugimendu-aginduez konpontzen da, baina agindu hauek, azken finean, eragiketa efektiboak ez direnez, programa luzeagoa eta motelagoa ere gertatzen da. Emaitza zer eragigaitan adierazten den aldatu egiten da konputagailuaren arabera. Gure adibideetan 1.eragigaian adierazten dela suposatuko dugu. Mugimendu-aginduek eragigai batek adierazten duen edukina beste eragigaiak adierazitako helbidean kopiatzen dute.

Adibidez, MUGI A,B aginduaren egikaritzapenaren ondorioa a=b izango litzateke (1.eragigaian emaitza adierazten delako) B-k adierazten duen helbideko edukina A-k adierazitako helbidean kokatzen delako.

Aurreko A=(B-C)/(B+C)*D formula kalkulatzeko programa, bi helbideko aginduen bidez, hauxe izango litzateke:

```
MUGI   A,B   ;a=b
KEN    A,C   ;a=a-c=b-c
MUGI   Z,B   ;z=b
GEHI   Z,C   ;z=z+c=b+c
BIDER  Z,D   ;z=z*d=(b+c)*d
ZATI   A,Z   ;a=a/z=(b-c)/(b+c)*d
```

Adibidean ikus daitekeenez aginduak aurreko kasuan baino motzagoak dira (eta arinagoak) baina agindu gehiago dago.

Agindu bakoitzaren luzera laburtzeko eta Memoriako atzipenak gutxiagotzeko (azken hau abiadura handitzeko) eragigai bakarreko aginduak diseinatzen dira.

3.2.3.- HELBIDE BAKARREKO AGINDUAK

Ondoko formatoa dagokie:

ERAGIKETA-KODEA/ERAGIGAIA

Non eragiketa burutzeko Unitate Aritmetiko-Logikoan kokaturik dagoen erregistro berezi bat erabiltzen bait da, METAGAILUA deritzana. Beraz, 1.eragigaia eta emaitza metagailuan kokatzen dira zehaztutako eragaia 2.datua izanik. Arrazoi horregatik, makina-agindu mota honetako konputagailuak Metagailu bidezkoak deitzen dira.

Adibidez, "GEHI X" aginduak Metagailuaren edukinari X-ek adierazitako datua gehitu behar zaiola, batura Metagailu bertan utziz, adierazten du.

Horrela agindu laburragoak eta arinagoak lortzen dira, baina Metagailuaren edukina aldatzeko eta gordetzeko aginduak behar dira. Programak egiterakoan agindu hauek programa luzatu eta motelduko dutelarik.

Bi agindu berri hauek ondokoak dira: KARGA (LOAD) aginduak eragigaiak adierazten duen datua Metagailuan kokatzen du eta GORDE (STORE) aginduak Metagailuaren edukina eragigaiak adierazten duen Memoriako helbidean kokatzen du.

Beraz, "KARGA X" egikaritzean X-ek adierazitako datua Metagailuan kokatzen da (Met=X) eta "GORDE Y" burutzean berriz, Metagailuan dagoena Y-k adierazitako helbidean gordetzen da (Y=Met).

Hona hemen aurreko formula kalkulatzeko agindu-mota hauek erabiltzen dituen programa:

```
KARGA  B   ; Met=b
KEN    C   ; Met=Met-c=b-c
GORDE  A   ; a=b-c
KARGA  B   ; Met=b
GEHI   C   ; Met=b+c
BIDER  D   ; Met=(b+c)*d
GORDE  Z   ; z=(d+c)*d
KARGA  A   ; Met=b-c
ZATI   Z   ; Met=(b-c)/(b+c)*d
GORDE  A   ; a=(b-c)/(b+c)*d
```

Programa hori hobetu daiteke lehenbizi (B+C)*D kalkulatuz eta ondoren (B-C) eta zatiketa eginez; GORDE eta KARGA agindu bana aurreratzen bait ditugu:

```
KARGA  B   ; Met=b
GEHI   C   ; Met=b+c
BIDER  D   ; Met=(b+c)*d
GORDE  Z   ; z=(b+c)*d
KARGA  B   ; Met=b
KEN    C   ; Met=b-c
ZATI   Z   ; Met=(b-c)/(b+c)*d
GORDE  A   ; a=(b-c)/(b+c)*d
```

3.2.4.- ERAGIGAIRIK GABEKO AGINDUAK

Agindu-mota hauetan agindu gehienek ez dute eragigairik eta dagokien formatoa hauxe da:

ERAGIKETA-KODEA

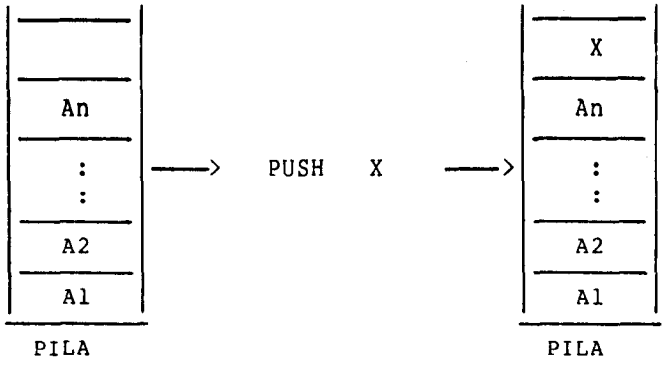
Eta orduan zer daturekin burutzen da eragiketa? Ba PILA datu egitura erabiltzen da, pilan kokatutako azken datuak eragigaiak izango direlarik.

Azter dezagun ba pila bat zer den. Pila, datu-multzo bat da, non nahi dugun informazio guztia gorde daiteke, baina informazioa eskuratzeko momentuan gordetako azken datua bakarrik lor dezakegu, pilako tontorrean dagoena. Izena, liburu-pila batekin duen antzetik dator, zeren eta liburuak bata bestearen gainean jartzen bait ditugu eta kentzerakoan gainetik hasi beharra dugu, tontorretik alegia.

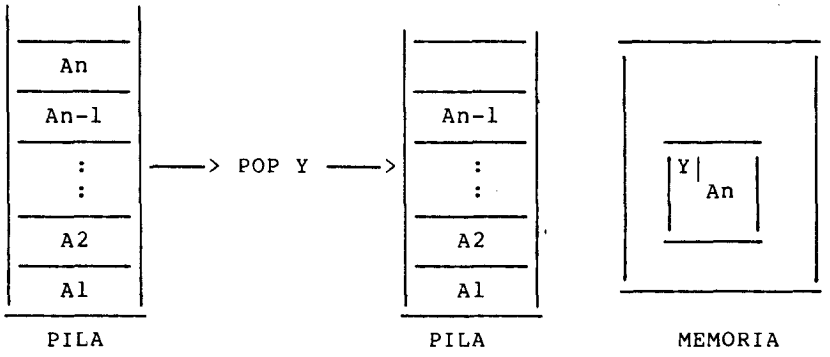
Pilari 2 eragiketa dagozkio PUSH (datu bat Memoriatik hartu eta pilan sartu) eta POP (tontorrean dagoena eskuratu eta Memorian gorde) eta haien ondorioak 3.1 irudian aztertzen dira.

Orduan, pila bidezko konputagailu batean agindu baten egikaritzapena horrelaxe gertatzen da: Pilaren azken bi datuen artean burutzen da eragiketa, datu hauek pilatik desagertuz eta emaitza pilaren tontorrean geldituz. Adibide bezala "KEN" aginduaren ondorioa 3.2 irudian ikus daiteke.

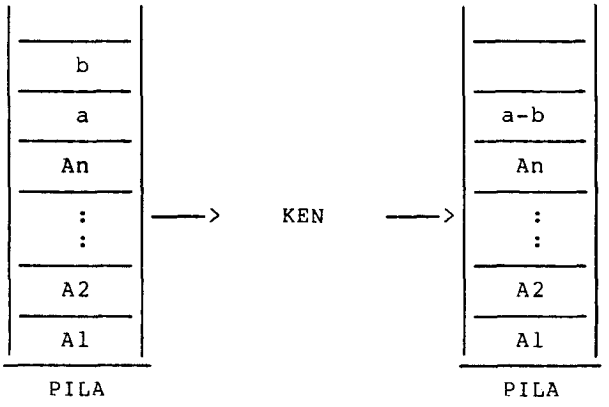
PUSH



POP



3.1 irudia.- PUSH ETA POP AGINDUAK



3.2 irudia.- PILA BIDEZKO KEN AGINDUA

Fisikoki konputagailu hauetan erregistro multzo batek osatzen du pila, memorian kokaturik baleude oso motela gertatuko litzateke eta.

Aurreko formula programatzeko programa honako hau litzateke pila bidezko konputagailu batean:

```
PUSH   B        ; pila: {b}
PUSH   C        ; pila: {b,c}
KEN             ; pila: {b-c}
PUSH   B        ; pila: {b-c,b}
PUSH   C        ; pila: {b-c,b,c}
GEHI            ; pila: {b-c,b+c}
PUSH   D        ; pila: {b-c,b+c,d}
BIDER           ; pila: {b-c,(b+c)*d}
ZATI            ; pila: {(b-c)/(b+c)*d}
POP    A        ; a=(b-c)/(b+c)*d
```

Mota honetako aginduak maneiatzen dituzten konputagailuek egitura berezia dute. Aginduen luzera laburtzea eta abiadura handiagotzea lortzen da, PUSH eta POP aginduak salbuespen izanik, baina PUSH agindua askotan erabili behar denez ez da aurrerapen argirik.

Konputagailu hauetan programa modu errazagoan idazteko formulak idazkera poloniar alderantzizkoa erabiliz idaztea guztiz komenigarria da.

Normalean erabiltzen dugun idazkerari in-finkoa deitzen zaio, datuen artean zehazten bait da eragiketa. Aldiz, idazkera poloniar alderantzizkoa edo post-finkoa eragiketa datuen atzean zehaztean datza. Bigarren idazkera honek hiru abantaila ditu:

- ez da parentesi behar formulak adierazteko.
- eragiketen arteko lehentasunik ez dago. Adibidez, a*b+c, (a*b)+c da eta ez a*(b+c). Idazkera poloniarrean lehenengoa ab*c+ da eta abc+* bigarrena.
- Pila bidezko konputagailuetan erabilpena oso egokia da.

Adibide batzu jar ditzagun:

IDAZKERA IN-FINKOA	IDAZKERA POLONIAR ALDERANTZIZKOA
a+b	ab+
(a+b)/c	ab+c/
a+b/c	abc/+
(a+b)/(c-d)	ab+cd-/
(b+c)/(b-c)*d	bc+bc-d*/

3.2.5.- AGINDU-MOTEN ARTEKO KONPARAKETA

Konputagailuari dagokion agindu-motak CPU-ren egitura, mikroprogramazioa eta programen ezaugarriak baldintzatzen ditu. Programen ezaugarriei begira, zenbat eta eragigai gutxiago izan hainbat eta laburragoak izango dira aginduak, hauen exekuzioa azkarragoa izanik, zeren eta eragigaiak adierazteko lekua eta eragigai horiei dagozkien Memoriara atzipenak aurreratzen bait dira. Eragigai-kopurua jaitsiz, alde batetik agindu bakoitza laburragoa eta azkarragoa bihurtzen da, baina beste alde batetik agindu laburrek agindu gehiago erakartzen dute problema bera ebazteko; ondorioz programa luzatu eta moteldu ere egiten da. Orduan, ezin da agindu-mota hoberena zehaztu, baizik eta programa eta konputagailu konkretuen aurrean aztertu beharko dira ezaugarri hauek.

Hala eta guztiz ere, gaur egunean agindu-mota arruntenak eragagai bakarrekoak eta bi eragigaikoak dira, lehenengoak konputagailu txikietan (mikroetan) eta bigarrenak handietan erabiltzen direlarik.

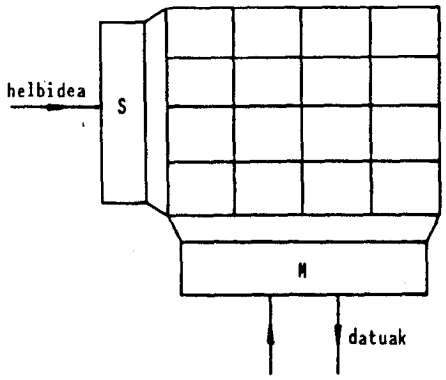
Bestalde konputagailu batzuk mota bat baino gehiagoko aginduak maneia dezaketela eta mota bakoitzean berezitasunak daudela kontutan hartzekoa da. Horrela, bi eragigaiko aginduak maneiatzen dituzten konputagailuetan eragigai bat askotan erregistro orokor bat izatea oso tipikoa da, Memoriari dagokion eragigai bat erregistro batek ordezkatzeak agindu laburragoak eta arinagoak erakartzen dituelarik. Horrela, kasu honetan agindu batek bi eragigai dauzka, baina metagailu bidezko konputagailuen antzeko abiadura behar du egikaritzeko.

3.3.- MEMORIA NAGUSIA

Esan dugunez, Memoria da konputagailuaren biltegia, informazioa pilatu eta gordetzen den lekua. Bi memoria-mota bereizten dira, Memoria Nagusia, CPU-rekin zuzenean loturik eta Memoria Laguntgarria, Sarrera/Irteera Unitateko osagaia dena. Memoria Nagusi azkar baina edukieraz mugatu baten ondoan Kanpo Memoria edo Memoria Lagunagarri zabal baina abiaduraz motelak egon ohi dira. Bestalde memorian bildutako informazioaren artean datuak eta aginduak bereizten direla ere jakina da. Azkenik, Memoriako eragiketak irakurketa eta idazketa dira, lehenengoaren bidez informazio bat eskuratzea eta bigarrenaren bidez datu edo agindu bat gordetzea lortzen delarik.

3.3.1.- MEMORIAREN EGITURA ETA ERAGIKETAK

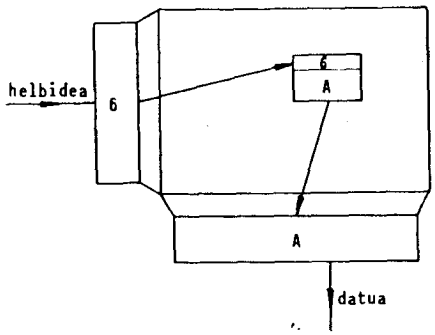
Memoriari lotuta bi erregistro agertzen zaizkigu (ikus 3.3. irudia): S erregistroa helbideetarako eta M erregistroa informazio-edukinetarako. S erregistroa, beraz, helbide-busarekin lotuta dago eta M erregistroa berriz datu-busarekin.



3.3 irudia.- MEMORIAREN EGITURA

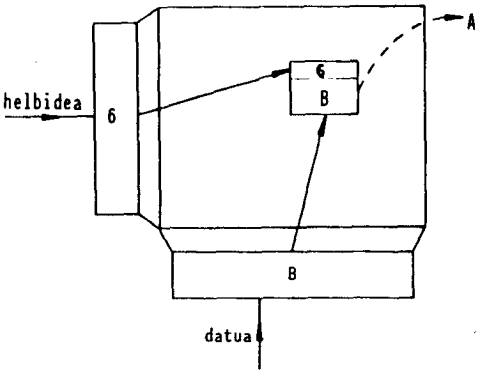
Azter ditzagun berriro Memorian egin daitezkeen eragiketak:

- IRAKURKETA: Memorian dagoen zerbait irakurri nahi dugunean informazio hori dagoen gelaskaren helbidea bidali behar da Memoriako S erregistrora eta irakurri ondoren, guk nahi genuen datua Memoriako beste erregistroan (M) edukiko dugu, Memoria zegoen zegoenean geratuko delarik. Datuak beraz, bere lekuan irauten du irakurria izan arren. (Ikus 3.4. irudia).



3.4 irudia.- MEMORIAKO IRAKURKETA

- IDAZKETA: Memorian zerbait idatzi nahi dugunean berriz, non eta zer idatzi nahi dugun adierazi behar da: lehenengoa, gelaskaren helbidea S erregistrora bidaliz eta bigarrena, hau da, gelaska horretan idatzi nahi duguna, M erregistroaren bitartez. Idatzi ondoren, helbide horretan zegoenaren ordez gure informazio berria egongo da, zaharra galduko delarik. (Ikus 3.5. irudia).



3.5 irudia.- MEMORIAKO IDAZKETA

3.3.2.- MEMORIA-MOTAK

Memoria Nagusien artean bi Memoria familia bereizten dira RAM Memoriak eta ROM Memoriak:

RAM Memoria.- Bertan irakurri eta idatzi egin dezakegu. Argindarraren etenaldi bat egonez gero, dagoen informazioa aldatu egiten da. RAM Memoriei memoria aktiboak deitzen zaie eta bigarren ezaugarriarengatik hegazkorrak direla esaten da.

ROM Memoria.- Memoria hauetan irakurri besterik ezin dugu egin, beraz, edukina finkoa da eta konputagailuaren bidez aldagaitza. Argindarraren etenaldia gertatutakoan informazioa mantendu egiten da. Horregatik, iraunkorrak direla esaten da. ROM Memoriei Memoria hilak ere deitzen zaie.

ROM eta RAM memoriak konparatuz ROM memoriak arinagoa eta iraunkorra izatea bere alde dauka baina eragozpen handi bat ere badu: informazioa aldatzeko ez duela balio eta beraz, funtzio konkretu batzuetarako bakarrik erabiltzen da; programa eta errutina bereziak egiteko batez ere. Horrela, Sistema Eragile mailako errutinak ROM memorian edukitzea oso tipikoa da eta mikroetan BASIC itzultzeko programa interpretatzailea ere bai. Dena den, konputagailu bat RAM memoriarik gabe ezin da ibili, baina askotan bi motatako memoriek Memoria Nagusia osatzen dute.

ROM memorien garapenaren ondorioz hiru mota sortu dira, ROM, PROM eta EPROM memoriak.

- ROM memoriak guztiz aldagaitzak dira eta frabrikanteari edukina zehaztuz eskatu behar zaizkio. Oso kopuru handiak bakarrik gertatzen dira errentagarriak.

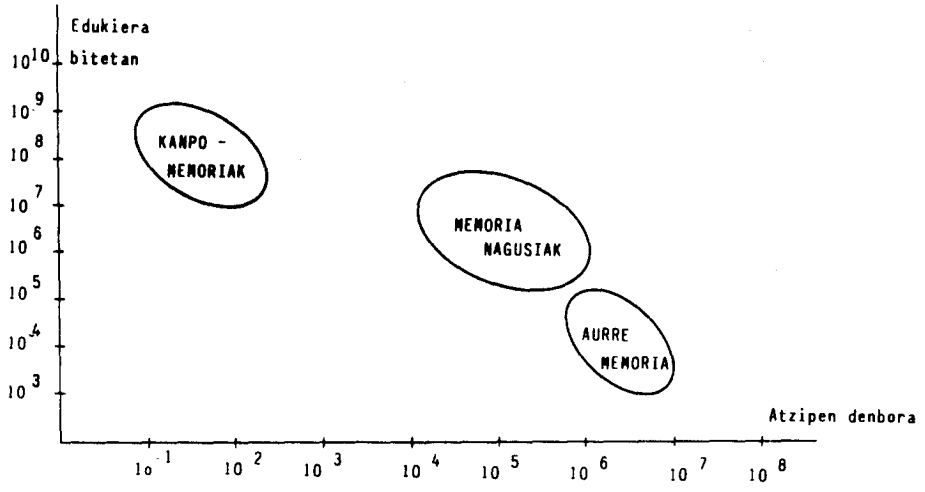
- PROM memoriak ROM programagarriak dira. Edukina behin betirako

alda dezakegu PROM memorien programagailua deritzon off-line dispositiboa erabiliz. Kasu honetan memoriako bit bakoitza fusible antzeko elementu bat da eta dispositibo programatzaileak, sartu nahi dugun informazioaren arabera, elementu bakoitza urtu egiten du ala ez. Beraz, PROM memoriak egin baino lehen ez da beren edukina ezagutu behar.

- EPROM memoriak ROM programarriak eta ezabagarriak dira, hau da, nahi adina bider alda dezakegu beren edukina EPROM memorien programagailua erabiliz. RAM eta EPROM memoriak antzekoak direla pentsatzea guztiz okerra da, zeren eta RAM memorietan edukinaren aldaketa CPU-k egiten bait du eta EPROM-etan berriz konputagailuarekin lotuta ez dagoen dispositibo batek, programagailuak.

Gaur egungo konputagailu handietan Memoria Nagusiak badu atal berezi bat Aurre-Memoria edo Cache Memoria.

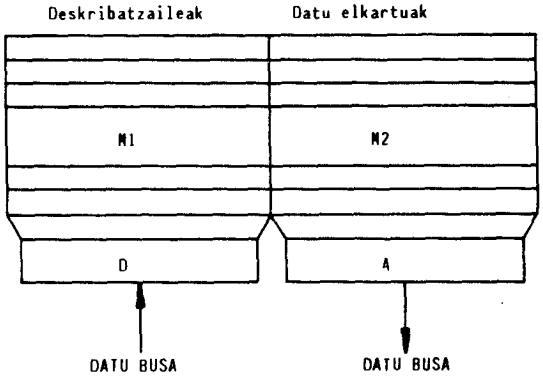
AURRE-MEMORIAK.- Memoria Nagusiak baino edukiera txikiagoa dute, baina askoz azkarragoak dira. Aurre-Memoria eta Memoria Nagusia lotuta daude haien arteko informazio-trukea oso arina izanik. Aurre-Memoriaren eta Memoria Nagusiaren arteko erlazioak Memoria Nagusiaren eta Kanpo-Memoriaren artekoarekin antza handia du, beraz, Memoria Nagusiko behar den informazioa Aurre-Memorian kokatzen da hasiera batean, CPU-k informazioa Aurre-Memorian eskuratu eta bilduko du eta Aurre-Memorian ez dagoen informazio bat behar denean, Aurre-Memoriaren eta Memoria Nagusiaren arteko informazio-trukea gertatuko da Aurre-Memorian dagoen informazioa Memoria Nagusian gordez eta beharrezkoa dena Aurre-Memorian kokatuz. Kanpo-Memoriaren, Memoria Nagusiaren eta Aurre-Memoriaren edukierari eta atzipen-denborari buruzko zehaztasunak daude 3.6 irudian.



3.6 irudia.- MEMORIA-MOTA DESBERDINEN EZAUGARRIAK

Badago beste memoria-mota berezi bat, Elkartze-Memoriak deitutakoena.

ELKARTZE-MEMORIAK.- Memoria hauetan informazio bat eskuratzeko ez da helbidea zehaztu behar edukina baizik. Beraz informazio bat, deskribatzailea deitutakoa, dagoen ala ez jakiteko eta dagoenean elkarturik daukan datu bat lotzeko balio dute. Edukinaren bidezko Memoriak helbideragarriak ere deitzen dira eta egitura berezia dagokie (ikus 3.7 irudia).



3.7 irudia.- ELKARTZE-MEMORIAREN EGITURA

Irakurri baino lehen, D erregistroan deskribatzailea kokatzen da eta irakurtzean D erregistroan dagoen informazioa M1 aldean dauden hitz guztiekin konparatzen da eta hitz batean edukin berbera baldin badago M2 aldean hitz horri dagokion hitza A erregistroan kokatuko da geroko erabilpenerako.

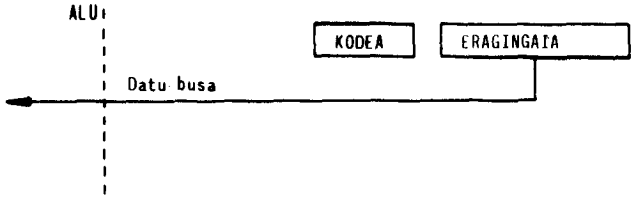
Elkartze-Memoriak zenbait erabilpen konkretutarako oso egokiak dira (tauletan bilaketak egiteko, parekatze-lanetan, ...) baina oraingoz garestiak dira. Adibidez, zenbakizko datak alfabetiko bihurtzen dituen programa batek hilabeteen bihurketa egiteko mota honetako taula bat erabiliko du Ø1/URTARRILA/Ø2/OTSAILA/.../12/ABENDUA. Taula hau Elkartze-Memoria batean kokatuz gero D, erregistroan hilabetearen zenbakia sartuz, dagokion izena lortzen dugu A erregistroan, Memoria Nagusia erabiliz askoz motelagoa gertatuko litzatekeelarik.

3.3.3.- HELBIDERATZE-MODUAK

2.mailako lengoaian aginduen eragigaiek gehienetan Memoriako helbide bat adierazten dute, erregistro orokor bat adieraztea posiblea bada ere. Memoriako helbidea modu desberdinetan zehaz dezakegu eragigaietan helbideratze-modu batzuren bidez. Hona hemen helbideratze-modu arruntenak:

A) BEREHALAKOA.- Eragigaiak datua bera zehazten du eta ez helbide

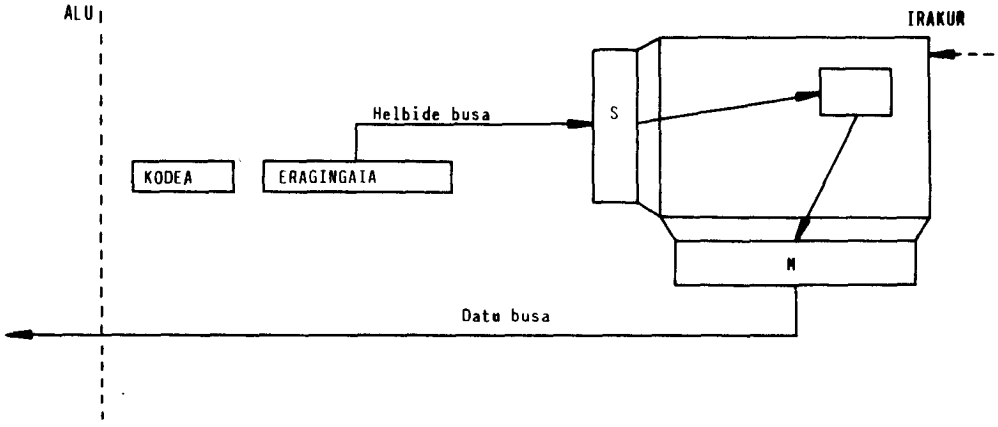
bat. Beraz, ez da helbideratze-modu propioa.
Eragigaiaren informazioa ALU-ra zuzenean joango da (ikus 3.8 irudia). Helbideratze-modu hau konstanteak adierazteko erabiltzen da.



3.8 irudia.- BEREHALAKO HELBIDERATZEA

B) ABSOLUTUA.- Eragigaiak datuaren helbidea zehazten du. Orain arte suposatu duguna da eta helbideratze-modu arruntena da. Helbideratze-modu zuzena ere deitzen da.

Eragigaiaren informazioa datu-busaren bidez erregistroan kokatuko da eta memoriako irakurketa bat eginez gero datua M erregistroan lortzen da, datu-busetik bidaltzeko prest (ikus 3.9 irudia).

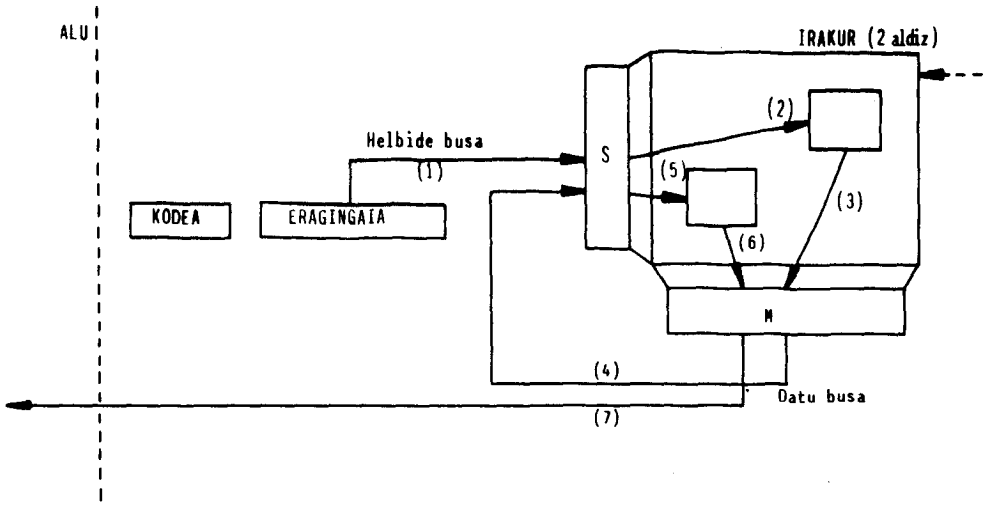


3.9 irudia.- HELBIDERATZE ABSOLUTUA

C) ZEHARKAKOA.- Eragigaiak datuaren helbidea edukitzen duen hitz baten helbidea zehazten du. Beraz, datua eskuratzeko, Memorian bi aldiz irakurri behar da, batean berezko helbidea irakurtzen da eta bigarrenean datua.

Eragigaiaren informazioa datu-busaren bidez S erregistroan kokatuko da eta Memoriako lehenengo irakurketa gertatutakoan M erregistroan datuaren helbidea kokatzen da. Informazio hau datu-busetik aterako da eta nolabait helbide-busera pasako da S erregistroan kokatzeko. S erregistroan kokatuz gero, bigarren irakurketa gertatzen da eta M erregistroan datua

agertuko da. (ikus 3.10 irudia).



3.10.- irudia.- ZEHARKAKO HELBIDERATZEA

Konputagailu batzuk N mailako zeharkako helbideratzea onartzen dute. N mailak, M erregistroan agertzen dena N bider helbidea izango dela adierazten du eta horren arabera datua eskuratzeko N+1 Memoriako irakurketa beharko dira.

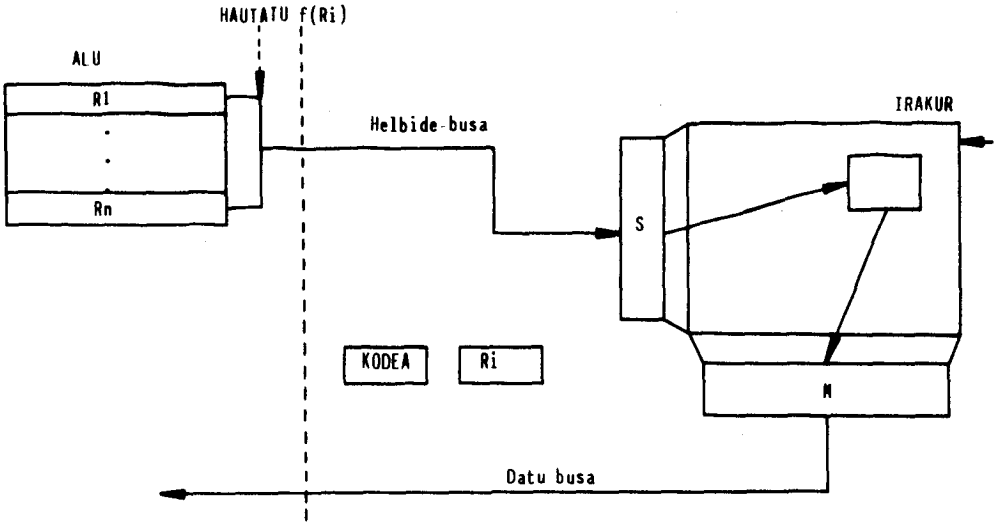
D) ERREGISTRO BIDEZKO ZUZENEKOA.- Eragigaiak erregistro orokor baten zenbakia zehazten du, erregistro horren edukina datua izango delarik. Beraz ez dago Memoriako irakurketarik eta datua ALU-k bere barnetik eskuratuko du.

E) ERREGISTRO BIDEZKO ZEHARKAKOA.- Eragigaiak datuaren helbidea daukan erregistro orokor baten zenbakia zehazten du. Beraz, datua lortzeko Memoriako irakurketa bat beharrezkoa izango da.

Eragigaiak zehazten duen erregistroa hautatzen da eta bere edukina helbide-busaren bidez S erregistroan kokatzen da, gero Memoriako irakurketa gertatzen da eta M erregistroan datua lortzen da, ondoren datu-busetik kanporatzeko (ikus 3.11 irudia).

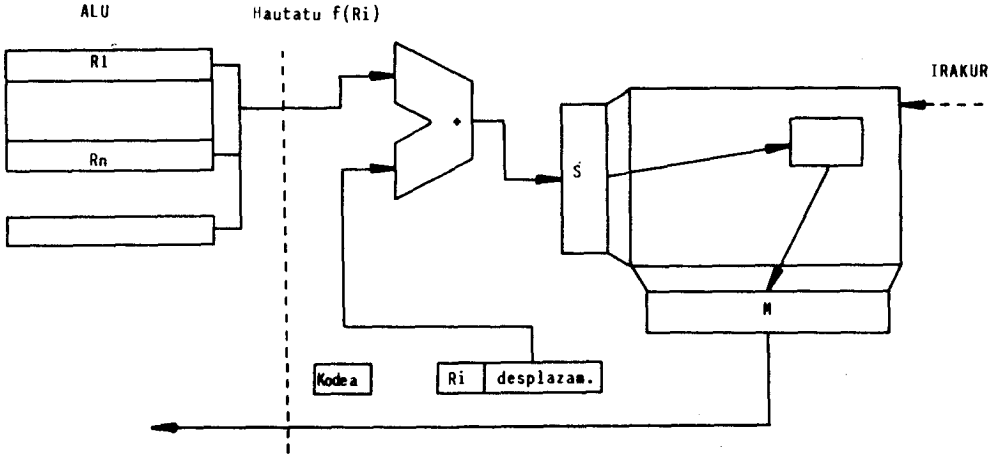
Batzutan erregistrozko zeharkako helbideratze N mailakoa onartzen da, hau da, memoriatik irakurtzen den informazioa (N-1) aldiz ez da datua, helbidea baizik eta Ngarren irakurketan datura atzitzen da. Adibidez VAXean erregistrozko zeharkako helbideratze 2.mailakoa erabiltzen da.

F) ERLATIBOA.- Eragigaiak bi zatitan banatzen dira, erregistro baten aipamena alde batetik eta desplazamendua bestetik. Datuaren helbidea, erregistroaren edukina eta desplazamendua gehituz lortzen da. Erregistroa orokorra edo berezia izan daiteke kasuaren arabera. Beraz, erregistroa hautatzen da eta dago-



3.11 irudia.- ERREGISTRO BIDEZKO ZEHARKAKO HELBIDERATZEA

kion edukina desplazamenduarekin gehitzen da, batura helbide-busaren bidez S erregistroan kokatuko da. Honez gero, Memoriako irakurketa bat gertatuko da, datua M erregistroan kokatuz. Ondoren, datu-busean zehar ALU-ra joango da (ikus 3.12 irudia).



3.12 irudia.- HELBIDERATZE ERLATIBOA

Helbideratze erlatiboak, betetzen dituzten funtzioen eta erabiltzen duten erregistroaren arabera, hiru motatakoak izan daitezke: oinarri erregistro bidezko erlatiboak, indize-erregistrozkoak eta programaren kontagailuzkoak. Haien arteko desberdintasuna 5. gaian aztertuko dugu.

Konputagailu batzuek helbideratze erlatibo zeharkakoa erabiltzen dute, non erregistroaren edukinaren eta desplazamenduaren artean ez bait da datuaren helbidea lortzen, datua-

ren helbidea daukan hitz baten helbidea baizik. Horren ondorioz, helbideratze erlatibo zeharkakoaren kasuak Memoriako bi irakurketa eskatzen ditu.

G) PILA BIDEZKOA.- Memorian dagoen pila egituran metatzen dira helbideak eta pilazko eragigai bat datorrenean, pilaren tontorrean dagoen informazioak datuaren helbidea zehazten du.

Ez dira pilazko konputagailuak eta pilaren bidezko helbideratzeak nahastu behar, lehenengoek ALU dagoen datu-pila erabiltzen dute, pilaren bidezko helbideratzeak berriz edozein konputagailu-motatako agindutan eragigai bezala erabil daitezke eta erabiltzen duten pila Memorian dago eta edukinak helbideak dira.

HELBIDERATZE-MODUA	ERAGIG.	MEMORIAKO HELBIDEA	DATUA
BEREHALAKOA	a	---	a
ABSOLUTUA	a	a->h	edukin(h)
ZEHARKAKOA	a	edukin(a)->h	edukin(h)
ERREGISTROZKO ZUZEN.	Ri	---	edukin(Ri)
ERREGISTROZKO ZEHARK.	Ri	edukin(Ri)->h	edukin(h)
ERLATIBOA	Ri,d	edukin(Ri)+d->h	edukin(h)
PILA BIDEZKOA	--	pop(pila)->h	edukin(h)

3.1 taula.- HELBIDERATZE-MODUEN XEHETASUNAK

Laburpen-eran eta kontzeptuak argitzeko 3.1 taulan helbideratze-modu guztiak zehazten dira.

3.3.4 HELBIDERATZE-MODUAREN ZEHAZTAPEN-MODUA

Eragigai baten helbideratze-modua ezagutuz gero, adierazten duen datua nola lortu aurreko pasartean aztertu dugu, baina nola daki CPU-k eragigai bati dagokion helbideratze-modua? Agindu barruan zehazturik egon beharko du, ondoko bi metodoen arteko bat erabiliz:

A) ERAGIGAIAN ZEHAZTURIK.- Konputagailu batzutan 2.mailako agindetan eragigai bakoitza bi zatitan banatzen da, helbideratze-modua adierazten duten bit batzu daude alde batetik eta helbideratzea bera adierazteko bitak bestetik (ikus 3.13 irudia).

B) KODEAK BEHARTURIK.- Beste konputagailutan irtenbide desberdina hartzen da, Agindu-kodeak helbideratze-modua edo moduak behartzen du (ikus 3.13 irudia). Horrela, konputagailu batean eragiketa bera burutzeko (batuketa adibidez) kode desberdinak egongo dira, bakoitzak helbideratze-modu desberdina erabiliz (adibidez batek erregistroen arteko batuketa, beste batek erregistro baten eta Memoriaren arteko batuketa eta abar).

AGINDU- -KODEA	1.ERAGIGAIA		2.ERAGIGAIA	
	Modua	Helbideratzea	Modua	Helbideratzea

(Eragigaian zehazturiko moduak)

AGINDU- -KODEA	1.ERAGIGAIA Helbideratzea	2.ERAGIGAIA Helbideratzea

(Kodeak beharturiko moduak)

3.13 irudia.- HELBIDERATZE-MODUEN ADIERAZPENA

Beraz, lehenengo kasuan eragiketa bakoitzari agindu orokor bat dagokio, eragigaiek helbideratze-modua zehazten dutelarik, bigarren kasuan berriz eragiketa eta helbideratze-modu bakoitzari kode desberdin bat dagokio, eragiketa bera adierazteko kode bat baino gehiago dagoelarik. Horren ondorioz, teorian behintzat, helbideratze-modua eragigaian zehazten denean, eragigai luzeagoa gertatzen da, kodeak helbideratze-modua zehazten duenean kode gehiago behar dira eta, noski, kodea adierazteko bit gehiago.

2. mailako aginduetan helbideratze-modua eragigaiaren bit batzuen bidez zehazten da aztertu dugun lehenengo kasuan, baina 4. mailako lengoaian, mihiztadura-lengoaian alegia, helbideratze-moduak adierazteko zenbait ikur erabiltzen dira. Adibidez, VAX-eko mihiztadura-lengoaian 2 zenbakiari, aurrean dauzkan ikurren arabera, 3.2 taulan aurkezten diren esangura desberdinak dagozkio.

IKURRA	ESANGURA
2	2 berehalakoa
@ 2	2 absolutua
R2	R2zko zuzena
(R2)	R2zko zeharkakoa
@(R2)	R2zko zeharkako 2.mailakoa

3.2 taula.- HELBIDE BATEN ESANGURAK

Hala eta guztiz ere, mihiztadura-lengoaian, helbideratzeak zehatz-mehatz adierazteaz aparte, aurrean aurkeztu dugun moduan hain zuzen ere, (helbideratze esplizitua erabili dugula esango dugu) beste era batean adieraz daiteke helbideratzea: sinbolo baten bidez (DATUA,ZENBAKI,...) eta orduan helbideratze inplizitua edo sinbolikoa erabiltzen dela esaten da. Helbideratze inplizitua 4. mailako lengoaiaren itzultzaileak, mihiztatzaileak alegia, bihurtuko du esplizitu.

3.4.- UNITATE ARITMETIKO-LOGIKOA

3.4.1.- EGITURA

Unitate aritmetiko-logikoak Kontrol-Unitatearen agindupean eragiketak burutzen ditu, Memoriako datuak erabiliz. Funtzio hau betetzeko, ondoko osagai hauek ditu:

- Hiru erregistro bereziak non bitan eragigaiek zehazten dituzten datuak kokatzen bait dira eta bestean emaitza. Hiru erregistro hauei 1.eragigaiaren erregistroa, 2.eragigaiaren erregistroa eta emaitzaren erregistroa deituko diegu. Hiru erregistro hauek datu-busarekin lotuta daude.

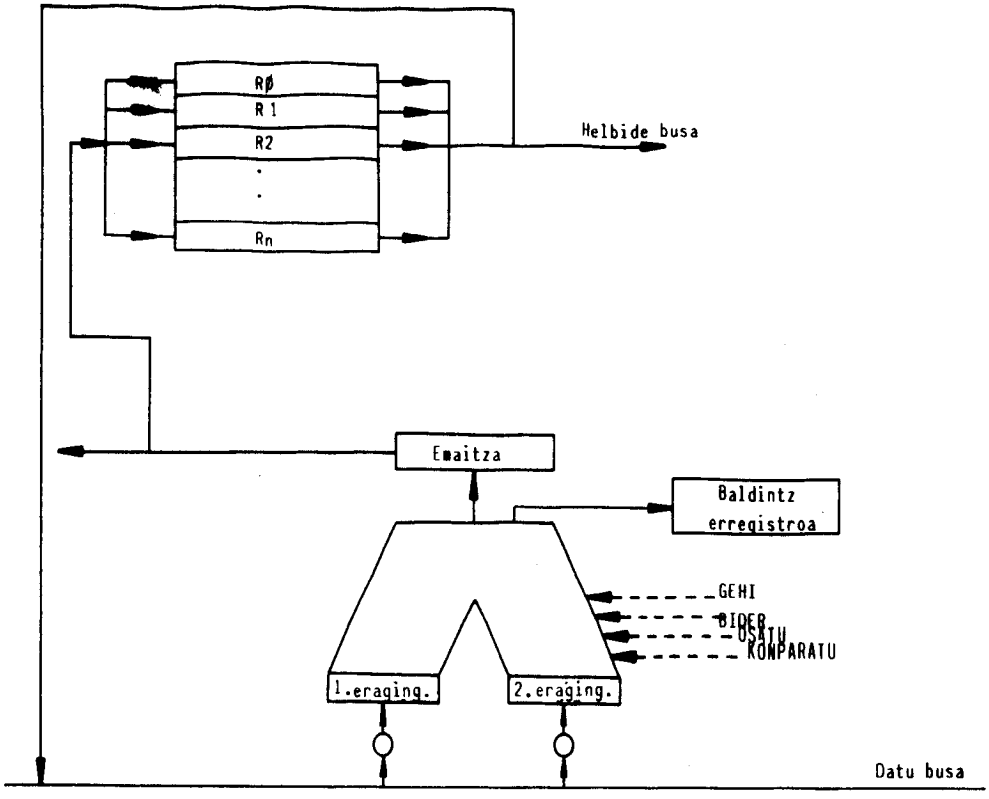
- Baldintz erregistroa. Erregistro honetan emaitza logikoak eta egoera bereziak adierazten dira, bit zehatzen multzo baten bidez. Adibidez, emaitzaren zeinua, gainezkada erroreak eta abar adierazten dira erregistro honetan. Erregistro honen informazioa Kontrol-Unitateak erabiliko du erabaki batzu hartzeko orduan eta horregatik ezin da ALU-ko erregistro bat dela esan, CPU-koa baizik.

- Erregistro orokorrak. Konputagailu gehienetan programa batek oso maiz erabiltzen dituen informazioak gordetzeko, Memorian kokatu ordez, erregistro orokorretan biltzen dira, programa azkarragoa izan dadin. Erregistro bakoitzean zenbakizko datu bat edo helbide bat sartzen da eta edukinik arruntena 16 edo 32 bitekoa da. Erregistro orokorren kopuru hedatuenak 8 eta 16 dira, zenbait konputagailutan ez badaude ere.

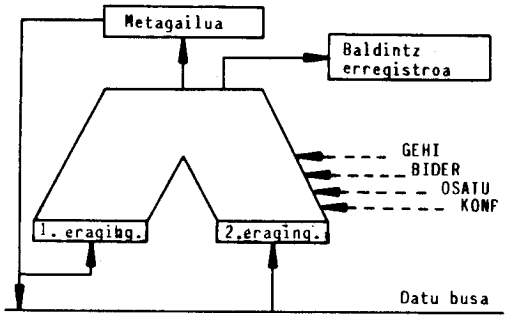
- Zirkuitu eragileak. Erregistroek aparte, ALU-a zirkuitu eragile batzuek ere osatzen dute, zeintzuk eragigaien erregistroen informazioa hartuta prozesatzen bait dute, Kontrol-Unitatearen aginduak jarraituz. Unitate aritmetiko-logikoa, oso oinarrizko eragiketak bakarrik burutzeko gai dela kontutan hartuz, ALU-a ondoko zirkuituek soilik osatzen dutela ulergarria izango da:

  -Batutzailea (batuketak burutzeko)
  -Biderkatzailea (biderkaketak burutzeko)
  -Osatzailea (osagarriak burutzeko)
  -Desplazatzailea(biderkaketan laguntzeko)
  -Konparatzailea (konparaketak burutzeko)

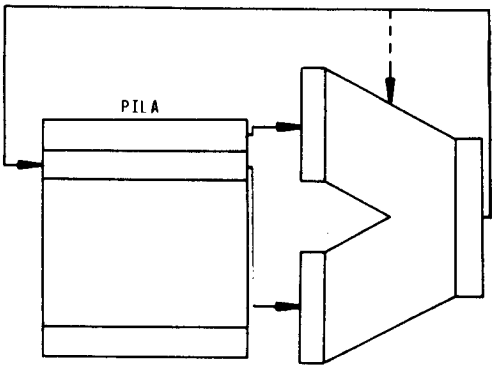
Hala eta guztiz ere, esan dugunez, maneiatzen duen 2.mailako agindu-motaren arabera egitura desberdina dagokio konputagailuari eta aurrerago aipatutakoan aldaketaren bat gerta daiteke konputagailu-motaren batean.



2. eta 3. helbideetako konputagailuen ALU ren egitura (erregistroarekin)



Metagailuzko konputagailuen ALU ren egitura (erregistrorik gabe)



Pila bidezko konputagailuen ALU-ren egitura

3.14 irudia.- ALUren EGITURA

3.14 irudian konputagailu-mota guztiei dagozkien egiturak zehatz-mehatz aurki daitezke.

3.4.2. ADIBIDE BAT: BATUTZAILEA

Zirkuitu eragilea 3.14 irudian unitate trinko bat bezala agertzen da baina batutzaileak, desplazatzaileak, osatzaileak, konparatzaileak eta batzutan biderkatzaileak osatzen dute unitatea eta hauetako bakoitzak bere osagarriak ditu. Adibide bezala batutzaile baten egitura aztertuko dugu. Zirkuitu eragile gehienak bezala AND, OR eta XOR ateen bidez eginda daude.

Ate hauek AND, OR eta XOR (ETA, EDO eta ALA) funtzio logikoak burutzen dituzte. Hona hemen eragiketa logiko hauen ondorioa adierazten duten taulak:

AND	Ø	1
Ø	Ø	Ø
1	Ø	1

OR	Ø	1
Ø	Ø	1
1	1	1

XOR	Ø	1
Ø	Ø	1
1	1	Ø

Beraz, hiru funtzioek hartutako bi biten balioaren arabera bit bat lortzen dute. AND funtzioak hartutako bi bitak 1 balioa badute soilik 1 balioa lortzen du, OR-ak berriz Ø balioa lortzeko hartutakoak Ø balioa izatea eskatzen du, XOR-ak azkenik hartutakoak berdinak baldin badira Ø balioa sortzen du eta desberdinak izanik 1 balioa.

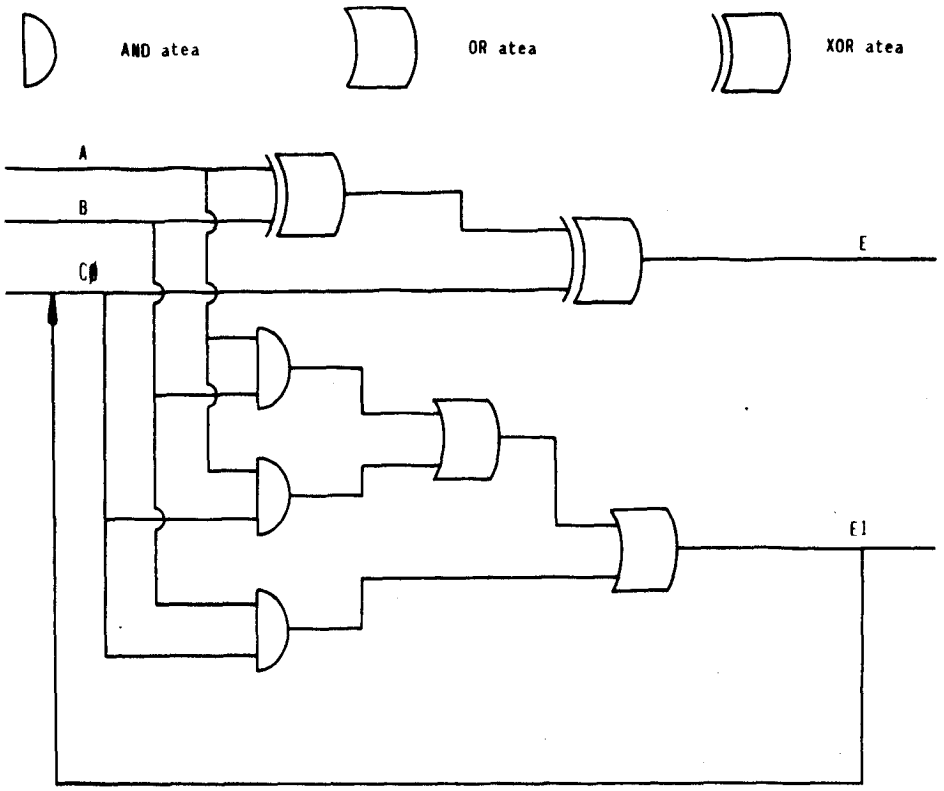
Orduan nola egin batutzaile bat ate hauen bidez? Azter dezagun batuketa bitarra. 1. eragigaiko bita (A), 2. eragigaiko bitarekin (B-rekin) gehitzeko aurreko bururakoa (CØ) kontutan hartu behar dela, emaitza (E) eta hurrengo bururakoa (C1) lortzeko gogoratuz gero, batuketa bitarrari dagokion egitaula 3.3 taulan dugu.

A (1.erag)	B (2.erag)	CØ (burur.)	E (emaitza)	C1 (burur.)
Ø	Ø	Ø	Ø	Ø
Ø	Ø	1	1	Ø
Ø	1	Ø	1	Ø
Ø	1	1	Ø	1
1	Ø	Ø	1	Ø
1	Ø	1	Ø	1
1	1	Ø	Ø	1
1	1	1	1	1

3.3 taula.- BATUTZAILE BATEN EGITAULA

Beraz E-k 1 balioa hartzen du A, B eta CØ artean duten 1 balioen kopurua bakoitia denean (1 edo 3) eta C1-en 1 balioen kopurua 2 edo gehiago denean.

3.15 irudian batutzaile baten egitura dugu.



3.15 irudia.- BATUTZAILE BATEN EGITURA

3.15 irudiaren egitura egiaztatzeko dagokion egitaula 3.4 taulan azter daiteke.

A	B	CØ	F	E	G	H	I	J	C1
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
Ø	Ø	1	Ø	1	Ø	Ø	Ø	Ø	Ø
Ø	1	Ø	1	1	Ø	Ø	Ø	Ø	Ø
Ø	1	1	1	Ø	Ø	Ø	1	Ø	1
1	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø
1	Ø	1	1	Ø	Ø	1	Ø	1	1
1	1	Ø	Ø	Ø	1	Ø	Ø	1	1
1	1	1	Ø	1	1	1	1	1	1

F: A XOR B
G: A AND B
I: B AND CØ
C1:((A AND B) OR (A AND CØ)) OR (B AND CØ) -------> BURURAKOA

E: (A XOR B) XOR CØ --> EMAITZA
H: A AND CØ
J: (A AND B) OR (A AND CØ)

3.4 taula.- BATUTZAILEAREN PAUSOAK

Ateak oinarrizko osagaiak izanik batutzaile bat egitea ez dela oso gauza zaila esan daiteke. Edozein elektronika-dendatan AND, OR eta XOR ateak aurki ditzakezu.

3.5.- KONTROL-UNITATEA

3.5.1.- EGITURA

Konputagailuaren funtzionamendu guztia gidatu eta kontrolatzen du Kontrol-Unitateak. Aginduak hartu, interpretatu eta betearazi egiten ditu, horretarako beharrezko diren mikroaginduak (kontrol-seinaleak) sortu eta dagozkionari bidaliz. Mikroagindu batzu ALU-ari igorriko dizkio eragiketak bete ditzan, beste zenbait Memoriara (irakurri edota idatzi behar denean), Sarrera/Irteerako unitateetara eta abar.

Labur esateko, KU-aren funtzioak hauexek dira:

- Une guztietan, bete beharreko agindua zein den ezagutu; aginduak Memorian daudenez gero, KU-ak jakin beharko du zein helbidetan dagoen une horretan egikaritu behar duen agindua.

- Exekutatu behar den agindua eskuratu.

- Agindu hori deskodetu, hau da ulertu eta interpretatu, exekutatua izan dadin beharrezko diren mikroaginduak sortu eta dagozkionari bidali.

Eta funtzio horiek bete ahal izateko KU-ak honako osagai hauek behar ditu:

- Programaren kontagailua (PC): Lehen aipatu ditugun funtzioen arteko aurrenekoa betetzeko erabiltzen dugu. Programaren kontagailua erregistro bat da, non beti une horretan exekutatzen ari den aginduaren helbidea gordetzen bait da. Horrela, agindu bat exekutatu eta gero lehen zeukan helbideari agindu horren luzera gehitu eta hurrengo aginduaren helbidea lortzen du. Helbide-busarekin lotuta dago bai PC-tik informazioa ateratzeko bai sartzeko.

- Aginduaren erregistroa: Bigarren funtzioari dagokio. Erregistro honetan gordetzen du agindua Memoriatik jaso ondoren. Eragiketaren informazio guztia biltzen da bertara: bai eragiketaren kodea (zer eragiketa den, alegia), bai eragigaiak. Datu-busarekin lotuta egongo da, aginduak Memoriatik eskuratzen bait dira.

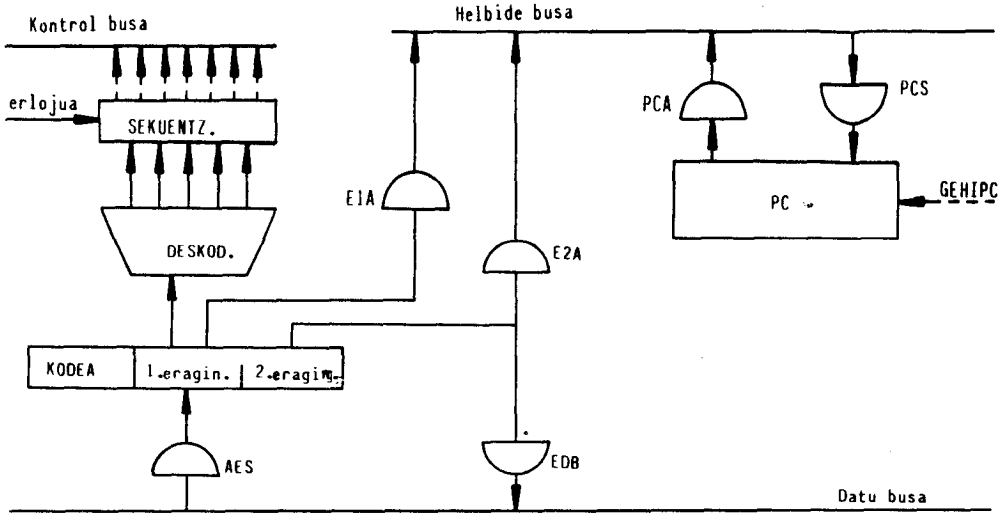
- Deskodetzailea: Aginduaren erregistrotik eragiketaren kodea hartu eta deskodetu egiten du, hau da, kode hori zein eragiketari dagokion ezagutu; gero mikroaginduak sortu eta bidali egingo ditu. Deskodetzaileak mikroaginduak sortzeko eragiketaren kodeaz aparte helbideratze-moduaren zehaztapena eta erregistroen zenbakiak zehaztuta daudenean erabiliko ditu.

- Sekuentziadorea: Deskodetzaileak sorturiko mikroagindu ho-

riek denboran zehar banatu eta datu-busean kokatzen ditu sekuentziadoreak. Mikroaginduak denboran zehar banatzeko denbora neurtzeko dispositibo bat erabiltzen du, erlojua. Erloju honek maiztasun finko batez seinale bat bidaltzen du. Erloju honen periodoari ziklo-denbora deritzo eta funtsezko ezaugarria dugu konputagailuetan.

Orain arte unitateak busekin zuzenean lotu izango balira bezala ikusi dugu, baina errealitatean ez da horrela gertatzen, baizik eta konexio bakoitzean AND ate bat bait dago. Ate bakoitza Kontrol-Unitateak sortutako mikroagindu baten bidez irekitzen da.

3.16 irudian 2 helbideko konputagailu bateko Kontrol-Unitatearen egitura aurkezten da, non ate bakoitzari dagokion izena mikroaginduaren izen sinbolikoa bait da. Beste motatako konputagailuetako Kontrol-Unitatetan desberdintasun bakarra eragigaien kopuruaren arabera busekin dituen konexioen kopurua izango da.



3.16 irudia.- 2 HELBIDEKO KONTROL-UNITATEAREN EGITURA

3.5.2.- MIKROAGINDUAK

Kontrol-Unitateak sortutako eta Kontrol-busetik zabaldutako Kontrol-seinaleak mikroaginduak edo mikroordenak deitzen dira. Mikroagindu hauek Memoriara, ALU-ra eta Kontrol-Unitatera bertara zuzendurik egongo dira. Mikroagindu hauek 1.mailako lengoaia osatzen dute eta pultsu elektronikoak badira ere guk sinbolo batzuren bidez adieraziko ditugu.

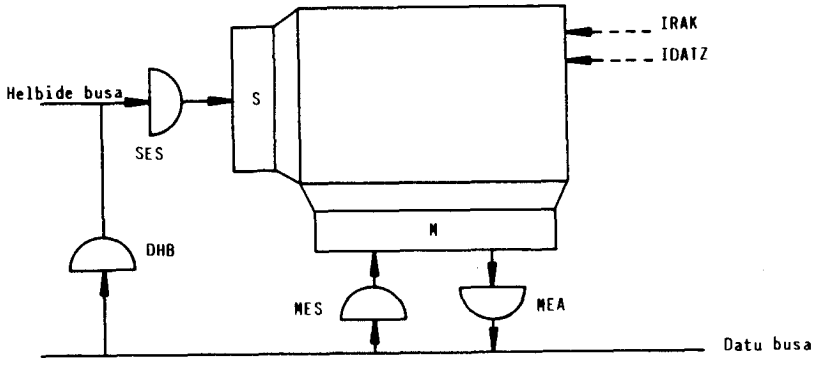
Bi mikroagindu-mota bereizten dira betearazten duten funtzioaren arabera: Mikroagindu eragingarriak eta bideratzaileak.

Mikroagindu eragingarriek eragiketa efektiboak burutzea agintzen dute. Mikroagindu hauek honako hauek dira:

IRAK.- Memoriatik irakurtzeko mikroagindua.
IDATZ.- Memorian idazteko mikroagindua.
GEHI.- ALU-n bi datuen arteko batuketa.
OSA.- Datu baten osagarria ALU-n.
DESPL.- Datu baten bitak desplazatu ALU-n.
KONP.- ALU-n bi datuen arteko konparaketa.
NOP.- Datu bat emaitzaren erregistroan utzi.
GEHIPC.- PC-ri aginduaren luzera gehitu Kontrol-Unitatean.

Beste batzu egon badaitezke ere, aipatutakoak arruntenak dira.

Mikroagindu bideratzaileek informazioa nondik nora bideratu behar den agintzen dute. Horretarako mikroagindu bakoitzak betetzen duen funtzio bakarra AND ate batetik informazioa pasatzen uztea da. AND ateak Memorian, ALU-n eta Kontrol-Unitatean kokaturik daude egitura azaltzean aipatu ez badugu ere. 3.16 irudian Kontrol-Unitatearen eskeman unitatre honi dagozkionak zehaztuta daude; 3.4 eta 3.14 irudietan berriz ez dira marraztu baina aipatuz gero 3.17 eta 3.18 irudietan marraztuta agertuko dira.



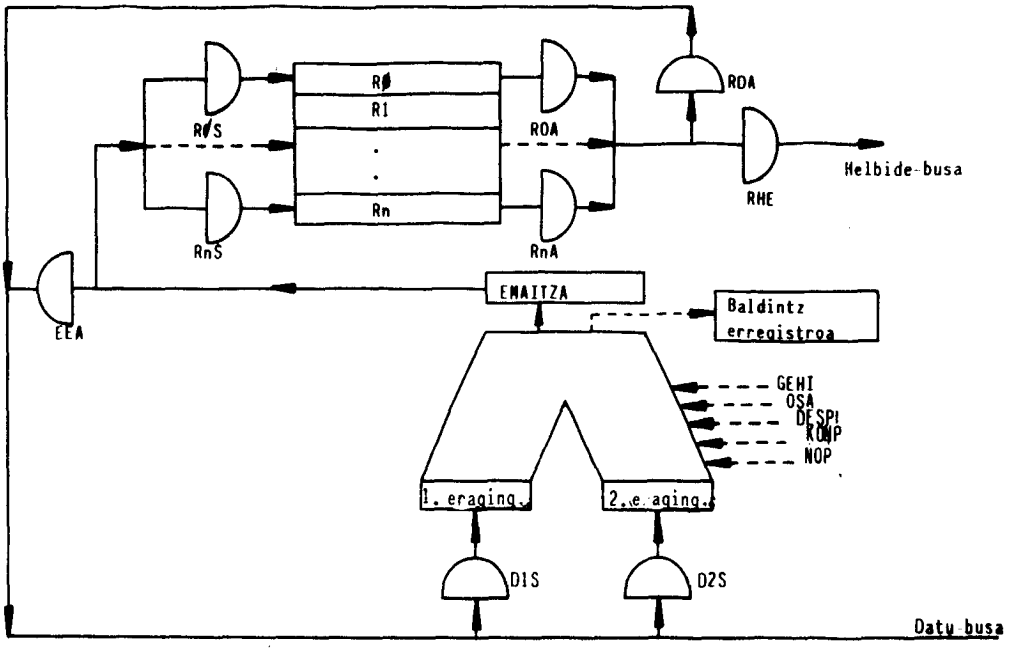
3.17 irudia.- MEMORIAKO EGITURA ZEHATZA

Mota honetako mikroaginduak honako hauek dira:

- Memoriari dagozkionak (ikus 3.17 irudia):

SES.- S erregistroan sarrera adierazten du helbide-busa eta S erregistroa lotuz.
MES.- M erregistroak sarrera adierazten du M erregistroan datu-busetik datorren informazioa sartzeko.
MEA.- M erregistrotik atera. Informazioa datu-busetik bideratuko da.
DHB.- Datuak helbide bihurtu. Helbideratze-modu zeharkakotan erabiltzeko ate honen bidez datu-busetik helbide-busera in-

formazioa pasarazten du Kontrol-Unitateak.



3.18 irudia.- ALU-KO EGITURA ZEHATZA

- Unitate aritmetiko-logikoari dagozkionak (2 eragigaiko konputagailutan, ikus 3.18 irudia):

  D1S.- 1.datuaren sarrera. 1.datua datu-busetik eskuratzeko.
  D2S.- 2.datuaren sarrera. 2.datua datu-busetik eskuratzeko.
  EEA.- Emaitzaren erregistrotik atera. Emaitza datu-busera bidaltzeko.
  RnS.- n. erregistroan sartu. Hautatu erregistro bat emaitza bat sartzeko. Erregistro bakoitzarengatik bat.
  RnA.- n. erregistrotik atera. Erregistro bat hautatu informazioa ateratzeko. Erregistro bakoitzarengatik bat.
  RDA.- Erregistrotik ateratakoa datu-busera. Erregistrozko helbideratze zuzenetan erabilia.
  RHE.- Erregistrotik ateratakoa helbide-busera. Erregistrozko helbideratze zeharkakotan erabilia.

- Kontrol-Unitateari dagozkionak (2 eragigaiko konputagailutan).

  AES.- Aginduaren erregistroan sarrera datu-busetik.
  E1A.- 1. eragigaitik atera helbide busera.
  E2A.- 2.eragigaitik atera helbide-busera.
  EDB.- Eragigaitik datu-busera 2. eragigaia berehalako helbideratzea denean.
  PCA.- PC-tik atera helbide-busera.
  PCS.- PC-n sartu helbide-busetik.

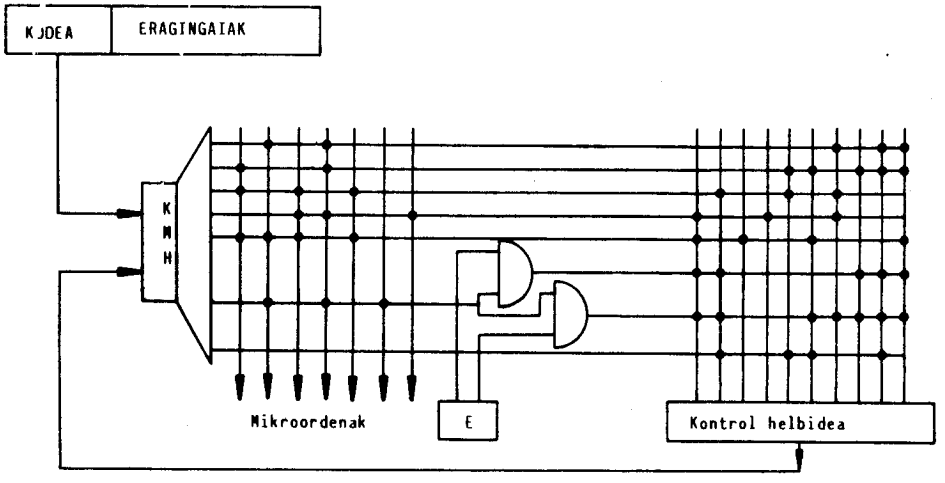
3.6.- MIKROPROGRAMAZIOA

Agindu bat mikroagindu eragingarri eta bideratzaileen segida baten bidez egikaritzen dela ikusi dugu. Agindu eta mikroaginduen arteko baliokidetasuna finkoa dela suposatu badugu ere, aldakorra izan daiteke. Horretarako Kontrol-Unitatean Kontrol-Memoria bat dago, non agindu bakoitzari dagozkion mikroaginduak agertzen bait dira.

Kableaturen ordez Kontrol-Memoria bat duten Kontrol-Unitatei eta dagozkien konputagailuei mikroprogramagarriak edo aldakordun eragiketa-kodekoa deitzen zaie mikoprogramazioaren bidez makina-lengoaiazko agindu berriak definitu daitezke eta. Beraz, kasu honetan 2.mailako lengoaia ez dago makinarekin guztiz lotuta, baizik eta 1.mailako aginduen bidez definitu bait daiteke.

Orain arte aginduak burutzeko sortzen diren seinaleei mikroaginduak edo mikroordenak deitzen genien, baina mikroprogramazioan berezi egin behar da zer den mikroagindua eta zer mikroordena. Agindu bati mikroprograma bat dagokio eta mikroprograma osatzen duten aginduei mikroaginduak deitzen zaie. Mikroagindu baten egikaritzapenak mikroordena bat edo gehiagoren exekuzioa erakarrtzen du. Beraz, agindu bat mikroagindu-multzo batek osatzen du, mikroagindua mikroordenez osatuta dagoelarik.

Mikroprogramazioaren funtsezko ideiak Wilkes-ek 1951.ean definitu zituen bere makina diseinatu zuenean. 3.19 irudian Wilkes-en makinaren eskema adierazten da.



3.19 irudia.- WILKES-EN MAKINAREN EGITURA

Wilkes-en eredua jarraituz, aginduaren eragiketa-kodeak aginduari dagokion mikroprogramako lehenengo mikroaginduaren helbidea adierazten du. Helbide hori KMH erregistrora (Kontrol-Memoriako Hautaketa) bidaltzen da. Helbide honen bidez mikroaginduari dagokion hitza aktibatzen da eta bi ondorio lortzen dira: Alde batetik ezkerreko aldean mikroordena bakoitzeko bit bat dago, zeinek 1 egoeran egonez dagokion mikroordena egikarituko dela adierazten bait du. Beraz bit hauen bidez mikroordenak sortuko dira. Eskuineko aldeko informazioaz mikroprogramaren hurrengo mikroaginduaren helbidea lortzen da, Kontrol-Helbidea izeneko erregistroan kokatuko delarik.

Beraz, bi alderdi daude, bata mikroagindu-kodea da eta zuzenean mikroaginduari dagozkion mikroordenak adierazten ditu, eta bestea mikroaginduen sekuentzia da, non hurrengo mikroaginduaren helbidea zehazten bait da.

E izeneko karratuan baldintzapeko aginduetarako mekanismoa adierazten da. E biegonkor bat da eta, balioaren arabera, hurrengo mikroaginduaren helbidea desberdina izango da. Baldintza bat betetzen denean hurrengo aginduaren helbidea PC-n dagoenaren ordez eragigaiak adierazten duen helbidea dela adieraztea baldintzapeko aginduen bidez egiten da.

Wilkes-ek asmatutako kontzeptuek mikroprogramazioaren oinarria badira ere, oso garapen handia izan dute. Horrela mikroaginduaren kodea adierazteko ondoko bi metodoak daude:

- Agindu motako kodeketa. Kodeketa honek mikroaginduari agindu baten formatoa ematen dio, beraz, kodeak eta eragigaiak mikroagindua osatuko dute. Oso mikroagindu laburrak lortzen dira baina deskodeketa konplexu samarra gertatzen da.

- Eremuen bidezko kodeketa. Metodo honetan mikroordenak taldeka banatzen dira eta mikroagindu guztiek gehienez talde bakoitzeko bat erabiliko dute. Talde bakoitzari eremu bat dagokio eta eremu horretan mikroaginduari dagokion mikroordena kodetzen da. Kodeketa honen deskodeketa askoz errazagoa da, baina luze samarra gertatzen da, Wilkes-en kodeketaren antza handia duelarik.

Mikroaginduen sekuentzia adierazteko ere bi metodo daude, bata Wilkes-ek erabilitakoa da eta sekuentzia inplizitoa deitutakoa bestea. Sekuentzia inplizitoa izeneko metodoan ez da ezer zehazten eta Kontrol-Unitateak hurrengo agindua Kontrol-Memoriako hurrengo helbidean dagoela badaki.

Kontrol-Memoriek Memoria Nagusia baino askoz azkarragoak izan behar dute, bestela aginduen egikaritzapena atzeratuko bait luteke. Kontrol-Memorietan gehienetan irakurri egiten da eta agindu berriak definitu edo aginduren bat aldatu nahi denean bakarrik idatziko da. Beraz, idazteko denbora ez da oso garrantzi handikoa

izango, baina irakurtzekoa bai.

3.7.- KONPUTAGAILU BATEN BARRUKO FUNTZIONAMENDUA

Kapitulu honetan unitateen egiturak banan bana aztertuko ditugu. Agindu bat egikaritzen denean informazioaren nondik norakoa aztertzea da oraingo helburua eta horretarako oso garrantzi handiko mikroaginduak edo ordenak dauzkagu, zeren eta, ikusi dugunez, konputagailuaren barruan mugimendu edo eragiketa bat gertatzeko Kontrol-Unitateak emandako ordena batek agindu behar bait du.

Barruko funtzionamendua aztertzen laguntzeko, 3.20 irudian unitateen egitura eta haien arteko loturak eta komunikazioa ikus daitezke.

Agindu bat konputagailuan egikaritzen denean bi fase berizten dira argi eta garbi, bilaketa-fasea eta exekuzio-fasea. Lehenengoa egikaritu behar den agindua eskuratu eta aginduaren erregistroan kokatzean datza, bigarrena berriz Kontrol-Unitateak agindua eskuratuz gero agindu hori burutzean.

3.7.1.- BILAKETA-FASEA

Fase honetan PC-k adierazten duen helbidean dagoen agindua irakurri eta aginduaren erregistroan kokatu behar da. Fase hau finkoa da agindu guztietarako konputagailu-mota guztietarako.

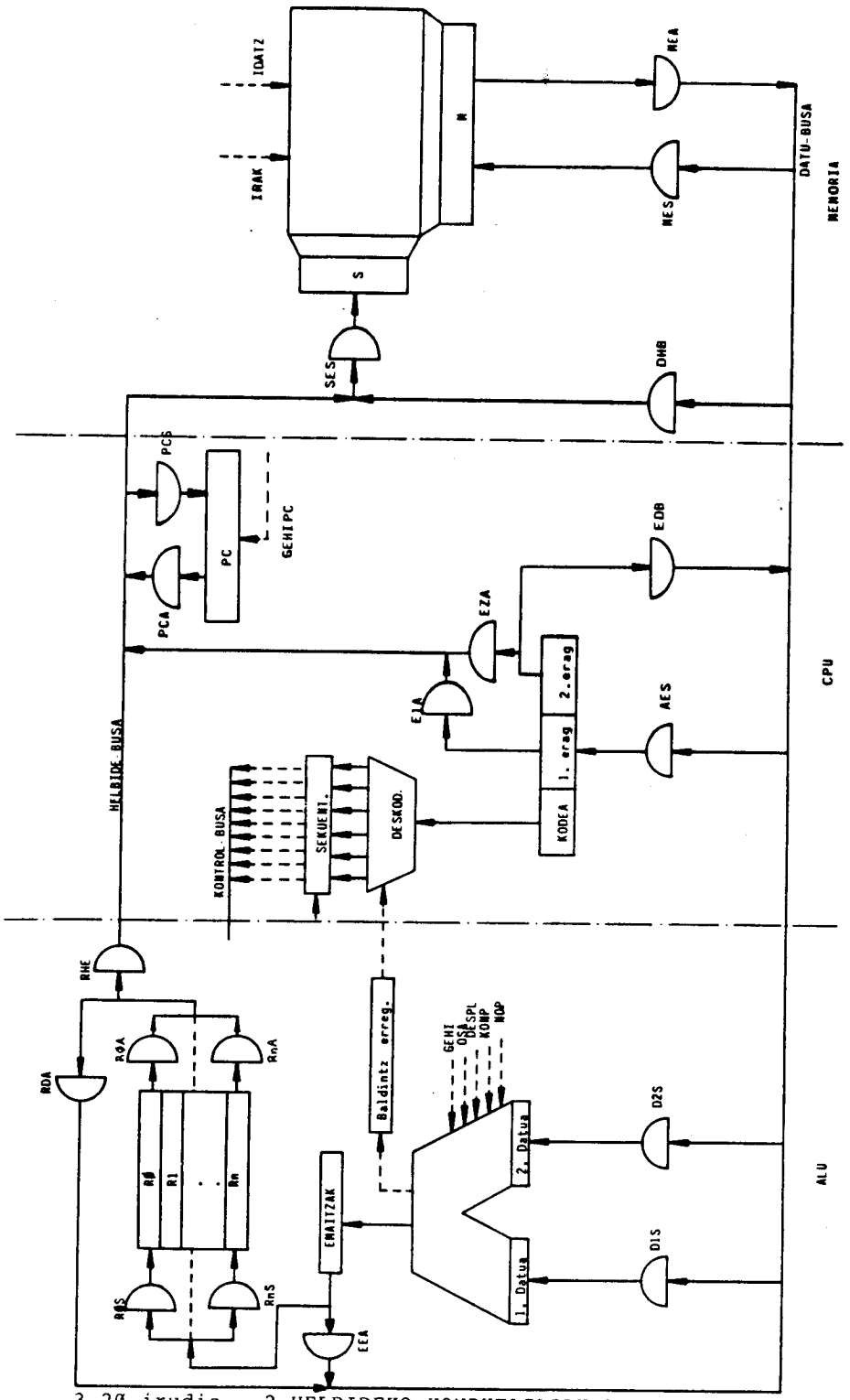
3.5.2. azpigalderan aipatu eta 3.20 irudian agertutako mikroaginduen izenak erabiliz, fase hau osatzen duten pausoak eta dagozkien mikroaginduak honako hauek dira:

a) PC erregistroan dagoen helbidea helbide-busera atera (PCA).
b) Helbide-busean dagoen helbidea S erregistroan sartu (SES).
c) Memoriatik irakurri, agindua M erregistroan gelditukо delarik (IRAK).
d) M erregistroko agindua datu-busera atera (MEA).
e) Datu-busean dagoen agindua aginduaren erregistroan sartu (AES).
f) PC erregistroan dagoen helbideari aginduaren luzera gehitu, hurrengo aginduaren helbidea lortzeko (GEHIPC).

Aginduaren erregistroan agindua kokatu ondoren, exekuzio-fasea hasteko prest dago konputagailua.

3.7.2.- EXEKUZIO-FASEA

Kontrol-Unitateak agindua burutzeko sortutako mikoraginduek osatzen dute fase hau. Aginduaren eta konputagailuaren motaren arabera fase hau desberdina da. Hurrengo adibideetan bi helbideko



3.20 irudia.- 2 HELBIDEKO KONPUTAGAILU BATEN EGITURA

konputagailu bat suposatuko dugu; 3.17 irudian agertzen dena.

Dena dela, fase honetan hiru azpifase bereiz daitezke: eragigaien bilaketa, eragiketa burutzea eta emaitza uztea.

A) ERAGIGAIEN BILAKETAK.

Eragigai baten bilaketa, eragigaiari dagokion datua datuaren erregistroan kokatzean datza jeneralean. Azpifase honek datua den eragigai bakoitzeko bilaketa bat eskatzen du, orduan, konputagailu-motaren arabera, bilaketa-kopuru desberdinak gertatuko dira. Bi helbideko konputagiluetan, emaitza adierazten duen ergigaia bilatzen den azkena da.

Bilaketa bakoitza helbideratze-moduak baldintzatzen du 3.3.3 galderan aztertzen zen bezala.

Hona hemen helbideratze-modu bakoitzari dagozkion pausoak (beti 2.eragigaia suposatuko dugu):

1) Berehalakoa

a) Eragigaia datu busera atera (EDB).
b) Datuaren erregistroan sartu datua (D2S).

2) Absolutua

a) Eragigaia helbide-busera atera (E2A).
b) Helbide-busean dagoena S erregistroan sartu (D2S).
c) Memoriako irakurketa (IRAK).
d) M erregistrokoa datu-busera atera (MEA).
e) Datuaren erregistroan sartu datua (D2S).

3) Zeharkakoa

Lehenengo lauak absolutuan bezala gertatzen dira (E2A, SES, IRAK, MEA).
e) Lortutako helbide-busean jarri (DHB).
f) Helbidea S erregistroan sartu (SES).
g) Memoriako irakurketa (IRAK).
h) M erregistrokoa datu-busera atera (MEA).
i) Datuaren erregistroan sartu datua (D2S).

4) Erregistrozko zuzenak

a) n.erregistrotik datua atera (RnA).
b) Ateratako datua datu-busean jarri (RDA).
c) Datuaren erregistroan sartu datua (D2S).

5) Erregistrozko zeharkakoa

a) n.erregistroko edukina atera (RnA).
b) Ateratako helbidea helbide-busean jarri (RHE).
Hemendik aurrera helbideratze absolutuari dagozkion azken lauak dira (SES, IRAK, MEA, D2S).

6) Erregistrozko zeharkako 2.mailakoa

Erregistrozko zeharkako kasuko lehenengo bostak eta memoriako zeharkako kasuko azken bostak (RnA, RHE, SES, IRAK, MEA, DHB, SES, IRAK, MEA, D2S).

B) ERAGIKETA BURUTZEA.

Datuen erregistroetan dauden datuak erabiliz aginduari dagokion mikroagindu eragigarria exekutatzean datza azpifase hau, emaitzaren erregistroan utziko delarik. Agindu aritmetiko eta logiko arruntetan mikroagindu bakar batek osatuko du azpifase hau. Jauzietan eta mugimendu motako agindu batzutan azpifase hau ez dago eta dagoen mugimenduetan NOP mikroaginduak osatzen du.

C) EMAITZA UZTEA.

Kalkulatutako emaitza aginduak adierazten duen erregistroan edo memoriako helbidean uztean datza azpifase hau. Metagailuzko konputagailuetan eragiketa burutzean, emaitza metagailuean zuzenean gelditzen denez, ez dago azpifase hau.

Hiru eta bi helbideko konputagiluetan emaitza erregistro orokor batean utzi behar denean, emaitza erregistrotik atera eta erregistro batean sartzeak osatzen du azpifase hau, beraz RnS mikroaginduarekin nahikoa izango da. Emaitza memorian utzi behar denean bi helbideko konputagiluetan, eragigaien bilaketa egindakoan emaitzaren helbidea S erregistroan kokatzen da eta azpifase hau Emaitz erregistrotik datu-busera ateratzeko aginduak (EEA-k), datu-busean dagoena M erregistroan sartzerkoak (MES-ak) eta idaztekoak (IDATZ aginduak) osatzen dute. Hiru helbidekoetan, berriz, mikroagindu hauek bidali baino lehen S erregistroan emaitzaren helbidea kokatzeko mikroaginduak beharrezkoak dira, zeintzuk helbideratze-moduaren arabera desberdinak bait dira.

3.7.3.- ADIBIDE BATZU

1) "GEHI absolutua,zeharkakoa" aginduaren bilaketa eta exekuzio-faseak osatzen dituzten mikroaginduak izendatu.

a) Bilaketa-fasea.

PCA, SES, IRAK, MEA, AES, GEHIPC.

b) Exekuzio-fasea.

b.1.- Eragigaien bilaketa.
E2A, SES, IRAK, MEA, DHB, SES, IRAK, MEA, D2S (2.datua)
E1A, SES, IRAK, MEA, D1S (1.datua)

b.2.- Eragiketa burutzea.
GEHI

b.3.- Emaitza uztea.
EEA, MES, IDATZ

2) "GEHI n.erregistrozko zuzena,m.erregistrozko zeharkakoa" agindua osatzen duten mikroaginduak izendatu.

a) Bilaketa-fasea.

PCA, SES, IRAK, MEA, AES, GEHIPC

b) Exekuzio-fasea.

b.1.- Eragigaien bilaketa.
RmA, RHE, SES, IRAK, MEA, D2S
RnA, RDA, D1S

b.2.- Eragiketa burutzea.
GEHI

b.3.- Emaitza uztea.
RnS

3) "MUGI n.erregistrozko zuzena, absolutua" agindua osatzen duten mikroaginduak izendatu.

a) PCA, SES, IRAK, MEA, AES, GEHIPC

b) E2A, SES, IRAK, MEA, D2A
NOP (2.datuaren erregistroan dagoen emaitzaren erregistrora)
RnS

4) "JAUZI (baldintza gabe) absolutua" agindua osatzen duten mikroaginduak izendatu.

a) PCA, SES, IRAK, MEA, AES, GEHIPC

b) Exekuzio-fasea kasu honetan berezia da, zeren eta agindu honen bidez programaren sekuentzia aldatzen bait da, hau da, agindu honen atzetik ez da PC-k adierazten duen agindua egikarituko, eragaigaiak adierazitakoa baizik. Beraz, 1. eragigaiaren bidez hurrengo aginduaren helbidea zehazten da eta helbide hori PC-n sartzea da exekuzio-fasearen helburua.

ElA (aginduaren helbidea helbide-busean kokatu)
PCS (PC erregistroan sartu helbide-busean dagoena)

3.8.- ARIKETA EBATZIAK

1.ARIKETA EBATZIA.

a) Konputagailu-mota desberdinetarako A=B*C+(D-E)**2 formula kalkulatzeko programa egin lengoaia sinbolikoa erabiliz.

b) Aginduetan eragiketa-kodearen luzera 8 bitekoa baldin bada eta eragigai bakoitzarena 16 bitekoa, seinalatu memoriako zati txikiena behar duen programa.

c) Memoriako irakurketa bakoitzak Ø.1 mksg irauten badu, idazketa Ø.2 mksg eta eragiketa burutzeak Ø.1 mksg (mugimenduak ezik) egikaritzeko denbora hobetzen duen programa seinalàtu helbideratze-modu guztiak absolutuak direla, zehaztu ez diren denborak Ø direla eta irakurritako zer edo zer idazteko Ø.13 mksg bakarrik behar direla kontutan hartuz.

EBAZPENA:

a) Programak.

- Hiru helbideko konputagailuan:

```
BIDER B,C,A      ; a=b*c
KEN   D,E,Z      ; z=d-e
BIDER Z,Z,Z      ; z=(d-e)**2
GEHI  A,Z,A      ; a=b*c+(d-e)**2
```

- Bi helbideko konputagailuan:

```
MUGI   A,B       ; a=b
BIDER  A,C       ; a=b*c
MUGI   Z,D       ; z=d
KEN    Z,E       ; z=d-e
BIDER  Z,Z       ; z=(d-e)**2
GEHI   A,Z       ; a=b*c+(d-e)**2
```

- Metagailuzko konputagailuan:

```
KARGA    B       ; met=b
BIDER    C       ; met=b*c
GORDE    A       ; a=b*c
KARGA    D       ; met=d
KEN      E       ; met=d-e
GORDE    Z       ; z=d-e
BIDER    Z       ; met=(d-e)**2
GEHI     A       ; met=b*c+(d-e)**2
GORDE    A       ; a=b*c+(d-e)**2
```

- Pilazko konputagailuan:

Idazkera poloniar alderantzizkoa erabiliz formula zera gelditukо litzateke: BC*DE-DE-*+

```
PUSH    B
PUSH    C
BIDER
PUSH    D
PUSH    E
KEN
PUSH    D
PUSH    E
KEN
BIDER
GEHI
POP     A
```

b) Programaren luzera.

- Hiru helbideko konputagailuan:

luzera(agindua) = luzera(kodea) + 3 * luzera(eragigaia) =
= 8 + 3 * 16= 56 bit
luzera(programa) = agindu - kopurua * luzera(agindua) =
= 4 * 56 = 224 bit

- Bi helbideko konputagailuan:

luzera(agindua) = 8 + 2 * 16 = 40
luzera(programa) = 6 * 40 = 240 bit

- Metagailuzko konputagailuan:

luzera(agindua) = 8 + 1 * 16 = 24
luzera(programa) = 9 * 24 = 216 bit

- Pilazko konputagailuan:

luzera(agindu-1) = 8 + 0 * 16 = 8
luzera(agindu-2) = 8 + 1 * 16 = 24

PUSH eta POP aginduak agindu-2 motakoak dira eragigai bat dagokielako, beste guztiak agindu-1 motakoak direlarik.

luzera(programa) = 7 * 24 + 5 * 8 = 208 bit

Beraz pilazko konputagailuari dagokion programa laburrena da kasu honetan, baina ezin da jeneralean baieztatu, programaren arabera desberdina da eta.

c) Programen exekuzio-denbora.

- Hiru helbideko konputagailuan:

```
denbora(agindua) = irakurketa - kopur*denb(agindua) +
                 + denb(eragiketa) + denb(idazketa) =
                 = 2 * 0.1 + 0.1 + 0.2 = 0.5 mksg
denbora(programa) = agindu-kop * denb(agindua) =
                  = 4 * 0.5 = 2 mksg
```

- Bi helbideko konputagailuan:

```
denbora(agindu-1) = 2 * 0.1 + 0.1 + 0.13 = 0.43
denbora(MUGI) = 0.1 + 0.2 = 0.3
```

Agindu arruntetan bi irakurketa, eragiketa eta aurretik irakurritako helbide batean idazketa bat gertatzen dira. Mugimendutan berriz, irakurketa bakarra dago, baina idazketa helbide berri batean gertatzen da.

```
denbora(programa) = 4 * 0.43 + 2 * 0.3 = 2.32 mksg
```

- Metagailuzko konputagailuan:

```
denbora(agindu-1) = denb(irakurketa) + denb(eragiketa) =
                  0.1 + 0.1 = 0.2
denbora(KARGA) = denb(irakurketa) = 0.1
denbora(GORDE) = denb(idazketa) = 0.2
denbora(programa) = 4*0.2 + 2*0.1 + 3*0.2 = 1.6 mksg
```

- Pilazko konputagailuan:

```
denbora(agindu-1) = denb(eragiketa) = 0.1
denbora(PUSH) = denb(irakurketa) = 0.1
denbora(POP) = denb(idazketa) = 0.2
denbora(programa) = 5*0.1+ 6*0.1 + 1*0.2 = 1.3 mksg
```

Beraz, kasu honetan pilazko konputagailuari dagokion programa arinena da.

2. ARIKETA EBATZIA.

Karakterekako konputagailu batean, non hitz bat bi byte osatzen bait dute eta datuen eta aginduen luzera hitz batekoa bait da ondoko programa dugu 0.helbidean hasita:

```
MUGI R3,14
MUGI R1,18
GEHI 26,(R1)
GEHI R1,16
KEN  R3,.1
JEZ  4      ;aurreko emaitza 0 ez bada jauzi (JEZ)
STOP
3
2
20
100
200
46
0
```

Rn erregistrozko helbideratze zuzenekoa, (Rn) erregistrozko zeharkakoa, n absolutua eta .n berehalakoa baldin badira, programa egikaritu ahala memoriako eta erregistroetako edukinen aldaketak aztertu (emaitza 1.eragigaian gelditzen dela kontutan hartuz).

EBAZPENA

Lehenbiziz aginduen eta datuen helbideak kalkulatuko ditugu. Karakterekako konputagailua denez, byte bakoitzari helbide bat dagokio eta aginduen eta datuen luzera 2 bytekoa izanik, lerro bakoitzak 2 helbide behar ditu. Beraz 0. helbidetik hasten denez aginduak 0., 2., 4., 6., 8., 10. eta 12. helbideetan kokatzen dira eta datuak 12.etik 26.era (ikus 3.5 taulan A zutabea).

Has gaitezen aginduz agindu:

a) (0.helb) MUGI R3,14

14.helbidean dagoen datua R3 erregistroan kokatuko da.
helb (datua) = 14
datua = 3 ====> eduk(R3) = 3

b) (2.helb) MUGI R1,18

helb (datua) = 18
datua = 20 ====> eduk(R1) = 20

c) (4.helb) GEHI 26,(R1)

helb(2.datua) = eduk(R1) = 20
2.datua = eduk(20) = 100
helb(1.datua) = 26
1.datua = 0
emaitza = 0 + 100 = 100 ====> eduk(26) = 100

d) (6.helb) GEHI R1,16

2.datua = eduk(16) = 2
1.datua = eduk(R1) = 20
emaitza = 22 ====> eduk(R1) = 22

e) (8.helb) KEN R3,.1

2.datua = 1
1.datua = eduk(R3) = 3
emaitza = 2 ====> eduk(R3) = 2

f) (10.helb) JEZ 4

Aurreko emaitza zero izan ez bada, 4. helbidera jauzi. Aurreko emaitza 2 izan denez, 4.helbidera jauziko da eta c, d, e eta f kasuei dagozkien aginduak berriro egikaritzen dira (ikus momentu honetako edukinak 3.5 taulako B zutabean).

c') (4.helb) GEHI 26,(R1)

2.datua = eduk(eduk(R1)) = eduk(22) = 200
1.datua = eduk(26) = 100
emaitza = 100 + 200 = 300 ====> eduk(26) = 300

d') (6.helb) GEHI R1,16

2.datua = eduk(16) = 2
1.datua = eduk(R1) = 22
emaitza = 22 + 2 = 24 ====> eduk(R1) = 24

e') (8.helb) KEN R3,.1

1.datua = eduk(R3) = 2
emaitza = 2 - 1 = 1 ====> eduk(R3) = 1

f') (10.helb) JEZ 4

Aurreko emaitza zero izan ez denez, 4. helbidera jauzten da berriro (ikus oraingo edukinak 3.5 taulan C zutabean).

HELBIDEA	A (hasieran)	B (1.bigizta)		C (2.bigizta)	D (3.bigizta)
0 - 12	aginduak				
14	3				
16	2				
18	20				
20	100				
22	200				
24	46				
26	0	100		300	346
R1	---	20	22	24	26
R3	---	3	2	1	0

3.5 taula.- MEMORIAKO ETA ERREGISTROTAKO DATUAK

c") (4.helb) GEHI 26,(R1)

2.datua = eduk(eduk(R1)) = eduk(24) = 46
1.datua = eduk(26) = 300
emaitza = 300 + 46 = 346 ====> eduk(26) = 346

d") (6.helb) GEHI R1,16

2.datua = eduk(16) = 2
1.datua = eduk(R1) = 24
emaitza = 24 + 2 = 26 ====> eduk(R1) = 26

e") (8.helb) KEN R3,.1

1.datua = eduk(R3) = 1
emaitza = 1 - 1 = 0 ====> eduk(R3) = 0

f") (10.helb) JEZ 4

emaitza 0 izan denez, datorren agindura pasatzen da.

g) (12.helb) STOP

Programa bukatu egin da (ikus edukin finalak 3.20 irudian D zutabean).

4.- D A T U E N A D I E R A Z P I D E A

4.1.- SARRERA

4.2.- DATUEN SAILKAPENA

4.3.- ZENBAKIEN ADIERAZPIDEA

4.4.- DATU ALFANUMERIKOAN ADIERAZPIDEA

4.5.- DATU-EGITURAK

4.1.- SARRERA

Aurreko kapituluetan informazioa gordetzeko osagaiak, Memoria Nagusia eta Memoria Lagungarriak alegia, aztertu dira, baina informazioa biltzeko modua ez da azaldu. Horregatik, kapitulu honen helburua, konputagailuetan informazioa nola adierazten den azaltzean datza.

Von Neumann-ek zioenez, konputagailuek bi informazio-mota desberdin gorde ditzakete: aginduak eta datuak. Informazio-mota desberdin hauek, memorian batera badaude ere, bereiztu egiten dira. Aginduek konputagailuak momentu bakoitzean bete behar dituen eragiketak adierazten dituzte, datuak berriz, eragiketa horien eragigaiak dira. Kapitulu honetan ez da aginduen adierazpidea azalduko, beren adierazpide propioa izan arren, datuak nola adierazten edo errepresentatzen diren agertzea izango bait da helburua.

Datuak Kanpo-Memorian gorde badaitezke ere, eragiketetarako behar dituen datuak Memoria Nagusitik eskuratzen ditu konputagailuak. Beraz, eragiketa batean erabili behar diren datuak Kanpo-Memorian baldin badaude, lehenbizi Kanpo-Memoriatik Memoria Nagusira higitu beharko dute eta Memoria Nagusian ezarri ondoren eragiketa buru daiteke.

Batzutan, Memoria Nagusian eta euskarrietan datuen adierazpidea berbera da, baina zenbait euskarritan ezin dira adierazpide jakin batzu erabili datuak adierazteko. Horrela, sarrerako euskarriek eta irteerakoek adibidez Karaktere formatoa (aurreago azalduko dugu) bakarrik erabil dezakete.

Gogora ditzagun orain 2.kapituluan azaldutako zenbait kontzeptu.

BITA.- Bi egoera desberdinetan egon daitekeen Memoriako elementuari bit deritzo. Egoera bati 0 balioa egokitzen zaio eta besteari 1 balioa.

Bit batean bi informazio desberdin sar daitezkeela jakinda, zenbat informazio desberdin gorde daiteke 2 bitetan? Lau, hau da bi aldiz bi (00, 01, 10 eta 11 konbinazioak). Oro har n bitetan 2^n (2 elementuren n-nakako konbinazio errepikadunei dagokiena) informazio desberdin adieraz daitezke.

BYTEA.- Karaktere bat adierazteko behar diren bitek byte bat osatzen dute. Askotan byte batek zortzi bit dauzka eta horrexegatik zortzikote ere deitzen zaio.

Karaktere bat adierazteko ez dira zortzi bit behar. Erabiltzen diren karaktereak honako hauek dira:

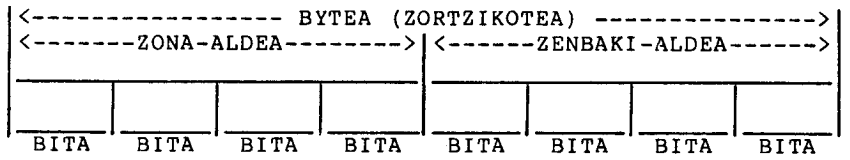
Letrak: A,B,C,D,...,Y,Z.
Zifrak: Ø,1,2,3,...,8,9.
Karaktere bereziak: ( ) . , - + * / [ ] @ : ...

Letrak 26 karaktere desberdin dira, zenbakiak 1Ø eta karaktere bereziak 29 baino gutxiago direnez gero, guztira ez dira 64 karaktere desberdinetara iristen. Lehen ikusi dugunaren arabera 6 bitekin 64 (2^6) informazio desberdin adieraz daitezkeela kontutan hartuta, edozein karaktere adierazteko 6 bitekin aski da.

Letren artean majuskulak eta minuskulak bereiztuko balira ere, 7 bitekin nahikoa izango litzateke (2^7=128).

Karaktere bat adierazteko behar-beharrezkoak direnak baino bit gehiago erabiltzen direnean erredundantzia dagoela esaten da. Erredundantziak erroreak detektatzeko eta konpontzeko balio du. Memoriako byteak zortzi bitez osaturik baldin badaude 256 (2^8) informazio desberdin sar daitezke byte batean, baina horien artean 64 bakarrik (2^6) izango dira egokiak, beraz, informazio desegoki bat agertzean errore bat gertatu dela detektatzen du makinak.

Memoria Nagusian eta Bitarteko Memorian byte batek 8 bit edukitzen ditu gehienetan. Zortzikote baten ezkerreko 4 bitek zona-aldea osatzen dute eta eskubiko beste 4ek zenbaki-aldea (ikus 4.1 irudia).



4.1 irudia.- BYTE BATEN OSAGAIAK

Konputagailuan informazio-mota desberdinak (zenbaki osoak, errealak, karaktere-kateak, ...) sar daitezkeenez gero, adierazpide desberdinak beharko ditugu eta horietako bakoitzari formatoa esango diogu. Formatoen arteko desberdintasuna ez dago ez ikurretan ez euskarrietan, informazioa ikur (bit) bihurtzeko moduan baizik.

4.2.- DATUEN SAILKAPENA

Konputagailuak erabiltzen dituen datuak ondoko bi mota hauetan sailka daitezke:

- ZENBAKIZKO DATUAK: Konputagailua gehienetan zenbakizko kalkuluak egiteko erabiltzen da gestio-lanetan nahiz zientzi-lanetan. Batuketek, kenketek, biderkaketek, zatiketek eta beste eragi-

keta matematikoek datu numerikoak erabiltzen dituzte eta, dudarik gabe, guzti horiek egiteko bidea izan behar du ordenadoreak.

Zenbakizko datuak batzutan zenbaki osoak dira, bestetan berriz, zenbaki errealak. Zenbaki osoek eta errealek adierazteko adierazpide desberdinak erabiltzen dituzte. Osoak adierazteko koma finkozko formatoak erabiltzen dira eta koma higikorrezko formatoa errealak adierazteko.

Bestalde, zenbakiak adierazteko guk 10 oinarria erabiltzen dugu, baina konputagailuak, aurrerago ikusiko dugunez, 2 oinarria erabiltzeko gaitasun handia du, bit bakoitzean zifra bitar bat sartzen bait da. Hau dela eta, bi formato desberdin daude zenbaki osoak errepresentatzeko; formato bitarra eta formato hamartarra, alegia.

- DATU ALFANUMERIKOAK: Eragiketa aritmetikoetan parte hartzen ez duten datuei alfanumerikoak deitzen zaie, letra, zifra eta karaktere berezien bidez osaturik egon daitezkeelarik.

  Datu-mota hau gestio-lanetan erabilktzen da batez ere, izenak, helbideak, telefonoak eta modu horretako beste informazio asko adierazteko. Adibidez hona hemen helbide bat: ARRASATE KALEA, 25-4.EZK. Letrak, zifrak eta karaktere bereziak erabiltzen direla ikus dezakegu.

  Datu alfanumerikoak karaktere-formatoan adierazten dira, memorian karaktere bakoitza byte batean ezartzen delarik. Sarrerako euskarrietan (txartel zulatua, teklatua, etab.)eta irteerako euskarrietan (paper inprimatua, pantaila, etab.) ere informazioa, numerikoa nahiz alfanumerikoa, karaktere-formatoan soilik erabil daiteke.

Datuen sailkapena ondoko hau dugu:

- DATUAK
  - ALFANUMERIKOAK --> Karaktere-formatoa
  - ZENBAKIZKOAK
    - OSOAK
      - BITARRA --> (Koma finkozko formato bitarra)
      - HAMARTARRA-> (form. hamartar paketatua edo form. hamartar zonaduna)
    - ERREALAK --> (Koma higikorrezko formato bitarra)

4.3.- ZENBAKIEN ADIERAZPIDEA

4.3.1.- OINARRI-ALDAKETA

Konputagailuari ondoen egokitzen zaion adierazpidea bitarra da, baina Sarrera-Unitatean zein Irteerakoan hamartarra erabili behar dugunez gero, hamartar-bitar bihurketa gertatuko da irakurtzean nahiz idaztean, hau da, irakurtzean 10 oinarritik 2 oinarrira bihurtuko dira datu numerikoak eta idaztean berriz 2 oinarritik 10era.

Bihurketa horren mekanismoa aztertu baino lehen, kasu orokorra aztertuko dugu, hau da, edozein oinarritatik 10 oinarrirako bihurketa eta alderantzizkoa.

n OINARRITIK 10 OINARRIRAKO BIHURKETA

Demagun $a_1\ a_2\ a_3 \ldots a_{m(n}$ zenbaki n-tar bat dela eta zenbaki hamartar bihurtu nahi dugula. Ondoko bide honi jarraitu behar gatzaizkio:

$$a_1\ a_2\ a_3 \ldots a_{m(n} = a_m + a_{m-1} \cdot n + \ldots + a_1 n^{m-1}$$

Metodo honi metodo polinomiala deitzen zaio.

1.Adibidea:

Bihurtu 8 oinarriko 23674 zenbakia 10 oinarrira.

Ebazpena:

$$23674_{|8} = 4 + 7.8 + 6.8^2 + 3.8^3 + 2.8^4 =$$
$$= 4 + 56 + 384 + 1536 + 8192 =$$
$$= 10.172_{|10}$$

10 OINARRITIK n OINARRIRAKO BIHURKETA

Demagun $a_1\ a_2\ a_3 \ldots a_{m(10}$ zenbaki hamartar bat dugula eta n-tar bihurtu nahi dugula. Ondoko bide honi jarraitu behar ga-

tzaizkio:

$$
\begin{array}{lllllll}
a_1\ a_2\ a_3 \ldots a_m & \underline{|n} & & & & & \\
\quad\quad\quad\quad r_1 & c_1 & \underline{|n} & & & & \\
\quad\quad\quad\quad\quad & r_2 & c_2 & \underline{|n} & & & \\
\quad\quad\quad\quad\quad & & \ldots & & & & \\
\quad\quad\quad\quad\quad & & & \ldots & & & \\
\quad\quad\quad\quad\quad & & & & c_{k-1} & \underline{|n} & \\
\quad\quad\quad\quad\quad & & & & r_k & 0 &
\end{array}
$$

Zatidura 0 izatera heltzen denean bukatzen da.

Orduan $a_1\ a_2\ a_3 \ldots m_{(10} = r_k\ r_{k-1} \ldots r_1{}_{(n}$ kontutan hartu hondarrak bukaeratik hasierarantz hartu behar direla.

Metodo honi zatiketaren metodoa deritzo.

2.Adibidea:

Bihurtu 10 oinarriko 10172 zenbakia 8 oinarrira.

Ebazpena:

$$
\begin{array}{lllllll}
10172 & \underline{|8} & & & & & \\
21 & 1271 & \underline{|8} & & & & \\
57 & 47 & 158 & \underline{|8} & & & \\
12 & 71 & 78 & 19 & \underline{|8} & & \\
4 & 7 & 6 & 3 & 2 & \underline{|8} & \\
 & & & & 2 & 0 &
\end{array}
$$

Orduan:

$$10172_{|10} = 23674_{|8}$$

Behar zuen bezala, bi ariketetan emaitza bera lortu dugula ikus dezakegu.

$$23674_{|8} = 10172_{|10}$$
$$10172_{|10} = 23674_{|8}$$

m OINARRITIK n OINARRIRAKO BIHURKETA

Zenbaki m-tar bat n-tar bihurtzeko ezin da zuzenean egin 10 oinarritik pasatu gabe: zenbaki m-tarra hamartar bihurtzen da metodo polinomialari jarraituz eta sortutako zenbaki hamartarra n-tar bihurtzen da zatiketaren metodoaren bidez.

$$z_{(m} \xrightarrow{\text{metodo polinomiala}} z_{(10} \xrightarrow{\text{zatiketaren metodoa}} z_{(n}$$

3.Adibidea:

Bihurtu 7 oinarriko 612 zenbakia 4 oinarrira.

Ebazpena:

a) $614_{\llcorner 7} = 4 + 1.7 + 6.7^2 = 4 + 7 + 294 = 305_{\llcorner 10}$

b)

$$\begin{array}{lllllll}
305 & \underline{|4} & & & & & \\
25 & 76 & \underline{|4} & & & & \\
\underset{\smile}{1} & 36 & 19 & \underline{|4} & & & \\
 & \underset{\smile}{0} & \underset{\smile}{3} & 4 & \underline{|4} & & \\
 & & & \underset{\smile}{0} & 1 & \underline{|4} & \\
 & & & & \underset{\smile}{1} & 0 &
\end{array}$$

$614_{\llcorner 7} = 305_{\llcorner 10} = 10301_{\llcorner 4}$

BITAR/HAMARTAR BIHURKETA

Bihurketa hau lehen ikusi ditugun bihurketen kasu partikular bat besterik ez da. Horregatik lehen metodo polinomiala eta zatiketaren metodoa erabiltzean n jarri beharrean orain 2 jarriko dugu.

4.Adibidea:

Bihurtu 2 oinarriko 1001001110 zenbakia 10 oinarrira eta 211 zenbaki hamartarra 2 oinarrira.

Ebazpena:

a) $1001001110_{\llcorner 2} = 0+1.2+1.2^2+1.2^3+0.2^4+0.2^5+1.2^6+0.2^7+$

$+ 0.2^8+1.2^9 = 2+4+8+64+512 = 590_{\llcorner 10}$

b) $211 \;|\; 2$ $\qquad 1001001110_{12} = 590_{10}$

$011 \quad 105 \;|\; 2$

$1 \quad 05 \quad 52 \;|\; 2$

$1 \quad 12 \quad 26 \;|\; 2$

$0 \quad 06 \quad 13 \;|\; 2$

$0 \quad 1 \quad 6 \;|\; 2$

$0 \quad 3 \;|\; 2$

$1 \quad 1 \;|\; 2$

$1 \quad 0$

$$211_{10} = 11010011_{12}$$

Konturatu zenbaki bitarren zifrak 0 eta 1 bakarrik izan daitezkeela. Jeneralean, n oinarrian zifrak 0 eta (n-1)-en artean daude. Adibidez 8 oinarrian zifrak 0 eta 7aren artean egon behar dute, 10 oinarrian 0 eta 9aren artean dauden bezala.

BITAR/ZORTZITAR ETA BITAR/HAMASEITAR BIHURKETA

Zenbaki bitarrak bitez bit adierazten ditugunean oso luzeak dira, horregatik Memoriako bitak paperean azaltzean, zenbaki bitarrak bitez bit idatzi beharrean, bitak bildu egiten dira eta zenbaki zortzitarrak edo hamaseitarrak idazten dira, hurrengo lerroetan ikusiko dugunaren arabera. Eta zergatik zortzitarrak edo hamaseitarrak eta ez hamartarrak edo beste oinarritakoak?

Hiru zenbaki bitar bilduta zenbaki zortzitar bat sortzen delako eta lau zenbaki bitar bilduta zenbaki hamaseitar bat. Hots, 2 oinarriko zenbaki bat 8 edo 16 oinarrira bihurtzeko metodo zuzen bat bait dago. Metodo zuzen hau aplikatzeko ondoko bi pauso hauek eman behar dira:

1.- Hasierako zenbakiaren zifra bitarrak eskuinetik ezkerretara hartu eta hirunaka (8 oinarrira bihurtzeko) edo launaka (16 oinarrira bihurtzeko) bildu taldetxoak osatuz. Ezkerretako taldetxoa zifraren baten faltan baldin badago, behar diren zeroak gehituko dira ezkerreko azken taldetxo hori osatu arte.

2.- Taldetxo bakoitza (hirukotea edo laukotea) zenbaki zortzitar edo hamaseitar bihurtu, azaldutako metodoei jarraituz edo 4.1 taulan erakusten diren baliokidetasunak erabiliz.

Konturatu 16 oinarria erabiliz 16 zifra daudela eta 10 zifra hamartar daudenez gero, beste 6 zifra berri asmatu behar dira. Horrexegatik lehenengo 6 letrak erabiltzen dira zifra moduan.

BITAR/ZORTZITAR	BITAR/HAMASEITAR
000 - 0	0000 - 0
000 - 1	0001 - 1
010 - 2	0010 - 2
011 - 3	0011 - 3
100 - 4	0100 - 4
101 - 5	0101 - 5
110 - 6	0110 - 6
111 - 7	0111 - 7
	1000 - 8
	1001 - 9
	1010 - A
	1011 - B
	1100 - C
	1101 - D
	1110 - E
	1111 - F

4.1 taula.- BITAR/ZORTZITAR ETA BITAR/HAMASEITAR ITZULPENAK

5.Adibidea:

Bihurtu 101001100 zenbaki bitarra 8 oinarrira eta 16 oinarrira.

Ebazpena:

a) $101001100_{|2} \longrightarrow$ *8 oinarrira*

1. pausoa $|101||001||100|$

2. pausoa $101 \quad 001 \quad 100$

$\downarrow \quad \downarrow \quad \downarrow$

$5 \quad 1 \quad 4$

$$101001100_{|2} = 514_{|8}$$

Lehen ikusi dugun zeharkako metodoa jarraituz:

$$101001100_{|2} = 0+0.2+1.2^2+1.2^3+0.2^4+0.2^5+1.2^6+0.2^7+1.2^8 =$$
$$= 332_{|10}$$

$$\begin{array}{l|l}
332 & 8 \\ \hline
12 & 41 \quad | \; 8 \\ \hline
4 & 1 \quad 5 \quad | \; 8 \\ \hline
 & \quad 5 \quad 0
\end{array}$$

$$101001100_{|2} = 332_{|10} = 514_{|8}$$

Metodo zuzena eta zeharkakoa erabiliz emaitza bera lortzen da baina metodo zuzena askoz motzagoa da, zenbakiak luzeak direnean batez ere.

b) $101001100_{|2} \longrightarrow$ *16 oinarrira*

1. pausoa |0001| |0100| |1100|

2. pausoa 0001 0100 1100

↓ ↓ ↓

1 4 C

$$101001100_{|2} = 14C_{|16}$$

ZENBAKI ERREALEN BIHURKETA n OINARRITIK 10 OINARRIRA

Orain arte osoen bihurketa besterik ez dugu aztertu, baina zer gertatzen da zenbaki errealekin?

Zenbaki erreal guztiak alde oso batek eta alde frakzionario batek osatzen dituzte. Zenbakia bihurtzeko, alde bakoitza (osoa eta frakzionarioa) bere aldetik bihurtzen da eta bi bihurketak bukatutakoan bi emaitzak lotzen dira. Beraz, osoak bihurtzen dakigunez gero/ alde frakzionarioa bihurtzeko metodoa ezagutu behar dugu.

Demagun $0.a_1a_2...a_{m(n}$ zenbaki n-tar bat dela eta hamartar bihurtunahi dugula. Osoei dagokien bide berbera segituko dugu, baina berretzaileak negatiboak direla kontutan hartuz.

$$0.a_1a_2...a_{m(n} = a_1n^{-1} + a_2n^{-2} + ... + a_mn^{-m}$$

6. Adibidea:

Bihurtu 5 oinarriko 0,342 zenbakia 10 oinarrira.

Ebazpena:

$$0{,}342_{\lfloor 5} = 3.5^{-1} + 4.5^{-2} + 2.5^{-3} =$$

$$= 3/5 + 4/25 + 2/125 =$$

$$= 0{,}6 + 0{,}16 + 0{,}016 = 0{,}776_{\lfloor 10}$$

ZENBAKI ERREALEN BIHURKETA 1Ø OINARRITIK n OINARRIRA

Lehen bezala alde osoa eta frakzionarioa bereizten dira eta bakoitza bere aldetik bihurtzen da. Osoak nola bihurtzen diren ikusita dagoenez, ondoren alde frakzionarioa bihurtzeko metodoa adieraziko dugu.

1Ø oinarritik n oinarrira bihurtzeko osoetan zatiketen bidez egiten zen, zenbaki frakzionarioak bihurtzeko berriz biderkaketen bidez egingo da.

Demagun $Ø.a_1a_2...a_{m(10}$ zenbakia n oinarriko bihurtu nahi dugula. Ondoko bide hau segitu behar dugu:

$$\begin{array}{l} Ø.\ a_1\ a_2\ \dots\ a_m \\ \hline x_1.\ a_1'\ a_2'\ \dots\ a_m' \\ \hline x_2.\ a_1''\ a_2''\ \dots\ a_m'' \\ \quad\ \dots \\ \quad\ \dots \end{array} \quad n$$

Orduan $Ø.a_1a_2...a_{m(10} = Ø.x_1x_2...(_n$ biderkaketak egiten direnean alde osoa kontutan hartzen ez dela ohartuz. Zifren kopurua infinitua izan daiteke, baina ondoko arau hau izan behar da gogoan zifra-kopuruaz jabetzeko: Zenbat eta oinarri txikiagoa hainbat eta zifra gehiago kalkulatu behar doitasuna mantentzeko eta alderantziz, oinarri handiagora bihurtzean zifra gutxiagokin doitasuna mantendu egiten da.

7.Adibidea:

Bihurtu 10 oinarriko 0,853 zenbakia 6 oinarrira.

Ebazpena:

$$
\begin{array}{r|l}
 & 0{,}853 \\
 & \quad 6 \\ \hline
5 & 118 \\
 & \quad 6 \\ \hline
0 & 708 \\
 & \quad 6 \\ \hline
4 & 248 \\
 & \quad 6 \\ \hline
1 & 488
\end{array}
$$

$$0{,}853_{\lfloor 10} = 0{,}5041\ldots_{\lfloor 6}$$

ZENBAKI ERREALEN BIHURKETA m OINARRITIK n OINARRIRA

Zenbaki m-tarraren alde osoa eta frakzionariota bereziz gero, alde bakoitza 10 oinarritik pasatuz n-tar bihurtzen da aztertu ditugun metodoak erabiliko direlarik. Azkenean alde osoa eta frakzionarioa batu egiten dira.

8.Adibidea:

Bihurtu 20,3124 zenbaki 5-tarra 7 oinarria.

Ebazpena:

$$20{,}3124_{\lfloor 5} = 20_{\lfloor 5} + 0{,}3124_{\lfloor 5}$$

a) $20_{\lfloor 5} = 0 + 2.5 = 10_{\lfloor 10}$

b)
$$
\begin{array}{l|l|l}
10 & 7 & \\
\underset{\smile}{3} & 1 & 7 \\
 & \underset{\smile}{1} & 0
\end{array}
$$

$10_{\lfloor 10} = 13_{\lfloor 7}$

c) $0{,}3124_{\lfloor 5} = 3/5 + 1/5^2 + 2/5^3 + 4/5^4 = 0{,}6 + 0{,}04 +$

$+0{,}016 + 0{,}0064 = 0{,}6624_{\lfloor 10}$

d) $0,6624$

$$\begin{array}{r} 7 \\ \hline 4 \rfloor\ 6368 \\ 7 \\ \hline 4 \rfloor\ 4576 \\ 7 \\ \hline 3 \rfloor\ 2032 \\ 7 \\ \hline 1 \rfloor\ 4224 \end{array}$$

$$0,6624_{\lfloor 10} = 0,4431\ldots_{\lfloor 7}$$

$$20,3124_{\lfloor 5} = 10,6624_{\lfloor 10} = 13,4431\ldots_{\lfloor 7}$$

Bitar/zortzitar eta bitar/hamaseitar bihurketa zuzenak erabil daitezke alde frakzionarioan baina taldetxoak egiteko orduan ezkerretik eskuinera jo behar dela eta zeroak gehitzean eskuinean gehitzen direla kontutan hartuz.

9.Adibidea:

Bihurtu 101000111, 1011100011 zenbaki bitarra 16 oinarrira.

Ebazpena:

1. pauso $\lfloor 0001 \rfloor$ $\lfloor 0100 \rfloor$ $\lfloor 0111 \rfloor$, $\lfloor 1011 \rfloor$ $\lfloor 1000 \rfloor$ $\lfloor 1100 \rfloor$

2. pauso 1 4 7 , B 8 C

Beraz $101000111,1011100011_{\lfloor 2} = 147,B8C_{\lfloor 16}$

4.3.2.- OSOEN ADIERAZPIDEA

Datuen sailkapena ikustean zenbaki osoak hiru formato desberdinetan errepresenta daitezkeela ikusi dugu. Hiru formato hauek koma finkozko bitarra, hamartar zonaduna eta hamartar paketatua dira. Banan-bana ikusiko ditugu.

A) KOMA FINKOZKO FORMATO BITARRA

Unitate Aritmetiko-Logikoak formato honetan eskatzen ditu gehienetan eragiketak burutzeko datuak. Zenbaki oso bat adierazteko bit-kopuru txikiena behar duen formatoa hauxe da.

Zenbaki oso bakoitza zenbaki bitar bezala hitz batean sartzen da, koma finkozko formato bitarra erabiltzen denean. Orduan, formato hau erabiliz, zenbaki oso baten memoriako luzera finkoa da beti, luzera hau hitzarena delarik hain zuzen ere, konputagailua

16 edo 32 bitekoa denaren arabera. Kapitulu honetako adibideetan eta ariketetan formato bitarrean adierazitako zenbaki oso batek 16 biteko luzera duela suposatuko dugu beti, 32 bitekoa izatea posible bada ere.

Zenbaki hamartar bat koma finkozko formato bitarrean memorian nola ezartzen jakiteko, ondoko bi pauso hauek eman behar dira:

1.- Zenbakia 10 oinarritik 2 oinarrira bihurtu.

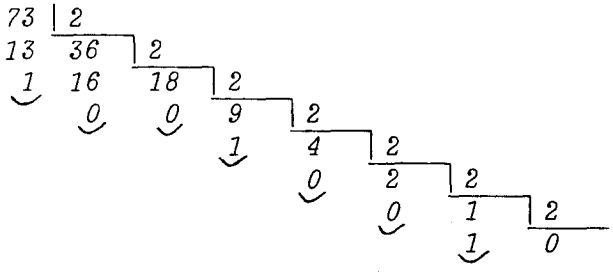
2.- Zenbaki bitarraren adierazpidearen luzerak eta hitzarenak berdinak izan behar dute, beraz, diferentziarik balego ezkerretatik zeroak gehituko dira hitzaren luzerara heldu arte (adibideetan 16 bitetara heldu arte).

10. Adibidea:

Nola geratuko litzateke Memorian 73 zenbaki hamartarra formato bitarrean sartuz.

Ebazpena:

1. pausoa: 73 hamartarra bitar bihurtu:



$$73_{|10} = 1001001$$

2. pausoa: 16 bitetan sartu

,0000,0000,0100,1001,

Baina orain arte koma finkozko formatoan, zenbaki positiboak bakarrik adierazi ditugu. Nola adierazten da zeinua formato hau erabiliz? Koma finkozko formato bitarrean ezker-ezkerreko bitak adierazten du zeinua, 0 denean plus zeinua adieraziz eta 1 denean berriz minus zeinua.

Hala eta guztiz ere, horrek ez du zenbaki positibo bat negatibo bihurtzeko ezker-ezkerreko bita zerotik batera aldatzea nahikoa denik esan nahi. Formato bitarra erabiliz, zenbaki negatibo baten adierazpidea ondoko bi pauso hauen bidez lortzen da:

- Zenbakia positibotzat hartu eta bere adierazpide bitarra lortu, lehen ikusitako metodoa erabiliz.

- Zenbaki positiboaren 2rako osagarria aurkitu ondoren azalduko den eran.

B) 2RAKO OSAGARRIA

Bit-multzo baten 2rako osagarria aurkitzeko ondoko bi pauso hauek eman behar dira:

- Bitez bit hartu eta bit bakoitza 0 denean 1 bihurtu eta 1 denean 0 bihurtu. Horrela, lerako osagarria lortzen da.

- Aurreko emaitzari bat zenbakia gehitu, kontutan hartuz 2 oinarrian gaudela eta oinarri honetan 1+1=10 dela ($10_2 = 2_{10}$).

11 Adibidea:

Aurkitu -66 zenbaki hamartarraren adierazpidea koma finkozko formato bitarra erabiliz.

Ebazpena:

1. pausoa: 66 formato bitarrera bihurtu

```
66 |2
06  33 |2
 0  13  16 |2
     1   0  8 |2
            0  4 |2
               0  2 |2
                  0  1 |2
                     1  0
```

$67_{|10} = 1000010_{|2}$

zeinua

$_{|}0|000_{|}0000|0100_{|}0010_{|}$ *(16 bitetan)*

2. pausoa: 2rako osagarria aurkitu:

a) 1erako osagarria kalkulatu:

0000 0000 0100 0010

↓

1111 1111 1011 1101

b) 1 gehitu

```
 1111 1111 1011 1101
                   + 1
 1111 1111 1011 1110
zeinua
        zeinua
-66 --> |1111 1111|1011 1110|
```

Zergatik erabiltzen du konputagailuak 2rako osagarria zenbaki negatiboak adierazteko? Ondoko bi arrazoi hauengatik:

- Zenbaki baten 2rako osagarriaren 2rako osagarria zenbaki berbera da (beraz -(-n) = n)

- Bi zenbakiren arteko kendura kalkulatzeko, kenkizunari kentzailearen 2rako osagarria gehitzea nahikoa da. Horrela Unitate Aritmetiko-Logikok ez dauka zeinu desberdineko zenbakiak berezi beharrik, zenbaki positiboak eta negatiboak era berean tratatuz, eta kenketa bat kalkutatu behar denean kendura kalkulatu beharrean osagarri bat eta batuketa bat burutuz.

Adibidez, 7-12 kalkulatzeko 7+(-12) burutuko du.

1. pausoa: 12ren 2rako osagarria kalkulatu.

```
                12 --> 0000 0000 0000 1100
            lerako
     osagarria --> 1111 1111 1111 0011
                                     + 1
               -12 --> 1111 1111 1111 0100
```

2. pausoa: 7 gehitu

```
                 7 --> 0000 0000 0000 0111
               -12 --> 1111 1111 1111 0100
          emaitza --> 1111 1111 1111 1011
```

Ezkerreko bitak zenbakia negatiboa dela adierazten digu. Emaitza hamartarra kalkulatzeko positibo bihurtzen dugu.

```
                    1111 1111 1111 1011
lerako
osagarria --> 0000 0000 0000 0100
                                 + 1
              ------------------------
              0000 0000 0000 0101 --> 5
```

Beraz emaitza -5 hamartarra ateratzen dela frogatzen da.

C) FORMATO HAMARTAR ZONADUNA

Formato hau erabiliz ez da oinarri-bihurketarik egin behar eta zifra hamartar bakoitza byte batean kokatzen da, kode baten arabera. Kode hau konputagailutik konputagailura desberdina da, baina guztiek ezaugarri amankomun bat daukate: byte guztien zona-aldeak (ezkerretako lau bitak) berdinak dira eta horregatik formatoari hamartar zonaduna deritzo.

Beraz, ez dago oinarri-bihurketarik zifraz-zifrako kodeketa baizik. Zeinua adierazteko eskuin-muturreko bytearen zona-aldea erabiltzen da, beraz zona-alde hori desberdina izango da.

Ikus dezagun 4 zifrako zenbaki oso baten adierazpide hamartar zonaduna:

zona-aldea	1. zifra	zona-aldea	2. zifra	zona-aldea	3. zifra	zeinua	4. zifra
<--1.BYTEA-->		<--2.BYTEA-->		<--3.BYTEA-->		<--4.BYTEA-->	

Zenbaki bat adierazteko formato hamartar zonaduna erabiliz, konputagailuaren arabera kode desberdinak daudela esan dugu lehen. Kode desberdin hauen artean ASCII eta EBCDIC kodeak dira garrantzitsuenak. EBCDIC kodea erabiliz, zenbaki hamartarrak 4.2 taulan azaltzen den bezala kodetuko dira.

12 Adibidea:

Aurkitu +409 eta -27 zenbaki hamartarren adierazpen hamartar zonaduna EBCDIC kodea erabiliz.

Ebazpena:

```
+409 --> |1111,0100|1111,0000|1100,1001|
          <-- 4 -->  <-- 0 -->  <-+->
                                  <-- 9 -->

-27  --> |1111,0010|1101,0111|
          <-- 2 -->  <-=->
                     <-- 7 -->
```

ZENBAKI HAMARTARRA	ADIERAZPIDEA ZONA-ALDEA	ZENBAKI-ALDEA
0	1111	0000
1	1111	0001
2	1111	0010
3	1111	0011
4	1111	0100
5	1111	0101
6	1111	0110
7	1111	0111
8	1111	1000
9	1111	1001
+ ZEINUA	1100 (eskuineko zona-aldean)	
- ZEINUA	1101 (ezkerreko zona-aldean)	

4.2 taula.- FORMATO HAMARTAR ZONADUNARI DAGOKION EBCDIC KODEA

Adierazpidearen luzera formato bitarrean finkoa zela ikusi genuen, formato hamartar zonadunean aldiz, luzera aldakorra da. Formato honetan, beraz, zifra-kopuruak luzera ematen digu, hau da 4 zifrako zenbaki batek 4 byte betetzen du eta 2 zifrakoak, berriz, 2 byte.

D) FORMATO HAMARTAR PAKETATUA

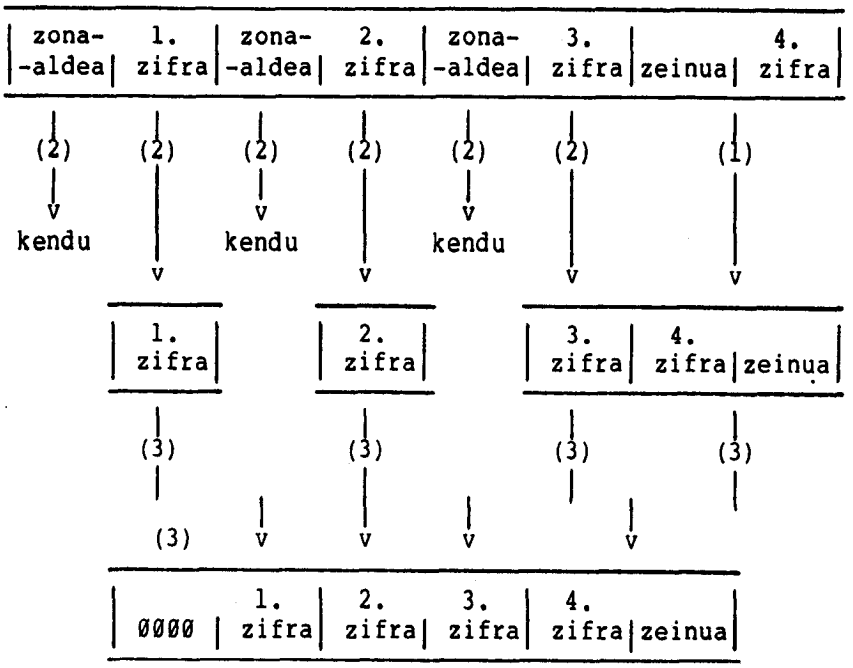
Formato honetan ere, hamartarra izateagatik, ez dago oinarri-bihurketarik eta zifra hamartar bakoitza 4 bitetan kokatzen da kode baten arabera.

Kodea formato hamartar zonadunarena bera da, baina zifra bakoitza kodetzean zona-aldea, finkoa denez gero, ez da kodetzen eta zenbaki-aldea besterik ez da kodetzen. Zeinua adierazteko ere kode bera erabiltzen da, baina eskuin-muturreko zenbaki-aldean ezartzen da, eskuineko 4 bitetan alegia.

Zenbakia kodetzean, zeinua kontaturik, byte-kopuru oso bat lortzen ez bada, ezkerretik zeroak gehituko dira ezkerreko bytea bete arte. Ikus dezagun, adibidez, 4 zifrako zenbaki oso baten adierazpena formato hamartar paketatuan:

0000	1. zifra	2. zifra	3. zifra	4. zifra	zeinua

Formato hamartar zonadunean adierazitako zenbaki bat formato hamartar paketatuko bihurtzeko ondoko hiru urrats hauek eman behar dira (ikus 4.2 irudia):



4.2 irudia.- ZONADUN/PAKETATU BIHURKETA

1.- eskuineko bytean zona-aldea eta zenbaki-aldea elkarrekin trukatu.

2.- Beste byte guztietan zona-aldea kendu.

3.- Bit guztiak bildu byte-kopuru oso bat osatuz.

13. Adibidea:

Aurkitu +409 eta -27 zenbaki hamartarren adierazpide hamartar paketatua EBCDIC kodea erabiliz.

Ebazpena:

+409 |0100,0000|1001,1100|
← 4 → ← 0 → ← 9 → ← + →
← BYTEA →

-27 |0000,0010|0111,1101|
← 2 → ← 7 → ← − →

Azter itzazu hemen eta 8. adibidean lortutako adierazpideen arteko diferentziak.

Formato honetan ere luzera aldakorra da. Zifren kopuruari bat gehituz eta batura hori birekin zatituz lortzen dena da luzera bytetan. Beraz:

$$\text{luzera}_{\text{bytetan}} = (\text{zifra-kopurua} + 1)/2$$

E) FORMATOEN ARTEKO KONPARAKETA

Unitate Aritmetiko-Logikoak formato bitarrean lan egiten duenez gero, zenbakiak Memoria Nagusian gordetzeko formato egokiena bitarra da. Informazioa formato bitarrean egonda ez da bihurketarik gertatuko eta eragiketak abiadura handiagoz egikarituko dira.

Konputagailuak eragiketa bat exekutatzeko erabili behar dituen datuak formato hamartarrean baldin badaude, lehenbizi bitar bihurtu beharko ditu datu horiek, gero eragiketa burutu eta azkenik emaitza hamartar bihurtu. Ondorioz, eragiketa egikaritzeko denbora luzatu egiten da.

Guk erabiltzen dugun zenbaki-adierazpidea hamartarra denez gero, sarrera eta irteerako euskarrietan zenbakiak gordetzeko formato egokiena, eta batzutan beharrezkoa, hamartarra da.

Bestalde, luzerari begira, euskarria hoberen probetxatzen duen formatoa bitarra da, hots zenbaki oso bat gordetzeko formato bitarrak hamartarrak baino luzera txikiagoa erabiltzen du. Formato hamartarren artean, formato paketatua zonaduna baino laburragoa da.

n bytetan zeintzu dira balio absolutuan sar daitezkeen zenbaki altuenak formatoaren arabera?

- Formato bitarrean: $2^{(\text{bit-kopuru} - 1)} - 1 = 2^{(8n-1)} - 1$.
  (Kontutan hartu 10 oinarrian 3 zifrakin adieraz daitekeen zenbakirik altuena 999 (10^3-1) dela, 4 zifrakin 9999 (10^4-1) eta, oro har, n zifrakin 10^n-1. Edozein oinarritara hedatuz oinarria $^{\text{zifra-kopurua})}-1$). Bitarrean n bytetan 8n bit daude baina bat zeinurako denez aurreko formula lortzen dugu.

- Formato hamartar zonadunean: 10^n-1.
  (formato honetan zifra-kopurua eta byte-kopurua zenbaki bera bait da)

- Formato hamartar paketatuan: $10^{(2n-1)}-1$.
  (formato honetan byte bakoitzean 2 zifra hamartar sartzen dira azken bytean ezik, azken bytean zifra bat eta zeinua sartzen direlarik. Beraz n bytetan zifra kopuruan 2n-1 da).

Beraz n bytetan zenbaki altuenak formatoaren arabera $2^{(8n-1)}-1$, $10^{n}-1$ eta $10^{(2n-1)}-1$ izango dira eta ondoko hau froga daiteke:

$$2^{(8n-1)}-1 > 10^{(2n-1)}-1 > 10^{n}-1$$

Honen ondorioz, lekua ondoen probetxatzen duen formatoa bitarra dela argi eta garbi ikusten da.

Adibidez, bi bytetan zeintzu izango dira zenbaki handienak formatoaren arabera?

- Formato bitarrean: $2^{(8*2-1)}-1 = 2^{15}-1 = 32767$.
- Formato hamartar zonadunean: $10^{2}-1 = 99$
- Formato hamartar paketatuan: $10^{(2*2-1)}-1 = 999$

F) DIGITU HAMASEITARREN BIDEZKO ADIERAZPIDEA

Zenbaki osoak memorian nola adierazten diren aztertu dugu, hau da, zenbaki oso bati dagokion bit-multzoa formatoaren arabera. Bit-multzo hori zehazteko bitak banan bana idatz daitezke (eta orain arte horrelaxe egin dugu), baina bitak launaka bildu eta lau biten taldetxo bakoitzak digitu hamaseitar batez adieraztea ere posible da. Beraz, memoriako edukina, bitez bit adieraztea edo digitu hamaseitarrez digitu hamaseitar adieraztea gauza bera da, digitu hamaseitar bat eta 4 bit guztiz baliokideak bait dira. Gainera memoriako edukina hamaseitarrez adieraziz irakurgarritasuna eta fidagarritasuna irabazten dugu, hau da, azkarrago ulertzen da eta hanka-sartze gutxiago gertatzen dira.

Ariketak egiteko orduan, zenbaki baten adierazpidea bitez bit ematea edo digitu hamaseitarren bidez ematea baliokidea da, azken hau azkarragoa izanik.

- Formato bitarrean.- Zenbaki positiboak adierazteko 2 oinarrira bihurtu beharrean 10 oinarritik 16 oinarrira bihur ditzakegu, luzera egokitzerakoan digitu hamaseitarren kopurua bit-kopuruaren laurdena izango delako. Zenbaki negatiboetan 2rako osagarria kalkulatzeko, digitu hamaseitarrak F hamaseitarrez osaturiko zenbaki bati kentzen zaio emaitzari bat gehituz, eragiketa hauek 16 oinarrian egin behar direlarik.

  Zeinua ezkerreko digituak adierazten du, 0-7 tartean dagoenean positiboa adieraziz eta 8-F tartean negatiboa.

14. Adibidea: *(*)*

Aurkitu -66 zenbakiaren digitu hamaseitarren bidezko adierazpidea formato bitarra erabiliz.

Ebazpena:

1. pauso

```
66   |16
02    4   |16
      4    0
```

$66_{10} = 42_{16} \rightarrow$ | 0,0 | 4,2 |

2. pauso

```
 FFFF
-0042
 FFBD  → 1erako osagarria
  + 1
 FFBE  → 2rako osagarri
```

-66 → | FF | BE | → | 1111,1111 | 1011,1110 |

* *11 adibidean agertutako enuntziatu bera da. Konparatu itzazu.*

- Formato hamartarretan.- Zonaduna erabiliz zona-aldea finkoa da eta F hamaseitar da, zenbaki-aldetan zifra bera kokatzen da digitu hamaseitar bezala. Zeinuak, C hamaseitarrak positiboa adierazten du eta D-k negatiboa.

  Hamartar paketatuan berdin gertatzen dazona-alde finkoak kenduta desagertzen bait dira.

15. Adibidea: *

Aurkitu +409 zenbakiaren digitu hamaseitarren bidezko adierazpidea formato hamartarrak erabiliz.

Ebazpena:

F. hamartar zonaduna +409 | F4 | F0 | C9 | →
(+) 4 0 9

→ | 1111,0100 | 1111,0000 | 1100,1001 |
4 0 9

F. hamartar paketatua +409 |40|9C| →

→ |0100,0000|1001,1100|
4 0 9 +

* *12. eta 13. adibideetan agertutako enuntziatu bera. Alderatu itzazu.*

4.3.3.- ERREALEN ADIERAZPIDEA

Zenbaki errealak memorian adierazteko formato bakarra koma higikorrezko formato bitarra da.

Zenbaki erreal bat memorian bi zenbakiren bidez errepresentatzen da: mantisa eta berretzailea. Mantisa (m) eta berretzailea (b) lortzeko ondoko formula hau erabili behar da:

$$Z_{erreala} = m * oinarria^{b} \quad non \quad 1/oinarria <= m < 1$$

$$Z_{erreala} \text{ ---> } (m,e)$$

Oinarria finkoa da, konputagailuari dagokio eta gehienetan 10a da (gure adibideetan 10 erabiliko dugu) 2 eta 16 erabiltzen badira ere.

MANTISA (adierazpide bitarra)	BERRETZAILEA (adierazpide bitarra)

Zenbaki positiboa denean mantisaren zeinua plus izango da, aldiz, negatiboa denean mantisari minus zeinua egokitzen zaio, beraz mantisaren zeinuak zenbakiaren zeinua adierazten du. Zenbakiari, balio absolutuz bat baino handiagoa denean berretzaile positiboa dagokio, aldiz 1 baino txikiagoa denean berretzaile zero edo negatiboa. Adibidez:

$$34{,}6785 = 0{,}346785 * 10^{2} \text{ ---> } (0{,}346785 \ , \ 2)$$
$$-34{,}6785 = -0{,}346785 * 10^{2} \text{ ---> } (-0{,}346785 \ , \ 2)$$
$$0{,}000035 = 0{,}35 * 10^{-4} \text{ -----> } (0{,}35 \ , \ -4)$$
$$-0{,}000035 = -0{,}35 * 10^{-4} \text{ -----> } (-0{,}35 \ , \ -4)$$

Koma higikorrezko formato bitarra erabiliz, zenbakiaren mantisa eta berretzailea bi zenbaki izango balira bezala gordetzen dira memorian, bakoitza formato bitarrean sartuz eta mantisa zenbaki frakzionarioa dela kontutan hartuz.

16. Adibidea:

0,00275 zenbaki harmartarrari dagokion mantisa eta berretzailearen adierazpidea asmatu (motza izateko suposa dezagun mantisa adierazteko 5 bit daudela eta berretzailerako 3 bit).

Ebazpena:

a) $0{,}00275 = 0{,}275 * 10^{-2} \rightarrow (0{,}275,-2)$

b) $0{,}275_{|10} \rightarrow |2$ *mantisa adierazteko 5 bitak 4 zifra bitar eta zeinuak osatuko dituzte*

```
0,275
    2
 0|.550
    2
 1| 100
    2
 0| 200
    2
 0| 400
```

$0{,}275_{|10} \rightarrow 0{,}0100\ldots_{|2}$

mantisa $\underbrace{0}_{\text{zeinua}}$ $\underbrace{0100}_{\text{mantisa absolutua}}$

c) $-2_{|10}$ $\quad 2 \;|\; 2$, $\;0 \;\; 1$ $\quad 2_{|10} = 010$ *berretzailea 3 bitek osatzen dutelako.*

2rako ogarria $010 \rightarrow 101 + 1 = 110$

berretzailea $\rightarrow \boxed{110}$

Zenbaki errealek sarri infinitu zifra hamartar dute eta denak errepresentatzea ezinezkoa da (gogoratu pi edo e zenbakiak). Zifra hamartarren kopuru maximo bat erabil daiteke eta kopuruaren arabera, doitasun-maila desberdina lortzen da. Zenbakia mantisa eta berretzailearen bidez adieraztean, mantisaren zifra-kopuruak zen-

bakiaren doitasuna ematen digu, beraz, zenbat eta zifra gehiago mantisarako hainbat eta doitasun handiago lortuko da. Konputagailuan formato bitarrean zifra-kopuruaren neurria bit-kopurua denez, mantisa adierazteko zenbat eta bit gehiago orduan eta doitasun handiagoa. Jeneralean koma higikorrezko formato bitarrean zenbaki errealak adierazteko bi aukera daude, batek besteak baino doitasun handiagoa eskainiz luzeagoa bait da, adiereazpide luzeenari doitasun bikoitzekoa esaten zaiolarik.

Koma higikorrezko formato bitarrean adierazten diren zenbakien magnitudea, berretzailea adierazteko bit-kopuruaren funtzioan dago, zenbat eta bit gehiago hainbat eta zenbaki handiagoak edo txikiagoak (berretzailearen zeinuaren arabera) adieraz daitezke.

17. Adibidea:

Berretzailea 8 bitek osatzen ba dute zein da koma higikorrezko zenbakiaren magnitudea.

Ebazpena:

$|zenb.| \leqslant mantisa\ handiena * 10^{berretz.handiena}$

$mantisa\ handiena < 1$

$berretzaile\ handiena = 2^{8-1} - 1 = 127$ *(Koma finkoz formato bitarrez adierazten bait da)*

$|zenb.| < 10^{127}$

$|zenb.| \geqslant mantisa\ txikiena * 10^{berretz.txikiena}$

$mantisa\ txikiena = 0,1$

$berretzaile\ txikiena = -2^{8-1} = -128$

$|zenb.| \geqslant 0,1 * 10^{-128} = 10^{-129}$

Beraz $10^{-129} \leqslant |zenbakia| < 10^{127}$

4.4.- DATU ALFANUMERIKOEN ADIERAZPIDEA

Karaktereak memorian adierazteko karaktere-formatoa erabiltzen da. Formato honen bidez karaktere bakoitza, kode bat jarraituz, memoriako byte batean kokatzen da (gogoratu bytearen definizioa). Hau da, karaktere bakoitzari zortzi biten konbinazio bat egokitzen zaio.

Egokitze hau aldakorra da kodearen arabera, izan ere konputagailuetan bi kode dira nagusi, ASCII kodea eta EBCDIC kodea, hauek 4.3 taulan ikus daitezkelarik.

18. Adibidea:

Aurkitu 'AMX2/'-ren Memoriako adierazpidea ASCII eta EBCDIC kodeak jarraituz.*

Ebazpena:

a) ASCII kodean:

A	M	X	2	*	/
10100001	10101101	10111000	01010010	01001010	01001111
A 1	A D	B 8	5 2	4 A	4 F

Launaka (ikus bitar-hamaseitar bihurketa)

b) EBCDIC kodean:

A	M	X	2	*	/
11000001	11010100	11100111	11110010	01011100	01100001
C 1	D 4	E 7	F 2	5 C	6 1

4.5.- DATU-EGITURAK

Aurreko pasartetan datu elemental edo oinarrizkoen adierazpena aztertu dugu, baina batzutan datu elementalak elkartu egiten dira egitura bat eratuz.

Datu-egitura erabilienak taulak, erregistroak eta fitxategiak ditugu. Azter ditzagun orain egitura hauek banan bana.

4.5.1.- TAULAK

Formato eta luzera finkoa duen datu-multzo batek taula bat osatzen du. Datu hauek gehienetan zenbakizkoak dira, alfanumerikoak izan badaitezke ere. Taula batean datu bat bilatzeko, datuari taulan dagokion lekua edo ordena zehaztu behar da.

Datu-egitura hau Memoria Nagusian erabiltzen direnen artean erabiliena da.

EBCDIC	Adierazpidea bitetan	ASCII-8
NUL	0000 0000	NULL
SOH	0000 0001	SOM
STX	0000 0010	EOA
ETX	0000 0011	EOM
PF	0000 0100	EOT
HT	0000 0101	WRU
LC	0000 0110	RU
DEL	0000 0111	BELL
	0000 1000	BKSP
	0000 1001	HT
SMM	0000 1010	LF
VT	0000 1011	VT
FF	0000 1100	FF
CR	0000 1101	CR
SO	0000 1110	SO
SI	0000 1111	SI
DLE	0001 0000	DC_0
DC1	0001 0001	DC_1
DC2	0001 0010	DC_2
TM	0001 0011	DC_3
RES	0001 0100	DC_4/STOP
NL	0001 0101	ERR
BS	0001 0110	SYNC
IL	0001 0111	LEM
CAN	0001 1000	S_0
EM	0001 1001	S_1
CC	0001 1010	S_2
CUI	0001 1011	S_3
IFS	0001 1100	S_4
IGS	0001 1101	S_5
IRS	0001 1110	S_6
IUS	0001 1111	S_7
DS	0010 0000	
SOS	0010 0001	
FS	0010 0010	
	0010 0011	
BYP	0010 0100	
LF	0010 0101	
ETB	0010 0110	
ESC	0010 0111	
	0010 1000	
	0010 1001	
SM	0010 1010	
CU2	0010 1011	

EBCDIC	Adierazpidea bitetan	ASCII-8
	0010 1100	
ENQ	0010 1101	
ACK	0010 1110	
BEL	0010 1111	
	0011 0000	
	0011 0001	
SYN	0011 0010	
	0011 0011	
PN	0011 0100	
RS	0011 0101	
UC	0011 0110	
EOT	0011 0111	
	0011 1000	
	0011 1001	
	0011 1010	
CU3	0011 1011	
DC4	0011 1100	
NAK	0011 1101	
	0011 1110	
SUB	0011 1111	
zurigunea	0100 0000	b (zurigunea)
	0100 0001	!
	0100 0010	"
	0100 0011	#
	0100 0100	$
	0100 0101	%
	0100 0110	&
	0100 0111	,
	0100 1000	(
	0100 1001	)
ç	0100 1010	*
.	0100 1011	+
<	0100 1100	,
(	0100 1101	-
+	0100 1110	.
|	0100 1111	/
&	0101 0000	0
	0101 0001	1
	0101 0010	2
	0101 0011	3
	0101 0100	4
	0101 0101	5
	0101 0110	6
	0101 0111	7

.../...

EBCDIC	Adierazpidea bitetan	ASCII-8
	0101 1000	8
	0101 1001	9
!	0101 1010	:
$	0101 1011	;
*	0101 1100	<
)	0101 1101	=
;	0101 1110	>
¬	0101 1111	?
-	0110 0000	
/	0110 0001	
	0110 0010	
	0110 0011	
	0110 0100	
	0110 0101	
	0110 0110	
	0110 0111	
	0110 1000	
	0110 1001	
	0110 1010	
,	0110 1011	
%	0110 1100	
-	0110 1101	
>	0110 1110	
?	0110 1111	
	0111 0000	
	0111 0001	
	0111 0010	
	0111 0011	
	0111 0100	
	0111 0101	
	0111 0110	
	0111 0111	
	0111 1000	
	0111 1001	
:	0111 1010	
#	0111 1011	
@	0111 1100	
,	0111 1101	
=	0111 1110	
"	0111 1111	
	1000 0000	
a	1000 0001	
b	1000 0010	
c	1000 0011	

EBCDIC	Adierazpidea bitetan	ASCII-8
d	1000 0100	
e	1000 0101	
f	1000 0110	
g	1000 0111	
h	1000 1000	
i	1000 1001	
	1000 1010	
	1000 1011	
	1000 1100	
	1000 1101	
	1000 1110	
	1000 1111	
	1001 0000	
j	1001 0001	
k	1001 0010	
l	1001 0011	
m	1001 0100	
n	1001 0101	
o	1001 0110	
p	1001 0111	
q	1001 1000	
r	1001 1001	
	1001 1010	
	1001 1011	
	1001 1100	
	1001 1101	
	1001 1110	
	1001 1111	
	1010 0000	@
	1010 0001	A
s	1010 0010	B
t	1010 0011	C
u	1010 0100	D
v	1010 0101	E
w	1010 0110	F
x	1010 0111	G
y	1010 1000	H
z	1010 1001	I
	1010 1010	J
	1010 1011	K
	1010 1100	L
	1010 1101	M
	1010 1110	N
	1010 1111	O

.../...

EBCDIC	Adierazpidea bitetan	ASCII-8
	1011 0000	P
	1011 0001	Q
	1011 0010	R
	1011 0011	S
	1011 0100	T
	1011 0101	U
	1011 0110	V
	1011 0111	W
	1011 1000	X
	1011 1001	Y
	1011 1010	Z
	1011 1011	[
	1011 1100	\
	1011 1101	]
	1011 1110	↑
	1011 1111	←
	1100 0000	
A	1100 0001	
B	1100 0010	
C	1100 00111	
D	1100 0100	
E	1100 0101	
F	1100 0110	
G	1100 0111	
H	1100 1000	
I	1100 1001	
	1100 1010	
	1100 1011	
	1100 1100	
	1100 1101	
	1100 1110	
	1100 1111	
	1101 0000	
J	1101 0001	
K	1101 0010	
L	1101 0011	
M	1101 0100	
N	1101 0101	
O	1101 0110	
P	1101 0111	
Q	1101 1000	
R	1101 1001	
	1101 1010	
	1101 1011	

EBCDIC	Adierazpidea bitetan	ASCII-8
	1101 1100	
	1101 1101	
	1101 1110	
	1101 1111	
	1110 0000	
	1110 0001	a
S	1110 0010	b
T	1110 0011	c
U	1110 0100	d
V	1110 0101	e
W	1110 0110	f
X	1110 0111	g
Y	1110 1000	h
Z	1110 1001	i
	1110 1010	j
	1110 1011	k
	1110 1100	l
	1110 1101	m
	1110 1110	n
	1110 1111	o
0	1111 0000	p
1	1111 0001	q
2	1111 0010	r
3	1111 0011	s
4	1111 0100	t
5	1111 0101	u
6	1111 0110	v
7	1111 0111	w
8	1111 1000	x
9	1111 1001	y
	1111 1010	z
	1111 1011	
	1111 1100	
	1111 1101	
	1111 1110	ESC
	1111 1111	DEL

4.3 taula.- ASCII ETA EBCDIC KODEAK

Taula bat maneiatzeko, makina-lengoaiaz, osagaien formatoa eta luzera, osagai-kopurua edo bukaerako kodea eta taularen hasierako helbidea jakin behar dira. I-garren osagaia aurkitzeko bere helbidea honako formula honen bidez kalkulatzen da:

helbidea(I) = helbidea(1) + (I-1)*luzera(osagaia)

Luzerak gelaskatan adierazita egon behar du, hau da, karakterekako konputagailuetan bytetan eta hizkakoetan hitzetan.

Adibidez, urteko hil guztien egun-kopuruekin taula bat osatzen dugu. Taularen osagai bakoitza zenbaki oso bat da eta 12 osagai egongo dira.

| 31 | 28 | 31 | 30 | 31 | 30 | 31 | 31 | 30 | 31 | 30 | 31 |

Tauletako osagaiak maneiatzeko indize erregistroaren bidezko helbideratze erlatiboa erabiltzen da eta horrelakorik ez dagoenean erregistro bidezko zeharkakoa.

Matrizeak taula bikoitz bezala sartzen dira Memoria Nagusian, hau da, errenkada bakoitza taula bat da eta matrizea errenkadaz osaturiko taula.

$$\left|\begin{array}{c|c|c|c} a_{11} & a_{12} & \cdots & a_{1n} \\ \multicolumn{4}{c}{\text{<--1. errenk-->}} \end{array}\right| \left|\begin{array}{c|c|c|c} a_{21} & a_{22} & \cdots & a_{2n} \\ \multicolumn{4}{c}{\text{<--2. errenk-->}} \end{array}\right| \begin{array}{c} \cdots \\ \cdots \\ \cdots \end{array} \left|\begin{array}{c|c|c|c} a_{m1} & a_{m2} & \cdots & a_{mn} \\ \multicolumn{4}{c}{\text{<--m. errenk-->}} \end{array}\right|$$

a_{ij} osagaiaren helbidea kalkulatzeko honako formula hau erabili behar da m*n dimentsioko matrize batean:

helbidea(I,J) = helbidea(1,1) + (I-1)*n*luzera * (J-1)*luzera

non luzera bait da osagai bakoitzaren luzera gelaskatan emanda.

Pilak eta iladak datu-egitura bereziak dira, baina taula gisako egitura dute, desberdinak taulan burutu daitezkeen eragiketak dira, tauletan eragiketetan mugarik ez dagoen bitartean piletan eta iladetan eragiketak mugatuta daude. Piletan PUSH eta POP dira onartzen diren eragiketa bakarrak eta iladetan antzekoak.

4.5.2.- ERREGISTROAK

Elementu bati dagozkion datuen multzoak erregistro bat osatzen du. Erregistroaren informazioek elkarren arteko erlazioa dute, baina ez dute derrigorrez ez formato, ez luzera berekoak izan behar.

Erregistro bateko datu bakoitzari eremua deitzen zaio.

Adibidez, ikasle bakoitzari dagokion datu-multzoak ikasle-erregistroa osatzen du. Erregistro honetan zein eremu desberdin izango genituzke? Ba, ikasle-zenbakia, izen-deiturak, helbidea, kurtsoa, sexua eta adina, adibidez. Konturatu, zenbakizko datuak daudela (adina eta kurtsoa) eta baita alfanumerikoak ere (helbidea, izen-deiturak) beraz, ez da gertatzen tauletan bezala, non osagai guztiek formato berbera bait zuten.

Sarrera/Irteerako unitateen eta Memoria Nagusiaren artean trukatzen den informazio-unitatea erregistroa da gehienetan.

4.5.3.- FITXATEGIAK

Erregistro homogenoen multzoak fitxategi bat osatzen du. Beraz, fitxategi baten erregistro guztiek egitura berbera dute eta erregistroz osaturiko taula bat bezala ikus daiteke.

Memoria Lagungarrietan informazioa fitxategitan gordetzen da, baina fitxategietako informazioa erregistroz erregistro irakurtzen edo grabatzen da.

Aurrean ikusi dugun adibidean, ikasle-erregistroen multzoak ikasleen fitxategia osatzen du. Erregistro bakoitzak ikasle baten informazioa du eta fitxategian ikasle guztien informazioa egongo da. Fitxategi honek, zenbat ikasle hainbat erregistro edukiko du, erregistro guztiek ikaslearen zenbakia, izen-deiturak, helbidea, kurtsoa, sexua eta adina adieraziko dutelarik.

Fitxategi bateko erregistro guztietan erregistroa identifikatzen edo bereizten duen eremu bat dago, eremu horri giltza deritzolarik.

Aurreko adibidean, ikasle-fitxategiaren adibidean alegia, ikasle-zenbakia izeneko eremua dugu giltza, ikasle-zenbakiak unibokoki identifikatzen bait du erregistroa, hau da, ez dira zenbaki bereko bi ikasle egongo. Beste adibide bat jarriz, liburutegi-fitxategia dugu eta bertan liburu bakoitzari erregistro bat dagokio, erregisto horretan liburuari buruzko informazio guztiak daudelarik. Adibidez, titulua, egilea, argitaletxea, orri-kopurua, argitalpen-urtea eta gaia izeneko eremuek osa dezakete liburu-erregistroa. Baina, liburu bakoitza bereizteko beste eremu bat egongo da, giltza, eta gehienetan zenbaki bat denez, liburu-zenbakia izango da.

Fitxategien artean mota desberdinak bereizten dira, honako hauek dira garrantzitsuenak:

- Fitxategi sekuentzialak: Erregistro bat eskuratzeko aurretik dauden guztiak atzitu behar direneko fitxategiak dira, hau da, fitxategi hauetan ez dago erregistro bat zuzenean eskuratzeko

modurik.

- Zuzeneko fitxategiak: Mota honetako fitxategietan erregistro bat zuzenean atzi daiteke eta aurrean dauden beste guztiak ez dira ikutu behar.

Erregistroak zuzenean eskuratzeko modu desberdinak daude, euskarri-helbidearen bidezko atzipena eta giltzaren bidezkoa arruntenak izanik.

Erregistro guztiak atzitu behar ditugunean ez dugu zuzeneko fitxategirik behar, baina fitxategi baten zenbait erregistroren informazioa eskuratu nahi dugunean, fitxategia sekuentziala baldin bada, denbora luzea beharko da prozesua ez-errentagarri bihurtuz.

5.- A G I N D U A K K O N P U T A G A I L U A N

5.1.- MAKINA-LENGOAIA/LENGOAIA SINBOLIKOAK

5.2.- ITZULTZAILEAK

5.3.- AGINDU-MOTAK

5.4.- AGINDUAK MAKINA-LENGOAIAZ

5.5.- AGINDUAK MIHIZTADURA-LENGOAIAZ

5.1.- MAKINA-LENGOAIA/LENGOAIA SINBOLIKOAK

Bigarren kapituluan ikusi genuenez konputagailu batean maila desberdinak bereiz daitezke, maila bakoitzari alegiazko konputagailu bat eta lengoaia bat dagokiolarik. Hirugarren kapituluan mikroprogramazioaren mailako lengoaia eta makina-lengoaiarekin daukan lotura aztertu genuen. Kapitulu honetan berriz makina-lengoaiak, mihiztadura-lengoaiak eta haien arteko lotura aztertuko ditugu.

Gogora eta zehatz ditzagun hirugarren kapituluan emandako definizio batzu:

MAKINA-LENGOAIA: CPUk zuzenean egikaritzen dituen aginduek osatzen dute makina lengoaia, agindu bakoitzak eragiketa aritmetiko edo logiko bat adierazten duelarik. Makina-lengoaiako aginduak memoriara biltzen direnez, bitarrean adierazi behar dira, hau da 0 eta 1 ikurren bidez. Programatzeko ez da batere atsegina lengoaia hau.

LENGOAIA SINBOLIKOA: Lengoaia hau osatzen duen agindu-multzoa ezin da zuzenean konputagailuan egikaritu, baina programatzaileek erabiltzen dituzte agindu hauek zeintzuk programa itzultzaile baten bidez makina-lengoaian bihurtzen bait dira egikaritzapena posible izan dadin.

MIHIZTADURA-LENGOAIA: Lengoaia sinbolikoa da zeinean beti agindu bakoitza itzuliz gero makina-lengoaiazko baliokidea agindu bakar bat bait da. Beraz makina-lengoaia eta mihiztadura-lengoaiaren artean oso erlazio estua dago, aginduak guztiz baliokideak bait dira haien adierazpidea desberdina izan arren.

Makina-lengoaiari dagokion konputagailua benetako konputagailua da eta horren ondorioz makina-lengoaiaren eta konputagailuaren artean oso lotura handia dago, beraz, konputagailu desberdinek makina-lengoaia diferenteak erabiliko dituzte. Mihiztadura-lengoaia eta makina-lengoaia balokideak direnez, mihiztadura-lengoaia konputagailuaren arabera desberdina izango da.

Goimailako lengoaiek berriz, hardwarearekin loturarik ez dute (edo oso txikia) eta konputagailu desberdinetako PASCAL-a, BASIC-a edota beste lengoaia jakin bat da. Horretarako itzultzaileak diferenteak izango dira makina-lengoaia desberdinetara itzuli behar dute eta.

Programazioa gehienetan goimailako lengoaiaz egiten da, kasu berezi batzutan mihiztadura-lengoaiaz egiten bada ere. Mihiztadura-lengoaia gure pentsamoldetik askoz urrutirago dago eta horregatik ikasteko zailagoa da, erabilpenak akats gehiago erakartzen du eta programak ulergaitzak gertatzen dira, hala ere memoria gutxiago hartzen duten programa arinagoak gertatzen dira mihiztadu-

EZAUGARRIAK	MIHIZTADURA-LENGOAIA	GOI-MAILAKO LENGOAIA
Makinaren pean	BAI	EZ (*)
Programatzeko zailtasuna	HANDIA	TXIKIA (*)
Ikasteko zailtasuna	HANDIA	TXIKIA (*)
Erroreak programatzean	ASKO	GUTXI (*)
Programen ulergarritasun	TXIKIA	HANDIA (*)
Exekuzio-abiadura	HANDIA (*)	TXIKIA
Memoriako beharra	TXIKIA (*)	HANDIA

(*)-k ezaugarria aldekoa dela adierazten du

5.1 taula.- MIHIZTADURA ETA GOI-MAILAKO LENGOAIEN EZAUGARRIAK

ra-lengoaiaz. Horregatik, programa bat memorian kokatzeko oso leku gutxi dagoenean edota exekuzio-denbora larria denean mihiztadura-lengoaiaz idazten da. Hala eta guztiz ere, mihiztadura-lengoaia bat menperatzeak konputagailuaren barne-funtzionamendua ezagutzen eskatzen digu.

Goimailako lengoaien eta mihiztadura lengoaien ezaugarriak alderatzen dira 5.1 taulan.

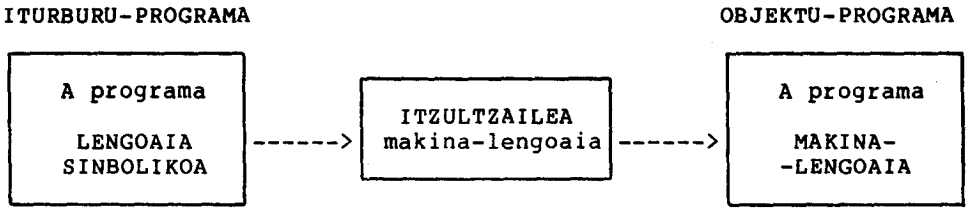
Makina-lengoaiak eta mihiztadura-lengoaiak konputagailu desberdinetan diferenteak dira eta desberdintasunak ondoko parametro hauen arabera finkatzen dira:

- eragigaien kopurua
- aginduen formatoa
- hitzaren eta gelaxkaren luzera
- erabil daitezkeen helbideratze-moduak
- helbideratze-modua zehazteko era (kodean edo eragigaietan)
- erabil daitezkeen aginduak
- erabil daitezkeen datuen formatoak
- memoriako tamaina
- erregistro orokorren kopurua

5.2.- ITZULTZAILEAK

5.2.1.- DEFINIZIOAK

L1 lengoaiaz idatzitako A programa bat hartuta L2 lengoaiaz idatzitako B programa baliokidea lortzen duen programa bati itzultzailea deritzo. A programari, berriz, iturburu-programa eta B-ri objektu-programa. Gehienetan L1 lengoaia lengoaia sinboliko bat da eta L2 lengoaia makina-lengoaia da.



5.1 irudia.- ITZULTZAILE BATEN ESKEMA

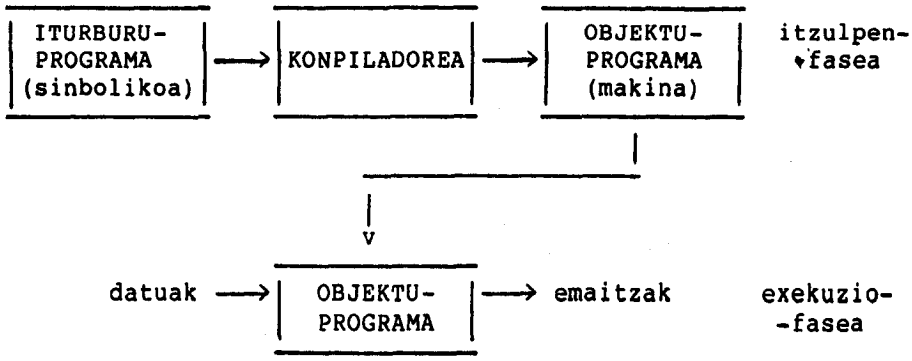
Beraz, itzultzaile bat programa bat da, zeinaren datuak goi-mailako lengoaiazko programa bat bait da eta lortzen duen emaitza makina-lengoaiazko programa baliokidea da (ikus 5.1 irudia). Egikaritzen den programa programa itzultzailea da, beraz, Memoria Nagusian kokaturik dago eta makina-lengoaiaz idatzita egongo da, bestela ezin bait litzateke egikaritu.

5.2.2.- ITZULTZAILE-MOTAK

Bi eratako programa itzultzaileak bereizten dira batez ere: konpiladoreak eta interpretatzaileak.

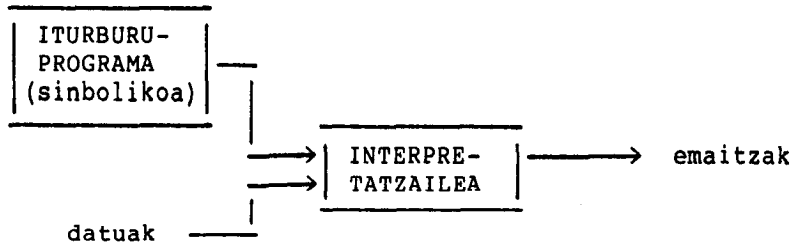
KONPILADOREAK.- Lengoaia sinbolikoz idatzitako programa bat hartuz makina lengoaiazko objektu-programa lortzen du, bitarteko memorian gordez geroztik objektu-programa nahi dugun adina bider egikaritzeko. Beraz, iturburu-programa bat idatziz gero, itzultzeko konpiladore bat erabiltzen bada bi fase bereizten dira, itzulpen-fasea objektu programa sortzeko eta egikaritzapen-fasea, benetako egikaritzapena lortzeko (ikus 5.2 irudia).

INTERPRETATZAILEAK.- Lengoaia sinbolikoz idatzitako programa bat hartuz agindu bakoitza itzuli egiten du eta itzuli ondoren egikarituerazten, objektu-programa lortu gabe eta agindu berbera behin baino gehiagotan egikaritu behar badu, horrenbeste aldiz itzuli beharko duelarik. Beraz, kasu honetan iturburu-programa idatziz gero egikaritzeko ez dira bi fase bereizten baizik eta aginduak itzuli ahala exekutatzen bait dira (ikus 5.3 irudia) eta programa bat egikaritu nahi dugun guz-



5.2 irudia.- KONPILADOREAREN FASEAK

tietan itzuli behar da.



5.3 irudia.- INTERPRETATZAILEAREN ESKEMA

Konpiladore eta interpretatzaileen arteko funtsezko desberdintasunek ondoko diferentzia hauek erakartzen dituzte:

- Programa baten garapen-faseari dagozkion probetan interpretatzaile bat edukitzea egokiagoa da, zeren eta akatsak edo erroreak detektatzeko ez da programa osoa itzuli behar eta gero probatzen hasi.

- Programa baten explotazio-fasean (ondo dagoela egiaztatuz gero) programa konpiladore batez itzulita edukitzea egokiagoa da, hau da, objektu-programa prest edukitzea, zeren eta exekuzioa arinagoa bait da itzultzeko denbora Ø denez.

- Interpretatzaileak momentu guztietan memoria nagusian egon behar duenez, berarentzat memoria gutxi hartzea komenigarria da, horregatik, lengoaia sinpleetan erabiltzen da batez ere, BASIC-en kasua hedatuena izanik. Arrazoi berberarengatik mikroetan interpretatzaileak ROM memoriatara biltzen dira.

EZAUGARRIAK	KONPILA-DOREAK	INTERPRE-TATZAILEAK
Bi fase bereiztuak	BAI	EZ
Objektu-programa	BAI	EZ
Egikaritzeko itzuli behar beti	EZ	BAI
Frogetarako egokiena	EZ	BAI
Explotaziorako egokiena	BAI	EZ

5.2 taula.- ITZULTZAILE-MOTEN EZAUGARRIAK

Aurreko diferentziak kontutan hartuz, iturburu-programa bat egin berria dagoenean progbtarako lengoaia sinbolikoari dagokion interpretatzailea erabiltzea eta ondo dagoela egiaztatuz gero dagokion konpiladorea erabiliz objektu-programa lortzea explotaziorako egoki-egokiena litzateke, baina horretarako lengoaia berberarentzat bi itzultzaile behar dira, bata konpiladore- erakoa eta interpretatzailea bestea, normalean horrela gertatzen ez delarik.

Laburpen moduan 5.2 taulan konpiladore eta interpretatzaileen ezaugarriak adierazten dira.

Konpiladoreen artean mota berezi batekoak bereizten dira: MIHIZTATZAILEAK, zeintzuren bereiztasuna mihiztadura-lengoaiazko programak itzultzea bait da. Hori dela eta, iturburu-programaren agindu bakoitza itzultzen duenean makina-lengoaiazko agindu bakar bat lortzen du mihiztatzaileak, bere lanik garrantzitsuena eta kasik bakarra sinboloak kode bitarreko bihurtzea delarik.

Beraz, mihiztatzaile bat beste edozein konpiladore baino sinpleago eta arinagoa izango da, goimailako lengoaiazko aginduak itzultzeko prozesua askoz zailagoa bait da (kontutan hartu goimailako agindu bati jeneralean agindu-multzo dexente dagokiola).

5.3.- AGINDU-MOTAK

Makina-lengoaia edo mihiztadura-lengoaia osatzen duten aginduak mota askotakoak izan daitezke eta lengoaia hauek makina bakoitzean desberdinak direnez, konputagailu bati dagokion makina-lengoaia edo mihiztadura-lengoaia mota batzutako aginduez horniturik egotea baina beste motatakoez ez egotea gerta daiteke.

Dena dela, maila orokorrean bi agindu-multzo bereizten dira: helburu orokorrekoak eta helburu berezikoak.

- helburu orokorreko aginduak edozein konputagailutako lengoaian agertzen dira, edozein erabilpen programatzeko beharrezkoak bait dira.

- helburu bereziko aginduak oso programa gutxitarako dira erabilgarriak eta, jeneralean, bakoitza helburu orokorreko agindu multzo baten baliokidea da.

Pasarte honetan helburu orokorreko aginduen mota garrantzitsuenak aztertuko ditugu.

Agindu baten egikaritzapenak ondoko aldaketa hauetakoren bat edo gehiago ondoriotzen du:

- Programaren kontagailuaren (PC-ren) edukinaren aldaketa. Agindu guztiek ondoriotzen dute aldaketa hau, NOP (ezer ez egitea) agindua sartuta ere.

- Baldintz erregistroaren edukinaren aldaketa.

- Memoriako edota erregistro orokorretako edukinaren aldaketaren bat.

Makina- edo Mihiztadura-lengoaiazko agindu mota garrantzitsuenak honako hauek dira:

- Datu-mugimenduak
- Datu-bihurketak
- Agindu aritmetikoak
- Agindu logikoak
- Konparaketak
- Jauziak
- Sarrera/Irteerak

5.3.1.- DATU-MUGIMENDUAK

Mota honetako aginduek datu bat toki batetik beste batera kopiatzen dute, kopia hau bitez bit egiten delarik. Beraz mugimendu izena ez da oso egokia zerbait mugitzen denean jatorritik desagertzen delako eta agindu hauek ez dute iturburu-datua desagertzea ondoriotzen. Emaitza bakarra helburu-helbidean zegoena desagertzea eta iturburu-datua kopiatzea da.

Horren ondorioz, geneukan datu bat bikoizten dugu beste tokian formatoa mantenduz. Mugitzen diren datuen luzera adierazi egin behar da baina hori beste pasarte batean aztertuko dugu. Hala eta guztiz ere, luzera bakar bat maneiatzen da mugimenduko agindu bakoitzean.

Datuen kokapena ez dugu zehaztu eta "tokia" esan dugunean erregistro orokorra, memoriako-helbidea edo pilako tontorra erabil daiteke han da, iturburu-toki eta helburu-toki bezala, erregistro

orokor baten zenbaki bat, memoriako helbide bat edo pila zehaztuko dugu.

Beraz, mugimendu batek ondoko tokien arteko datu-trukea erakar dezake, nahiz eta lengoaia batean denak posible ez izan:

- Memoriatik Memoriara
- Erregistrotik Memoriara
- Memoriatik Erregistrora
- Pilatik Memoriara
- Memoriatik Pilara
- Erregistrotik Erregistrora
- Erregistrotik Pilara
- Pilatik Erregistrora

Mugimendu batean bi eragigai daude: batzuetan inplizituki. Horrela, 3. kapituluan aipaturiko KARGA, GORDE, PUSH eta POP agindua mugimenduak dira, eragigai esplizitu bakar bat eduki arren.

5.3.2.- DATU-BIHURKETAK

Datu-mugimenduen bariante bat dira. Datu baten kopia ekartzen badute ere, ez da bitez bit kopiatzen mugimenduetan bezala, luzera- edota formato-aldaketa bat sortzen bait da, baina datuaren esanahia mantendu egiten da adierazpidea aldatuz.

Datu-bihurketak zenbakizko datuekin erabiltzen dira eta bi eragigaik halabehartzen dute: iturburuak eta helburuak.

Bihurketa-mota desberdinak daude, bai datuak non kokaturik daudenaren arabera bai zer formatotatik zer formatotara bihurtu denaren arabera.

Hona hemen datu-bihurketen mota desberdinen adibideak: Koma finkozko bitarretik koma higikorrezko bitarrera, bitarretik hamartar zonadunera, hamartar zonadunetik paketatura, hamartar paketatutik bitarrera, etab.

5.3.3.- AGINDU ARITMETIKOAK

Agindu hauek hiru eragigaik behartzen dituzte konputagailu motaren arabera, batzutan gutxiago esplizitatzen badira ere. Bi datu eta emaitza bat adierazten duten eragigaiek. Mota honetako aginduek bi datuen arteko eragiketa aritmetikoa burutzen dute, emaitza gorderik geldituz.

Agindu aritmetiko arruntenak batuketa, kenketa, biderkaketa eta zatiketa dira eta denek datu numerikoak erabiltzen dituzte.

Datuen formatoen arabera mota desberdin hauek bereizten dira:

- Koma finkozko formato bitarreko agindu aritmetikoak. Ia ia konputagailu guztitan ezarrita daude.

- Koma higikorrezko formato bitarrezko agindu aritmetikoak. Erabilpen zientifikotarako konputagailuetan behar beharrezkoak dira.

- Formato hamartarrezko agindu aritmetikoak. Jestio-lanetarako konputagailuetan erabiltzen dira, baina aginduak geldiagoak gertatzen dira.

Batzutan oso erabiliak diren agindu aritmetikoak arinagoak eta motzagoak izateko, agindu desberdinak balira bezala definitzen dira agindu aritmetiko bakunak deitutakoak, eragigai esplizitu bakarra halabehartzen dutelako.

Hauen artean hedatuenak handitu (inkrementatu) eta txikitu (dekrementatu) aginduak dira.

HANDITU agindua, zehaztutako datuari bat gehitu eta emaitza datuaren ordez kokatzean datza.

TXIKITU agindua, datuari bat kendu eta emaitza datuaren ordez kokatzean datza.

5.3.4.- AGINDU LOGIKOAK

Agindu mota honetan biten arteko eragiketak adierazten dituzten aginduak sartzen ditugu. Hauen artean agindu hedatuenak algebra boolearreko AND, OR eta XOR eragiketak, bit-desplazamenduak eta bit-biraketak dira.

AND LOGIKOA.- Hiru eragigai halabehartzen ditu agindu honek, OR eta XOR aginduek bezala haietako bi datuak izango direlarik eta hirugarrenak emaitza adierazten duelarik.

AND aginduak bi datuen bitak banan banan hartzen ditu, bi bitak 1 egoeran baldin badaude dagokien emaitzaren bita 1 egoeran ipintzen du eta beste kasutan Ø egoeran.

Agindu honek AND eragiketa logikoa burutzeaz gain, datu baten zenbait bit Ø egoeran ipintzeko ere balio du. Horretarako, datu bat konstante bat da, maskara deituko zaiolarik eta datuaren edukineko Ø egoeran jarri nahi ditugun bitei dagozkien maskarako bitak Ø egoeran definitzen ditugu, maskarako beste bit guztiak 1 egoeran ipiniz (kontutan hartu Ø AND X = Ø eta 1 AND X = X dela).

Adibidez, 16 biteko datu batean lehenengo eta azken lau bitak zero egoeran ipini nahi baditugu, AND eragiketa egingo dugu 0000111111110000 (0FF0) maskarekin. Horrela datua 1011 0111 1011 1100 baldin bada, ondoko emaitza hau lortuko da:

```
datua ------> 1011 0111 1011 1100
maskara ----> 0000 1111 1111 0000
              ---------------------
emaitza ----> 0000 0111 1011 0000
```

Hau da, emaitzaren 5.bitetik 12.bitera bitartekoak datuarenak bezalakoak dira eta lehenengo eta azken bita berriz zeroak.

OR LOGIKOA.- Bi daturen artean OR eragiketa boolearra burutzen du emaitza gordez, beraz, bi datuen bitak banan banan hartzen dira, bi bitak 0 egoeran baldin badaude dagokien emaitzaren bita 0 egoeran gelditzen da eta beste kasu guztietan 1 egoeran.

Agindu honek OR eragiketa logikoa burutzeaz aparte, datu baten zenbait bit 1 egoeran ipintzeko ere balio du. Horretarako ere, maskara bat erabiltzen da, zeinetan 1 egoeran jarri nahi diren bitei dagozkien bitak 1 egoeran definitzen bait dira, beste bitak 0 egoeran definituko direlarik, dagozkien datuaren bitak mantenduz.

Adibidez, lehengo adibidean datuaren lehenengo eta azken lau bitak 1 egoeran jarri nahi baditugu, beste guztien egoera mantenduz, maskara 1111 0000 0000 1111 balioaz definituko dugu:

```
datua -------> 1011 0111 1011 1100
maskara -----> 1111 0000 0000 1111
               ---------------------
emaitza -----> 1111 0111 1011 1111
```

XOR LOGIKOA.- Agindu honek bi daturen bitak banan banan hartzen ditu eta bi bitak egoera berberean baldin badaude, dagokien emaitzaren bita 0 egoeran jartzen du, bitak egoera desberdinetan badaude berriz 1 egoeran jartzen du.

XOR eragiketa logikoa burutzeaz gain, honen bidez, datu baten zenbait biten egoera alda daiteke. Horretarako, egoeraz aldatu nahi diren datuaren bitei dagozkien maskarako bitak 1 egoeran definitzen dira, beste guztiak 0 egoeran definitzen direlarik.

Adibidez OR eta AND aginduekin erabilitako datuaren lehenengo eta azken lau biten egoera aldatu nahi badugu beste bit guztien egoera mantenduz 1111 0000 0000 1111 maskara erabiliko dugu:

```
datua -------> 1011 0111 1011 1100
maskara -----> 1111 0000 0000 1111
               ------------------
emaitza -----> 0100 0111 1011 0011
```

BIT-DESPLAZAMENDUA.- Agindu honen bidez datu baten bit guztiak zenbait posizio eskuinetara edo ezkerretara desplaza daitezke. Teorikoki hiru eragigai dauzka: datua, posizio-kopurua eta emaitzaren tokia, baina batzuetan posizio-kopurua batean finkatzen da eta emaitza datuaren lekuan kokatzen da agindu bakun bihurtuz.

Ezkerretarako desplazamenduetan desplazatzen diren posizio-kopurua hainbat bit galtzen dira ezkerraldean eta eskuinaldean beste horrenbeste bit zeroz betetzen dira. Eskuinetarako desplazamenduetan eskuinaldetik galtzen dira bitak eta ezkerraldetik betetzen dira 0z eta 1ez, desplazamendu motaren arabera.

Desplazamenduak, ezkerretakoaz edo eskuinetakoaz izateaz aparte bi motatakoak izan daitezke: aritmetikoak eta logikoak.

Desplazamendu aritmetikoa.- Ezker-ezkerreko bita zeinua dela suposatzen da eta ez da desplazatzen. Eskuinetarako desplazamenduetan ezkerreko bitak zeinu-bitez betetzen dira. Ezkerretarako n biteko desplazamendu aritmetikoa, 2n eta datuaren arteko biderkaduraren baliokidea da eta eskuinetarako desplazamendu aritmetikoa berriz, datuaren eta 2nren arteko zatiduraren baliokideak.

Desplazamendu logikoa.- Bit guztiak desplazatzen dira eta betetzeko beti zeroak erabiltzen dira.

Ikus dezagun desplazamendu mota guztietako adibide bat, datua 1001 0111 1011 1100 balioa izanik eta 2 posizio-kopurua:

- Eskuinetarako desplazamendu aritmetikoa.
  1001 0111 1011 1100 ------> 1110 0101 1110 1111
- Ezkerretarako desplazamendu aritmetikoa.
  1001 0111 1011 1100 ------> 1101 1110 1111 0000
- Eskuinetarako desplazamendu logikoa.
  1001 0111 1011 1100 -------> 0010 0101 1110 1111
- Ezkerretarako desplazamendu logikoa.
  1001 0111 1011 1100 --------> 0101 1110 1111 0000

BIT-BIRAKETA.- Bit-desplazamenduaren antzekoa da agindu hau, baina desplazamenduetan bit batzuren informazioa galdu eta beste batzurena betetzen den bitartean, biraketetan berriz, alde batetik galtzen diren bitak beste aldetik sartzen dira.

Datua osatzen duten bitek zirkulu batean baleude bezala jokatzen dute eta horregatik biraketa deitzen zaio aginduari.

Ezkerretarakoa edo eskuinetarakoa izan daiteke biraketa eta logikoa bakarrik hartuko dugu kontutan, biraketa aritmetikoak ez bait du zentzu handirik.

Lehengo adibidea biraketa bezala egin dezagun:

- Eskuinetarako biraketa (2 posizio).
  1001 0111 1011 1100 ------> 00 1001 0111 1011 11

- Ezkerretarako biraketa (2 posizio).
  1001 0111 1011 1100 ------> 01 0111 1011 1100 10

5.3.5.- KONPARAKETAK

Konparaketa bat burutzen denean sortzen den emaitza bakarra baldintz erregistroko aldaketa da. Gehienetan agindu honen atzetik baldintzapeko jauzi bat dago konparaketaren emaitza erabiltzeko.

Konparaketa batek bi eragigai halabehartzen ditu, eragigai hauek konparatzen diren datuak adierazten dituztelarik.

Konparaketak aritmetikoak eta logikoak izan daitezke. Aritmetikoetan datuak zenbakitzat hartzen dira, formatoak eta luzerak kontutan hartzen direlarik, konparaketa burutzen denean, baldintz erregistroan lehenengo datua bigarrena baino handiagoa, txikiagoa edo berdina den adieraziko da emaitza gisa. Konparaketa logikoak burutzean, bi datuen bitak banan banan aztertzen dira ezkerretatik eskuinetara, baldintz erregistroan sortuko den emaitzak berdintasuna adieraziko du datuen bit guztiak berdinak baldin badira eta desberdintasuna gainerantzekotan.

Konputagailu batzutan 0 zenbakiaren konparaketa egiteko agindu bakuna existitzen du.

5.3.6.- JAUZIAK

Izen honekin ezagutzen diren aginduek programaren segida normala aldatzen dute (ikus 5.4 irudia). Horretarako, PC erregistroaren edukina aldatzen dute aginduaren exekuzioaren ondorioz.

Beraz, segida normala alda dezakeen agindu-mota bakarra jauzia da, beste mota guztietako aginduen egikaritzapenaren ondoren segidan dagoen hurrengo sententzia egikaritzen bait da.

```
(ag1) ---------
(ag2) ---------
(ag3) ---------
(ag4) JAUZI ag8---        Egikaritzapenaren segida:
(ag5) ---------  |        ag1, ag2, ag3, ag4, ag8, ag9
(ag6) ---------  |
(ag7) ---------  |
(ag8) ---------<--
(ag9) ---------
```

5.4 irudia.- JAUZI AGINDUAREN ESKEMA

Agindu hauek eragigai bat gutxienez halabehartzen dute; eragigai honek PC-an kokatu behar den helbidea adierazten duelarik, hau da, egikaritu nahi dugun hurrengo aginduaren helbidea. Helbide horri jauzi-helbidea deritzo.

Jauziak mota desberdinetakoak izan daitezke, alde batetik baldintzagabekoak ala baldintzapekoak eta bestetik itzulera gabekoak ala itzuleradunak.

BALDINTZAGABEKO ETA BALDINTZAPEKO JAUZIAK. Baldintzagabeko aginduek segida normalaren aldaketa erakartzen dute beti. Baldintzapeko jauziek, berriz, lehenbizi baldintza bat egiaztatzen dute baldintz erregistroa erabiliz eta baldintza betetzen bada segidaren aldaketa gertatuko da. Bestela ez dago ondoriorik eta segida normalari jarraitzen zaio (ikus 5.5 irudia).

```
                                   Egikaritzapenaren segida:
(ag1) --------------                        ag1
(ag2) --------------                        ag2
(ag3) --------------                        ag3
(ag4) JAUZI bald,ag8      bald EGIAZK<---ag4--->bald FALTSUA
(ag5) --------------          ag8                   ag5
(ag6) --------------          ag9                   ag6
(ag7) --------------                                ag7
(ag8) --------------                                ag8
(ag9) --------------                                ag9
```

5.5 irudia.- BALDINTZAPEKO JAUZIAREN ESKEMA

Beraz, baldintzagabeko jauziek eragigai bakarra eskatzen duten bitartean (jauzi-helbidea), baldintzapeko jauziek bi eragigai halabehartzen dituzte: baldintza eta jauzi-helbidea.

Orain arte ikusitako jauziak jauzi arruntak edo itzuleragabekoak dira eta jauzi hauetan jauzi-helbidera joanez gero exekuzioaren segida era normalean egikaritzeari jarraitzen zaio konputagailua.

ITZULERADUN JAUZIAK (AZPIPROGRAMARAKO JAUZIAK). Itzuleradun jauzietan, berriz, azpiprograma (azpirrutina edo prozedura) baten lehenengoa den agindura jauzi egiten da eta azpiprogramaren egikaritzapena bukatzean itzulera-helbidera bueltatzen da (ikus 5.6 irudia). Beraz, agindu berezi bat behar da azpiprogramaren amaia adierazteko.

```
                                        Prozedura
(ag1) ------------        ----->(agA) ----------
(ag2) ------------        |     (agB) ----------
(ag3) ------------        |     (agC) ----------
(ag4) JAUZI proze -------       (agD) ----------
(ag5) ------------<-----------(agE) AMAIA
(ag6) ------------
(ag7) ------------          Egikaritzapenaren segida:
(ag8) ------------        ag1,ag2,ag3,ag4,agA,agB,agC,
(ag9) ------------        agD,agE,ag5,ag6,ag7,ag8,ag9
```

5.6 irudia.- ITZULERADUN JAUZIAREN ESKEMA

Itzulera-helbidea itzuleradun jauzi aginduari segida normalean jarraitzen zaion aginduaren helbidea da eta nonbait (memoriako hitz batean, erregistro batean edo pilako tontorrean) gorde beharko da helbide hori geroago PC-an kokatuz, itzulera lortzen bait da. Askotan, itzulera-helbidea gordetzeko, beste eragigai batez zehazten da.

Beraz, azpirrutina edo prozedura baterako jauzi baten egikaritzapenak bi aldaketa ekartzen ditu:

- PC-ren edukina gordetzen da nonbait (eragigai batek zehaztutako lekuan edo inplizituki gordetzen den tokian). Honekin itzulera-helbidea biltzen da.

- Jauzi-helbidea PC-an sartzen da. Honekin azpiprogramaren lehenengo agindura jauztea lortzen da.

Honez gero azpiprograma egikaritzen da eta amaiaren agindua aurkitzen denean ondoko aldaketa gertatzen da:

- Itzulera-helbidea gorderik dagoen tokitik PC-ra mugitzen da. Honekin jauziaren ondorengo agindura joaten da.

5.3.7.- SARRERA/IRTEERAKO AGINDUAK

Agindu hauen bidez Unitate Zentralaren eta Sarrera/Irteerarako dispositiboen artean trukatzen da informazioa. Bereizten dira sarrerako aginduak edo irakurketak (dispositibotik memoriara informazioa bidali) eta irteerako aginduak edo idazketak (memoriatik dispositibora informazioa bidali).

Agindu guztiak konputagailu batetik bestera asko aldatzen badira ere, Sarrera/Irteerako aginduak gehien aldatzen direnak dira, dudarik gabe. Hala eta guztiz ere, ondoko bi multzo handietan bana daitezke sarrera/irteerako aginduak konputagailuen arabera:

- Sarrera/Irteerako agindu berezirik ez duten konputagailuak, zeinetan dispositiboak memoriako posizio edo gelaskatzat jotzen bait dira programatzeko orduan. Mikro gehienetan horrela gertatzen da. Kasu honetan sarrera/irteerako aginduak gelaska berezi hauek maneiatzen dituzten mugimenduak dira eta horregatik agindu bakoitzaz irakur edo idatz daitekeen informazioa karaktere edo zenbaki oso bat da.

- Sarrera/Irteerako agindu bereziez horniturik dauden konputagailuak. Konputagailu hauetan irakurketa edo idazketa bat egitea zail samarra gertatzen da, agindu multzo bat (multzo honetan sarrera/irteerako agindu bereziak eta bestelako aginduak daudelarik) idatzi behar bait da. Dena dela, agindu-multzo hauetako batez erregistro bat irakurtzen edo idazten da. Sarrera/irteerako agindu berezietan, besteen artean, ondoko parametroak zehaztu beharko dira: irakurketa edo idazketa, dispositiboaren identifikazioa eta memoriako helbidea (informazioa hartzeko edo uzteko).

Sarrera/irteera programatzea oso konplexua gertatzen dela eta konputagailu askotan Sistema Eragileak laguntzen du makroagindu batzuren bidez. Makroagindu bakoitza mihiztadura-lengoaiaz agindu bakar bat izango balitz bezala erabiltzen da, baina itzulitakoan makina-lengoaiazko agindu-multzo bat sortzen da (kasu honetan makromihiztadura-lengoaiaz hitz egiten da).

Sarrera/irteerarako Sistema Eragilearen erreztasunak erabiltzeari Alegiazko sarrera/irteera deitzen zaio.

5.4.- AGINDUAK MAKINA-LENGOAIAZ

Makina-lengoaia baten ezaugarriak ondoko parametro hauek aztertuz definitzen dira:

- Maneiatzen dituen agindu-motak
- Aginduaren formatoa
- Maneiatzen dituen helbideratze-moduak
- Maneiatzen dituen datuen formato eta luzerak
- Programaren kontrola

Pasarte honetan, parametro hauek, lehenbizikoa ezik, banan banan aztertuko ditugu, VAX eta IBM-370 konputagailuen makina-lengoaiak erabiliko ditugularik adibideak jartzeko.

5.4.1.- AGINDUAREN FORMATOA

Makina-lengoaiazko aginduen formatoa konputagailua diseinatzen denean erabakitzen da, zeren eta formato hauek CPU-ren egiturarekin zerikusi handia bait dute.

Dakigunez, agindu batean bi alderdi bereizten dira eragiketaren kodea eta eragigaiak. Eragigaiaren kodeak egikaritu behar den eragiketa adierazten du eta eragigaiak edo eragigaiek berriz eragiketa horretan parte hartzen duten datua edo datuak zehazten ditu. Eragigaien kopurua konputagailu-motak baldintzatzen du eta hutsetik hirura bitartekoa izan daiteke.

Aginduaren formatoa erabakitzen denean honako erizpide hauei jarraitu behar zaie:

a) Aginduek ahalik eta laburrenak izan behar dute. Agindu laburrak erabiltzeak memori hartze txikiagoa eta batez ere abiadura handiagoa ekartzen ditu, zeren eta dendora berean agindu gehiago eskura bait dezake Kontrol Unitateak.

b) Aginduan behar diren agindu desberdin guztiak adierazteko behar den adina bit utzi behar da.

c) Aginduaren luzera byte kopuru oso bat izatea komenigarria da.

d) Memoriako edozein helbide adierazteko behar den adina bit gordeko da eragigaietan. Edukin berbereko konputagailuetan hizkako batek bit-kopuru txikiagoa behar du karakterekako batek baino. Izan ere karakterekakoak bereizmen handiagoa bait du.

   Hizkako konputagailuak helbide gutxiago erabiltzen du, baina karaktere batera iristeko karaktere hori daukan hitzera helbideratu behar da eta gero hitz horretan karakterea aukeratu.

Beraz, erizpide hauei jarraituz, aginduaren luzera erabakitzen da, kode eta eragigai bakoitzaren luzera zehaztuz.

Aginduen luzera bi motatakoa izan daiteke:

- Luzera finkoko aginduak. Lengoaiako agindu guztiek luzera finkoa dute eta, ondorioz, kodearen luzera eta eragigaien luzera finkoak dira.

- Luzera aldakorreko aginduak. Agindu batzu besteak baino luzeagoak dira kodearen luzera desberdina dutelako edota eragigaien luzera desberdina dutelako.

Luzera finkoko aginduek CPU-ren arkitektura eta funtzionamendu sinpleagoa ekartzen dute, aginduaren erregistroan beti agindu bakar oso bat sartuko da eta.

Izan ere, luzera finkoko aginduak luzeagoak gertatzen dira, zeren eta aukeratzen den luzera horrek agindu luzeena adierazteko behar dena izan behar bait du.

Aurreko arrazoiak direla medio, eraikitzen diren konputagailu gehienek luzera aldakorreko aginduak maneiatzen dituzte, baina luzeren aukera mugatuta dago formato ezagutu batzuren artean, arkitektura eta funtzionamendua konplexuegiak gerta ez daitezen.

5.4.2.- KODEAREN FORMATOA

Dena dela, konputagailu askotan aginduen luzera aldakorra bada ere, eragiketaren kodeak luzera finkoa du, horrela, Deskodetzaileak beti bit-kopuru berbera erabiliko bait du.

Kodearen luzera finkoa denean luzera hau aginduen kopuruak baldintzatzen du.

Zenbat bit beharko du n agindu desberdin maneiatzen duen konputagailu batek kodea adierazteko, kode hori luzera finkokoa baldin bada?

Biten kopurua edo luzera bitetan 1 izango da 2^1=n betez. Horrela, 4 agindu desberdin adierazteko kodeak 2 bit behar du, 64 agindu adieraztekoak 6 bit, 256 agindu adieraztekoak 8 bit, etab.

8 biteko luzera finkoa oso hedatua dago konputagailu askotan (VAX, IBM-370, ...), beraz, konputagailu hauek 256 agindu desberdin maneia dezakete.

Kodea luzera aldakorrekoa denean, maiztasun handieneko aginduek kode laburrena dute. Eragigaien luzera helbideratze-moduaren menpean dago eta ondoko azpigalderan aztertuko dugu.

5.4.3.- AGINDUAREN FORMATOEN ADIBIDEAK

Ikus dezagun VAX eta IBM-370 konputagailuen aginduen formatoa.

A) VAX-EN ADIBIDEA.- Ondoko ezaugarriak ditu bere makina-lengoaiak:

- Karakterekako konputagailua.
- Luzera aldakorreko aginduak.
- Luzera finkoko agindu-kodea. Luzera 8 bitekoa. 256 agindu desberdin.
- Eragigaien kopuru aldakorra. Eragigai 1,2,3 edo gehiago agin-

AGINDU-KODEA	1. ERAGIGAIA MOTA H.M.	Rn	HELBIDEA	BESTE ERAGIGAIAK (egonez gero bakoitza 1.eragigaiaren format
<-8 bit->	<4+4 bit>		<0,8,16 o 32 bits>	

* H.M. --> Helbideratze-modua

5.7 irudia.- VAX-eko AGINDUEN FORMATOA

duaren arabera.
- Eragigai bakoitzean bi datu zehazten dira, helbideratze-modua eta helbidea.
- Helbideratze-modua 4 bitez adierazten da.
- Helbidea 0,8,16 edo 32 bitez adierazten da helbideratze-moduaren arabera.

5.7 irudian konputagailu honetako aginduaren formatoa ikus dezakegu.

B) IBM-370-EN ADIBIDEA.- Ondoko ezaugarriak ditu:

- Karakterekako konputagailua.
- Luzera aldakorreko aginduak.
- Aginduak bost formato desberdinetan bereizten dira, RR, RX, RS, SI eta SS izeneko formatoetan hain zuzen. RR formatoari 16 biteko luzera dagokio; RX, RS eta SI formatoei 32 bitekoa eta SS formatoari 48 bitekoa.
- Agindu guztiek kodearen luzera finkoa dute: 8 bitekoa.
- Agindu guztiek beti 2 eragigai erabiltzen dute.
- Helbideratze-modua kodeak adierazten du inplizituki.

5.8 irudian IBM-370 konputagailuko aginduaren formatoak ikus ditzakegu.

5.4.4.- HELBIDERATZE-MODUEN ADIERAZPIDEA

Eragigaien formatoa helbideratze-moduaren menpean dagoela esan dugu. Konputagailu batzuetan (VAX-ean adibidez), 3. kapituluan ikusi genuenez, helbideratze-modua eragigaian bertan adierazten da, bestetan (IBM-370ean adibidez) kodeak behartzen du eragigaien helbideratze-modua. Lehenengo kasuan helbideratze-moduaren adierazpidearen luzera finkoa izan ohi da eta, maneiatzen diren moduen kopuruaren arabera, luzera finkatzen da. Horrela, VAX-ean 16 helbideratze-modu desberdin maneiatzen dira eta ondorioz 4 bit behar dira helbideratze-modua zehazteko (4 bit => 16 informazio desberdin).

Helbidea adierazteko behar den bit kopurua helbideratze-moduaren menpean dago, nahiz modua eragigaian zehaztu nahiz kodearen

RR form.	agindu-kodea	R1	R2		
	< 8 bit >	< 4 +	4 >		
RX form.	agindu-kodea	R1	X2	O2	D2
	< 8 bit >	< 4 +	4 >	<- 4	+ 12 bits->
RS form.	agindu-kodea	R1	R3	O2	D2
	< 8 bit >	< 4 +	4 >	<- 4	+ 12 bits->
SI form.	agindu-kodea	datu bereh-koa		O1	D1
	< 8 bit >	< 8 bit >		<- 4	+ 12 bits->
SS form.	agindu-kodea	luzera(k)	O1	D1	O2 | D2
	< 8 bit >	< 8 bit >	<- 4 + 12 bits->		<- 4 + 12 bits->

* R1,R2,R3 ikurrek erregistro orokorrak adierazten dituzte
* O1,O2 ikurrek oinarri erregistroak adierazten dituzte
* D1,D2 ikurrek desplazamendu positiboak adierazten dituzte
* X2 ikurrak indize-erregistro bat adierazten du

5.8 irudia.- IBM-370eko AGINDUEN FORMATOAK

arabera behartuta egon. Hona hemen luzera desberdinak dagozkien helbideratze-moduak:

- Berehalako datuak.- Datuaren edota formatoaren arabera luzera desberdinak sor daitezke. Konputagailu gehienetan karaktere bat edo koma finkozko formato bitarraren bidezko zenbaki bat bakarrik onartzen da berehalako datu bezala, aipatutako datuak byte edo hitz bakar batean sartzen bait dira.

- Helbideratze absolutuak eta zeharkakoak.- Eragigaian memoriako helbide bat zehaztu behar denez gero, memoriako edozein helbide adierazteko adina bit beharko izango dira. Beraz, n gelaska duen memoriako edozein helbide adierazteko l=log n (logaritmo bitarra) bit behar dira, gelaska bakoitzari helbide bat bait dagokio eta helbideak zeinu gabeko formato bitarrez adierazten bait dira.

Adibidez, zenbat bit behar dira 4 Kbyte duen memoriako helbide bat adierazteko?

Karakterekako konputagailua baldin bada: 4*1024 helbide desberdin dugu, beraz 12 bit behar dira ($4*1024=4096=2^{12}$).

Hizkako konputagailua baldin bada eta 4 bytek hitz bat osatzen dutela suposatuz, 1024 helbide desberdin dugu beraz 10 bit behar dira.

- Erregistro bidezko helbideratze zuzenekoak edo zeharkakoak.- Eragigaian erregistro orokor baten zenbakia zehaztu behar da, beraz, erregistro-kopuruaren arabera bit kopuru desberdina beharko da zenbakia adierazteko. Erregistro kopuru hedatuena 16koa 0tik 15era zenbaturik. Honen ondorioz, kasu hauetan 4 biteko luzera beharrezkoa izango da ($2^4=16$) edozein erregistro zehazteko.

- Helbideratze erlatiboak.- Kasu honetan, eragigaian erregistro baten zenbakia eta desplazamendua zehaztu beharra dago. Beraz, erregistroaren zenbaki altuena eta desplazamendu handiena adierazteko behar den biteko luzera gordeko da.

Helbideratze erlatiboak 3 motatakoak izan daitezke eta 3 motei funtzio eta luzera desberdinak dagozkie.

- Oinarri-erregistro bidezko erlatiboa.- Helbideratze-modu honetaz ez da memoriako edozein helbidetara iristen, tarte batean dagoen helbide batera baizik. Eta hori desplazamenduaren luzerak edozein helbide adierazteko balio ez duelako.

Oinarri-erregistroen erabilpenaren funtsa hau da: Memoria zenbait alderditan banatzen da, alderdi bakoitzari alderdiaren hasierako helbidea edukiko duen erregistro bat egokitzen zaiolarik. Erregistro horri oinarri-erregistroa deritzo. Beraz datu batera helbideratzeko, lehenbizi dagoen alderdia ezagutu behar da, alderdiari dagokion erregistroa zehaztu beharko dugularik eta gero alderdi horren hasierarekiko desplazamendua kalkulatuko da.

Sistema honen abantailarik garrantzitsuena memoriara helbideratzeko bit-kopurua gutxitzea da.

Adibidez, IBM-370 konputagailuan memoria 4 K-ko alderditan banatzen da. Memoriak 2^{24} gelaska ditu beraz helbideratze absolutu bat zehazteko 24 bit beharko dugu. Oinarri-erregistrozko helbideratze erlatiboa erabiliz, berriz, erregistroaren zenbakirako 4 bit eta desplazamendurako 12 bit ($2^{12}=4K$) behar ditugu, beraz, guztira, 16 bitez memoriako edozein helbide zehatz daiteke.

Momentu batean memoriako helbide guztietara iritsi ezina da eragozpenik handiena. Lehengo adibidean, 4096 alderdi daudenez, 4096 erregistro beharko genituzke, baina 16 daudelako, gehienez 16 alderditako informaziotara hel gaitezke momentu berean.

- Indize-erregistro bidezko erlatiboa.- Desplazamenduan memoriako edozein helbide adieraz daiteke, beraz, desplazamenduaren luzera helbideratze absolutuari dagokion adina bitekoa izango da. Horren ondorioz, helbideratze absolutuarena baino luzeagoa da luzera, oinarri-erregistroa adierazten duten biten kopurua diferentzia delarik.

Modu hau tauletara helbideratzeko erabili ohi da, desplazamenduan taularen hasierako helbidea kokatuz eta indize-erregistroaren edukin bezala osagaiaren helbidea eta hasierako helbidearen arteko diferentzia erabiliz.

- PC erregistro bidezko erlatiboa.- Kasu hauetan erabiltzen den erregistroa ezin da edozein erregistro orokor izan, baizik eta PC erregistroa izatera behartuta dago. Desplazamenduak positiboak edo negatiboak izan daitezke (aurreko kasutan beti positiboak izan behar zuten). Luzera aldetik ezin da arau orokor bat eman, konputagailuaren arabera desberdina bait da.

Modu honetan adierazitako helbideratzeak automatikoki birkokagarriak izatea da abantailarik garrantzitsuena, baina esplizituki erabiltzeko oso astuna da, gauden helbidetik datuaren helbidera dagoen desplazamendua kalkulatu behar bait da, datu bera agindu desberdinetan desplazamendu desberdinez adieraziko delarik. Eragozpen honengatik ez da helbideratze-modu hau asko erabiltzen. Hala ere, jauzi-aginduei ondo doakie modu hau erabiltzea, jauzietan gertu dauden aginduetara helbideratzen da eta.

5.4.5.- ERAGIGAIEN ADIERAZPIDEAREN ADIBIDEAK

Ikus dezagun VAX eta IBM-370 konputagailuetan zer helbideratze-moduk erabiltzen diren eta nola adierazten diren.

A) IBM-370AREN ERAGIGAIAK

Konputagailu honetan lau helbideratze-modu besterik ez dira onartzen; berehalakoa, erregistrozko zuzenekoa, oinarri erregistrozko erlatiboa eta indize erregistrozko erlatiboa deritzenak. Gainera, modu hauek ezin dira agindu guztietan erabili, aginduaren formatoak behartzen bait ditu. Horrela 5.8 irudian aipatutako RX formatoa duten agindu guztiek, 1. eragigaian helbideratze erregistrozko zuzenekoa eta 2.ean indize-erregistrozko erlatiboa behartzen dute. Azter ditzagun helbideratze-moduak banan banan.

Berehalakoa.- SI motako aginduetan 2. eragigaian soilik onartzen da. Luzera 8 bitekoa da 5.8 irudian ikus daitekeenez, beraz, karaktere bakarreko berehalako datuak bakarrik erabil daitezke.

Erregistro bidezko zuzenekoa.- RR motako aginduetan 1. eta 2. eragigaiak, RS motakoetan 1. eta 3. eragigaiak eta RX motakoetan

HELBIDERATZE-MODUA	AGINDU-MOTA	ERAGIGAIA (1. 2. 3.)	LUZERA bitetan	ADIERAZPIDEA mihiztadura-len
Berehalakoa	SI	2.	8	C'c' edo X'nn'
Erregistrozko zuzenekoa	RR RX RS	1. 2. 1. 1. 3.	4 4 4	n n n
Oinarri erregis bidezko erlatiboa	RS SI SS	2. 1. 1. 2.	16(4+12) 16(4+12) 16(4+12)	d(n) d(n) d(n)
Indize-erregistrozko erlatibo	RS	2.	16 (4+4+12)	d(x,n)

5.3 taula.- IBM-370eko HELBIDERATZE-MODUAK

1. eragigaia erregistrozko zuzeneko helbideratzeaz adierazi behar dira nahitaez. Konputagailu honetan 16 erregistro orokor daudenez gero, erregistro bat adierazteko 4 bit behar dira.

Oinarri-erregistro bidezko erlatiboa.- RS aginduetan 2.eragigaian, RI aginduetan 1.eragigaian eta SS aginduetan 1. eta 2.eragigaietan mota honetako helbideratze bat zehaztu behar da. Konputagailu honetan erregistro bat adierazteko 4 bit aski direnez eta desplazamendua 12 bitek osatzen dutenez, 16 bit beharrezkoak dira oinarri-erregistrozko helbideratze erlatibo bat zehazteko.

Indize-erregistro bidezko erlatiboa.- RX aginduetan 2.eragigaian oinarri-erregistroaren eta desplazamenduaren zehaztapenaz gainera indize-erregistro bat adierazten da, horrela, desplazamenduaren eta oinarri-erregistroaren bidez lortutako memoriako helbidea indize batez alda daiteke. Berezko helbidea, oinarri- eta indize-erregistroen edukinak desplazamenduari gehituz burutzen da. Helbideratze honetako luzera 20 bitetakoa da 12 desplazamendurako eta 4 erregistro bakoitzerako.

Laburpen gisa 5.3 taulan IBM-370aren helbideratzeen ezaugarriak azaltzen dira.

B) VAX-EN ERAGIGAIAK

IBM-370 konputagailuaren moduekin konparatuz oso diferentzi nabarmenak dauzkagu, alde batetik askoz modu gehiago onartzen direlako eta bestetik agindu gehienetan eragigaiei dagozkien helbideratze-moduak mugaturik ez daudelako, baina eragigai bertan helbideratze-modua zehaztu behar da.

HELBIDERATZE-MODUA	Makina-lengoaiaz MOTA	ERREGIST	HELBIDEA	ADIERAZP. mihiz-len
Berehalakoa	8	15	datua(8,16,32 bit)	#n
Absolutua	9	15	helbidea (32 bit)	@#n
Erreg.zuzenekoa	5	n	-----	Rn
Erreg.zeharkak.	6	n	-----	(Rn)
Erreg.zeharkako autohanditua	8	n	-----	(Rn)+
Erreg.zeharkako autotxikitua	7	n	-----	-(Rn)
Erreg.zeharkako 2.mail.autohand	9	n	-----	@(Rn)+
Erlatiboa	10 12 14	n n n	desplazam.(8 bit) desplazam.(16 bit) desplazam.(32 bit)	d(Rn) d(Rn) d(Rn)
Erlatibo zeharkakoa	11 13 15	n n n	desplazam.(8 bit) desplazam.(16 bit) desplazam.(32 bit)	@d(Rn) @d(Rn) @d(Rn)
Indize-erregist erlatiboa	4	n	8,16,24 edo 40 bit beste eragigai bat	...[Rn]

5.4 taula.- VAX-eko HELBIDERATZE-MODUAK

Beraz, 5.7 irudian azaltzen denaren arabera, eragigai bakoitzean 2 edo 3 informazio eman behar dira: helbideratze-modua, erregistro bat eta batzutan helbide bat. Horren ondorioz, eragigaiaren luzera, aipatutako 3 informazioen luzerak gehituz lortuko dugu eta helbideratze-modua 4 bitez, erregistroa beste 4ez eta helbidea behar denean, 8, 16 edo 32 bitez adierazten direla kontutan hartuz, eragigai-luzera 8, 16, 24 edo 40 bitekoa da, hau 1, 2, 3 edo 5 bytekoa.

Helbideratze-moduen zehaztapena ez luzatzearren 5.4 taulan azaltzen da.

5.4.6.- DATUEN FORMATOA ETA LUZERA

Konputagailuak eragigai batez badaki zehatz-mehatz datua non dagoen, hau da memoriako zer helbidetan edo zer erregistrotan kokaturik dagoen, baina behin datu hori atzituz gero, zer formatotan interpretatzen du eta luzera eskuratzen du? Galdera honi pasarte honetan erantzungo diogu.

Beraz, bi erabaki hartu behar dira erlazionaturik badaude ere, bata formatoa nola erabaki eta bestea luzera nola erabaki.

A) DATUEN FORMATOA

Konputagailu gehienetan agindu bati dagozkion datuen formatoa aginduaren kodearen araberakoa da, beraz, eragiketa berdina burutzeko, datuen formato bakoitzari agindu-kode desberdina dagokio. Adibidez, formato bitarrean dauden datuen arteko batuketa bat adierazteko 8Ø (hamaseitar) kodea zehaztu behar bada, formato hamartar paketaturik dauden datuen arteko batuketa adierazteko ezin da 8Ø kodeaz adierazi, beste kode batez baizik.

Aurrean esandakoaren ondorioz, Kontrol-Unitatea, agindu-kodea aztertuz, datuek izango duten formatoaz jabetzen da.

B) DATUEN LUZERA

Formatoa jakinez gero, formato hau luzera finkokoa baldin bada, datuaren luzera behartuta dago, beraz, Kontrol-Unitateak ez du luzera asmatu beharrik. Formatoaren luzera aldakorra baldin bada berriz, KU-k nolabait luzeraz jabetu beharko du eta horretarako eragigaietan zehaztu behar da luzera.

DATUEN FORMATOA	DATUEN LUZERA	AGINDU-KODEA SINBOLIKOA	AGINDU-KODEA MAK-LENG	ERAGIGAIAK
Koma finkozko bitarra	8 bit	ADDB2	8Ø	eragl,erag2
	16 bit	ADDW2	AØ	eragl,erag2
	32 bit	ADDL2	CØ	eragl,erag2
Koma higikorrezko bitarra	32 bit	ADDF2	4Ø	eragl,erag2
	64 bit	ADDD2	6Ø	eragl,erag2
Hamartar paket	?	ADDP4	2Ø	luzl,erl,luz2,er2

5.5 taula.- VAX-eko BATUKETAK EGITEKO AGINDUAK ETA DAGOZKIEN DATUAK

Beraz, formato bitarra maneiatzen duten aginduetan ez da datuen luzera zehaztu beharrik. Formato hamartarra edo karakterea maneiatzen duten aginduetan,, berriz datuen luzera adierazi behar da eragigaietan, bai beste eragigai bezala (VAX-en kasuan adibidez) bai eragigai baten alderdi bat bezala (IBM-37Øaren SS motako aginduetan, adibidez).

Adibide gisa, 5.5 eta 5.6 tauletan, bi konputagailuetan batuketak egiteko dauden agindu desberdinak eta dagozkien datuen for-

DATUEN FORMATOA	DATUEN LUZERA	AGINDU-KODEA SINBOLIKOA	AGINDU-KODEA MAK-LENG	AGINDUAREN FORMATOA
Koma finkozko bitarra	32 bit	AR	1A	RR
	16 bit	AH	4A	RX
	32 bit	A	5A	RX
Koma higiko-rrezko bitarra	64 bit	ADR	2A	RR
	64 bit	AD	6A	RX
Hamartar paket	?	AP	FA	SS

5.6 taula.- IBM-370eko BATUKETAK EGITEKO AGINDUAK ETA DAGOZKIEN DATUAK

matoa eta luzera adierazten dira.

5.4.7.- ARIKETA EBATZIA

Konputagailu batean makina-lengoaian aginduen luzera finkoa da eta 16 bitek osatzen dute agindu bakoitza. Helbideratze-modu bakarra absolutua da eta 6 bit beharrezkoak dira helbide bat adierazteko.

Aginduak bi formatotakoak izan daitezke:

- (a) formatoa non aginduan kodea eta eragigai bakarra bait dauden.
- (b) formatoa non aginduan kodea eta bi eragigai agertzen bait dira.

(b) motako 15 kode desberdin baldin badaude, kalkulatu (a) motako zenbat kode desberdin egon daitezkeen.

EBAZPENA:

A) Lehenbizi agindu-kodeen luzerak kalkulatuko ditugu

(a) motako aginduetan
luzera (kodea) + luzera (eragigaia) = 16
luzera (kodea) = 16 - luzera (eragigaia) = 16 - 6 = 10 bit
(b) motako aginduetan
luzera (kodea) + 2 * luzera (eragigaia) = 16
luzera (kodea) = 16 - 2 * 6 = 4 bit

Beraz, kodea luzera aldakorrekoa da, (a) formatoko aginduetan 10 bitekoa eta (b)-koetan 4koa.

B) (a) motako agindu-kopurua kalkulatu

Kontutan hartzekoa da Deskodetzaileak agindua aztertzen duenean ez dakiela zer motatakoa den eta horregatik (a) motako agindu-den lehenengo 4 bitak eta (b) motakoenak ezin direla berdinak izan.

Kontrol-Unitateak, enuntziatuak dioenez, (b) motako 15 agindu desberdin ezagutu eta onartzen ditu, beraz, lehenengo 4 bitak aztertuz, Kontrol-Unitateak 16 informazio posibleen artean 15 kasutan (b) motako agindua ulertuko du eta kasu bakar batean (a) motako agindutzat hartuko du.

Esandakoarekin ateratzen dugun ondorioa hauxe da: (a) motako agindu guztiek kodean lehenengo 4 bitak berdinak dituzte, bestela Kontrol-Unitateak (a) motako agindutzat hartuko bait lituzke.

A) atalean kalkulatu dugunez (a) motako aginduetan kodearen luzera 10 bitekoa da, baina 10 bit hauen artean lehenengo lauak finkoak dira, beraz, agindu desberdinak adierazteko konbina daitezkeen bitak 6 besterik ez dira eta honen ondorioz (b) motako 2 ber 6 kode desberdin ezagutu eta onartuko du Kontrol-Unitateak, hau da, (a) motako 64 agindu desberdin adieraz daitezke.

C) Adibidea.- Konputagailu honen aginduak honako hauek lirateke:

(b) motakoak (15 agindu)

0000	erag.1	erag.2
0001	erag.1	erag.2
0010	erag.1	erag.2
...	...	...
1110	erag.1	erag.2

(a) motakoak (64 agindu)

1111	000000	erag.1
1111	000001	erag.1
1111	000010	erag.1
...	...	...
1111	111111	erag.1

5.5.- AGINDUAK MIHIZTADURA-LENGOAIAZ

Aginduak makina-lengoaiaz nola adierazten diren aztertu dugu aurreko kapituluan, izan ere, gaur egun programak ez ditu inork makina-lengoaiaz idazten goimailako lengoaien bidez edo mihiztadura-lengoaiaz programatzen bait da.

Goimailako lengoaien erabilpenak programazioa errazten du, baina memoria gutxi edukitzen denean edota egikaritzapenak ahalik

eta arinena izan behar duenean mihiztadura-lengoaia erabiltzen da. Mihiztadura-lengoaiaren erabilpenak programaren nondik norakoak zehatz-mehatz ulertzea ahalbidetzen digu, hau da, guk mihiztadura-lengoaiaz programatzean erabakitzen dugu datuak non eta nola kokatzen diren, nola atzitzen diren eta objektu-programaren ideia zehatz bat hartzen dugu, mihiztadura-lengoaiaz idatzitako agindu bakoitza makina-lengoaiazko agindu bakar baten baliokidea bait da.

5.5.1.- SENTENTZIEN FORMATOA

Beraz, ikus dezagun mihiztadura-lengoaia osatzen duten agindu en formatoa. Mihiztadura-lengoaiaz aginduak karaktereen bidez idazten dira, ondoko formato honi jarraituz:

[ETIKETA] AGINDU-KODEA ERAGIGAI1[,ERAGIGAI2 ...] [;OHARRA]

non makoek dagozkien eremuak hautazkoak direla adierazten bait dute eta eremuen artean gutxienez txurigune bat bait dago.

Agindu-kodeak zer eragiketa burutu behar den adierazten du kode sinboliko batez. Jeneralean eragiketari dagokion izen ingelesaren laburdura bat da kode sinboliko hori (ikus 5.5 eta 5.6 tauletan batuketari dagozkion kode sinbolikoak VAX eta IBM-370ean) eta mihiztatzaileak makina-lengoaia bihurtuko du.

Eragigai bat baino gehiago baldin badago, hauek komen bidez bereizten dira. Eragigai bakoitza makina-lengoaiara itzuliko du mihiztatzaileak, kasu batzutan helbideratze-modu eta helbide bihurtuz (VAX-ean adibidez) eta bestetan helbide bakarrik (IBM-370ean adibidez). Eragigaien arazoa 5.5.2 pasartean sakonduko dugu.

Etiketa eta oharra izeneko eremuak hautazkoak dira, beraz, agindu batzutan agertuko dira eta bestetan ez.

Etiketak aginduari deitzeko izen sinboliko bat definitzen du, ez da makina-lengoaiara itzuliko, baina mihiztatzaileak izen honi helbide bat asignatzen dio eta izen hau eragigai batean aurkitzen duenean izenari asignatuta daukan helbidea ordezkatzen dio. 5.5.2 pasartean honetáz gehiago sakonduko dugu. Etiketak letra batez hasten dira eta luzera mugatua dute.

Oharrak iturburu-programan dauden aginduak argitzeko erabiltzen dira, horrela, programa idatzi ez duen baten batek irakurtzen badu, errazago ulertuko du eta programatzaile berak etorkizunean aldaketa bat egiteko begiratuko ditu. Goimailako lengoaietan irazkinak jartzea komenigarria bada ere, mihiztadura-lengoaiaz idatzitako programetan behar beharrezkoa bihurtzen da, bestela programa ulergaitza izango da. Dena dela, oharrez mihiztatzailea ez da batere urduratzen, hauek dokumentazio gisa gelditzen direlarik.

5.5.2.- ERAGIGAIAK MIHIZTADURA-LENGOAIAN

Mihiztadura-lengoaiazko aginduetan zehazten den eragigai bakoitza bi motatakoa izan daiteke: esplizitua ala inplizitua.

A) ERAGIGAI EXPLIZITUAK

Eragigaian, kode sinboliko batez, makina-lengoaian eragigaiari dagozkion kontzeptu guztiak zehazten dira. Horrela, helbideratze-modua, erregistro edota helbidea, desplazamendua, behar denean, etab. zehatz-mehatz adierazten dira, baina ez bitez bit, zenbakiak hamartarrean (edo nahi dugun oinarrian) eta helbideratze-moduak kode batzuen bidez baizik.

Horrela 5.3 eta 5.4 tauletan mihiztadura-lengoaiaz helbideratzeak nola adierazten diren azaltzen da (ikus MIHIZTADURA-LENGOAIAZ zutabea).

Eragigai esplizituak itzultzeko orduan mihiztatzailearen funtzioa kode-bihurketa bat da.

B) ERAGIGAI INPLIZITUAK

Eragigai hauetan ez dira helbideratze-modua eta helbidea zehatz-mehatz aipatzen, baizik eta adierazpen sinboliko bat jartzen bait da. Adierazpen sinboliko honi dagozkion helbideratze-modua eta helbidea mihiztatzaileak kalkulatzen du itzulpena egiten duenean.

Adierazpen sinbolikoa izen sinboliko batek edo izen sinbolikoen eta konstanteen konbinazio aritmetiko batek osatzen du. Programa batean eragigaietan agertzen diren izen sinboliko guztiek etiketa moduan definituta egon behar dute, bestela, mihiztatzaileak ezingo luke izen sinbolikoari dagokion berezko helbidea sortu.

Mihiztatzaileak jeneralean eragigai sinboliko guztiak helbideratze-modu bakar bat bihurtzen dituzte, horrela, VAX-ean eragigai sinboliko guztiak PC erregistrozko modu erlatibo bihurtzen dira eta IBM-370ean oinarri-erregistroaren eta desplazamenduaren ordez adierazpen sinboliko bat jar daiteke edozein eragigaietan. Berezko helbideak kalkulatzeko mihiztatzaileak sinbolo-taula eraikitzen du, non etiketa berri bat agertzen denean izen sinbolikoa eta dagokion helbidea definitzen bait dira.

Eragigai sinboliko edo esplizitu bat itzuli behar denean mihiztatzaileak sinbolo-taulatik hartzen ditu adierazpen bakoitzean agertzen diren izen sinbolikoei dagozkien helbideak.

C) ARIKETA EBATZIA

Ondoko eragigaiak VAX-eko makina-lengoaia bihurtu 5.7 irudia eta 5.4 taula kontsultatuz eta DATU-ari mihiztatzaileak 900 helbidea asignatu diola eta PC-ren edukina 800 dela jakinez:

R5, 10(R5), DATU, DATU+20, @#500, #1000

Ebazpena:

ZEHAZTURIKO ERAGIGAIA	ERAGIGAI ESPLIZITUA	MAKINA-LENGOAIAZKO MOTA	ERREG.	HELB./DESPL.
R6	R6	5	6	---
10(R5)	10(R5)	10	5	10 (despl)
DATU	100(PC)	10	15	100 (despl)
DATU+20	120(PC)	10	15	120 (despl)
@#500	@#500	9	15	500 (helb.)
#1000	#1000	8	15	1000 (datua)

5.5.3.- SASIAGINDUAK

Eragigaietan erabiltzen diren izen sinbolikoak etiketak definitzen dituztela esan dugu, baina etiketa bat agindu baten izena denez, ezin zaio datu bati izen sinboliko bat eman eta eragigai gehienetan datuak aipatu behar dira. Orduan, nola eman izen sinboliko bat datu bati? Horretarako eta beste zenbait helburu lortzeko sasiaginduak daude.

Makina-lengoaiara itzultzen ez den mihiztadura-lengoaiaz idatzitako aginduari SASIAGINDUA deitzen zaio. Beraz, mihiztadura-lengoaiaz idatzitako agindu bakoitza ez da beti baliokidea makina-lengoaiazko beste batekiko. Sasiaginduen kasuan ez.

Dena dela, makina-lengoaiaz sasiaginduek agindu arrunten egitura mantentzen dute (etiketa, kodea eta eragigaiak) baina mihiztatzaileak modu berezi batez tratatzen ditu. Badaude bi sasiagindu-mota, mota bakoitzak helburu desberdina duelarik eta mihiztatzaileak tratamendu berezia ematen dielarik. Alde batetik sasiagindu artekariak bereizten dira eta bestetik datuak definitzeko sasiaginduak.

A) DATUAK DEFINITZEKO SASIAGINDUAK

Makina-lengoaiaz programatu beharko bagenu aldagaiak ezingo genituzke definitu, memoriako helbideak erabiliko genituzke aldagai bezala. Mihiztadura-lengoaia gehienetan, berriz, aldagaiak defini daitezke. Horretarako, datuak definitzeko sasiaginduak erabiltzen dira, zeinetan etiketa bezala aldagaiaren izena idazten

bait da. Azken finean, sasiagindu batez aldagai bat definitzea da mihiztatzaileari esatea zenbait gelaska jauzi behar duela eta sinbolo-taulan izen bat sartu behar dela, zenbat gelaska jauzi behar diren eragigaian zehazten delarik.

Izan ere, sasiagindu hauen artean mihiztadura-lengoaia gehienetan bi motatako sasiagindu bereizten dira: datuak definitzeko bakarrik balio dutenak eta datuak definitzeko aparte balio bat sartzeko ahalmena eskaintzen digutenak. Bigarren mota honetakoetan mihiztatzaileak lekua gorde eta izena taulan sartzeaz gainera. Lehenbiziko kasuan aldagai bat definitu besterik ez dugu egiten, bigarrenean berriz, aldagaia definitzen dugu eta hasierako balioa duen aldagai bat definitzen dugu edo konstante bat, balioa programak aldatzen ez badu.

Balioa sartzeak arazo bat sortzen du: zer formatotan eta luzeratan adieraziko den memorian balioa. Horretarako mihiztadura-lengoaiek sasiagindu desberdinak eskeintzen dituzte, bakoitzak formato desberdin bat emanez.

SASIAGINDUA		DEFINITURIKO DATUA		
KODEA	ERAGIG.	LUZERA(byte)	BALIOA	FORMATOA
.BLKB	n	n	---	---
.BLKW	n	2*n	---	---
.BLKL	n	4*n	---	---
.BYTE	n	1	n	Koma
.WORD	n	2	n	finkozko
.BYTE	n	4	n	bitarra
.FLOAT	n	4	n	Koma Higik.
.DOUBLE	n	8	n	bitarra
.PACKED	n	behar dena	n	Ham.Paketat
.ASCII	/string/	behar dena	string	Karakterea

5.7 taula.- VAX-eko DATUAK DEFINITZEKO SASIAGINDUAK

Hona hemen sasiaginduen egitura mihiztadura-lengoaian:

a) Datuak definitzeko sasiaginduetan

ETIKETA SASIAGINDUA GELASKA-KOPURUA

non batzuetan sasiagindu posible bat baino gehiago bait dago, bat eragigaian zehazten dena byte-kopurua dela adierazteko, beste bat hitz-kopurua, beste bat hitz-bikoitzen kopurua, eta abar.

b) Datuak definitzeko eta balioa sartzeko sasiaginduei ondoko egitura dagokie:

ETIKETA SASIAGINDUA BALIOA

non sasiaginduaren arabera balioa formato eta luzera desberdinetan sartzen bait dira. Batzutan balio bat baino gehiago aipa daiteke, balioak komen bidez bereizten direlarik. Horrela, hasierako balioa duten taulak definitzea posiblea da. VAX konputagiluko datuak definitzeko sasiaginduak azaltzen dira 5.7 taulan.

B) SASIAGINDU ARTEKARIAK

Sasiagindu artekariek itzulpen-prozesua gidatzen dute, hau da, parametro batzuren bidez mihiztatzaileari zehazten diote programa baten itzulpena nola egin behar den.

Mihiztatzaileak sasiagindu artekariak aurkitzen dituenean sasiaginduetan azaldutako guztia itzulpen-prozesurako kontutan hartzen da, baina sasiagindu hauei objektu-programan ez zaie lekua gordetzen, sasiagindu definitzaileetan gertatzen zen bezala.

Sasiagindu desberdin asko daude funtzio desberdinak burutzeko, hala eta guztiz ere, ondoko hauek dira burutu ditzaketen funtzio garrantzitsuenak (parentesi artean VAX-eko konputagailuan funtzioari dagokion sasiagindu artekarien kodea):

- Iturburu-programaren hasiera (.TITLE)
- Iturburu-programaren amaiera (.END)
- Egikaritzen hasteko aginduaren helbidea (.ENTRY)
- Oinarri-erregistroaren funtzioa duten erregistro orokorrak
- Makroagindu baten definizioaren hasiera (.MACRO)
- Makroagindu baten definizioaren amaiera (.ENDM)
- Kanpo-aldagaien izen sinbolikoak (.EXTERNAL)
- Itzulpen-listatuaren aukera (.LIST,.NLIST)
- Erreferentzia gurutzatuen listatuaren aukera (.CROSS,.NOCROSS)
- Makroen hedapenaren listatuaren aukera (.SHOW,.NOSHOW)

Konputagailuen artean sasiagindu artekarien kodeak oso desberdinak badira ere, sasiagindu hauek betetzen dituzten funtzioak funtsean berdinak dira eta aipatutako funtzio gehienak adierazteko sasiaginduak konputagailu guztietan daude.

5.5.4.- PROGRAMAREN EGITURA

Goimailako lengoaiaz programak egitura finko bat jarraituz idatzi behar dira, mihiztadura-lengoaiaz berriz egitura hori malguagoa da, baina malguntasun horren truke erroreak sartzeko probabilitateak handitu egiten dira. Errore-gehikuntza hori gerta ez

dadin mihiztadura-lengoaiaz programatzen denean ondoko egitura honi jarraitzen saiatzen dira programatzaileak:

```
PROGRAMAREN HASIERAKO SASIAGINDUA
[BESTE SASIAGINDU ARTEKARI BATZU]
{              . . .
 DATUAK DEFINITZEKO SASIAGINDUAK
               . . .                 }
EGIKARITZAPENAREN HASIERAKO SASIAGINDUA
{              . . .
           AGINDUAK
               . . .                 }
ITURBURU-PROGRAMAREN AMAIERAKO SASIAGINDUA
```

Non makoen artean dagoena hautazkoa bait da eta giltzek barruan dagoena behin baino gehiagotan ager daitekeela adierazten bait dute.

PASCAL lengoaiarekin konparatuz, lehenengo sasiagindua PROGRAM sententziaren baliokidea izango genuke, datuak definitzeko sasiaginduak erazagupen-atalarena, egikaritzapenaren hasierako sasiagindua hasierako BEGIN sententziarena eta azken sasiagindua amaierako puntuarena.

Ez da nahastu behar iturburu-programaren amaiera eta egikaritzapenaren amaiera, zeren eta goimailako lengoaietan askotan baliokideak izan arren, batek itzulpen-denboran eragiten bait du eta besteak berriz egikaritzapen-denboran, mihiztadura-lengoaian lehenengoa sasiagindu artekaria eta bestea agindua direlarik.

Batzutan ez da azaldutako egitura errespetatzen eta orduan datuak definitzeko sasiaginduak, definitutako memoriako tokiak egikaritzeko agindutzat har ez daitezen, arreta handia jarri behar da. Horretarako, aginduen artean agertzen diren sasiagindu definitzaileak baldintza gabeko jauzi agindu bat izan behar dute aurretik, hemen ondoan ikusten denez:

```
        agindua
          ...
        agindua
        jauzi   GERO
        sasiag.
          ...
        sasiag.
GERO    agindua
          ...
```

Jauzi agindua baldintza gabekoa da eta eragigaian sasiaginduen ondoko lehendabiziko aginduaren helbidea zehazten da, gehienetan modu sinbolikoz.

6.- S A R R E R A / I R T E E R A K O O S A G A I A K

6.1.- SARRERA

6.2.- PERIFERIKO ETA EUSKARRIEN SAILKAPENA

6.3.- SARRERAKO PERIFERIKOEN DESKRIPZIOA

6.4.- IRTEERAKO PERIFERIKOEN DESKRIPZIOA

6.5.- BITARTEKO MEMORIAK

6.6.- UNITATEEN ARTEKO KONPARAKETA

6.7.- PERIFERIKO/KONPUTAGAILU KONEXIOA

6.8.- ETENAK

6.1.- SARRERA

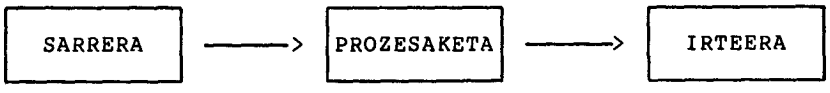
Hogeigarren mendearen erdira arte (1950) zientzia eta teknikazko datu guztiak liburutan bildurik zeuden. Edonork informazioa lortu nahi bazuen liburutara jo behar zuen, hau da, paperezko euskarritara. Desabantailarik nagusienak bi ziren: Liburutegien tamaina izugarria batetik, edifizio osoak behar bait ziren liburuak gordetzeko eta bestetik edozein informazio bilatzeko denbora luzea igaro beharra liburutegiko fitxategiak arakatzen.

Euskarri modernoek, magnetikoek esan nahi da, Gutenberg aroari bizitza laburtu diote, beste aro berri bati hasiera emanez: Teleinformatikaren aroari. Ehun orrialde dituen liburu baten hitz guztiak gaur egun orrialde erdiko pelikula magnetikoan gorde daitezke.

Gainera liburuaren edozein kapitulu bilatu nahi bagenu, adibidez, aurkibidean gaia bilatzea eta gero orrialdea ikustea, gutxi gorabehera hamar segundo beharko luke pertsonak eta denbora berean erdi mailako ordenadore batek hirurehun bilaketa egingo lituzke, liburua disko magnetikoan jasota balego (atzipen-denbora=30ms). Gainera paperezko euskarrietan behin bakarrik idatz dezakegu; ez dago paperean berridazterik. Euskarri magnetikoetan mila aldiz berridatz dezakegu, noski, aurretik zegoen informazioa galduz eta azken idatzi duguna kontserbatuz.

6.1.1.- EUSKARRI ETA PERIFERIKOAK

Gogora ditzagun ordenadorearen hiru atal nagusiak (ikus 6.1 irudia)



6.1 irudia.- DATU-PROZESAKETAREN ESKEMA

Prozesaketa, CPU (Central Procesing Unit) Prozesurako Unitate Zentralean gidatzen da, aurreko kapituluan aztertu dugun bezala. Oraingo honetan azalduko duguna SARRERA eta IRTEERAri dagokiena izango da.

Informazioaren sarrera eta irteera gailu batzuren bidez lortzen da; aparatu hauei Periferikoak esaten zaie. Izenak argi azaltzen digu, periferikoak CPUa inguratzen duten gailuak direla.

Periferikoen funtzioa informazioa CPUra sartu edota handik ateratzea da eta horretarako euskarriak behar ditugu. Euskarriak beraz, informazioa ordenadoreari sartu, bertatik atera edo gorde-

tzeko erabiltzen dira. Material desberdinetakoak izan daitezke: papera, material magnetikoa, kablea, etab.

Laburpen gisa eskema hau izan behar duzue gogoan:

PERIFERIKOAK	EUSKARRIAK
Informazioa irakurtzen edo idazten duen APARATUA	Informazioa errepresentatzeko erabiltzen den MATERIALEA

6.1 taulan zenbait periferiko eta beraiei dagokien euskarria jaso ditugu.

PERIFERIKOAK	EUSKARRIAK
TXARTEL-IRAKURGAILUA	TXARTEL ZULATUA
ZINTA-IRAKURGAILUA	ZINTA ZULATUA
INPRIMAGAILUA	PAPER JARRAIA
ZINTA-UNITATEA	ZINTA MAGNETIKOA
DISKO-UNITATEA	DISKO MAGNETIKOA
DISKET-UNITATEA	DISKETA edo DISKO MALGUA
DANBOR-UNITATEA	DANBOR MAGNETIKOA

6.1 taula.- PERIFERIKO ETA EUSKARRIAK

6.2.- PERIFERIKO ETA EUSKARRIEN SAILKAPENA. EZAUGARRIAK

6.2.1.- SAILKAPENA

Periferiko eta euskarriek ezaugarri desberdinak eduki ditzakete. Hona hemen berezitasun batzu.

a) On-line eta off-line periferikoak.

Unitate Zentralaren inguruan dauden periferikoak ordenadorearekin konektaturik (kable baten bidez) baldin badaude, on line edo lineako periferikoak direla esaten dugu. Ondorioz, periferikoak informazioa bidal/jaso dezake ordenadorera/tik. Konektaturik ez baldin badago off-line edo lineaz kanpo dagoela esaten da, hau da, ez dezake inola ere informazioa zuzenki bidal/jaso ordenadorera/tik. Enpresa askotan langileen sarrera/irteerako fitxagailuak lineaz kanpo (off-line) daude. Periferiko honek langile guztien sarrera eta irteerako lan orduak zinta edo txingola zulatuan erregistratzen ditu. Hurrengo egunean, txingola-irakurgailuan sartzen dira (on-line) beronek CPUari bidal diezaion soldatak eta primak kalkulatzeko.

b) Sarrerako eta irteerako unitateak eta bitarteko memoriak.

Sailkapen hau garrantzitsuena da. Lehen ikusi dugun eskemari jarraituz, periferikoak hiru motatakoak izan daitezke:

- Sarrerako periferikoak: Informazioa ordenadorera bidaltzeko bakarrik balio dutenak.

- Irteerako periferikoak: Ordenadoretik ateratzen den informazioa euskarritan idazteko balio dutenak.

- Bitarteko memoriak: Bai sarrerarako eta baita irteerarako ere erabiltzen direnak (periferiko magnetiko gehienak). Sarrerako eta irteerako periferikoen erabiltzaileak eta konputagailuak beren artean komunikatzeko balio dute, bitarteko memoriak berriz memoria nagusiari laguntzeko, hau da, informazioa memoria nagusian beharrezkoa ez denean bitarteko memorian gordetzen da.

c) Periferiko helbideragarriak eta ez-helbideragarriak.

Periferiko batzuk zuzenean atzi dezakete edozein informazio, informazio horren helbidea emanez gero; hau da, giltza emanez gero ordenadoreak helbide bihurtzen du eta informazio hori aurkitu egingo du, bere aurreko informazioa irakurri gabe.

Disko magnetikoa mota honetako periferikoa da eta helbideragarria dela esaten da.

Zinta zulatua, zinta magnetikoa eta txartel zulatua aldiz, periferiko ez-helbideragarriak dira, zeren eta informazio bat eskuratzeko euskarri honetan aurretik dagoen informazio guztia irakurri behar bait dute.

d) Euskarri berrerabilgarriak eta ez-berrerabilgarriak

Periferiko batzuren euskarriek behin erabiliz gero ez dute beste informaziorik errepresentatzeko balio, bota edo jaso egin behar dira. Txartel zulatua eta txingola zulatua, adibidez, horrelakoak dira, hau da, ez dira berrerabilgarriak. Periferiko magnetikoen euskarriak aldiz, berrerabil ditzakegu informazio berria gainean grabaturik. Noski, lehen grabaturik zegoen informazioa ezabatua geratzen da.

6.2.2.- EZAUGARRIAK

Mota bereko euskarri edota periferikoen artean desberditasun nabarmenak daude, batzuk garrantzi handikoak. Periferiko edo euskarri bat aztertzen dugunean kontutan hartu behar diren ezaugarri garrantzitsuenak pasarte honetan azalduko ditugu.

a) Euskarriaren edukiera

Euskarrian adieraz daitekeen informazio-kantitatea adierazten du edukierak. Euskarriaren arabera karakteretan, K-tan (1 K=1024 karaktere) edo M-tan (1M=1024 K = 1.000.000 karaktere). Oso ezaugarri garrantzitsua da bitarteko memorietan batez ere.

b) Transferentzi abiadura

Ezaugarri hau periferikoari dagokio eta memoria eta periferikoen arteko informazio trukearen abiadura adierazten du. Bitarteko memorietan transferentzi abiadura desberdina izaten da irakurtzen denean eta grabatzen denean, azken hau motelagoa gertatzen delarik. Karaktere/segundutan edo batez ere K/segundu unitateetan neurtzen da. Dispositibo guztien abiadura Memoria Nagusiko abiadurarekin konparatuz gero izugarri motelak gertatzen dira.

c) Atzipen-denbora

Informazio zehatz bat euskarri batean bilatzeko dispositiboak zenbat denbora behar duen adierazten digu ezaugarri honek. Euskarri ez-helbideragarrietan ez du zentzu handirik ezaugarri hau erabiltzeak, zeren eta aurrean dagoen informazio guztia irakurri behar bait da eta beraz kokaturik dagoen lekuaren arabera oso desberdina izango da.

Horren ondorioz, euskarri helbideragarrietan erabiltzen da eta helbide bat emanda helbide horretan dagoena atzitzeko behar den denborari atzipen-denbora esaten zaio. Atzipen-denbora, jeneralean, milisegundutan neurtzen da.

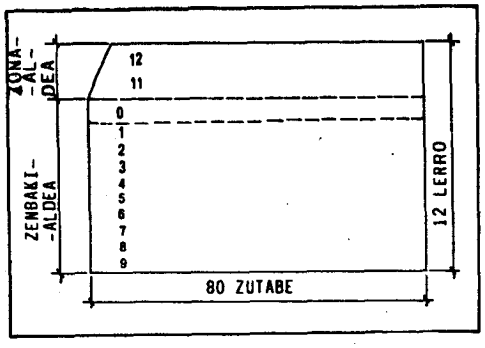
Badaude beste ezaugarri batzu garrantzi txikiagoa dutenak. Horietako bat prezioa da eta, kontzeptu bezala horrenbesteko garrantzirik ez badu ere, askotan kontutan hartzekoa da. Prezioa pezeta/karakteretan neurtzen da.

6.3.- SARRERAKO PERIFERIKOEN DESKRIPZIOA

6.3.1.- TXARTEL ZULATUA (fitxa)

A) EUSKARRIA

Lehenengo kapituluan esan den bezala, Hollerith Doktoreak 1890. urteko Estatu Batuetako zentsuko datuak biltzeko asmatu zuen. Gehien erabiltzen zen sarrerako euskarria dugu hau baina gaur egun gutxi erabiltzen da, gero eta baztertuago geratuz doa.



6.2 irudia.- TXARTELAREN EGITURA

Txartel bakoitzak 8Ø zutabe eta 12 lerro ditu. Letrak, zenbakiak eta abar, hau da karaktereak, zulotxoak eginez adierazten dira. Beraz, byte bakoitza 12 bitek osatzen dute eta txartel baten edukiera 8Ø karakterekoa da.

B) HOLLERITH KODEA

Karaktereak adierazteko, zutabe bakoitzak dituen 12 posizioak erabiltzen ditu eta posizio hauek bi zatitan bana genitzake: fitxaren oinarritik hasita, lehenengo bederatzi posizioei zenbaki-aldea deritze eta 9,8,7,6,5,4,3,2,1 eta Ø balioak hartzen dituzte. Beste hirurak Ø,11,12 zulaketak dira eta zona-aldea esaten zaie.

Zenbakiak (Ø-9) zenbaki-aldeko zulotxoen bidez adierazten dira. Adibidez, 5 zenbakia, zenbaki-aldeko boskarren posizioan zulotxoa eginez adierazi ohi da.

Letrak bi zulotxoren bidez adierazten dira, bata zona-aldean eta bestea zenbaki-aldean eginez. Beste karaktereak bat, bi, hiru zulotxo eginez adierazten dira. Kontutan eduki behar da karaktere bera adierazteko ez dela inoiz bi konbinazio desberdin erabiltzen ez eta alderantziz ere.

C) PERIFERIKOA

Txartelen informazioa sartzeko TXARTEL-IRAKURGAILUA izeneko dispositiboa dugu. Dispositibo honetan txartelak, sarrerako gordelekuan jarriz gero, kanal batetik pasatzen dira eta kanal horretan kokaturik dauden irakurtzeko buru batzuren bidez zulaketak detektatzen dira, azkenik txartelak irteerako gordelekuan gelditzen direlarik.

Karakterea	Hollerith kodea	Karakterea	Hollerith kodea	Karakterea	Hollerith kodea
¢	12-8-2	:	8-2	0	11-6
.	12-8-3	#	8-3	P	11-7
<	12-8-4	@	8-4	Q	11-8
(	12-8-5	'	8-5	R	11-9
+	12-8-6	=	8-6	S	0-2
|	12-8-7	"	8-7	T	0-3
&	12		12-0	U	0-4
!	11-8-2	A	12-1	V	0-5
$	11-8-3	B	12-2	W	0-6
*	11-8-4	C	12-3	X	0-7
)	11-8-5	D	12-4	Y	0-8
;	11-8-6	E	12-5	Z	0-9
¬	11-8-7	F	12-6	0	0
	11	G	12-7	1	1
/	0-1	H	12-8	2	2
,	0-8-3	I	12-9	3	3
%	0-8-4	J	11-1	4	4
-	0-8-5	K	11-2	5	5
>	0-8-6	L	11-3	6	6
?	0-8-7	M	11-4	7	7
		N	11-5	8	8
				9	9

6.2 taula.- HOLLERITH KODEA

Txartel-irakurgailuaren ezaugarririk garrantzitsuena abiadura da. Irakurgailuaren abiadura 200-2000 txartel/minutu tartekoa izan ohi da, irakurtzeko teknikaren arabera. Irakurgailu moteletan irakurtzeko buruak eskuila mekanikoak dira, lasterretan, berriz, zelula fotoelektrikoen bidez egiten da irakurketa.

Txartelak zulatzeko dispositibo bat dago, txartel-zulagailua baina gehienetan dispositibo hau lineaz kanpo egoten da. Txartel-zulagailuaren bidez transkipzio lana egiten da, hau da, paperean idatzitakoa edo konputagailuak zuzenean ulertzen ez duen euskarri batean dagoen informazioa euskarri eta kode egoki batean jartzea. Txartel-zulagailuan teklatu baten bidez informazioa sartzen da eta emaitza bezala informazio hori daukaten fitxak lortzen ditugu.

6.3.2.- ZINTA ZULATUA

A) EUSKARRIA

Zinta zulatua, fitxa zulatuaren moduko zulotxo biribilak dituen paperezko zinta da. Zulaketa-kanal deituriko lerro batzuri jarraituz egiten dira zulaketak.

Zinta zulatuetan informazioa adierazteko zutabe bakoitzean karaktere bat gordetzen da. Karaktere bat gordetzeko 5,6,7 edo 8 zulaketa posibleen bidez egiten da, dispositiboaren arabera. Zulaketa posible hauek direla eta, zinta zulatuak 5,6,7 eta 8 kanalekoak direla esaten da, byte bakoitza 5,6,7 edo 8 bitek osatzen dutelarik.

Zinta zulatua gero eta gutxiago erabiltzen da, gaùr egun ia ia baztertuta dagoelarik.

B) PERIFERIKOA

Zinta zulatuaren informazioa konputagailuan sartzeko ZINTA-IRAKURGAILUA erabiltzen da. Dispositibo honen abiadura 100-200 karaktere/segundu tartekoa izan ohi da eta irakurtzeko metodo mekanikoa edo optikoa erabiltzen dira optikoa lasterragoa baino garestiagoa gertatuz.

Transkipzioa egiteko zinta-zulagailua deritzon dispositiboa erabiltzen da. Dispositibo hau lineaz kanpo dago eta txartel-zulagailu antzekoa da.

6.3.3.- TEKLATUA

Idazmakina elektroniko bat konputagailuarekin zuzenean lotuta dela esan daiteke. Teklatua idazmakina arrunten teklez eta beste tekla berezi batzuz horniturik dago.

Teklatuan ez dago oso argi euskarria zein den. Teklatu batzutan karaktere bakoitza sakatu ondoren karaktere hori konputagailura bidaltzen da; bestetan berriz teklatuan memoria txiki bat dago, buffer edo indagetzaile deritzana, non bete arte karaktereak biltzen bait dira eta bufferra betetzen denean automatikoki bertan dagoen informazio guztia konputagailura bidaltzen da. Lehenengo kasuan euskarria linea da bigarrenean berriz bufferra.

Teklatuaren abiadura oso handia da baina sakatzen duen pertsonaren abiadurak oso motela bihurtzen du.

Teklatu oso erabilia da gaur egun. Ezaugarri garrantzitsuenak honako hauek ditugu:

- ez du aurreko transkripzioa behar (transkripzioa linean egiten delako) eta, ondorioz, informazioa sartzen duenak sar dezake zuzenean konputagailuan.

- berrerabilgarria da eta gaur egun nahiko merkea.

- transkripzioa linean egiteagatik konputagailura informazioa bidaltzen den bakoitzean CPU-k bere lana utzi behar du eragiketa hau kontrolatzeko, beraz, CPUren errendimendua jeisten da. Hau eragozpen handia da, batez ere oso informazio luzeak sartu behar direnean eta kasu horietan ez da komenigarria informazioa teklatuaz sartzea.

Teklatu batek eta pantaila batek terminal bat osatzen dute, baina gehienetan elkarrekin badaude ere bi dispositibo dira, bata sarrerakoa (teklatua) eta irteerakoa bestea (pantaila).

6.3.4.- IRAKURKETA OPTIKO ETA MAGNETIKORAKO DOKUMENTUAK.

Oinarrizko dokumentuak euskarritara pasatu behar dira datu horiek prozesatuak izan daitezen, bestela ezinezkoa izango bait litzateke ordenadore bidez tratatzea. Pauso hau transkripzioa deitzen da eta eman beharrak baditu bere eragozpenak.

- Gehigarrizko kostua, zeren eta prozesu osoaren kostua handitu egiten bait du.

- Zenbait errore sortzea, eskuzko lanak beti dira osoki kontrolatzen zailak.

- Dokumentuak sortu diren unetik emaitzen lorpenerako denbora luzatzea.

Arazo guztiok gainditzeko teknika berriak sortzen ari dira dokumentuak bete ahala, hau da, transkripzio berririk egiteke, ordenadorean zuzenean sartzea posible izan dadin. Noski, dokumentu horiek irakurtzeko, Sarrerako Unitatea behar-beharrezkoa da.

A) EUSKARRIA

Dokumentu-euskarrietan erabiltzen diren karaktereak bi motatakoak izan ohi dira:

1.- Karaktere magnetikoak.
2.- Karaktere optikoak.

Lehenengoak tinta berezi batzuren bidez marrazten dira; karaktere hauek magnetizatu egiten dira irakurgailura pasa baino lehen, irakurgailuak ezagutu ditzan. Karaktere optikoak aldiz, kontrastez bereizten dira, hau da, paperaren zuritasuna eta karaktereen beltztasuna bereziz. Printzipio hau erabiliz, marka sentikorrezko txartelak eta dokumentuak sortu dira.

Karaktere magnetikozko dokumentu gisan taloiak, txekeak eta antzekoak ditugu. Banketxeek beren artean trukatzen dituztenez, lehenbiziko banketxean beheko lerroan karaktere magnetiko batzu idazten dira geroko transkripzioak beharrezkoak izan ez daitezen.

Marka sentikorrezko txartelak, txartel zulatuen antzekoak dira, hau da, zutabe eta lerrotan banaturik daude. Adibide garbi-garbia kinielak ditugu; kinielak marka sentikorrezko txartelak dira.

B) DISPOSITIBOA

Karaktere magnetikoen irakurgailuak:

Irakurgailu hauek ez dituzte karaktereak beraien itxuragatik identifikatzen, baizik eta karaktere bakoitza osatzen duten zutikako lerroen eta beraien arteko distantziak identifikatuz. Irakurketatik sortutako pultsuak irakurgailuak daukan taularekin alderatzen ditu bertako ikurrekin erlazionatzeko. Alderaketan errore-marjina bat onartzen da, baina beti muga batzuren barruan. Muga hori gainditu gero makinak ez du karakterea onartuko. Karaktere magnetikoen irakurketa lerroz-lerro egiten da.

Karaktere optikoen irakurgailuak:

Karaktere hauek begi-bistaz irakurgarriak dira. Makinak irakur ditzan lente-sistema eta argi indartsu bat erabiltzen du, belzgune eta txuriguneak puntu txikitako zonak bihurtu arte. Puntu hauek pultsu elektriko bihurtzen dira eta hauek dira makinak ezagutzen dituenak eta berak dituen ereduekin alderatzen dituenak.

Dena dela idazmakinaz idatzitako orriak irakurtzeko dispositibo fidagarriak egon arte ezin da transkripziorik ez dagoenik esan zeren eta pasarte honetan ikusitako dokumentutan transkripzioa beharrezkoa bait da, baina transkripzioa informazioa sartzen denean egiten da eta konputagailura sartzeko momentuan.

6.3.5.- BESTEAK

Badago beste sarrerako unitaterik, baina erabilpen berezitarako direlako edo oraindik guztiz garatu gabe daudelako ez dira oso ezagunak eta gutxi erabiltzen dira. Lehenengo kasuan grafikoak

sartzeko taula grafikoa dugu (arkitektura eta injinerutzako lanetan erabilzen da) eta bigarren kasuan ahotsezko sarrera (abiadura handiz garatzen ari da).

6.4.- IRTEERAKO PERIFERIKOEN DESKRIPZIOA

6.4.1.- INPRIMAGAILUAK

Irteerako dispositibo arruntenak eta erabilienak dira, lortzen duten informazioa iraunkorra eta zuzenean ulergarria bait da.

A) EUSKARRIA

Inprimagailu batez informazioa paperean idazten da, Inprimagailuek erabilitako paperak jeneralean etengabekoa izan behar du paper jarraia deitutakoa. Honen bi ertzetan zulaketa bana bait dago, inprimagailuak papera lineaz linea mugi dezan. Paper jarrai standard bat dago, pijama papera deitutakoa.

Pijama papereko orri bakoitzean 66 edo 72 lerro idatz daitezke lerro bakoitzean, gehienez 132 karaktere sartzen direlarik. Beraz, edukiera teorikoa 72*132 litzateke, baina ulergarria izan dadin itxura minimo bat behar da eta horretarako zurigune asko uzten dira hain zuzen ere.

B) PERIFERIKOA

Paperean idazten duen dispositiboa inprimagailua deitzen da. Inprimagailuen ezaugarriengatik hiru mota bereizten dira: karakterekako inprimagailuak , lerrokakoak eta bereziak.

a) Karakterekako inprimagailuak

Karakterekako inprimagailuek idazmakina elektriko baten tankeran idazten dituzte karaktereak, banan banan eta bata bestearen atzetik. Idazmakinetan bezala idazgailu bat dago, burua deritzona, eta buruaren eta paperaren artean zinta tintaduna dago. Bi norantzetan idazteko gai izatea da idazmakinarekin daukan desberdintasun handiena.

Buru-mota desberdinak daude: orratz-matrizea, margarita, etab., mota bakoitzak kalitate eta abiadura desberdina eskaintzen duelarik. Dena dela, batezbesteko abiadura 40-200 karaktere/segundu tartekoa izaten da.

b) Lerrokako inprimagailuak

Konputagailu handi samarrentzat karakterekako inprimagailuak motelak gertatzen dira eta lerrokako inprimagailuak erabiltzen dituzte.

Lerrokako inprimagailuek kolpe bakoitzean lerro osoa idazten dute, hau da 132 karaktere batera idatz dezakete. Gaur egungo mota honetako inprimagailuak katea baten bidez idazten dute eta kateazko inprimagailuak ere deitzen dira. Sei bider edo gehiago errepikaturik dauden karaktereek osatzen dute katea eta behar den karakterea posiziotik pasatzen denean, dagokion mailuari eragiten zaio, zinta tintadunaz karakterea idatziz. Katea oso abiadura handiz mugitzen da eta lerro guztia batera idazten dela ematen du. Lortzen duten abiadura 500-3000 lerro/minutu tartekoa izaten da eta kalitate aldetik karakterekakoak baino txarragoa eskaintzen dute.

c) Inprimagailu bereziak

Oso konputagailu handitarako eta batez ere zenbait erabilpenetarako aurreko inprimagailuak motelak dira, teknologia mekanikoa erabiltzen bait dute.

Gaur egun badaude teknologia mekanikoa erabiltzen ez duten beste inprimagailu-mota batzu. Inprimagailu laster hauek tinta magnetizatuaren bidez edo laser izpien bidez funtzionatzen dute, baina oso oso garestiak dira, inprimategietan besterik erabiltzen ez direlarik.

6.4.2.- PANTAILAK

Gaur egun oso zabalduak daude periferiko hauek eta, esan dugun bezala, teklatu bati atxikirik egon ohi dira.

Pantailari dagokion euskarria zein den ez dago oso argi. Hala eta guztiz ere gehienetan buffer batez horniturik egoten da.

Bi pantaila-mota bereizten dira: karakterekakoak eta grafikoak. Lehenengotan edukiera desberdinetakoak daude, baina oso arruntak dira 80 karaktere/lerro eta 24 lerro dauzkatenak. Pantaila grafikoetan marrazkiak adieraz daitezke eta edukinak ez du garrantzi handirik, bereizmena (puntu-kopurua/azalera-unitate) eta askoz inportanteagoa izanik.

Abiadura aldetik, pantaila aztertu genuenean esandakoa, azkarra da baina, informazioak zenbait denboraz finko geratu behar duenez, motela kontsideratzen da.

Oso hedatuak daude pantailak eta hauen erabilpenik garrantzitsuenak sistemaren mezuak ikustaraztea eta emaitza laburrak adieraztea dira, teklatuaren bidez sakaturikoa pantailaratzearen gain.

6.4.3.- BESTEAK

Pantailak eta inprimagailuak dira, diferentzia handiz, gehien erabiltzen diren irteera-unitateak. Hala ere, badaude erabilpen zehatzetarako erabiltzen diren beste batzu.

A) MIKROFILM-UNITATEA

Idatzi behar diren txostenak oso oso luzeak direnean, papere-an idaztea luzea eta garestia gertatzen da eta gainera idatzitako datu bat aurkitzea gehienetan astuna bihurtzen da. Kasu hauetan oso egokia da informazioa mikrofilmean grabatzea, metodo fotografikoak erabiltzen dituen mikrofilm-grabagailu batez. Mikrofilmean informazioa papereko egituraz grabatzen da, baina behar den lekua 100 aldiz txikiagoa da, beraz erosoago maneiatuko da eta bilaketa errazagoa izango da. Eragozpen bi daude: alde batetik oso garestia dela unitatea eta bestetik lorturiko informazioa ulergaitza dela zuzenean, lineaz kanpoko dispositibo bat erabili beharko delarik.

B) MARRAZLEA (PLOTERRA)

Unitate hau grafikoak eta planoak paperean egiteko erabiltzen da. Beso mugikor baten muturrean arkatz bat dago eta beso hau mugituz marrazkiak lortzen dira.

Oso erabilia da diseinu-aplikaziotan, hau da injinerutzan, arkitekturan eta antzeko erabilpenetan planoak lortzeko. Sarrerako dispositibo bat elkarturik du batzutan, grafikoak sartzeko TAULA GRAFIKOA.

C) ENTZUNGAILUAK

Oraindik garatzen ari den dispositibo honetaz emaitzak hitzez eta hotsez kanpora daitezke.

6.5.- BITARTEKO MEMORIAK

Bitarteko memorien artean ez-helbideragarriak eta helbideragarriak bereizten dira. Lehenengoen artean zinta magnetikoa eta kaseta dira erabilienak, helbideragarrien artean diskoa batez ere, disketa eta danbor magnetikoa. Banan banan azter ditzagun.

6.5.1.- ZINTA MAGNETIKOA

A) EUSKARRIA

Euskarri hau bi geruzaz osaturik dago; azpiko geruza, poliesterrezkoa da eta hauskaitza. Goiko geruza magnetizagarria den burdin oxidoz konposatua. Bi material horien proportzioa asko aldatzen da marka batetik bestera.

Gehien erabiltzen diren zintek 600, 1200 eta 2400 oinbeteko (0.3048 m) luzera dute eta 13 mm-ko zabalera.

Informazioa geruza magnetizatuan erregistratzen da. Azala zenbait pistaz osaturik dago, pista hauek zinta zulatuan kanaleek betetzen zuten funtzio bera dutelarik. Karaktere bakoitza zeharkako lerro batean erregistratzen da eta ez pista batean zehar.

Zintak 7 edo 9 pista ditu, nahiz eta, informazioa 6 edo 8 pistatan egon, zeren azkeneko pista kontrolerako erabiltzen bait da. Posizio batean 7 pistako zinta batean 2**6 = 64 karaktere desberdin errepresenta daitezke, bederatzikoa erabiltzen badugu, berriz, 2**8 = 256 karaktere. Azken hauek dira gehien erabiltzen direnak zeren karaktereak zortzikoteka errepresentatzen bait dira ordenadore gehienetan.

Grabazio-dentsitatea: luzera-unitatean sartzen den informazio-kantitatea da eta BPI unitatean neurtzen da. BPI izena ingelesezko "bit per inch" esapidearen laburdura da, euskaraz bit hazbeteko esango genukeena eta azken finean hazbeteko karaktere-kopurua adierazten duena. Merkatuan dauden zintak 800, 1600 eta 3200 BPI-koak dira.

Teorikoki hutsunerik utzi gabe graba dezakegu informazioa zintan, baina praktikan, hau ezinezkoa da, zeren informazioa zintan grabatu aurretik ordenadorearen Memorian gorderik egon behar bait du eta honen ondorioz, grabazio-eragiketa batean gehienez ere Memoria Nagusiaren edukiera graba dezakegu, hau da informazioa bloke bat osaturik grabatzen da. Blokeak bata bestearen atzetik grabatzen dira tartean hutsuneak utziz. Hutsune horietan (IRG, Inter-Record-Gap, esaten zaie), ez da informaziorik grabatzen eta zintaren abio eta geldialdiengatik sortzen dira. Grabatzeko edo irakurtzeko zinta abiadura konstantez higierazi behar da eta bloke biren grabaketen artean dagoen geldialdiek eta geroko abioek behar duten bidea da IRG-a. Normalean hutsuneok 3/4 hazbeteko luzera dute. Ikusten dugunez, zinta grabatuaren zati bat alferrik galduta dago.

Blokeak handiagotuz zinta gehiago probetxatzen dugu, gainera blokeak txikiak direnean abio eta geldiketa gehiago egin behar dira zinta irakurtzeko eta, ondorioz, irakurketa-abiadura txikiagoa izango da. Dakusagunez, bloke luzeak erabiltzea da egokiena.

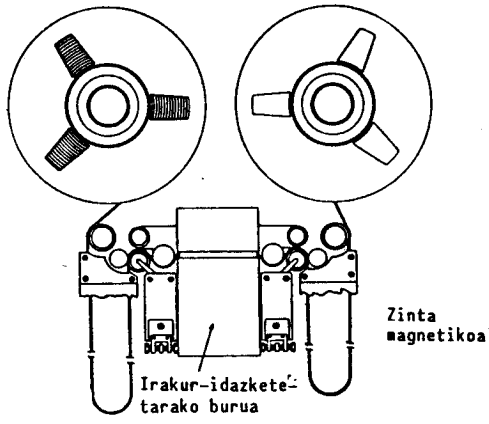
Bestetik, badaude beste arrazoi batzu bloke txikiak erabiltzea gomendatzen dutenak; adibidez, Memoria Nagusia mugatua dela eta ez dezakegula blokea osorik gordetzeko erabil eta honen ondorioz, tratamendu-denbora luzatu egingo litzatekela. Praktikan, blokeek 200 karakteretik 5000 karaktererako tartean aurkitzen dira. Ikusten dugunez, zintaren edukiera eta abiadura erreala %70ean txiki daitezke.

B) ZINTA MAGNETIKOEN IRAKUR/IDAZKETARAKO GAILUAK

Lehen ikusi ditugun euskarriek, txartel eta zinta zulatuek, bi aparatu behar zituzten, bata irakurketarako eta bestea zulaketarako. Euskarri magnetikoekin ez da horrelakorik gertatzen, dispositibo berak irakurketa eta grabaketa egiten bait ditu.

Elektroimanez osaturiko buru bat edo gehiago duten dispositiboen bidez egiten dira bi eragiketak.

Grabaketarako, buruari korrontea ematen zaio, euskarri magnetizagarriaren (zinta, disko, danbor edo orri magnetikoa) puntu batean eremu magnetiko bat norantza batean edo bestean induzituz; bi norantza horiek ematen dute bitaren definizioa.



6.3 irudia.- ZINTA MAGNETIKOAREN UNITATEA

Irakurketan, lehen grabaturiko geruza irakurketarako buruaren aurretik pasatzen da korronte induzitu bat sortuz. Jakina, korrontearen norantza, imantazio-puntuaren norantzak baldintzapeturik dago.

Zinta magnetikorako erabiltzen diren unitateek pista bakoitzeko buru bat dute, bai irakurketarako bai grabaketarako, ira-

kur-grabazio prozesua denbora berean pista guztietan egiten delarik.

Blokea irakurtzen edo grabatzen hasten denean, karaktereen irakur/idazketa abiadura, hau da, At transferentzi-abiadura, Aa arraste-abiaduraren eta d zintaren grabazio-dentsitatearen funtzio zuzenean dago. Beraz :

$$At = Aa * d$$

Erabiltzen diren unitateek segundu bakoitzean 40-200 K karaktere (edo K zortzikote, zinta 9 pistakoa baldin bada) transferitzen dituzte. (K karaktere = 1024 karaktere).

Txartel zulatuaren eta zinta zulatuaren ordez azken urte hauetan, ekipo askok, lineaz kanpo, zinta magnetikoa erabiltzen dute. Grabaketa, esan bezala lineaz kanpo (off-line) egiten da eta bitarteko memoria sarrerako memoria bihurtzen da.

6.5.2.- KASETA

Euskarri hauek ezagutzen ditugun musikazko kaseten antzekoak dira itxuraz. Informazioa pistetan grabatzen da zintetan egiten denaren antzera.

Zinta magnetikoarekin alderatuz, euskarri honek askoz ere luzera txikiagoa du eta irakurketa-denbora aldiz, luzeagoa. Ordenadore txikietan (ordenadore pertsonaletan) asko erabiltzen da euskarri hau informazioa gordetzeko.

6.5.3.- DISKO MAGNETIKOA

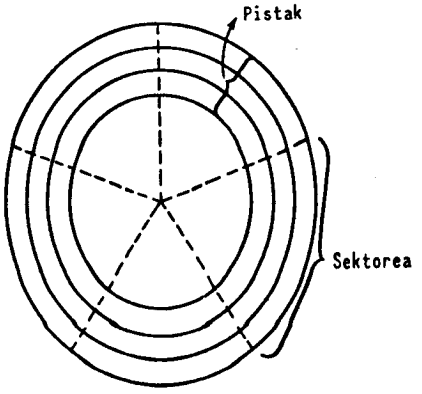
A) EUSKARRIA

Disko magnetikoa, bi azalak material magnetizagarriz estaliak dituen metalezko (aluminiozkoa gehienetan) disko bat da. Datuak bitez bit, bata bestearen ondoren, gordetzen dira, puntuak zirkunferentzia zentrukideak diren pistetan magnetizatuz.

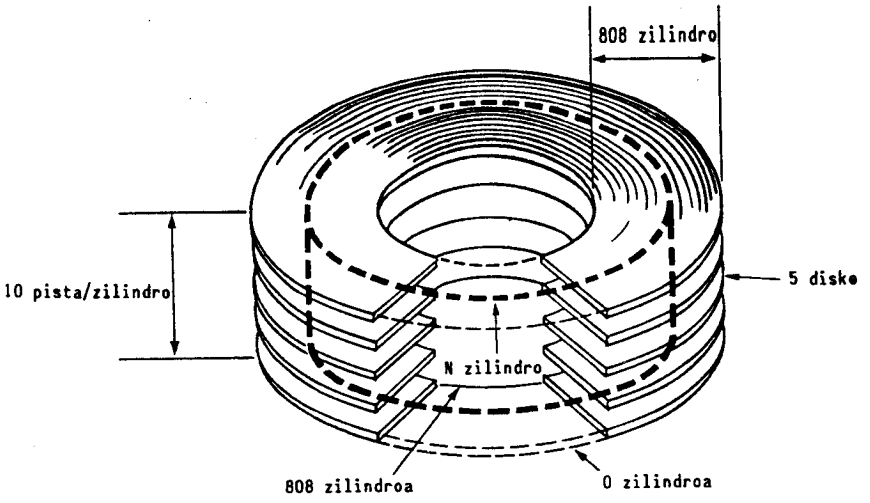
Kanpoko itxura kontutan edukiz bi motatako diskoak daude: disko bakunak (single) eta disko metatuak (disk-pack), hau da, ardatz baten inguruan zenbait disko bakun loturik dituztenak.

B) DISPOSITIBOA

Disko-Unitatea, funtsean diskoa sartzen den ardatz batez, eta irakur/idazketarako buruez osaturik dago.



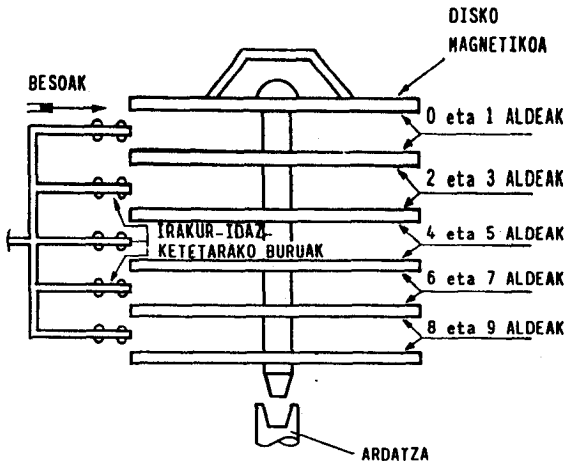
6.4 irudia.- DISKO BAKUNA



6.5 irudia.- DISKO METATUAK

Diskoaren biraketa-abiadura minutuko 1000 bira baino handiagokoa izaten da. Irakur/idazketarako buruak beso batzuri loturik daude, disko-azalaren gainean erradialki higitzen direlarik. Buru hauek irakur/idazketarako pista detektatzean, besoak geldiarazten dituzte, informazioa irakurri edo grabatuz. Normalean, honenbeste informazio-azal hainbeste buru izaten dituzte disko metatuek, hau da buru bat diskoaren alde bakoitzean.

Batzutan disko handietan buru bat baino gehiago jartzen da alde bakoitzean; orduan buru bakoitzari koroa bat osatzen duten pista-multzoa atxekitzen zaio.



6.6 irudia.- DISKO-UNITATEA

Irakur/idazketaren prozesuan, buru-multzotik bakar bat aktibatzen da. Beso guztiak euskarri bati loturik daude eta ondorioz, higidura bera dute, guztiek posizio erradiala denbora berean lortuz. Buruak pareko pistetan, diskoaren azal guztietan, kokatzen dira.

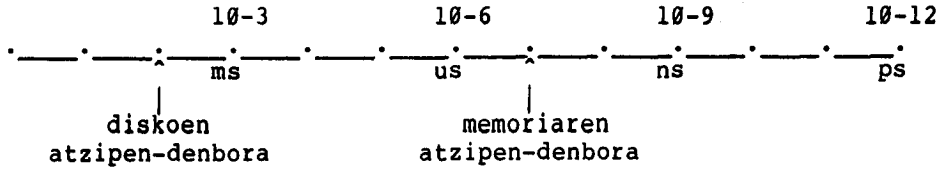
Irakur/idazketako posizioak, hau da, buruek posizio batean irakur/idatz ditzaketen pista guztiek zilindroa osatzen dute. Beste era batera adieraziz, zilindroak buru adina pista ditu eta alde bakoitzean buru bakarra dagoenean, zenbaki berdineko alde guztietako pistek osatzen dute zilindroa.

Bi motatako disko metatuak daude: finkoak, dispositibotik atera ezin daitezkeenak, eta higigarriak, erabili ahala unitatean sartu eta handik atera daitezkeenak.

Diskoa memoria helbideragarria (atzipen zuzenekoa) da. Informazioa non kokatua dagoen, hau da helbidea, jakinez gero zuzen-zuzenean atzi dezakegu beste aurreko informazioak irakurri gabe. Beste euskarrietan, zinta zulatu eta magnetikoan, aurreko informazioa irakurri egin behar zen helbidera iritsi baino lehen.

Informazio baten helbidea jakinez gero, atzipen-denbora oso laburra da, 20-50 ms (1 ms = 10-3 s.) tartekoa; baina kontutan eduki behar dugu ferrita eta erdieroalezko memoriak 100.000 aldiz laterragoak direla (300 ns = 3 * 10-7 s)

Atzipen-denbora funtsean hiru zatitan bana dezakegu: kokapen-denbora, buruen selekziorako denbora eta biraketa-denbora.



6.7 irudia.- DISKOEN ATZIPEN DENBORAREN ESKALA

Kokapen-denbora, buruek, pisten gainean jartzeko behar dutena da. Denbora hau oso aldakorra da; buruak bilatzen ari diren helbidearen (pistaren) gainean badaude ez dute higitu beharrik, baina bilatzen ari diren helbidea (pista) diskoaren beste muturrean baldin badago kokapen-denbora maximoa izango da. Hiru denboren artean pisurik handiena duena kokapen-denbora da.

Buruen selekziorako denbora, besteen konparazioan arras laburra da. Buru-multzoa kokatuta dagoenean, informazioaren gainean dagoen burua aktibatzeko behar den denbora da.

Biraketa-denbora, burua aukeratu den unetik irakurri behar den informazioa azpitik pasatu arte pasatzen den denbora-tartea da. Batezbeste, bira-erdia izaten da, zeren diskoak pistetan helbideratzen bait dira eta informazioaren hasiera pistaren hasiera da. Gaur egun, badaude beste disko-mota batzu, sektoreen bidez helbideratzen direnak, hau da helbidea zehazteko zilindroa, pista, sektorea eta erregistroa aipatu behar dira.

Burua irakurri behar duen informazioaren gainean jarri ondoren, datuok Memoriara transferitzen dira. Transferentzi abiadura, biraketak, abiadurak eta grabaketa-dentsitateak baldintzatzen dute.

Transferentzi abiadura konstantea da, hau da, erradioa aldagarria izanik, grabaketa-dentsitateak desberdina behar du izan pista batetik bestera eta, logigoki, handitu egiten da barruko pistetan.

Laburpen gisa, gaurko diskoen ezaugarri nagusien balioak emango ditugu:

- Disko metatuen edukiera: 7-100 M karaktere.
- Kokapen-denbora: 30-75 ms (batezbestekoa).
- Batezbesteko biraketa-denbora: 8-12.5 ms.
- Transferentzi abiadura: 100-800 K karaktere/s.

6.5.4.- DISKETA EDO DISKO MALGUA. (floppy-disk)

Euskarri honek arrakasta izugarria lortu du azken urteotan. Erdi-mailako ordenadoreetan informazioa lineaz kanpo (off-line)

gordetzeko erabiltzen da eta ordenadore txikitan, hau da, mikroordenadoretan, diskoaren ordez, bitarteko memoria bezala (on-line).

Euskarri honek, musika-diskoen antza handia du, baina material magnetikoa azal batez estalirik aurkitzen da beti. Neurri askotako disketak daude merkatuan baina gehien erabiltzen direnak hauexek dira:

Disket standarda --> 8" (zortzi hazbetekoa)
Disket txikia -----> 5 1/4" (bost eta laurden hazbetekoa)

Disketa euskarri helbideragarria da, pista bakoitza sektoretan zatitua egonik. Modelo askotan sektoreak 256 byte ditu. Edukieraren arabera bi motatako disketak daude:

- Dentsitate sinplea dutenak; disket bakoitzean 500.000 karaktere sartzen direlarik (8"koa denean).

- Dentsitate bikoitzekoak; disket bakoitzean 1.000.000 karaktere sartzen dira (8"koa denean).

Diskoekin alderatuz disketak edukiera askoz ere txikiagoa dute eta atzipen-denbora luzeagoa.

6.5.5.- DANBOR MAGNETIKOA

Danbor magnetikoa, informazio-euskarri gisa, zilindro batez osaturik dago, goialdean material magnetizagarria duelarik. Disko magnetikoan ikusi dugun bezala, hemen ere datuak bata bestearen jarraian gordetzen dira, bit baten ondoren beste bit bat gordez. Bitak zirkunferentzia zentrukidetan magnetizatzen dira, pistak osatuz.

Danbor magnetikoaren Unitate Periferikoa, funtsean eragingailuaz eta diskoaren antzeko irakur/idazketetarako buruez osaturik dago. Eragingailuak biraketa-higidura eragiten dio danborrari. Danborrak ez dira inoiz higigarriak, finkoak baizik. Jeneralean, modelo guztiek pista bakoitzeko irakur/idazketetarako buru bat dute.

Dispositibo honen abantaila nagusia hauxe da: buruen kokapen-denbora desagertu egiten dela osorik eta denbora hau zen luzeena diskoetan. Danborretan atzipen-denborak bi osagai ditu: buruen selekziorako denbora (hutsaren hurrengoa) eta biraketa-denbora, bira-erdia batezbeste. Normalean, danborren biraketa-abiadura diskoarenaren aldean handia izaten da eta, ondorioz, biraketa-denbora diskoetan baino laburragoa.

Danborrek, diskoek baino edukiera txikiagoa eta atzipen-denbora laburragoak dituzte. Danborren batezbesteko ezaugarri nagusiak honela laburtuko genituzke:

- Edukiera: 0,2-4 milioi karaktere.
- Atzipen-denbora: 3-10 milisegundo.
- Transferentzi abiadura: 200-1.500 K karaktere/s.

6.6.- UNITATEEN ARTEKO KONPARAKETA

Pasarte honetan 6.3, 6.4 eta 6.5 taulen bidez Unitateen ezaugarriak eta erabilpenak aztertu eta konparatuko ditugu. Konparaketa hau osoa ez da, zeren eta adibidez Sarrera dispositiboak Bitarteko memoriakin alderatzea zentzu handirik ez bait du. Horregatik sarrerako dispositiboak, irteerakoak eta bitarteko memoriak aztertu dira bakoitza bere aldetik.

SARRERA-UNITATEA	EZAUGARRIAK				ERABILPENAK
	KONPUT-MOTA	TRANS-KRIPZ.	BERRERA-BILGARR.	ABIA-DURA	
TXARTEL ZULATUA	HANDIAK	BAI	EZ	MOTELA	-Programak eta datuak -gero eta gutxiago
TEKLATUA	GUZTIAK	EZ	BAI	MOTELA	-Programak, mezuak -Kontrol-sententziak
DISKETA	HANDIAK	BAI	BAI	AZKARRA	-Datuak, prog.luzeak
BESTEAK	GUTXI	--	--	MOTELA	-Bereziak

6.3 taula.- SARRERA-UNITATEEN ARTEKO KONPARAKETA

IRTEERA-UNITATEA	EZAUGARRIAK					ERABILPENAK
	KONPUT-MOTA	ULER-GARRIA	BERRERA-BILGARR.	IRAUN KORRA	ABIA-DURA	
INPRIMA-GAILUA	GUZTIAK	BAI	EZ	BAI	MOTELA	-Txostenak
PANTAILA	GUZTIAK	BAI	BAI	EZ	MOTELA	-Mezuak
MIKROFILM	HANDIAK	EZ	EZ	BAI	AZKARRA	-Txosten luzeak
DISKETA	HANDIAK	EZ	BAI	BAI	AZKARRA	-Inform.-trukea
BESTEAK	GUTXI	--	--	--	MOTELA	-Bereziak

6.4 taula.- IRTEERA-UNITATEEN ARTEKO KONPARAKETA

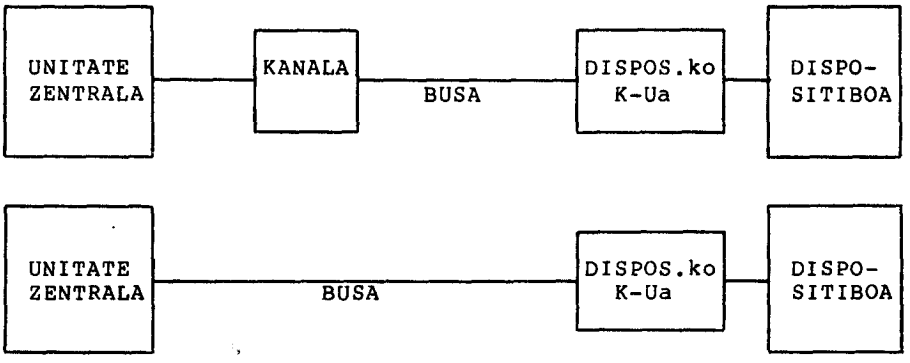
Disket eta zinta magnetikoak, deskribapenean bitarteko memoriak aztertu baditugu ere, konputagailu handietan sarrera-unitate bezala erabiltzen dira (OFF-LINE grabagailu bat egonez) eta beste konputagailuetara eramango den informazioa ateratzeko (konputagailuen arteko trukerako) irteera-unitate bezala.

BITARTEKO MEMORIA	EZAUGARRIAK KONPUT-MOTA	HELBI-DERAG.	EDU-KIERA	PRE-ZIOA	ATZIPEN DENBORA	FITXATEGI-MOTA
ZINTA M.	GEHIEN.	EZ	HAND.	MERKEA	---	-Sekuentzialak
DISKOA	GEHIEN.	BAI	HAND.	GAREST	ERDIXKA	-Zuzenekoak -Sekuentz.motzak
DANBOR M.	HANDIAK	BAI	TXIK.	GAREST	TXIKIA	-Zuzeneko berezia
KASETA	MIKROAK	EZ	TXIK.	MERKEA	---	-Sekuentzialak
DISKETA	MIKROAK	BAI	TXIK.	MERKEA	HANDIA	-Zuzenekoak

6.5 taula.- BITARTEKO MEMORIEN ARTEKO KONPARAKETA

6.7.- PERIFERIKO/KONPUTAGAILU KONEXIOA

Periferikoen ezaugarriak aztertu ondoren periferiko horiek konputagailuarekin nola lotzen diren azalduko dugu pasarte honetan. Konexioa bus baten bidez egiten da, baina Unitate Zentralaren eta dispositibo baten artean busaz aparte dispositiboaren Kontrol-Unitatea dago eta batzutan Sarrera/Irteerako prozesatzailea edo kanala (ikus 6.8 irudia). Beraz, konexioa egiteko 3 unitate edo osagai bereizten dira: Sarrera/Irteerako busa, dispositiboaren kontrol-unitatea eta kanala. Azter ditzagun banan banan.



6.8 irudia.- DISPOSITIBO ETA KANALAREN ARTEKO KONEXIO-MOTAK

6.7.1.- SARRERA/IRTEERAKO BUSAK

Bus hauetan barrena Memoriatik periferikorako eta alderantzizko informazio-trukeak egiten dira. Busean zehar mugitzen den informazioa datuek eta kontrol-informazioek osatzen dute.

Norantzaren arabera bi bus-mota bereizten dira: norantza bakarrekoak eta bi norantzatakoak bitarteko memoriak azken hauekin loturik egon behar duelarik.

Zabaleraren arabera ere bi bus-mota bereizten dira, seriezko busak eta paralelozkoak. Lehenengotan informazioa bitez bit igortzen da eta bigarrenetan berriz karakterez karaktere. Seriezko busetan zirkuitu bakar bat aski da paralelozkotan karaktere bat adierazteko bit adina zirkuitu gutxienez. Jeneralean dispositibo azkarrak paralelozko busekin konektatzen dira.

6.7.2.- DISPOSITIBOAREN KONTROL-UNITATEA

Osagai hau dispositibo eta konputagailuaren arteko interfacea da, hau da, bien arteko elkargunea. Dispositiboaren kontrol-unitatearen funtzioen artean ondoko hauek dira garrantzitsuenak:

- Kode-bihurketa.- Konputagailuek karaktereak adierazteko ASCII edo EBCDIC kodea erabiltzen duten bitartean, dispositibo batzuk beste koderen bat erabil dezakete. Kasu honetan irteeran ASCII kodetik beste kodera itzuli behar da eta sarreran alderantziz. Adibidez, txartel-irakurgailuak informazioa Hollerith kodean irakurtzen du eta memoriara iristerako EBCDIC edo ASCII kodean egon behar du. Bihurketa hori dispositiboaren kontrol-unitateak burutzen du.

- Datuen egiaztapena.- Erroreak detektatzeko eta batzutan zuzentzeko dispositiboaren kontrol-unitatea erabiltzen da. Adibidez, paretasun-bita aztertuz, bitarteko memoria batean gertaturiko aldaketa bat detekta dezake.

- Kontrol-seinaleen detekzioa unitate zentraletik datozenean edo seinale horien sarrera-dispositiboan baldintza bereziren bat gertatu denean. Adibidez, errore bat detektatzen duenean dispositiboaren kontrol-unitateak seinale bat bidaliko dio unitate zentralari.

- Bufferren gestioa (buffering-a).- Bufferra beterik dagoenean, hau hustu eta zegoen informazioa konputagailura edo dispositibora bidaltzen du, sarrera edo irteera denaren arabera.

6.7.3.- KANALAK

Kanalak sarrera/irteerako eragiketak burutzeko diseinaturiko dispositiboak dira eta horregatik sarrera/irteerarako prozesatzaileak ere deitzen zaie. Beraz, kanalez horniturikо konputagailu batean sarrera/irteerako agindu bat egikaritu behar denean ez du CPU-k egikaritzen kanal batek baizik.

Kanalek ondoko bi ezaugarri oso garrantzitsu hauek dituzte:

- Sarrera/irteerarako eragiketak burutzen dituzten bitartean CPU-k beste aginduren batzu egikari ditzake.

- Memoria Nagusia atzi dezakete.

Horren ondorioz, programa batean CPU-k Sarrera/Irteerako agindu bat aurkitzen duenean kanalari ematen dio berak egikari dezan eta kanalak egikaritzen duen bitartean CPU-k beste aginduak exekutatzen segitzen du, horrela CPUren errendimendua hobetuz.

A) KANALEN ZERGATIA

Sarrera/Irteerarako prozesatzaileak eta ez beste eragiketetarakoak jartzeak erakartzen du sarrera/irteerako aginduak egikaritzeko behar den denbora eta beste aginduek behar dutenaren arteko diferentziak. Sarrera/Irteerako aginduak 500 edo 1000 aldiz motelagoak dira besteak baino eta kanalik ez badago CPU-k denbora gehiena sarrera/irteerako aginduak burutzen emango du.

CPU betidanik osagai garestientzat hartu da, gaur egun hori hain argi ez badago ere eta eragiketa aritmetiko eta logikoak burutzeko diseinaturik dago, beraz, denborarik gehiena sarrera/irteerako aginduak burutzen ari baldin bada CPUa ez da behar bezala probetxatzen.

B) KANALAREN FUNTZIONAMENDUA

Kanalek, sarrera/irteerako aginduak egikaritzeko, ondoko betebehar hauek dituzte:

- Elkarhizketa behar da kanalaren eta CPUren artean, CPUk agindu bat egikaritzen hasi behar dela abisatzeko eta kanalak CPUri noiz bukatzen duen oharterazteko. Elkarhizketa hau kontrol-seinale batzuren (etenak) eta memoriako hitz batzuren bidez egiten da.

- Kanalak, prozesatzaile bat izateagatik programa bat egikaritu behar du, kanal-programa, beraz, memoriako atal batean sarrera/irteerako agindu bakoitzari dagokion kanal-programa kokatu beharko da. Kanal-programaren aginduari kanal-komandoa deitzen zaio eta komando bakoitzean beste zehaztapenen artean eragiketa (irakurketa, idazketa, bilaketa), memoriako helbidea eta dispositiboko helbidea adierazten dira.

Betebehar hauek modu desberdinetan ebazten dira konputagailuaren arabera, baina funtsean kanal baten funtzionamendua honako hau dugu:

CPU-k sarrera/irteerako agindu bat aurkitzen duenean eten batez kanal bati agindua bete behar duela ohartarazten dio, kanalak etena jasotzen duenean memoriako helbide finko batean duen programa egikaritzen du eta bukatzen duenean eten batez CPUri eragiketa burutzea bukatu duela ohartarazten dio. CPU-k, beste aginduak exekutatzen ari zen bitartean etena jasotzen duenean, etenaz jabetzen da errutina berezi bat egikarituz eta ari zenarekin segituko du. Dena dela, ezbeharren bat gerta liteke eta orduan eten baten bidez kanalak CPUri ohartaraziko lioke, CPU-k beharrezko eginkizunak bete ditzan .

Sarrerako agindu bat aurkitzen duenean CPU-k eragiketa kanalari bidaliz gero jeneralean ezin du hurrengo aginduak egikaritzen jarraitu zeren eta kanala irakurtzen ari den datua beharrezkoa bait da. Horregatik, kanalak dituzten konputagailuak kanalei probetxua ateratzeko multiprogramazioa behar dute (7.gaian begiratu).

C) KANAL-MOTAK

Kanalek zenbait ezaugarri desberdin eduki dezakete eta ezaugarri hauek bereizten dituzte.

Loturik daukaten dispositibo kopuruari begira dispositibo bakarrarekin loturiko kanalak eta bat baino gehiagorekin loturik daudenak bereizten dira, azken hauek arruntenak direlarik.

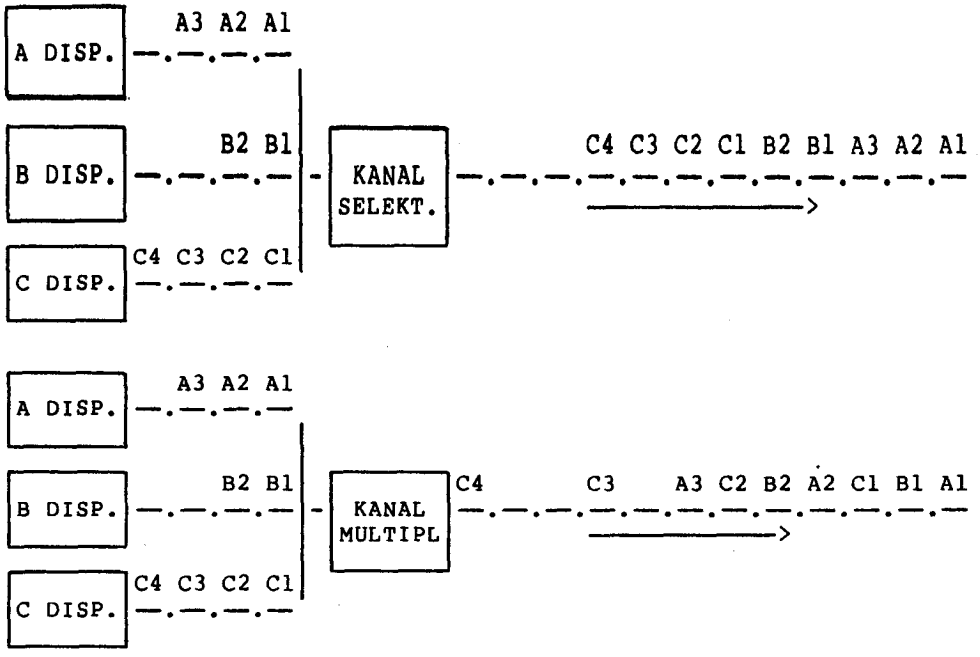
Zenbait dispositiborekin loturiko kanalen artean, funtzionamenduaren arabera, kanal selektoreak eta multiplexoreak bereizten dira:

Kanal selektoreak.- Loturiko dispositiboren batek kontrola hartzen duenean, hau da kanala dispositiboren bati zerbitzatzen hasten zaionean, dispositibo horrek kanala monopolizatzen du, beste dispositibo guztiei zuzenduriko eragiketek, martxan dagoena bukatu arte, itxoin beharko dutelarik (ikus 6.9 irudia).

Kanal multiplexoreak.- Loturiko dispositiboen artean banatzen da kanalaren denbora, dispositibo bakoitzari tarte bat egokituz. Denbora-tartea hitz bat bidaltzeko denbora baldin bada, hizkako multiplexorea dugu, aldiz, tartean erregistro edo bloke bat bidal badaiteke, blokekako multiplexorea (ikus 6.9 irudia).

Burutu behar diren eginkizunen arabera selektore edo multiplexore gisa lan egin dezaketen kanalei hibridoak deitzen zaie. Kanal hibridoak oso egokiak dira baina garestiak gertatzen dira.

Eskuarki dispositibo lasterrak kanal selektoreekin edo blokekako multiplexoreekin lotzen dira, dispositibo motelak berriz hizkako kanalekin.



6.9 irudia.- KANAL-MOTEN ARTEKO DIFERENTZIAK

D) PARALELISMO FUNTZIONALA

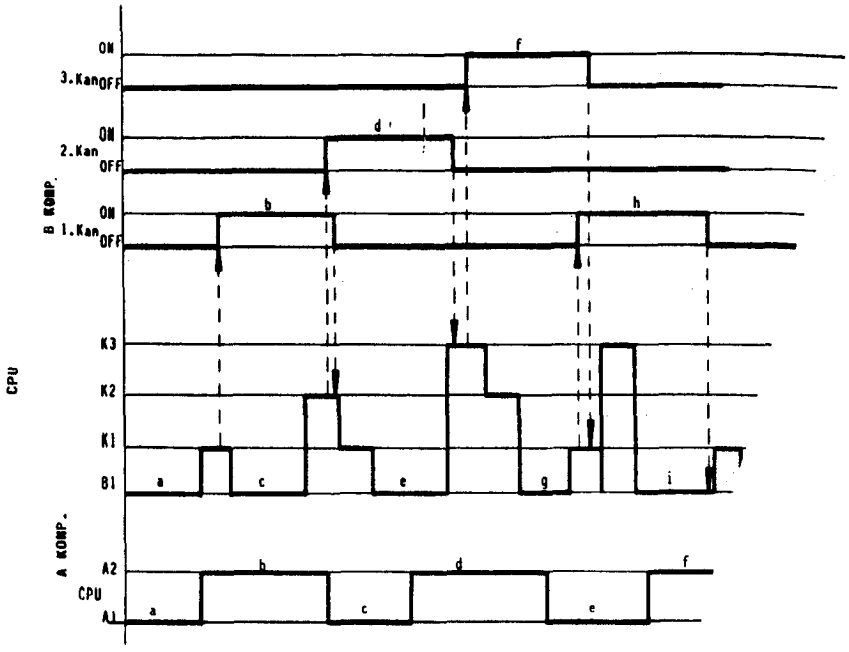
Programa zehatz bat kanalik gabèko konputagailu batean (A) edo hiru kanalez hornituriko beste batean (B) egikaritzean gertatzen diren desberdintasunak azter daitezke 6.10 irudian.

Argi ikusten da programa bera kanalez hornituriko konputagailuan askoz arinago burutzen dela. Kanalik ez badago, CPU bi egoeratan bereizten da: agindu arruntak exekutatzen (A1) eta sarrera/irteerako aginduak egikaritzen (A2). Hiru kanalez horniturik badago CPU agindu arruntak egikaritzen ihardun daiteke (B1) edo kanalen bat zerbitzatzen (K1,K2,K3), hau da lau egoeratan kanal bakoitzari bi egoera dagozkio, geldirik (OFF) ala lanean (ON).

6.8.- ETENAK

Etenak kontrol-seinale baten ondorioz gertatutako exekuzio-sekuentziaren aldaketak dira. Jeneralean, kanalek sortzen dituzte etenak, beraz, sarrera/irteerako eragiketek sortzen dituzte eten gehienak beste eten motak ere badauden arren; makina-erroreek eta programa-erroreek sortutakoak adibidez.

Itzuleradun jauzietan bezala, eten bat gertatzen denean martxan dagoen programaren egikaritzapena geldiarazten da eta etena-



6.10 irudia.- KANALEN PARALELISMO FUNTZIONALA

ren tratamendua egiten duen errutinara jauzi egiten da, tratamendu hori bukatu ondoren martxan zegoen programara itzuliz. Programa etenera itzultzean, etena gertatzean zegoen egoera berdinean egon behar du prozesuak, hau da, PC erregistroak, baldintz erregistroak eta erregistro orokorrak zeuzkaten edukinak berreskuratu beharko dituzte.

Itzuleradun jauziak sinkronoak dira, etenak berriz asinkronoak; hauxe da haien arteko desberdintasunik handiena. Programa bat datu berdinekin egikaritzen den guztietan jauzi berberak gertatzen dira, baina eten desberdinak gerta daitezke. Horregatik jauziak sinkronoak eta etenak asinkronoak direla esaten da. Beraz, programa bat idazten denean datu zehatz batzutarako zenbat jauzi gertatuko diren kalkula daiteke, zenbat etenaldi gertatuko diren kalkulatzea berriz, ezinezkoa da, etenak programatik kanpo sortzen bait dira.

6.8.1.- ETENEN TRATAMENDUAK

Eten bat gertatzen denean, kasu gehienetan, burutzen ari zena utzi eta etena tratatzen du CPU-k. Ondoko urrats hauek dira tratamendua osatzen dutenak:

- PC eta baldintz erregistroen edukina gordetzea. Batzutan erregistro orokor batzuren edukina ere gorde egiten da.

- Etena tratatzeko errutinaren helbidea PC-an sartzea. Honen ondorioz errutinara jauzi egingo da.

- Errutina burutzea.

- Errutina burutuz gero, lehenengo urratsean gordetako balioak, dagozkien erregistrotara atzera sartzea. Honen ondorioz programa etenera itzultzen da.

Gerta liteke etenaldi baten tratamendua burutzen den bitartean beste eten bat sortzea. Horrelako kasuetan konputagailuaren arabera irtenbide desberdinak daude; funtsean bi hauek:

1) Lehengo etenaren tratamendua baztertu, eten berria onartuz eta beste edozein eten bezala tratatuz. Honetarako, erregistroen edukinak gordetzeko, pila erabili behar da, bestela, memoriako helbide finkoetan gordez, eten bati dagozkion balioek aurrekoari dagozkionak ezabatzen dituzte, eten zaharraren itzulera galerazten delarik (ikus 6.11 irudian etenaldi-sekuentzia baten ondorioak).

2) Eten baten tratamendua bukatu arte CPU ez da beste etenez arduratzen, sortzen diren etenek itxaron behar dutelarik. CPU-k etenak galerazten dituela esaten da. Kasu honetan eten batzuren informazioa galtzeko arriskua da eragozpen handiena. Hala ere, CPU-k nolabait jakin beharko du egikaritzen ari den agindua programari ala eten baten tratamenduari dagokion. Horretarako baldintz erregistroaren bit bat erabiliko litzateke 1 egoerak programa egikaritzen ari dela eta Ø egoerak etenak tratatzen ari dela adieraziz. Bit honi maskara deitzen zaio eta Ø egoeran dagoenean etenak maskaraturik daudela esaten da. Dena dela, bi kasuetan ez dugu inoiz lehentasuna aipatu.

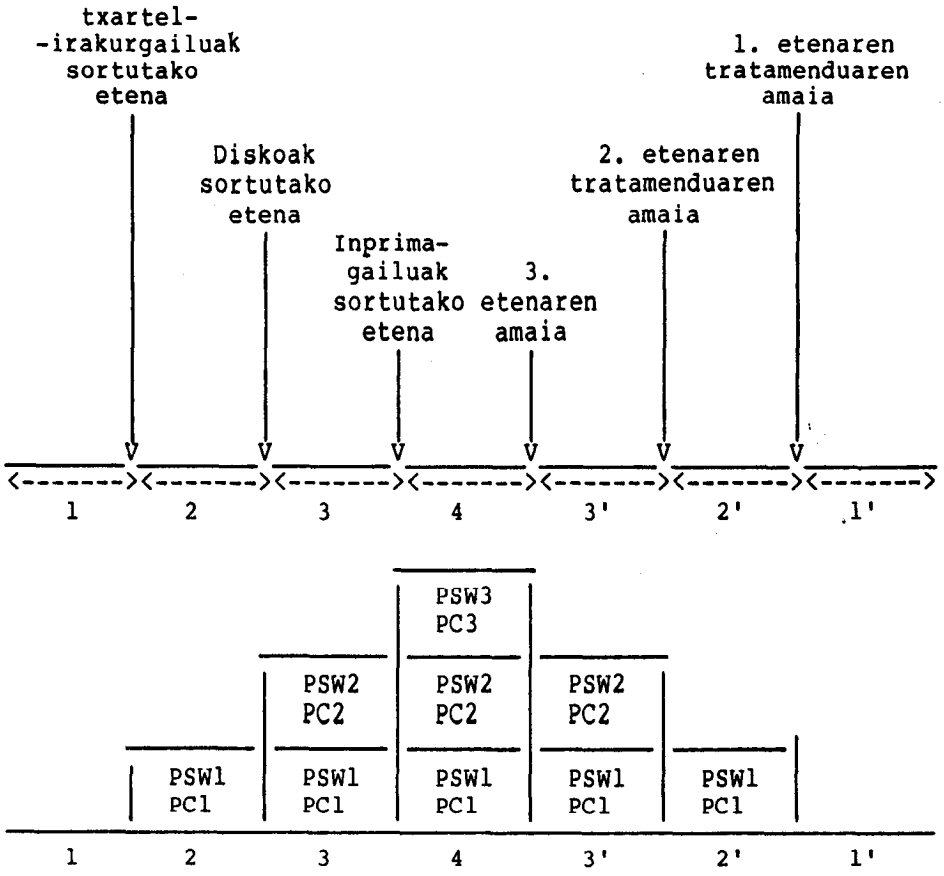
Eten guztiek ez dute lehentasun edo premia berdina eta eten bat tratatzen denean lehentasun handiagoko beste bat sortzen bada eten hori tratatzea logikoa dirudi; eten berriak premia edo lehentasun txikiagoa baldin badu berriz, itxoin eraztea izango litzateke egokiena.

Horregatik, errealitatean funtsezko ebazpenei lehentasuna gehitzen zaie tratamendua konplexuagotuz.

6.8.2.- ADIBIDEAK

A) IBM-370 KONPUTAGAILUAN

IBM-370ean etenak bost motatan sailkatzen dira: Sarrera/irteeratakoak, makina-erroreengatikoak, programa-erroreengatikoak, kanpo-etenak eta gainbegiraleari egindako deiengatikoak. Eten bat



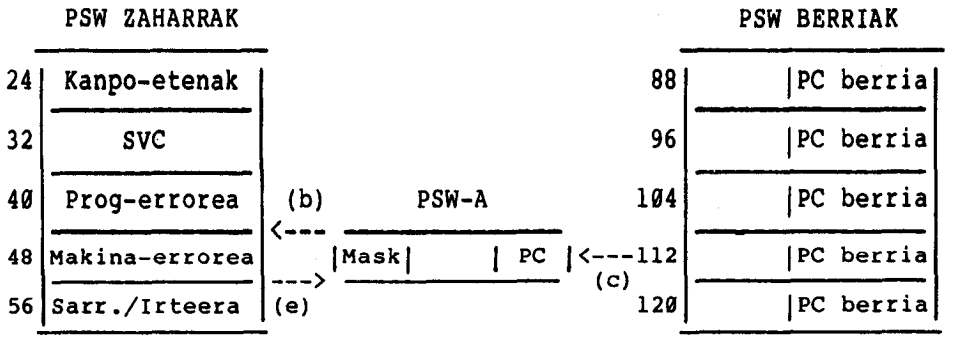
6.11 irudia.- ETENEN KONTSEKUTIBOEN ONDORIOAK

gertatzen denean mota horretako beste bat tratatzen ari bada, eten berria zain geldituko da, bestela tratatuko da.

IBM-370ean baldintz erregistroa eta PC-a toki berean daude PSW (programa-egoeraren hitza) izeneko erregistro batean. PSW erregistro honetan PC-ren informazioaz aparte eten mota bakoitzari bit bat asignatzen zaioeta 5 bit hauei maskara-bitak deritze. Maskara-bita 1 egoeran baldin badago mota honetako etenak onartzen direla esan nahi du eta 0 egoeran badago mota honetako etenak maskaraturik daude eten berriak galerazten direlarik.

Eten bat tratatzeko PSW-ren edukina gorde egiten da eta horretarako memoriako helbide finko batean 5 hitz bikoitz daude, eten-mota bakoitzari dagozkionak, hitz hauei PSW zaharrak deitzen zaielarik.

Bestalde, errutinara jauzteko beste 5 hitz bikoitz daude, PSW berriak, zeinetan PCri dagozkion bitetan errutinaren helbidea bait dago.



6.12 irudia.- IBM-KO ETEN-TRATAMENDUREN PAUSOAK

Eten bat gertatzen denean IBM-370eko CPUa ondoko urratsei jarraitu behar zaie (ikus 6.12 irudia):

a) Etenak PSW-ean etena sortu duen unitatearen informazioa sartzen du.

b) PSW-aren edukina eten motari dagokion PSW zaharrean kokatzen da.

c) Eten motari dagokion PSW berrriaren edukina PSW-ean kokatzen da, honen ondorioz mota horretako etenak galerazten dira eta tratamendurako errutinara jauzi egiten da.

d) Errutina egikaritzen da. Aginduren batek PSW zaharra aztertuko du, etenaren arrazoia jakiteko (a) urratsean informazio hori sartu bait da.

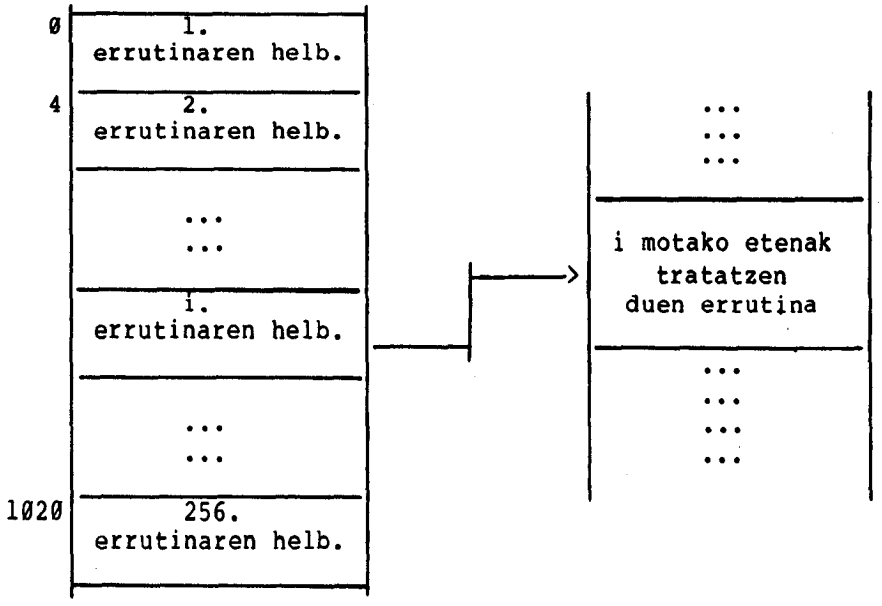
e) Errutinaren azken aginduaren ondorioz PSW zaharraren edukina PSW-ean kokatzen da eta ondorioz aurreko prozesura itzultzen da eta mota horretako etenak onartzen dira.

B) VAX KONPUTAGAILUAN

Konputagailu honetan 31 lehentasun maila bereizten dira PSL (programa-egoeraren hitz bikoitza) izeneko erregistroaren 5 biten egoerei dagozkienak. Bost bit hauek osatzen duten balioari IPL (etenaren lehentasun-maila) izena ematen zaio. Lehentasun hauek letik 31ra zenbatzen dira eta zenbat eta zenbaki handiagoa izan hainbat eta lehentasun handiagoa adierazten da. Lehentasun txikiena dagokien 15 zenbakiak softwareak sortutako etenek erabiltzen dituzte eta 16tik 31ra doazen lehentasunak hardwareak sortutakoek. Sarrera/Irteerako dispositiboek sortutako etenei 16tik 23ra doazen lehentasunak dagozkie.

Bestalde, etenen tratamendua bektorizaturik dago, hau da, eten-mota bakoitzarengatik memoriako helbide finko batean etenari dagokion errutinaren helbidea dago bektore deitutako hitz bikoitz batean. 0tik 1023 memoriako helbideraino dagoen informazioa bektoreak dira, beraz, 256 bektore desberdin daude 256 eten mota dife-

renteri dagozkienak (ikus 6.13 irudia).



6.13 irudia.- VAX-EKO ETEN-BEKTOREAK

Konputagailu honetan eten bat gertatzen denean ondoko urrats hauei jarraitzen zaie:

a) IPLR izeneko erregistro batean etenari dagokion lehentasunaren balioa kokatzen da.

b) Martxan dagoen programa edo errutinari dagokion lehentasuna eta etenaren lehentasuna adierazten dira, horretarako PSL-ko IPL-aren bitak IPLR-arenekin konparatzen dira. Zegoen lehentasuna baino handiagoa baldin badagokio etenari, hau onartu egiten da, bestela, galerazi egiten du CPU-k.

c) Etena onartzean PSL eta PC-ren balioak pilan gordetzen dira, IPLR-aren balioa PSL-n kokatzen da eta etenari dagokion bektorearen edukina PC-n sartzen da. Honen ondorioz oraingo lehentasuna igo egiten da eta tratamendurako errutinara jauzi.

d) Errutinaren bukaeran PC-n eta PSL-n lehengo balioak berreskuratzen dira, alboko ondoriorik gabeko itzulera gertatuz.

7.- S I S T E M A E R A G I L E A

7.1.- KONTZEPTUA ETA OSAGAIAK

7.2.- HASIERAKO PROGRAMA-KARGATZAILEA

7.3.- GAINBEGIRALEA. SISTEMAREN KARGATZAILEA

7.4.- EGINKIZUN-KUDEATZAILEA. KONTROL-SENTENTZIAK

7.5.- ITZULTZAILEAK ETA ESTEKATZAILEA

7.6.- LIBURUTEGIAK

7.7.- SISTEMA ERAGILEAREN FUNTZIONAMENDUA

7.8.- PROZESU-MOTAK

7.9.- BESTE ZENBAIT KONTZEPTU

7.1.- KONTZEPTUA ETA OSAGAIAK

7.1.1.- KONTZEPTUA

Aurreko gaietan konputagailuaren egituraz eta konputagailu horrek egikaritzen dituen aginduez hitz egin da, baina programa bat egikaritzeko betebeharrak eta bideak ez ditugu aipatu.

Dakigunez, programa batek egikaritzeko, makina lengoaiara itzulita egon behar du, eta programa horren exekuzioan gerta daitezkeen etenak Memoria Nagusian sarturik maneiatu behar dira, ... Eginkizun hauek (itzulpena, karga, maneiua) eta beste asko software bidez burutzen dira eta erabiltzaile guztiek programatu beharko lituzkete OINARRIZKO SOFTWAREA edo SISTEMA ERAGILEA deritzana ez balego. Beraz, oinarrizko softwarerik gabe erabiltzaileak bere beharrak ebazten dituzten programak egiteaz gain, konputagailuaren funtzionamendurako eginkizun asko programatu beharko lituzke, konputagailuaren explotazioa oso zaila eta garestia gertatuz. Hori dela eta, konputagailuaren funtzionamendurako beharrezkoak diren programak erabiltzaile guztientzako behin bakarrik programatzen dira, oinarrizko softwarea osatzen duten programa hauek konputagailuarekin batera saltzen direlarik.

Esandakoaren arabera oinarrizko softwarea edo sistema eragilea horrela defini daiteke: Konputagailuaren efektibotasuna handiagotuz haren programazioa eta operazioa errazten duten programa- eta errutina-multzoak oinarrizko softwarea osatzen du.

Oinarrizko softwarea osatzen duten programak makina-lengoaiaz idatzirik eta EGOILIAR izeneko euskarri batean (disko magnetikoan jeneralean) gordeta daude. Hasiera batean oso sinpleak baziren ere, gaur egun oso sofistikatuak dira, gero eta laguntza gehiago eskaintzen dutelarik.

Beste ikuspegi batetik, konputagailu batean dauden programa guztiak ondoko hiru motatakoak izan daitezke:

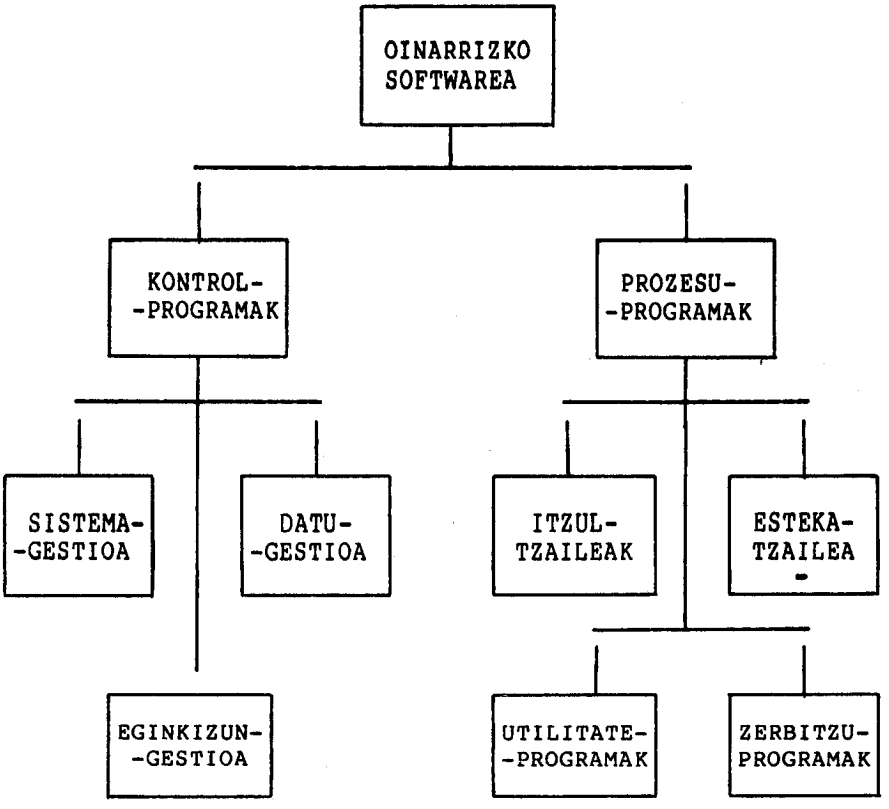
- Oinarrizko softwarea.- Erabiltzaile guztiek behar dituzten programek osatzen dute, konputagailuarekin batera erosten delarik. Beraz, ez du erabiltzaileak programatzen.

- Aplikazio-programak.- Erabiltzaile bakoitzak programatu egiten ditu, zeren eta bere behar zehatzak ebazteko programak bait dira.

- Paketeak.- Erabiltzaile multzo baten beharrei erantzuten dien programa-multzo batek pakete bat osatzen du. Software garapenerako enpresek egiten dituzte gehienetan, kontabilitaterako paketeak, produkziorakoak, estadistiketarakoak eta egituraren kalkulurakoak oso arruntak direlarik.

7.1.2.- OSAGAIAK ETA SAILKAPENA

Oinarrizko softwarea osatzen duten programak bi multzotan banatzen dira: kontrol-programak eta prozesu-programak.



7.1 irudia.- OINARRIZKO SOFTWAREAREN OSAGAIAK

KONTROL-PROGRAMAK.- Programa eta errutina hauen funtzioa beste programen programazio eta operazioan laguntzea da. Programa hauen exekuzioa ez du erabiltzaileak eskatzen, baizik eta sistemak, beharra daukanean, erabakitzen du. Kontrol-programen multzoari SISTEMA ERAGILEA deitzen zaio, nahiz eta batzutan oinarrizko softwarea eta sistema eragilea baliokidetzat hartu. Kontrol-programek hirugarren mailako lengoaia osatzen dute. Adibidez, Sarrera/Irteerarako agindu sinpleak definitzen dira eta gure programan agindu-multzo konplexu batez idatzi behar zena, agindu bakar batez idazten da, exekuzioan agindu hori agertzen denean automatikoki makina-kodean baliokideak diren aginduak egikarituerazikо direlarik.

PROZESU-PROGRAMAK.- Erabiltzaile guztiek lan eroso baterako beharrezkoak dituzten programak dira. Programa hauek kontrol-programen zerbitzuez baliatzen dira eta erabiltzaileak erabakitzen du noiz exekutatu behar diren.

Azken finean, aplikazio-programak bezalakoak dira, baina ebazten dituzten eginkizunak guztiz orokorrak dira; adibidez, programazio-lengoaiak itzultzea, euskarrien artean informazioa trukatzea, fitxategia sortu eta ezabatzea, ... Kontrol-programak ebazten dituzten funtzioen arabera, hiru multzotan sailkatzen dira, sistema-gestioa, eginkizun-gestioa eta datu-gestioa.

SISTEMA-GESTIOA.- Konputagailuaren funtzionamendu zuzenaz arduratzen diren errutinen multzoak osatzen du. Bere funtzio garrantzitsuenak ondoko hauek ditugu:

- Erroreen detekzioa eta maneiua.
- Etenen tratamendua.
- Programak memorian kargatzea.
- Programen birkokapena.
- Baliabideen banaketa (CPU, memoria, dispositiboak).

Funtzio hauen garrantzia dela eta, errutina gehienek memoria nagusian egon behar dute beti, bestela prozesuak atzeratuko bait lirateke.

EGINKIZUN-GESTIOA.- Programa-multzo honek erabiltzaileek sistemari bidaltzen dioten informazioa maneiatu egiten du, beharrezkoa den informazioa sistema-gestioari bidaliz. Sistema-gestioan memoriaratzen eta martxan jartzen ditu programa hauek, behar dituenean.

DATU-GESTIOA.- Errutina hauek sarrera/irteerako eragiketetan laguntzen dute, programazioa errazagoa gerta dadin. Atzipen-metodo desberdinak eta zenbait erraztasun gehiago eskaintzen ditu datu-gestioak oso konplexuak liratekeen sarrera/irteerako aginduak beste aginduen mailatara erraztuz.

Prozesu-programak mota desberdinetakoak dira, itzultzaileak, estekatzailea, zerbitzu-programak eta utilitate-programak bereizten direlarik. Itzultzaileak eta estekatzailea aurrerago aztertuko ditugu.

ZERBITZU-PROGRAMAK eta UTILITATE-PROGRAMA.- Erabiltzaile guztiak zenbait eginkizun orokorrekin (informazioa sailkatu, fitxategiak sortu, kopiatu eta ezabatu, testuak apaindu, ...) oso maiz topatzen dira. Eginkizun orokor hauek ebazten dituzten programak zerbitzu edo utilitate-programak deitzen dira.

Zerbitzu- eta utilitate-programen arteko bereizketa ez dago oso argi, dena dela sistema eragilearekin lotura daukatenean zerbitzu-programak deitzen dira eta erabiltzailearekin lotura handiagoa daukatenean utilitate-programak. Lehenengoen artean programa liburuzaina dago eta azkenen artean kopiatzeko programak eta editorea.

Sistema eragile bat baloratzeko ondoko aztergaiak daude:

- Egoiliarraren kokapena.
- Eskainitako atzipen-metodoak.
- Eskainitako prozesu-motak.
- Eskainitako itzultzaileak.
- Eskainitako utilitate-programak.

7.2.- HASIERAKO PROGRAMA-KARGATZAILEA (IPL)

Gaur egungo konputagailu guztiek sistema eragilea erabiltzen dute. Sistema eragilerik ez balego konputagailuaren errendimendua askoz txikiagoa izango litzateke. Beraz, konputagailuak funtzionatzeko, sistema eragilea behar da, baina sistema eragilea (SE-a) euskarri batean gordetzen da, diskoan jeneralean eta sistema eragile (SE) hori, dagozkion funtzioak bete ditzan, memoria nagusira eraman behar da, sistema-gestioa behinik behin. Beraz, konputagailua pizten denean SEaren muina egoiliar euskarritik memoriara eraman behar da eta horretaz arduratzen den programa txiki batek memoria nagusian egon behar du. Konputagailu batzutan hardwarearen bidez memoriaratzen da programa txikia, bestetan berriz, ROM memoria txiki batean.

Aurrekoa kontutan hartuz, konputagailua pizten denean sistema eragilearen muina memoriaratzen duen programa txiki bat da hasierako programa-kargatzailearen definizio bat. Konputagailu batzutan hasierako programa-kargatzailea ez dago, zeren eta SEren muina, gainbegirale deritzana, ROM memorian dago.

7.3.- GAINBEGIRALEA. SISTEMAREN KARGATZAILEA

Gainbegiralea (SPVa), edozein programa exekutatzen denean, programen funtzionamendu zuzenaz eta konputagailuaren probetxamenduaz arduratzen den SEaren atala da eta, ondorioz, beti egongo da memoria nagusian kokaturik; hau da, SEaren muina da.

Gainbegiralearen funtzioen artean ondoko hauek ditugu:

- Erroreen detekzio eta maneiua .- Programa-errore edo makina-erroreei adi egongo zaie gainbegiralea eta horrelako bat gertatzean exekutatzen ari den programa eten eraziko du eta errorearen maneiurako errutina egikarieraziko du, errutina hori gainbegiralean dagoelarik.

- Etenen maneiua .- Sarrera/irteerako dispositiboek sortutako etenen bat iristen denean, aurreko kapituluan azterturiko tratamendua eman behar dio.

- Baliabideen banaketa .- Programa bat baino gehiago paraleloan egikaritzen badira (ikus multiprogramazioa eta denbora bana-

tua), programa bakoitzari bere memoria-zatia eta CPUaren tartea edo lehentasuna asignatzen dio prozesu-motaren arabera, programen exekuzioak denboran tartekatuko dituelarik. Alegiazko memoria dagoenean helbideen itzulpena ere gainbegiralearen funtzio bat da.

- Programa-kargatzea .- Momentu guztietan memorian dagoen atala denez, bere eginkizunen artean programa-kargatzea dago. Bere funtzioak burutzeko edota erabiltzailearen eskakizunei erantzuteko gainbegiraleak programak memoriaratu ditzake. Horretarako SISTEMAREN KARGATZAILEA izeneko programa bat dauka.

Sistemaren kargatzailea, beraz, beste programak memorian kargatzeko balio duen gainbegiratzailearen osagai bat dugu. Kargatze prozesuan kargatzaileak programa birkoka dezake programa birkokagarria baldin bada, bestela, helbide guztiak itzuli beharko dira.

7.4.- EGINKIZUN-KUDEATZAILEA. KONTROL-SENTENTZIAK

EGINKIZUN-KUDEATZAILEA (Job Scheduler) eginkizun-gestioaren programa nagusia da. Erabiltzaileak egikaritu nahi dituen programen izenen eta exekuziorako baldintzen berri kudeatzaileari ematea da bere funtzioa. Horretarako, gainbegiraleak, sistemaren kargatzailearen bidez, memoriaratzen du eginkizun-kudeatzailea eta programa honek kontrol-sententziak irakurtzen ditu, sententzia hauen informazioa gainbegiraleari bidaltzen diolarik. Kontrol sententzien informazioa, gainbegiraleak horretarako duen datu-alderdi batean gelditzen da, gainbegiraleak informazio hori maneia dezan.

KONTROL-SENTENTZIAK erabiltzaileren eta SEaren arteko komunikabidea dira, komando izenarekin ere ezagutzen direlarik.

Kontrol-sententzien bidez, erabiltzaileak egikaritu nahi dituen programen izenak eta ezaugarriak zehazten ditu eta sententzia hauek operatzaileak dispositibo batean kokatzen dituenez gero, eginkizun-kudeatzaileak irakurtzen ditu, informazioa gainbegiraleari bidaliz.

Kontrol-sententziak edo komandoak idazteko, SEari datxekion komando-lengoaia bat dago. Komando hauek karaktere berezi batez hasten dira beti (normalean $ batez) ondoan kode bat eta zenbait eragigai zehazten direlarik.

Komando-mota asko dago, baina hedatuenak hauexek ditugu:

- Programaren exekuziorako komandoa.- Komando honek kode finko bat du (normalean RUN edo EXEC) eta eragigai batez egikaritu nahi dugun aplikazio-programaren izena zehazten da. Beraz, sententzia honen bidez gainbegiraleak izendaturiko programa kargatu eta egikaritu dezan lortzen dugu. SEaren prozesu-pro-

grama bakoitza egikarierazteko komando berezi bat dago.

- Asignazio-komandoa.- Komando honetaz izen sinbolikoak dispositibo fisikoei asignatzen zaizkie, baliokidetasunak gainbegiraleak kontutan har ditzan. Horrela, programa orokorrak lortzeko, programatzean, ez ditugu dispositibo fisikoen helbide zehatzak aipatu behar, baizik eta izen logikoak erabiliko ditugu, egikaritzapena burutzeko orduan izen logikoei berezko helbideak asignatuko dizkiegularik. Batch moduan (ikus 7.8 pasartean) lan egiten duten konputagailutan komandoak dispositibo bakar batez sar daitezke; sistemaren kontsola (txartel-irakurgailua edo teklatua) deitutakoaz. Prozesu elkarreragiletan (ikus 7.8 pasartean) erabiltzaile bakoitzak bere komandoak barneratzen ditu bere teklatua erabiliz.

7.5.- ITZULTZAILEAK ETA ESTEKATZAILEA

Oinarrizko softwarearen prozesu-programak dira, beraz, kontrol-sententzien bidez eskatu behar da programa hauen exekuzioa.

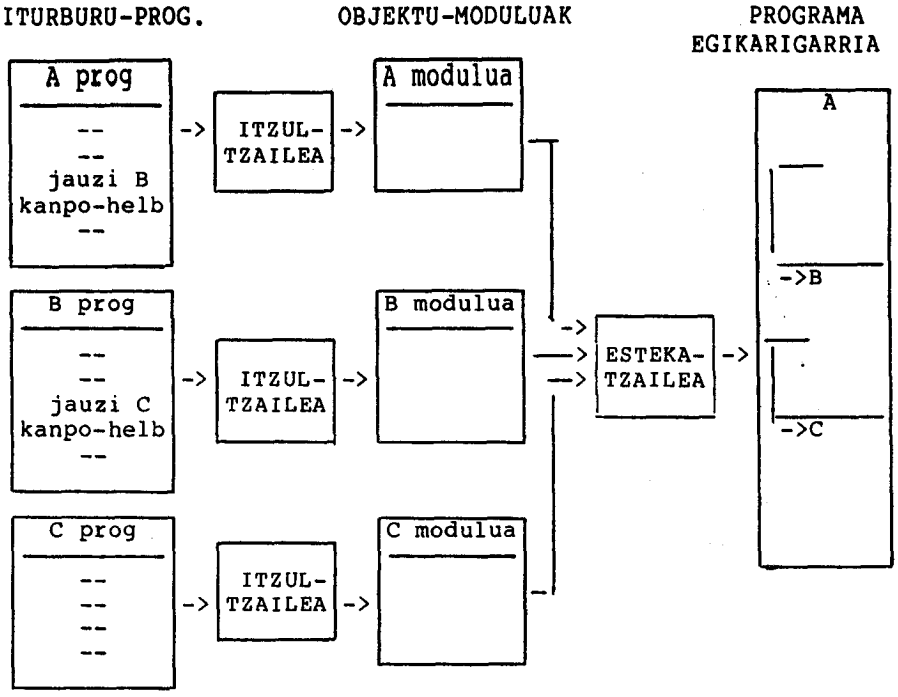
Itzultzaileen funtzioak eta motak bostgarren gaian aztertu genituen. Itzultzaileek, itzulpena egiteaz gain, itzulpen-erroreak pantailaratu edota inprimatzen ditu, denbora berean iturburu programaren listatu bat eginez. Itzulpen-erroreak programazio-lengoaiaren sintaxiaren kontrako erroreak dira, baina itzulpen-errorerik ez badago ere, ezin da programa zuzena denik esan, exekuzio-erroreak egon bait daitezke. Mihiztatzaileak nabaritzen dituen itzulpen-errore arruntenak honako hauek dira: sinbolo ez-definitua, sinbolo behin baino gehiagotan definitua, kode okerra, eragigai gehiegi edo gutxiegi, helbide okerra eta antzekoak.

Aipatzekoa da konpiladore edo mihiztatzaileak lortzen duen objektu-modulua edo objektu-programa ezin dela zuzenean exekutatu, lehenago estekatu egin behar bait da.

ESTEKATZAILEA (Linker) objektu-moduluak elkartu eta haien arteko lotura-helbideak ebazten dituen oinarrizko softwarearen programa da.

Estekatzailearen beharra kanpo-aldagaiak eta kanpo-azpiprogramen erabilpenak sortzen du. Guk programa bat idazten dugunean programatutako eta itzulitako beste programak erabil ditzakegu. Horretarako, programa horien izen sinbolikoak kanpo-sinboloak direla erazagutu behar diogu itzultzaileari, honek itzulpen-errore bat (sinbolo ez-definitua) sor ez dezan. Hala ere, errorerik sortuko ez badu ere, ezingo ditu kanpo-sinbolo horiek ebatzi, hau da, berezko helbide bihurtu, zeren eta ez bait ditu sinbolo horiek inon definitzen. Beraz, itzultzaileak lorturiko objektu-programa, objektu-modulua eta modulu birkokagarriei deitzen diela ere, ezin da egikaritu lehenago deitutako moduluak gehitu eta kanpo-sinboloak ebazten ez badira. Horretaz, estekatzailea arduratzen da eta, objektu-modulu bat edo gehiago datu gisa hartuz, moduluak elkartu,

moduluen arteko helbideak ebatzi eta programa egikarigarria sortu egiten du (ikus 7.2 irudia).



7.2 irudia.- PROGRAMA BATEN MODULUAK ITZULIZ ETA ESTEKATUZ.

Estekatzaileak bere eginkizuna burutzeko ondoko urratsak segitzen ditu:

- Moduluei dagozkien izenak eta luzerak biltzen ditu taula batean.

- Modulu bakoitzari karga-helbide bat asignatzen dio, modulua birkokatzeko balioko duelarik. Urrats honetarako, aurreko urratsean eraikitako taula erabiltzen da.

- Modulu guztien Sarrera-puntuak, hau da beste moduluetatik erabil daitezke izen sinbolikoak, eguneratzen dira dagozkien moduluen karga-helbidea gehituz. Helbide guztiak birkokatu behar dira, baina birkokagarriak baldin badira ez da beharrezkoa.

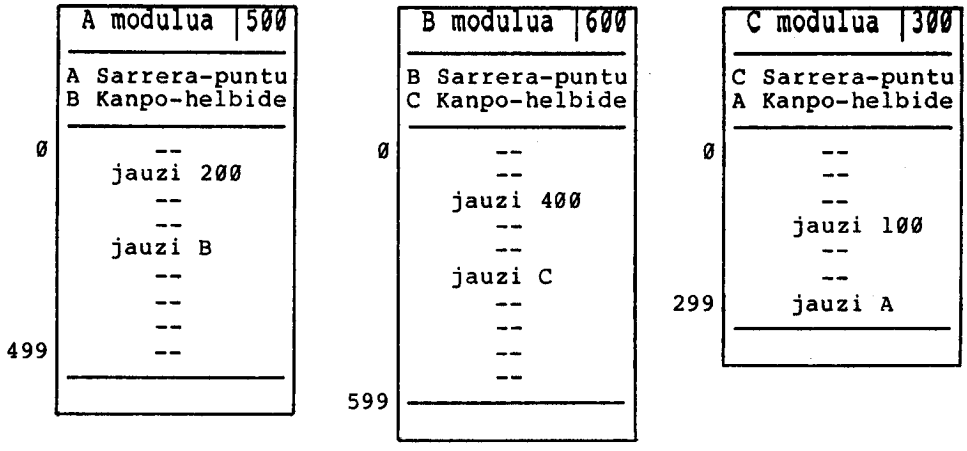
- Moduluen arteko helbideak, hau da, modulu guztien kanpo-helbideak, ebazten dira, modulu baten kanpo-helbide batek beste modulu baten sarrera-puntua izan behar bait du. Azken baldintza hau betetzen ez bada, errore bat detektatzen da eta estekaketa ez da burutzen.

Estekatzaileak urrats hauei jarraitzeko objektu-moduluen informazioa erabiltzen du eta bere eginkizuna errazteko itzultzaileak, objektu-modulua sortzen duenean, aginduen itzulpena burutzeaz gain estekatzailearentzako kontrol-informazioa lortzen du.

Objektu-modulua, beraz, ez dute makina-lengoaiazko aginduek bakarrik osatzen, beste zenbait informazio ere badago. Hona hemen objektu-moduluen egitura:

a) Moduluaren izena.
b) Sarrera-puntuak.
c) Kanpo-helbideak.
d) Aginduak (makina-lengoaiaz).
e) Birkokapenerako zerrenda.
f) Moduluaren amaiera.

Adibide bat azter dezagun. 7.3 irudian A, B eta C moduluak dauzkagu eta, modulu bakoitzean, izena, luzera, sarrera-puntu bat, kanpo-sinbolo bat eta helbide ez-birkokagarri bat daude. Modulu hauek estekatu behar dira.



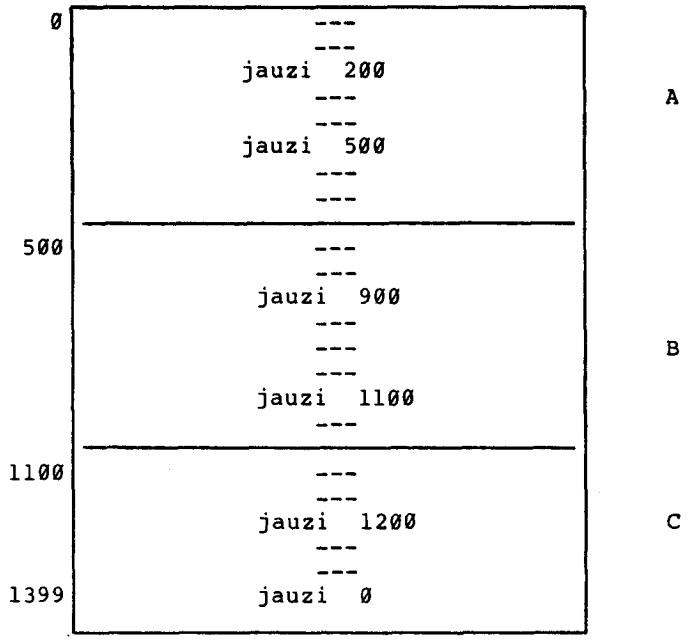
7.3 irudia.- A, B ETA C OBJEKTU-MODULUEN EGITURA

Kasu honetan estekatzaileak ondoko eragiketak burutuko ditu:

1) Modulu-taula eraikitzen du, taula horretan moduluen izenak eta luzerak bilduko direlarik. Ondoko taula izango litzateke A/500/B/600/C/300.

2) Modulu bakoitzari karga-helbidea kalkulatzen dio aurreko moduluen luzera gehituz horretarako aurreko puntuan eraikitako taula erabiliz. Adibidean B moduluari 500 helbidea egokitzen zaio eta C moduluari 1100-a.

3) Sarrera-puntuak eguneratzen ditu, beraz B eta C moduluen izenak izan aparte, sarrera-puntuak dira; beraz 0, 500 eta 1100 helbideak asignatzen zaizkio A, B eta Cren arabera. Bestalde, helbide ez-helbideragarriak ere eguneratzen dira, moduluari dagokion karga-helbidea gehituz. Horrela A programan dagoen 200 helbidea berdin gelditzen da B-ren 400 helbidea 900 bihurtu eta C-ren 100 helbidea 1200 helbidea.



7.4 irudia.- 7.3 IRUDIAREN MODULUEK OSATURIKO PROGRAMA EGIKARIGARRIA

4) Moduluen arteko helbideak ebazten dira, hortaz, A moduluan aipatzen den B helbideak 500 balioa hartuko du, B programaren C izen sinbolikoaren ordez 1100a eta C programaren A-ren ordez 0 helbidea kokatu da.

Pausu hauek burutu ondoren, 7.4 irudian dagoen programa egikarigarria sortzen du.

7.6.- LIBURUTEGIAK

7.6.1.- KONTZEPTUA

Liburutegiak programak gordetzeko fitxategiak dira. Zerbitzu-programa batek liburutegiko programak gestionatu egiten ditu programa honi LIBURUZAINA deitzen zaiolarik.

Badaude bi liburutegi mota, sistemarena eta erabiltzailearena. Sistemaren liburutegian oinarrizko softwarea osatzen duten programak gordetzen dira, nahiz kontrol-programak, nahiz prozesu-programak. Erabiltzailearen liburutegietara aplikazio-programak eta paketeak biltzen dira. Konputagailu batean sistemaren liburutegi bakar bat dago, egoiliar izeneko bitarteko memoria batean, erabiltzailearenak, berriz, nahi adina egon daitezke.

Liburutegi batean programa-mota desberdinak egon daitezke, iturburu-programak, objektu-moduluak, programa egikarigarriak edota prozedurak. Programa-mota bakoitza azpiliburutegi batean kokatzen da, horrela, iturburu-programak iturburu-liburutegian kokatzen dira, objektu-moduluak modulu-liburutegian, programa egikarigarriak egikarigarrien liburutegian eta prozedurak prozedura-liburutegian. Ez dira nahastu behar orain aipaturiko prozedurak eta programazioan erabilitako prozedura izeneko azpiprogramak. Lehenbizikoak komando-multzo batean dautza eta sistema eragileari zuzendutako programatzat har daitezke.

7.6.2.- LIBURUZAINA

Liburutegietan (ikus 7.5 irudia) KATALOGOAn programen alderdian dagoen programa bakoitzarengatik erregistro bat dago eta erregistro horretan, programa horren izena, liburutegiaren helbidea, luzera eta beste zenbait ezaugarri zehazten dira. Horrela liburutegian izen ezaguneko programa bat atzitu nahi denean ez da programaz programa sekuentzialki bilatu behar, baizik eta katalogoan bilatu sekuentzialki, eta programari dagokion helbidea jakinez gero, programa zuzen atzitzen bait da.

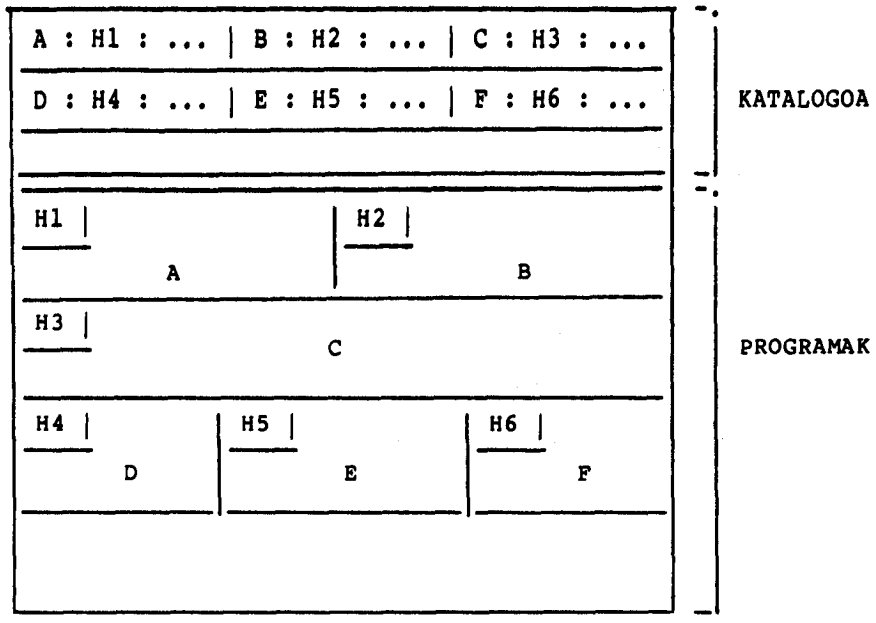
Esan dugunez, liburutegien maneiuaz liburuzain izeneko zerbitzu-programa bat arduratzen zen. Beste funtzioen artean, liburuzainak ondoko eragiketak burutzen ditu:

- Programa bat katalogatzea.- Horretarako, alde batetik, programa programen alderdian kopiatzen du eta, bestetik, programari dagozkion ezaugarriak bildu eta katalogoan erregistro bat osatuz kokatu.

- Programa bat ezabatzea.- Horretarako, programari dagokion katalogoko informazioa ezabatu egiten du.

- Programa bati izena aldatzea.- Programari dagokion katalogoko erregistroan programaren izena aldatu egiten du.

- Liburutegiaren programak birkokatzea.- Ezabaturiko programak ez dira programen alderditik kendu, katalogotik soilik baizik, eta ondorioz, programen alderdian toki asko degradaturik dago zeren eta dagoen informazioa ezabaturiko fitxategiei bait dagokie. Noizean behin komenigarria da programak birkokatu eta degradaturiko lekua berreskuratzea, beste programak sartu ahal izateko.



7.5 irudia.- LIBURUTEGI BATEN EGITURA

7.6.3.- EGOILIARRA

Liburutegiak azaletik aztertu ondoren, sistemaren liburutegia sakonago adieraziko dugu.

EGOILIARRA sistemaren liburutegian kokatuta dagoen euskarriari deitzen zaio eta jeneralean instalazioaren disko azkarrena erabiltzen da egoiliar gisa. Egoiliarrean lau alderdi desberdin hauek bereizten dira:

- Egikarigarrien liburutegia edo memori irudien liburutegia (CIL-Core Image Library).- Alderdi honetan oinarrizko softwarearen programa-egikarigarri guztiak daude, horrek esan nahi du, oinarrizko softwarearen atal garrantzitsuena han dagoela. Horrela, hasierako programa kargatzailea, gainbegiralea, eginkizun-kudeatzailea, itzultzaileak, estekatzailea, zerbitzu eta utilitate-programak, etab., egikarigarriak izan arren, liburutegi honetan daude.

- Modulu-liburutegia (RL-Relocatable Library).- Alderdi hau osatzen duten modulu gehienak datu-gestioaren osagaiak dira, hau da, sarrera/irteerako programazioan laguntzeko moduluak, beste eragiketa-motak programatzen laguntzeko moduluak badaude ere.

- Lotura-atala.- Alde hau itzultzaileak, estekatzaileak eta kargatzaileak erabiltzen dute haien eginkizunak burutzeko.

- Beste zenbait lanerako atal.- Alderdi hau bitarteko emaitzak eta fitxategiak kokatzeko erabiltzen du SEak.

7.6.4.- LOTURA-ATALA

Itzultzaileak, konpiladore eta mihiztatzaileen kasuan behinik behin, itzultzeko iturburu-programa iturburu-liburutegian bilatzen du, konputagailu batzuetan sarrera-unitatean egon badaiteke ere, eta lorturiko objektu-modulua lotura-atalean kokatzen du. Maiz lortutako modulua automatikoki katalogatzen da baina SE batzutan katalogatzea erabiltzaileak eskatu behar du.

Estekatzailea, erabiltzaileak zehazturiko moduluak, modulu-liburutegian bilatu egiten ditu nahiz eta konputagailu batzutan erabiltzaileak, nahi badu, modulu bat lotura-atalean dagoela adierazteko aukera izan. Estekatzaileak lortzen duen programa egikarigarria, lotura-atalean eraikitzen da eta itzultzaileen kasuan bezala automatikoki edo erabiltzailearen eskakizuna dela medio, egikarigarrien liburutegian kataloga daiteke.

Kargatzailea, erabiltzaileak programaren izena adierazten badu, programa egikarigarrien liburutegitik eskuratzen saiatuko da. Programaren izena zehazten ez bada berriz, lotura-atalean dagoen programa memoriaratuko du.

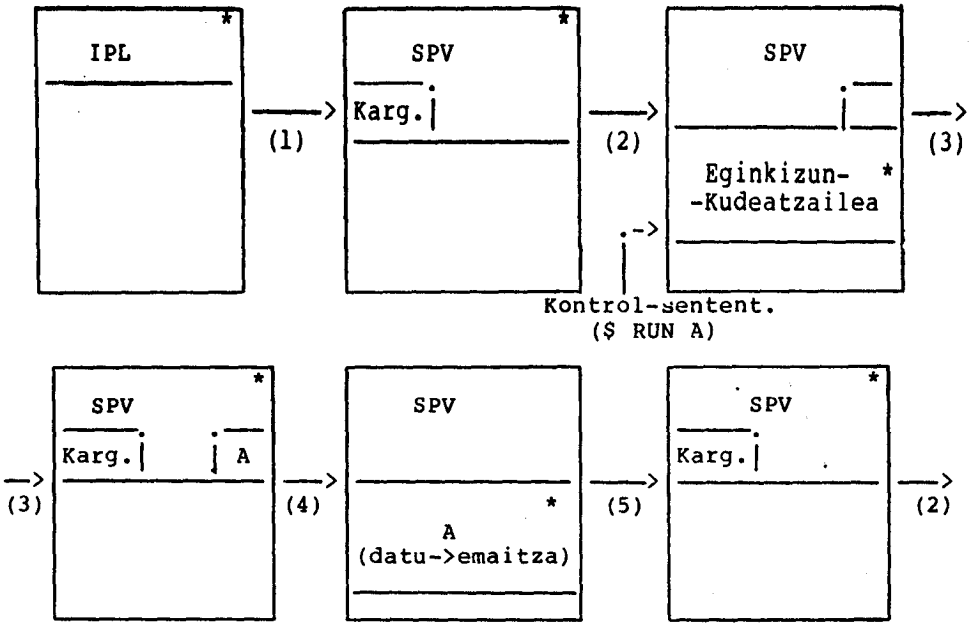
7.7.- SISTEMA ERAGILEAREN FUNTZIONAMENDUA

Konputagailua pizten denean guk nahi ditugun programak egikari daitezen abiarazten den mekanismoa edo prozedura azaltzea da pasarte honen helburua. Mekanismo hori ondoko pauso hauek osatzen dute (ikus 7.6 irudia):

1) Hasierako programa kargatzailea egikaritzen da. Horretarako programa horrek memorian egon behar du, nahiz ROM memorian dagoelako, nahiz operatzaileak kargatzeko tekla bat sakatu duelako eta, ondorioz, hardwarearen bidez kargatu egin da. Dakigunez, IPL programaren egikaritzapenaren ondorioz, Gainbegiralea (SPVa) memoriaratzen da eta abiarazi.

2) Gainbegiralea egikaritzen da, baina ez daki zein den erabiltzaileak egikaritu nahi duen programa. Horregatik, kontrol-sententziak irakurri behar dira, baina gainbegiralearen funtzioen artean kontrol-sententzien irakurketa ez dagoenez, gainbegiraleak, funtzio hori burutzen duen programak, hau da, eginkizun-kudeatzaileak memoriaratu eta abiarazi behar du. Eginkizun-kudeatzaile programaren karga gainbegiralearen atala den sistemaren kargatzaileak egiten du sistemaren egikarigarrien liburutegitik eskuratuz.

3) Eginkizun-kudeatzailea egikaritzen da. Programa honek erabiltzaileak sarturiko kontrol-sententziak irakurtzen ditu eta sententzien informazioa gainbegiraleari uzten dio, gainbegiralearen memoriako datu-alderdi batean (kontutan har ezazu gainbegiralea beti memorian dagoela). Eginkizun-kudeatzaileak exekuziorako komando bat aurkitzen duenean (adibidean RUN A) gainbegiraleari in-



* --> momentu bakoitzean egikaritzen den programa

7.6 irudia.- PROGRAMAK EGIKARITZEKO SISTEMAK SEGITZEN DUEN MEKANISMOA

formazioa bidaliz gero, komandoak irakurtzen uzten du eta gainbegiraleari kontrola itzultzen dio.

4) Gainbegiraleak tratatzen du eginkizun-kudeatzaileak bidalitako informazioa eta exekuziorako komandoari dagokion informazioa, hau da, erabiltzaileak egikaritu nahi duen programaren izena, maneiatu behar duenean, sistema kargatzailaren bidez zehaztutako programa egikarigarrien liburutegietan bilatu eta memoriaratzen du. Kargaturiko programa (adibidean A programa) erabiltzailearen edozein programa edo oinarrizko softwarearen edozein prozesu-programa izan daiteke.

5) Erabiltzaileak eskaturiko programa egikaritzen da edozein programa bezala: datu batzu irakurriko ditu, datu horiek prozesatu eta emaitzak kanporatu. Egikaritzapena bukatzen denean, kontrola gainbegiraleari uzten zaio eta bigarren pausoan gaude berriro.

Beraz, bigarren pausotik bostgarreneraino dauden prozesuek amaiera gabeko bigizta bat osatzen dute eta errepikatzen dira konputagailua itzali arte.

Hirugarren pausoan eta batez ere bostgarrenean, egikaritzapena eten egiten duen zenbait gertaera (makina-erroreak, progra-

ma-erroreak, etenak,...) jazo daiteke, kasu hauetan gainbegiralea abiaraziko litzateke eta etena maneiatuz gero exekutatzen ari zen programari kontrola utziko litzaioke.

Pasarte honetan ikusitakoaren ondorioz, erabiltzaileak programa bat egikarierazten duenean ez da bakarrik programa hori exekutatzen, baizik eta automatikoki SEren beste programak (gainbegiralea eta eginkizun-kudeatzailea) ere egikaritu bait dira. Beraz, Sistema Eragilerik ez balego konputagailuaren programazioa (programazio-lengoaia sinbolikoen eza) eta operazioa astun-astuna izango litzateke, konputagailuaren errendimendua izugarri jaitsiko litzatekelarik.

7.8.- PROZESU-MOTAK

7.8.1.- ZENBAIT KONTZEPTU

MONOPROGRAMAZIOA/KONKURRENTZIA

Aurreko pasartean gainbegiraleaz gain beste programa bakar bat memorian egon eta egikari daitekeela suposatu badugu ere, konputagailu askotan, gainbegiraleaz aparte, memorian programa bi edo gehiago egon daitezke. Memorian programa bakarra karga daitekeenean (gainbegiralea kontutan hartu gabe), konputagailuak MONOPROGRAMAZIO moduan lan egiten duela esaten da eta programa bat baino gehiago karga daitezkeenean KONKURRENTZIA dagoela esaten da.

Memorian programa bat baino gehiago kargatzen direnean ez dira kargatzen lehenbizi programa bat osoro exekutatzeko eta gero beste bat etab. (horretarako monoprogramazioa aski da), baizik eta kargaturiko programen aginduen egikaritzapena tartekatzen bait da programa guztiek batera aurreratzen dutelarik. Horregatik, konkurrentzia dagoela esaten da, konkurrentzia hau erreala ala itxurazkoa izan badaiteke ere. Konkurrentzia errealean momentu berean agindu bat baino gehiago egikaritzen dira, beraz, CPU bat baino gehiago behar dira, itxurazko konkurrentzian, berriz, CPU bakar bat dago eta momentu berean agindu bakar bat egikaritzen da, baina programen aginduen exekuzioak tartekatzen dira.

Memorian programa bat baino gehiago karga daitezkeenean, kargatzen diren programen planifikazioa bi modutan egin daiteke: zentralizatuan edo elkarreragilean.

MODU ZENTRALIZATUA/MODU ELKARRERAGILEA

MODU ZENTRALIZATUAn programen exekuzioa dispositibo bakar batetik eska daiteke, hau da, kontrol-sententziak operatzaile bakar batek sartzen ditu eta operatzaile honek momentu bakoitzean egikaritu behar diren programak zeintzu diren erabakitzen ditu.

MODU ELKARRERAGILEan erabiltzaile desberdinek eskatzen dute haien programen exekuzioa dispositibo desberdinetatik (jeneralean terminaletatik), beraz, bakoitzak bere kontrol-sententziak sartuko ditu. Gaur egun modu elkarreragileak izugarrizko garapena du. Konkurrentzia dagoenean argi dago SEak konplexuagoa izan behar duela, zeren eta programak birkokatu behar ditu eta programen aginduen exekuzioa tartekatu behar du, tartekatze hau noiz egin behar den erabakiz.

BATCH-A/DENBORA ERREALA

Azkenik monoprogramazioan edo konkurrentzian egikaritzen diren programak bi motatakoak izan daitezke: batch programak eta denbora errealeko programak.

BATCH programak erabiltzaileak erabakitzen duenean memoriaratu eta egikaritzen dira, programa arruntak direlarik.

DENBORA ERREALeko programek erantzun-denbora kritikoa dute, hau da haien exekuzioak ahalik eta azkarrena izan behar du. Erantzun-denbora larriaz aparte, denbora errealeko programek egikaritzapen asinkronoa dute, hau da, datua iristen denean exekutatzen dira eta ez erabiltzaileak eskatzen dituenean. Beraz, programa hauek memorian kargatuz gero, memorian mantentzen dira eta datu bat iristen zaien bakoitzean egikaritzen dira lehenbailehen.

Bankuetxetako kontu korronteen maneiurako programa denbora errealeko betiko adibidea dugu. Programa memorian kargaturik dago eta bulego batetik bezeroren batek dirua atera edo sartzen duen bakoitzean programa abiarazten da programaren exekuzioa ahalik eta azkarrena izanik bulegotan iladak osa ez daitezen.

Denbora errealeko programen programazioa konplexuagoa gertatzen da, sarrera asinkronoa bait da eta, beraz, ezin da irakurtzea agindu, sarrerako eragiketak zailagoak bihurtzen direlarik. Horretan SEak ere laguntzen du.

Kontzeptu hauek aztertu ondoren, SEen prozesu-mota desberdinak aztertuko ditugu, bakoitzari programen planifikazioa eta beraz SE diferentea dagokiolarik. Multiprogramazioa, Multiataza, Denbora banatua eta Multiprozesua izango dira aztertuko ditugun motak. Monoprogramazioa ez dugu azalduko, orain arte suposatu dena bait da.

7.8.2.- MULTIPROGRAMAZIOA

CPU bakar batez eta modu zentralizatuan programa batzuren egikaritzapen konkurrente edo paraleloari multiprogramazioa deritzo.

Multiprogramazioaren helburua, konputagailuaren atal garrantzitsu eta garestiena, hau da, CPUa hobeto aprobetxatzea da. Kanalez horniturikо konputagailu gehienetan erabiltzen dira, zeren eta kanalen bidez CPUa hobeto aprobetxatzen bada ere, maiz, kanalak irakurtzen duen datua programak behar du programa hori etenik gelditzen delarik. Programa bakarra baldin bada, kanalak irakurtzen duen bitartean CPUk ezin du agindurik egikaritu, lanik egin gabe gelditzen delarik, multiprogramazioa baldin badago, berriz, programa bat etenik dagoenean, CPUk beste programen aginduak egikari ditzake eta programa guztiak etenik dauden kasuan bakarrik geldituko da lanik egin gabe.

Beraz multiprogramazioaren ezaugarri nagusienak ondoko hauek dira:

- Memorian programa egikarigarri bat baino gehiago dago, beraz, memoria zatitu egiten da programa bakoitza zati batean kokatuz. Memoriako zati bakoitzari PARTIZIOA deitzen zaio.

- Programak itxurazko konkurrentzian egikaritzen dira, hau da, CPU bakar bat dago eta ondorioz momentu berean agindu bakar bat egikaritzen da.

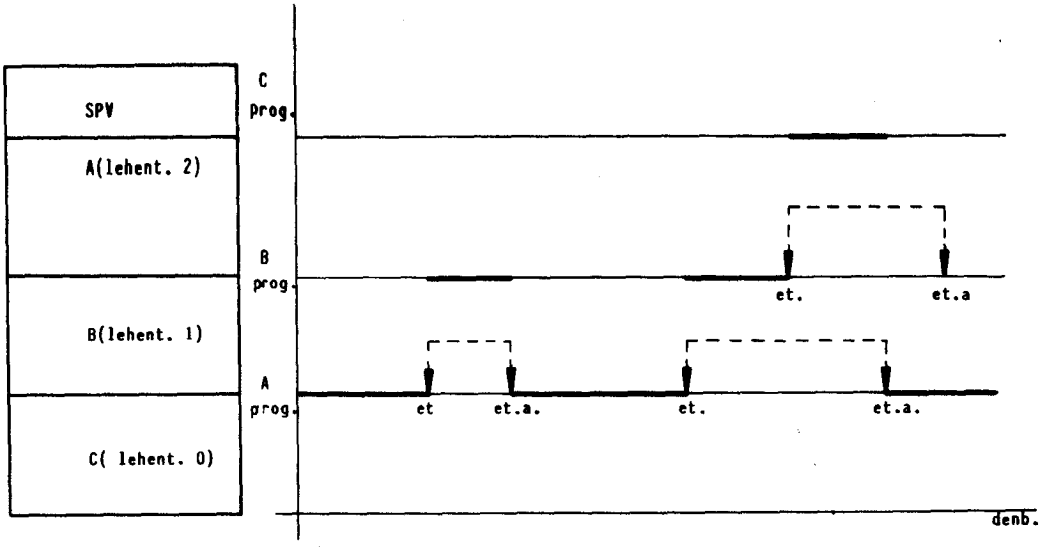
- Planifikazioa zentralizaturik dago operatzaileak egikaritu behar diren programak zeintzu diren zehazten duelarik.

- Memorian dauden programek lehentasun desberdinak dituzte kokaturik daudeneko partizioen arabera. Beraz konkurrentzian ez dute programa guztiek aukera berdina, baizik eta lehentasun handiko partizioetan kokaturiko programek lehenago egikaritzeko aukera izango dute.

Azter dezagun zehatzago multiprogramazioaren mekanismoa, multiprogramazioan SEak jarraitzen duen prozedura hain zuzen ere.

Demagun 7.7 irudian dauden programak multiprogramazioz egikaritu behar direla eta operatzaileak lehentasun handieneko partizioan A programa kokatu duela, lehentasun ertainean B programa eta lehentasun txikienekoan C-a. Hasiera batean A programaren aginduak egikarituko dira, baina oso maiz gertatzen den legez (jeneralean S/I agindu bat dela medio) programa etenik gelditukо da. Orduan gainbegiraleak bitartean B programaren aginduak egikaritu behar direla erabakiko du eta horrela gertatuko da A programaren etena amaitu arte edo B programa etenik gelditu arte. A programaren etena amaitu bezain laster, gainbegiraleak A programari kontrola uzten dio, hau da, beti lehentasun handiko programen aginduak egikaritzen saiatzen da. A programa etenik dagoenean B programa etenik gelditzen bada, C programaren aginduak egikarieraziko ditu gainbegiraleak, beraz, lehentasun txikieneko programaren aginduak egikaritzen dira bakarrik, beste partizio guztietako programak etenik daudenean.

Programa bat geldirik geratzearen zergatia gehienetan S/Iko eragiketak direnez, S/I eragiketa utzi duten programak lehentasun handieneko partizioetan jartzea ez da komenigarria, CPUa monopoli-



7.7 irudia.- A, B ETA C PROGRAMAK MULTIPROGRAMAZIOAZ

zatuko bait lukete.

Bestalde, denbora errealeko programa bat beste programekin multiprogramazioan egikaritzen bada, denbora errealeko programa lehentasun handieneko partizioan kokatu behar da, bestela erantzun-denbora motela gertatuko bait litzateke.

Multiprogramazioak CPUren errendimendua hobetzen du baina zenbait eragozpen aurkezten du; hona hemen hauetako garrantzitsuenak:

- Sistema Eragilea konplexuago eta luzeago bihurtzen da. Gainbegiraleak, bere eginkizunez gain, momentu bakoitzean zer programa exekutatzen den kontrolatu behar du eta programa batetik bestera kontrola aldatzen duenean kontrola galtzen duen programaren kontrol-informazioa gorde behar du, abiarazten duen programaren kontrol-informazioa berreskuratu. Bestalde, SEren programaren batek partizioen kontrola eraman behar du, hau da partizioak sortu, dimentsionatu eta lehentasunak egokitzeko gai den prozesu-programa bat egon behar du.

- Memoria-premia handiagoa dago monoprogramazioan baino. Programa bakar bat kokatzen denean 3, 4 edo programa gehiago kokatzen dira eta gainera gainbegiralea luzeagoa da, beraz memoriako beharra handitu egiten da. Bi irtenbide daude, gehienetan konbinaturik erabiltzen badira ere, benetazko memoria gehiago jarri (oso garestia dena) edo alegiazko memoria erabili (ikus 7.8 pasartea).

- Dispositiboen beharra biderkatzen da, batez ere txartel-irakurgailuarena eta inprimagailuarena. Konkurrentzian egikaritzen diren programa guztiek inprimagailu bat erabili nahi le-

zakete eta inpimagailu bakarra (edo programa-kopurua baino gutxiago) baldin badago programa guztien txostenak nahasten dira, txosten desberdinen lerroak tartekatuz. Konponbidea ez da dispositibo gehiago erostea (oso garestia bait da) baizik eta spooling delakoa erabiltzea (ikus 7.8 pasartea).

- Multiprogramazioaz egikaritzen diren programek birkokagarriak izan behar dute, helbide desberdinetan (partizio desberdinei dagozkionak) karga bait daitezke.

Partizioen banaketaren arabera, bi multiprogramazio mota bereizten dira: luzera finkoko partizioak (partizio estatikoak) dituen multiprogramazioa eta luzera aldakorreko partizioak (partizio dinamikoak) dituenak.

LUZERA FINKOKO PARTIZIOAK (PARTIZIO ESTATIKOAK).- Programak egikaritzen hasi baino lehen partizioak dimentsionaturik daude eta beraz partizio-kopuru finkoa dago. Partizioen dimentsioak edota partizio-kopurua alda daitezke SEren zerbitzu-programa batez baina bitartean ezin dira programak egikaritzen aritu.

Luzera finkoko partizioen erabilpenak memoriako probetxamendu txar bat erakartzen du, zeren eta programa baten luzera partizioarena baino laburragoa bada, programa-luzeratik gorako partizio-zati osoa ezin bait da erabili eta programa partizioa baino luzeagoa bada, ezin da partizio horretan kargatu.

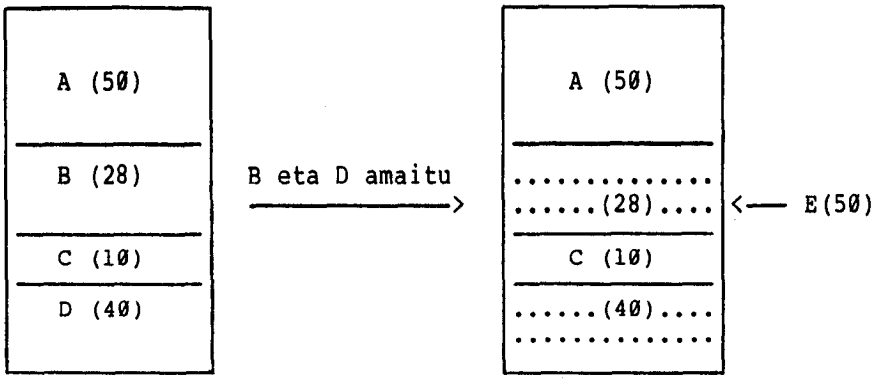
LUZERA ALDAKORREKO PARTIZIOAK (PARTIZIO DINAMIKOAK).- Programa bakoitzak, memorian kargatzen denean, partizio bat osatzen du dinamikoki, beraz, momentu bakoitzean exekutatzen diren programen luzeren arabera, programa- edo partizio-kopuru aldakorra egongo da.

Partizio dinamikoek partizioak maneiatzeko SE sofistikatuago eta luzeagoa eskatzen dute. Memoriako probetxamendua hobetzen da, baina piskanaka memoria degradatzen da, hau da programa bat kargatzeko, memoriako luzera egonda, ezin da programa hori kargatu, dagoen memoria guztia albokoa ez delako (ikus 7.8 irudia).

Azken eragozpen hau ebazteko SE sofistikatu batzuek partizio segmentatuak (albokoak izan gabekoak) onartzen dituzte eta, horrela, memoria osoa probetxa daiteke momentu guztietan. Partizio segmentatuak maneiatzeko, SEa zaildu eta luzatu egiten da, segmentazioak programako helbideak aldatu egiten bait ditu eta SEak helbideen itzulpenaz arduratu behar bait du.

7.8.3.- MULTIATAZA (MULTITASKING)

CPU bakar batez programa baten ataza desberdinen egikaritzapen konkurrente edo paraleloari Multiataza deitzen zaio.



7.8 irudia.- B ETA D PROGRAMAK AMAITU ONDOREN 68 k LIBRE EGONDA EZIN DA 50 k DUEN E PROGRAMA KARGATU.

Ataza, programa baten exekuzio bat da eta, beraz, multiataza dagoenean programa berbereko exekuzio desberdinak paraleloan burutzen dira, baina paralelotasuna edo konkurrentzia itxurazkoa da. Multiataza beharrezkoa da Denbora Errealeko programetan. Adibidez, bankuetxe batean kontu korronteak maneiatzeko programan bi eragiketa ia ia batera iristen badira, multiataza egon gabe bigarren eragiketa ez da prozesatuko lehenengoa prozesatzen bukatu arte, erantzun-denbora luzatuz, multiataza badago, berriz, lehenengo eragiketa prozesatzen den bitartean gertatzen diren eten guztien denbora bigarren eragiketa prozesatzeko erabiltzen dira, erantzun-denbora laburtuz.

Hori dela eta, multiataza, programa baten exekuzio desberdinen arteko multiprogramaziotzat jo daiteke, multiatazan egikaritzen den programak jeneralean beste programekin multiprogramazioan lan egiten duelarik.

Programa batek, multiatazan egikaritzeko, BERRABIAGARRIa izan behar du. Programa berrabiagarri batean, exekuzio desberdinek datu-alderdi diferenteak erabiltzen dituzte, agindu-alderdia bakarra bada ere. Lehen aipatutako adibidean, kontu korronteak maneiatzeko programa berrabiagarria ez balitz, lehenengo exekuzioan eten bat gertatzean, bigarren exekuzioak bere datuak bereganatuko lituzke lehenengo exekuzioarenak ezabatuz eta, honen ondorioz, lehenengo exekuzioa okerki burutuko litzateke.

Bestalde multiataza kontrolatzeko SEa zailago eta luzeagoa bihurtzen da memoria-premia handituz. Multiprogramazioan bezala, progamaren exekuzio desberdinetako sarrera edota irteerako dispositiboek diferenteak izan behar dute edo spooling-a erabili behar da.

7.8.4.- DENBORA BANATUA (TIME SHARING)

CPU batez modu elkarreragilean abiarazitako programa batzu paraleloan egikaritzen direnean denbora banatuaz lan egiten dela esaten da.

Beraz, multiprogramazioaren antzekoa da baina paraleloan egikaritzen diren programak erabiltzaile desberdinek eskatu eta memoriaratzen dute. Multiprogramazioan bezala, memoria zatitzen da, konkurrentzia itxurazkoa da, baina planifikazioa elkarreragilea da eta ez dago lehentasun-sistemarik MULTIPLEXAZIOA baizik.

Denbora banatuaz, erabiltzaile bakoitzak bera konputagailuaren erabiltzaile bakarra izango balitz bezala lan egiten du, baina konputagailuaren potentzia erabiltzaileen artean banatzen da. Horretarako erabiltzaile bakoitzari SEan PROFIL bat egokitzen zaio.

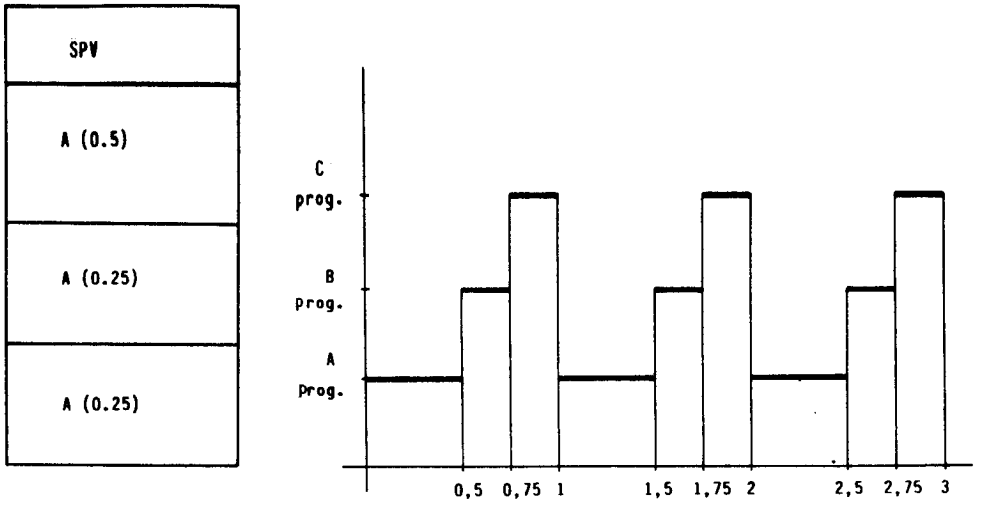
Profilean erabiltzaileak erabil ditzakeen konputagailuaren baliabideak zehazten dira, hau da, CPUko denbora-tartea, memoriako edukiera, dispositiboak, oinarrizko softwarea etab. Ezaugarri garrantzitsuena CPUko denbora-tartea da, multiplexazioan zerrikusi handia bait du.

Esan dugunez, denbora banatuaz ez dago lehentasun-sistemarik, hau da ahal den guztietan egikaritzen den lehentasun handieneko programa bat, hori etenik dagoenean egikaritzen den beste bat, etab., ez dago, baizik eta multiplexazioaren bidez tartekatzen dira programen egikaritzapena. Adibidez memorian A, B eta C programak egikaritzeko prest badaude SEak programa bakoitzari dagokion profilaren arabera, CPUko denbora-tartea begiratuz hain zuzen ere, denbora-unitate bakoitza hiru programen artean banatzen du bakoitzari tarte bat asignatuz.

Demagun A programari 1/2 bat asignatzen zaiola eta B eta C-ri 1/2 bat eta denbora-unitatea ms-a dela. Gertatuko dena hauxe da (ikus 7.9 irudia): CPU-k 0.5 ms-tan A programaren aginduak egikaritzen ditu, baina denbora hori bukatzen denean, nahiz eta programa etenik ez egon, gainbegiraleak B programa abiarazten du eta 0.25 ms-tan B programaren aginduak exekutatzen dira. Tarte hau bukatzen denean C programaren aginduak egikarituko dira gainbegiralearen kontrolpean eta C-ri dagozkion 0.25 ms-ak bukatutakoan gainbegiraleak A'programari kontrola utziko dio, esan duguna errepikatuz.

Programa baten exekuzio-tartean etenik gertatuko balitz CPUa ezer egin gabe gera ez dadin, gainbegiraleak kontrola beste programa bati emango lioke.

Denbora banatuan programen lehentasunak erabiltzaileen profilaren funtzioan finkatu egiten dira, baina lehentasun hauek ez dira absolutuak, multiprogramazioan gertatzen zen bezala, baizik eta



7.9 irudia.- HIRU PROGRAMAREN DENBORA BANATUZKO EGIKARITZAPENA

lehentasun handiagoak denbora-tartea luzeagoa erakartzen du.

Denbora banatua, gero eta erabiliagoa da, gaur egun prozesu-mota erabiliena bihurtzen ari delarik. Kalkulu-zentru askotan programatzaile bakoitzak bere teklatua eta pantaila ditu, bere programak itzuli, estekatu eta probatzeko, beste erabiltzaileekin denbora banatuan lan eginez.

Beste kasu tipikoa, erabiltzaile independienteak konputagailu zentral batekin lotuta daudenekoa da. Erabiltzaile bakoitzak sarrera eta irteerako zenbait dispositibo du datuak sartzeko eta emaitzak lortzeko eta, linea telefonikoaren bidez, konputagailu zentrala erabiltzen dute programak garatu eta egikaritu ditzaketelarik. Konputagailuaren erabilpenarengatik erabiltzaileak pagatzen du. Modu honen bidez injineru eta arkitekto askok lan egiten dute, zeren eta haien programa konplexu eta luzeetarako mikro bat ez bait da nahikoa eta konputagailu handi bati ez bait lioteke errendimendu handirik aterako, oso garestia gertatuz.

Denbora banatuaren abantaileei, konputagailu bat zenbait erabiltzailek modu elkarreragilean erabil dezala alegia, zenbait eragozpen kontrajartzen zaie. Oro har, eragozpen hauek multiprogramazioak zeuzkan berberak dira, SE konplexuagoa, programa birkokagarriak izatearen beharra, memoria nagusi handiagoaren beharra eta dispositiboekiko gatazka, azken bi eragozpen hauek gainditzeko alegiazko memoria eta spool-a erabiltzen direlarik. Hala eta guztiz ere, denbora banatuan jeneralean erabiltzaile bakoitzak bere sarrera eta irteerako zenbait dispositibo eduki behar du, ondorioz dispositibo-gatazka handiena bitarteko memoriarena da, eragozpen hau profilaren bidez ebazten delarik.

Beraz, denbora banatuak eta multiprogramazioak ezaugarri asko amankomunean dute, baina honako ezaugarri hauek bereizten dituzte:

- Denbora banatuan modu elkarreragilean lan egiten da, aldiz, monoprogramazioan modu zentralizatuan, beraz, denbora banatuan erabiltzaileek zuzenean kargatzen dituzte programak.

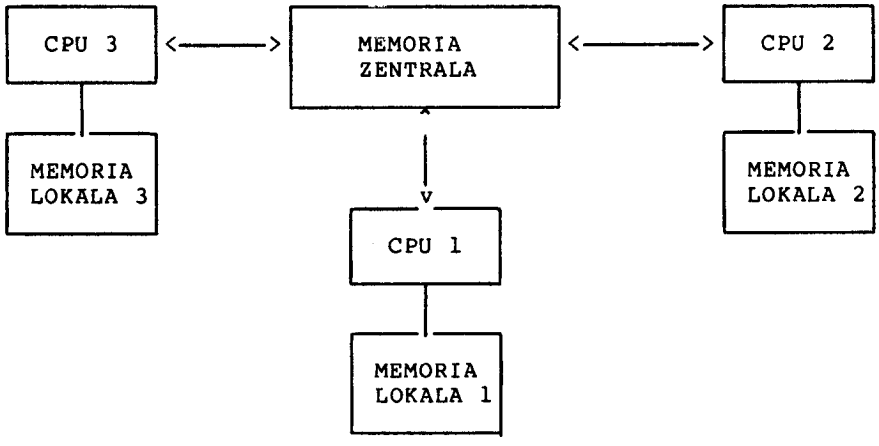
- Denbora banatuan CPUko denbora multiplexatzen da kargatutako programen artean eta multiprogramazioan berriz momentu guztietan lehentasun handiagoko programak egikaritzen saiatzen da sistema.

7.8.5.- MULTIPROZESUA

CPU batzuren arteko lan konbinatuari multiprozesua deitzen zaio.

CPU bat baino gehiago dagoenez konkurrentzia erreala gertatzen da, hau da, momentu berean eragiketa edo agindu bat baino gehiago burutzen dira.

CPU hauek lana konbinaturik egiteko elkarri informazioa bidali behar diote eta horretarako memoria nagusi bat amankomunean dute, bakoitzak beste memoria lokal bat eduki badezake ere (ikus 7.10 irudia). Beraz, multiprozesua ez da konputagailu batzuek batere loturarik gabe lan egitea, baizik eta hauek monoprogramazioan lan egiten duten konputagailu batzu lirateke.



7.10 irudia.- MULTIPROZESUAREN ESKEMA 3 CPU ERABILIZ

CPUek egikaritzen dituzten eragiketen arteko erlazioaren arabera ondoko multiprozesu-mota desberdin hauek bereizten dira:

- ATAZEN ARTEKO MULTIPROZESUA.- CPU bat programa bat egikaritzen hasten denean ataza hori CPU horrek bakarrik exekutatzen du, bitartean beste CPUek beste programak edo programa bera beste datu batzurekin (beste ataza bat) egikarituko dituztelarik. Honetarako programak memoria zentralean edo lokaletan egon daitezke eta datuak eta emaitzak memoria lokaletan, ataza bakoitza zer CPUk egikarituko duen SEk erabakiko duelarik. Multiataza eta multiprogramazioa konkurrentzia errealean lirateke.

- AGINDUEN ARTEKO MULTIPROZESUA.- Programa bat egikaritzen denean prozesu horren zenbait agindu batera egikaritzen dira programa horren exekuzioa laburtuz. Kasu honetan CPUen arteko sinkronizazioa zaila gertatzen da zeren eta agindu batzu egikaritzeko beste batzuren emaitza behar bait da. Beraz bi irtenbide daude, programatzaileak zer agindu exekuta daitezkeen batera zehaztea programan edo SEan azken hau erabakitzea oraindik ondo garatu gabe dagoelarik. Mota honetan ez da memoria lokalik erabili ohi CPU bakoitzak datuak memoria zentraletik eskuratzen bait ditu eta emaitza berriro utzi.

- MIKROAGINDUEN ARTEKO MULTIPROZESUA.- Agindu bat burutzeko unitate bakoitzak mikroagindu bat egikaritzen du, beraz, CPUa zenbait unitate berezik osatzen dute, teknologia honi PIPE-LINE deritzolarik. Agindu bakoitzaren exekuzioa muntai-katea batean bezala aurreratzen du. Adibidez, bilatze-faseaz unitate bat arduratzen bada eta exekuzio-faseaz beste bat, lehenengo aginduaren exekuzio-fasea burutzen den bitartean bigarren aginduaren bilatze-fasea burutu daiteke, aginduak azkarrago egikarituz.

Azken mota hau ez da multiprozesutzat hartzen CPUan zenbait unitatek osatzen badute ere, CPU azkarrago bakar bat ematen bait du.

Multiprozesuaren helburua abiadura eta fidagarritasuna handitzea da, nahiz eta horrek kostuaren igoera erakarri. Sinkronizaziorako denbora (sinkronizazioa SEaren funtzioa da) eta memoria zentralerako atzipenengatik abiadura ez da izango CPU bakarrarekin zena bider CPUen kopurua, baina ez da izango hori baino askoz motelagoa.

Fidagarritasunaren handitzeak, CPU bat hondatzen bada sistemak hala ere lanean jarrai dezakeela esan nahi du, CPU bakarra dagoenean berriz, CPUa hondatuz gero sistema geldirik geratzen da. Kostuaren gehikuntza kontutan hartzekoa da, baina memoria eta dispositibo gehienak amankomunean daudenez CPU batzuren kostuaren gehikuntza hau beste horrenbeste konputagailuen erosketarena baino askoz txikiagoa da.

Azkenik SEa zaildu egiten dela ere nabaria da, zeren eta SEk CPUen arteko sinkronizazioaz eta memoria zentralaren kontrolaz arduratu behar bait du, CPUren bat hondatzen bada eginkizunak berbanatuz.

7.9.- BESTE ZENBAIT KONTZEPTU

Aurreko pasartean azaldutako kontzeptu berriak aztertuko ditugu, Alegiazko Memoria eta Spooling-a alegia.

7.9.1.- ALEGIAZKO MEMORIA (VIRTUAL STORAGE)

Konputagailu batean kargaturik dauden programek dagoen memoria nagusiaren edukina baino gehiago hartzen dutenean Alegiazko Memoriaren bidez lan egiten da.

Alegiazkoa edo birtuala deitzen da zeren eta egikaritzeko prest dauden programak memoria nagusian osoki egon gabe, baleude bezala funtzionatzen bait dute. Horretarako programak memoria nagusian egon beharrean, bitarteko memoria berezi batean kokatzen dira (diskoan), programa hauen informazioa behar ahala memoria nagusira barneratuko delarik.

Alegiazko memoria dagoenean Sistema Eragileak kudeatzen du ondoan adieraziko den funtzionamendua jarraituz. Bitarteko memorian dagoen informazioa luzera finkoko zatitan banatzen da zati bakoitzari ORRIA deitzen zaiolarik. Memoria nagusia ere orrien luzera duten zatitan banatzen da, memoria nagusiko zati bakoitzari BLOKEA deituko diogularik.(ikus 7.11 irudia). Orrien luzera hauta daiteke baina jeneralean K bat eta 8 K arteko luzera erabiltzen da.

Behar diren neurrian bitarteko memorian dauden orriak memoriako blokeetan kokatzen dira (ORRI-BARNERAKETA). Memoria nagusia beterik dagoenean eta beste orriren bat barneratu behar denean bloke batean dagoen orri bat kanporatu behar da, bitarteko memorian gordez (ORRI-KANPORAKETA).

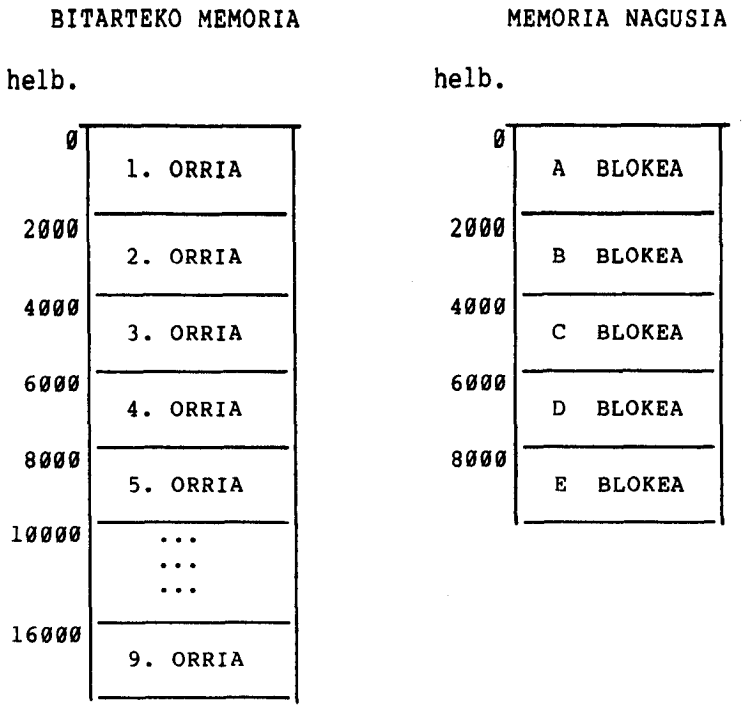
Funtzionamendua aztertzeko 7.11 irudian dagoen adibideari jarraituko gatzaizkio, beraz, bitarteko memorian 9 orri daude, baina memoria nagusian 5 bloke besterik ez dago. Normalean programa bat baino gehiago egongo badira ere, azalpena errazteko 9 orriek programa bakar bat osatzen dutela suposatuko dugu, funtsean funtzionamendua berdina da eta.

Hasiera batean memoria nagusiko blokeetan lehenengo orriak kokatuko dira eta adibidean urrats hau emanez gero 7.12 irudian dagoen egoera gertatuko da.

Programa egikaritzen hasten da eta pentsa dezagun 2. orriko agindu batek 7. orriko helbide batera jauztea agintzen duela. Emango diren pausoak ondoko hauek izango dira (ikus 7.13 irudia):

1) Memoria nagusiko zer orri kenduko den erabaki (adibidez 3.a).

2) Erabakitako orria bitarteko memorian gorde beharrezkoa denean



7.11 irudia.- ALEGIAZKO MEMORIAKO ADIBIDEA (9 orri, 5 bloke eta 2000 gelaskako orriak)

(ORRI-KANPORAKETA).

3) Barneratu behar den orria (7.a gure kasuan) bitarteko memoriako zer helbidetan dagoen jakin.

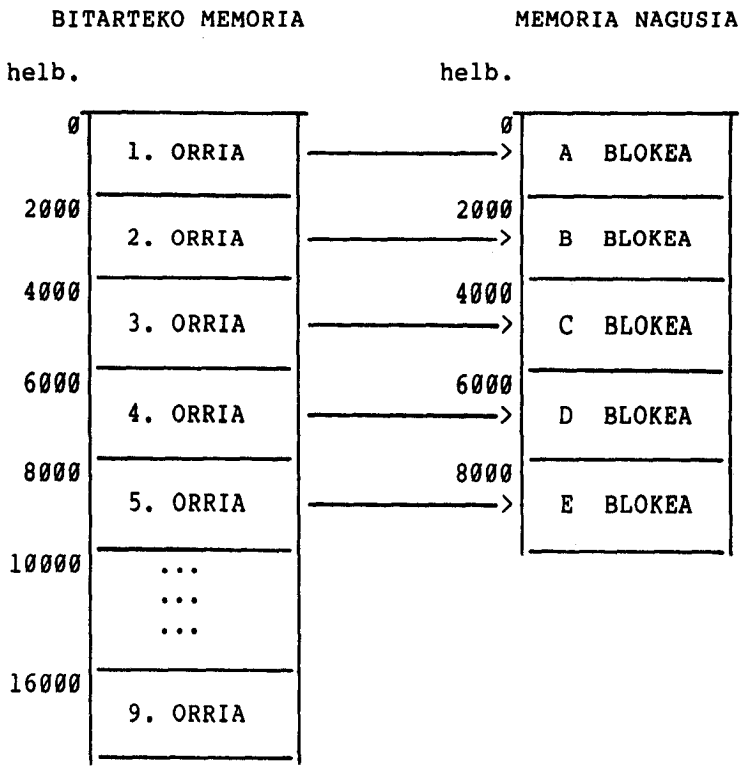
4) Orria bitarteko memoriatik memoria nagusira eraman (ORRI-BARNERAKETA).

5) Helbide birtualak edo alegiazkoak erreal bihurtu (HELBIDE-ITZULPENA), bestela sistemak ez bait luke funtzionatuko.

Lehenengo 4 urratsek ORRI-ALDAKETA osatzen dute eta SEak softwarearen bidez burutzen ditu, helbide-itzulpenean berriz hardwareak lagutzen dio SEari. Orri-aldaketa burutzeko, SEak bi taula erabiltzen ditu orri-taula eta bloke-taula.

ORRI-TAULAn alegiazko memoriako orri bakoitzari osagai bat egokitzen zaio, osagai bakoitzean ondoko eremu edo datu hauek daudelarik:

- Orri-zenbakia (identifikadorea).
- Egoera (memoria nagusian dagoen ala ez).
- Bloke-zenbakia (orri hau memoria nagusian badago zein bloketan).
- Bitarteko memoriako helbidea.



7.12 irudia.- ALEGIAZKO MEMORIAKO ADIBIDEAREN HASIERAKO EGOERA

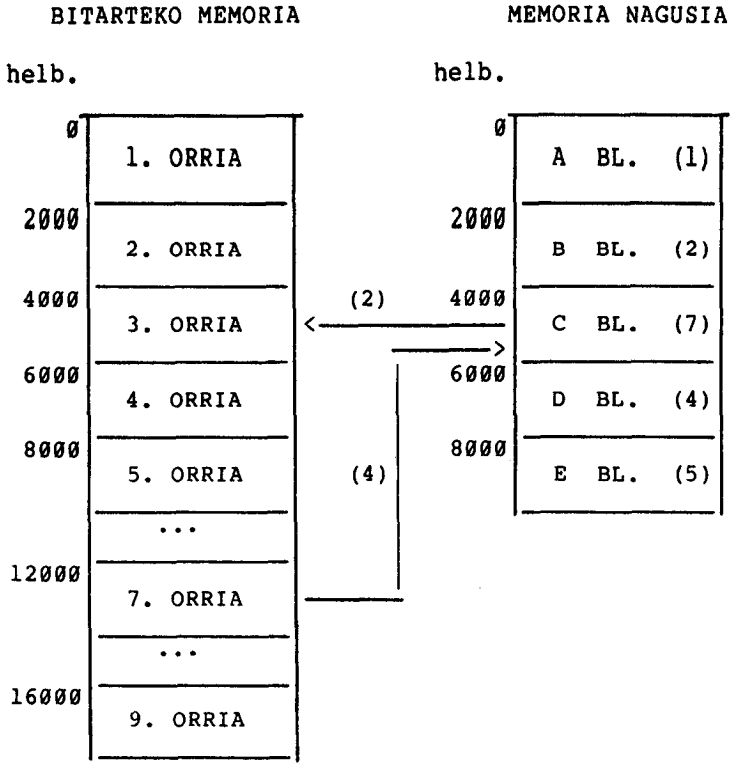
Taula honen bidez orri-aldaketaren 2. 3. eta 4. pausoetako eragiketak burutzen dira.

BLOKE-TAULAn benetako memoriako bloke bakoitzari osagai bat egokitzen zaio, osagai bakoitza ondoko eremu hauek osatzen dutelarik:

- Bloke-zenbakia (identifikadorea).
- Orri-zenbakia (blokean kokaturik dagoen orria).
- Gertaturiko aldaketak (bloke honetan edukina aldatu den ala ez, aldaketarik egon ez bada bloke honetako orria kendu behar denean ez bait da salbatu behar).
- Atzipen-kopurua (kontagailu bat da eta normalean atzipen-kopuru txikiena daukan blokearen orria kanporatuko da).

Taula hau orri-aldaketaren 1. eta 2. urratsetan kontsultatuko da.

Alegiazko memoriaren erabilpenak memoria nagusiaren beharra murriztu egiten du multiprogramazioaren eta denbora banatuaren erabilpena merkeagotuz, alegiazko memoriarik gabe prozesu-mota hauek oso garestiak bihurtzen bait dira. Izan ere, SEa konplexuagoa bihurtzeaz gain alegiazko memoriaren erabilpenak sistemaren abiadura moteltzen bait du, zeren eta orri-aldaketak burutzen dituenean CPUk ez du erabiltzailearen programarik egikaritzen. Hala



7.13 irudia.- ALEGIAZKO MEMORIAKO ADIBIDEAN, ORRI BATEN ALDAKETA (3.aren ordez 7.a)

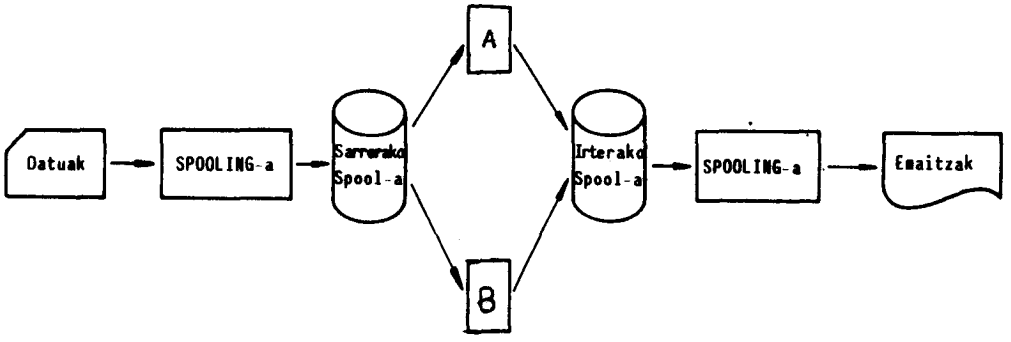
eta guztiz ere, alegiazko memoriaren edukina eta benetazko memoriarenaren arteko proportzioa zaintzen bada (alegiazko memoriaren edukinak ez du izan behar 4 aldiz benetako memoriarena baino handiagoa) abiaduraren oso degradazioa txikia nabaritzen da, alegiazko memoriaren erabilpena guztiz egokia izanik.

7.9.2.- SPOOLING-A

Programak egikaritzen direnean bere Sarrerako dispositiboetatiko irakurketak bitarteko memoriatik burutzen direnean edota Irteerako dispositiboetarako idazketak bitarteko memoriara, Spooling-sistema batean gaudela esaten da.

Sarrerako eta irteerako dispositiboak (inprimagailuak batez ere, txartel-irakurgailuak, etab.) oso motelak, eskasak eta erabiliak dira. Beraz, multiprogramazio edo denbora banatuaren bidez lan egiten denean egikaritzen diren programek dauden dispositibo motel hauen kopurua baino handiagoaren beharra dute. Mota honetako dispositibo gehiago erosi beharrean spooling-a erabiltzen da, alegiazko dispositiboen sistema ere deitzen zaiolarik.

Programek sarrerako dispositiboetatik datuak eskatu baino lehen Spoolingak datuak irakurtzen ditu bitarteko memorian (diskoan) utziz. Programak egikaritzen direnean sarrerako dispositiboetatiko irakurketak spool-diskotik burutzen dira automatikoki (programa aldatu gabe) eta irteerako dispositiboetarako irteerak, hauek ere automatikoki, spool-diskora kanporatzen dira, geroago Spoolingak behar diren dispositiboetara kanporatuko dituelarik (ikus 7.14 irudia).



7.14 irudia.- SPOOLING-AREN EGITURA

Spooling programa beste programekin konkurrentzian egikaritzen da, multiprogramazio edo denbora banatuaren bidez. Hortaz, spoolinga erabiltzen denean partizio bat gehiago dago, memoriako beharra handiagotuz.

Spoolingaren abantailak, alde batetik dispositibo motelen beharra murriztea, multiprogramazioa edo denbora banatuaren erabilpena merkeagotuz, eta beste alde batetik dispositibo hauen errendimendua igotzea dira. Irteerako spoolinga askoz gehiago erabiltzen da sarrerakoa baino, batez ere inprimagailuekin, non txostenak errepika bait daitezke.

Spoolingaren eragozpen handiena alegiazko memoriarena bera da, alegia, sistema-abiaduraren degradazioa. Programak, dispositibo motelak erabitzen ez dutenez, arinago exekutatzen dira, baina spoolingh programaren exekuzioa kontatuz, denbora luzatzen da. Izan ere, abiadura ez bait da asko degradatzen eta konputagailu gehienetan erabiltzen da spoolinga.