

SARRERA-IRTEERA AZPISISTEMA

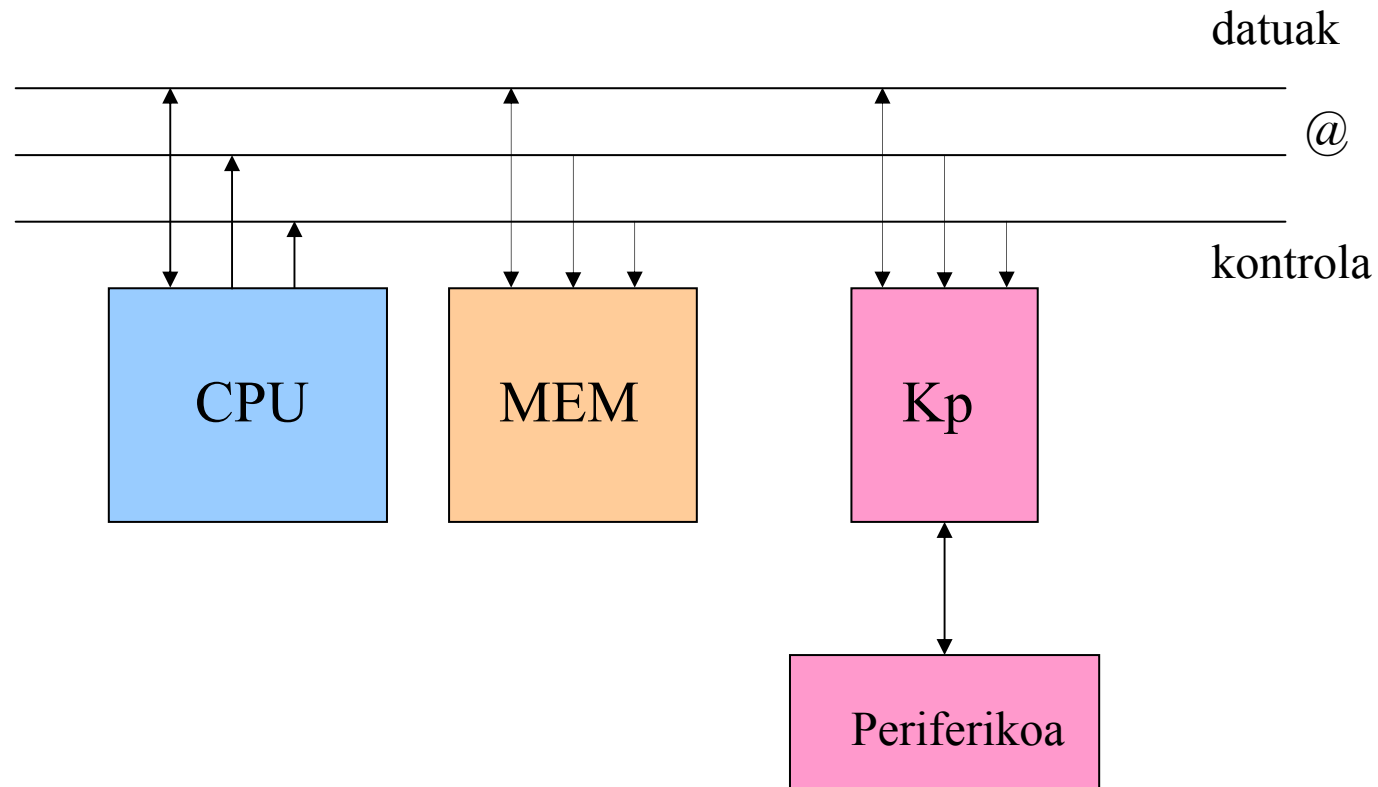
Konputagailuen Arkitektura I

4. gaia

S/I interfasearen deskribapena

- Helburua
 - Nola komunikatzen den CPU kanpoaldearekiko
 - Nola kontrolatu komunikazio hori
- Periferikoak
 - Datuak aurkezteko (pantaila, inprimagailua,...)
 - Datuak eskuratzeko (teklatura, sentsoreak,...)
 - Datuak gordetzeko (diskoak, zintak,...)
 - Bestelakoak (motoreak, balbulak,...)
- Konexioa (bus bakarra, bi bus)

S/I interfasearen eskema



Kontroladorea

- Justifikazioa

periferiko mota asko, busetara konexio fisiko desberdinekin eta elektronikoki era desberdinetan kontrolatuak

- Kontroladorea

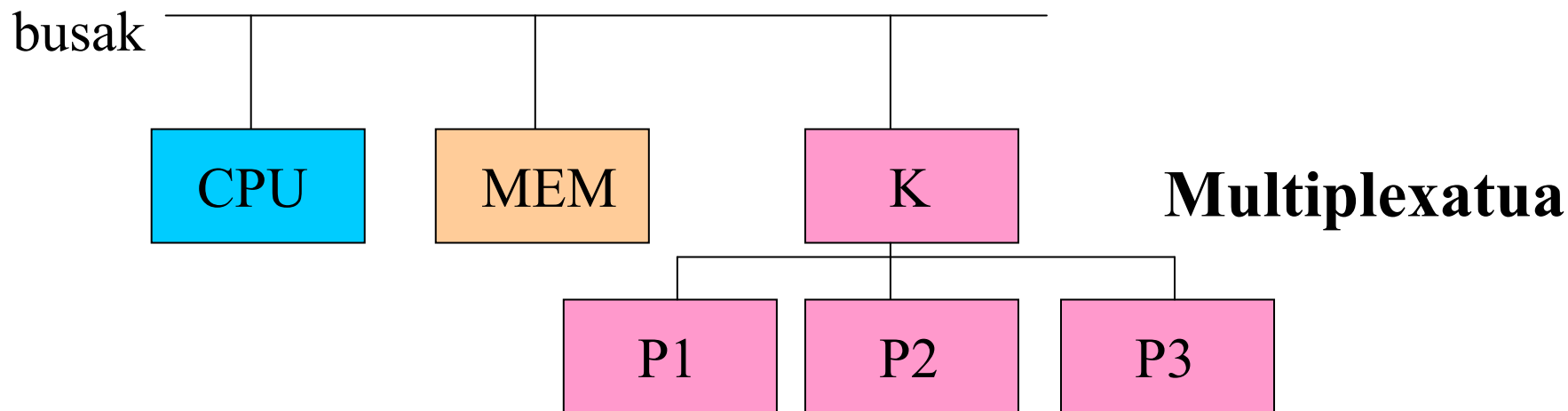
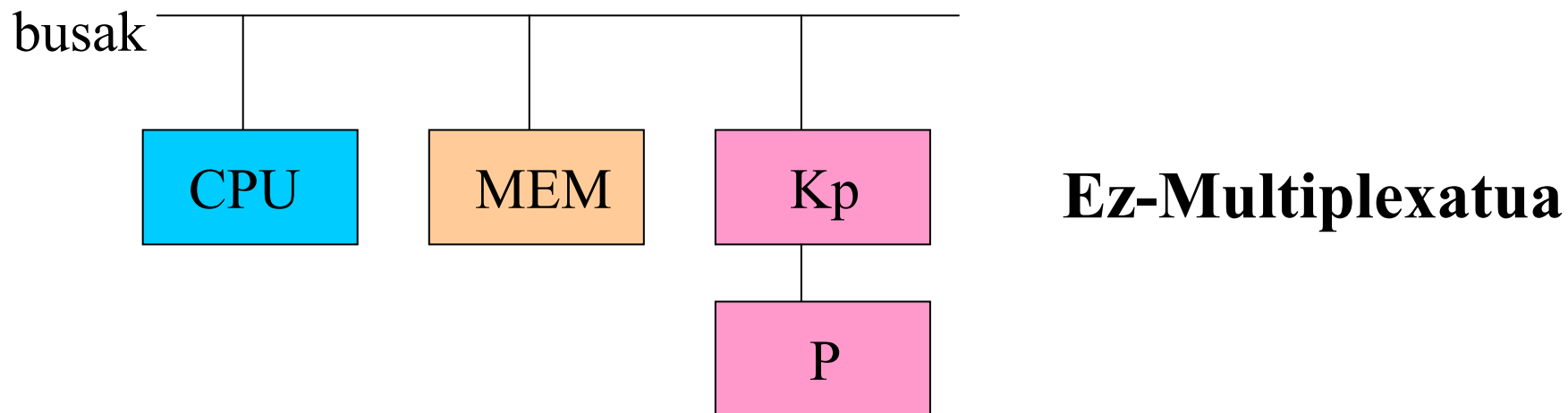
Busak eta periferikoaren arteko tartekari elektronikoa da. ML-ren mailatik erregistro batzuk bezala ikusten da. Kontroladoreak lana kentzen dio CPU-ri, honek ez baitu zuzenki kontrolatu behar periferikoa

CPU eta periferikoaren arteko komunikazioa kontroladorearen erregistroen bitartez egiten da.

S/I-ko kontroladoreen sailkapena

- Ez-multiplexatua
 - Periferiko bat kontrolatzen du soilik: periferiko azkarrak
- Multiplexatua
 - Periferiko batzuk kontrolatzen ditu
 - Karakterez multiplexatua: periferiko motelak
 - Blokez multiplexatua: erdi-mailako abiadura duten periferikoak

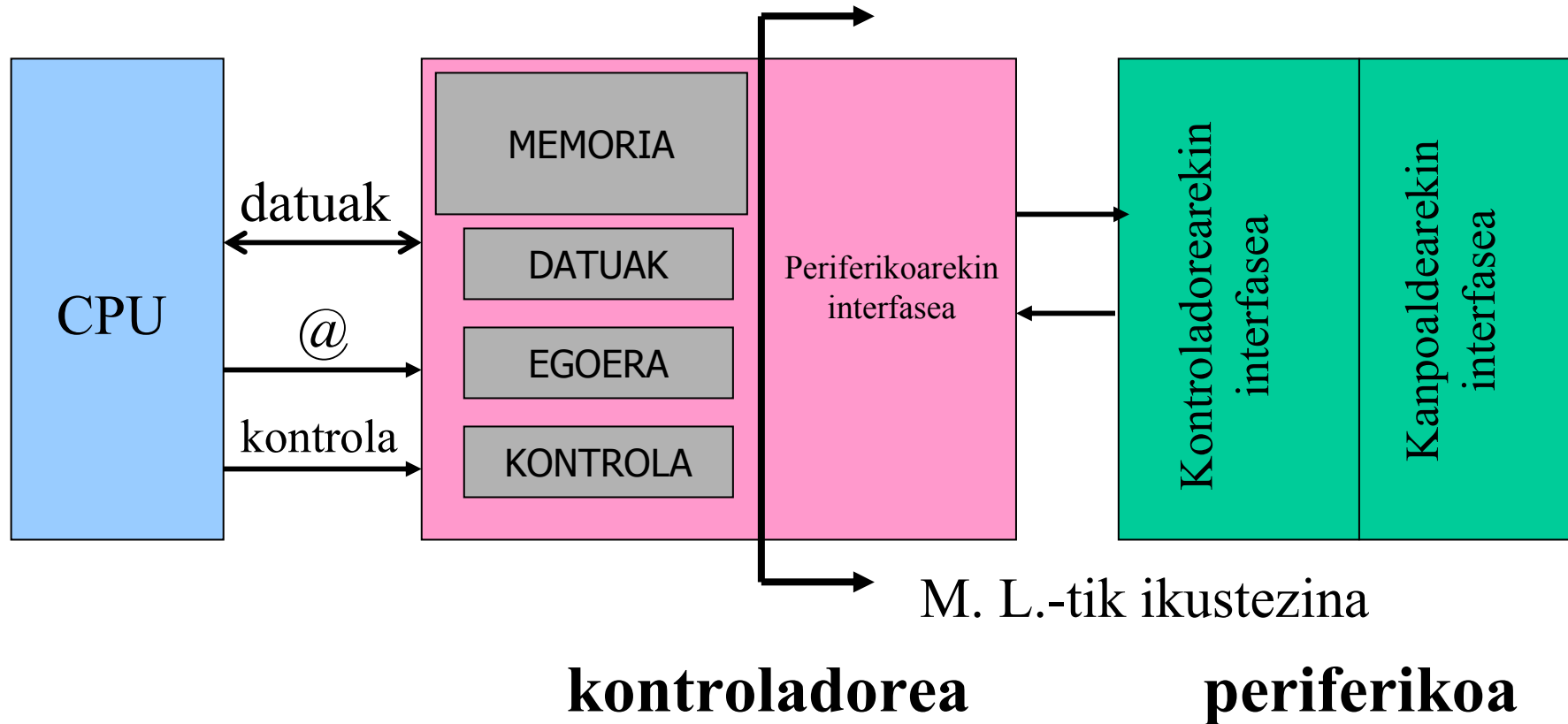
S/I-ko kontroladoreen sailkapena



S/I-ko kontroladorearen funtzioak

- CPU-rekin elkarrizketa
 - S/I-ko eragiketak betetzeko eskaerak jasotzen ditu
 - Periferikoen egoerari buruz informatzen du
- Periferikoaren kontrola
 - CPU-k eskatutakoa bete dezan
- Periferikoa eta CPU-ren arteko transferentzia errazten du

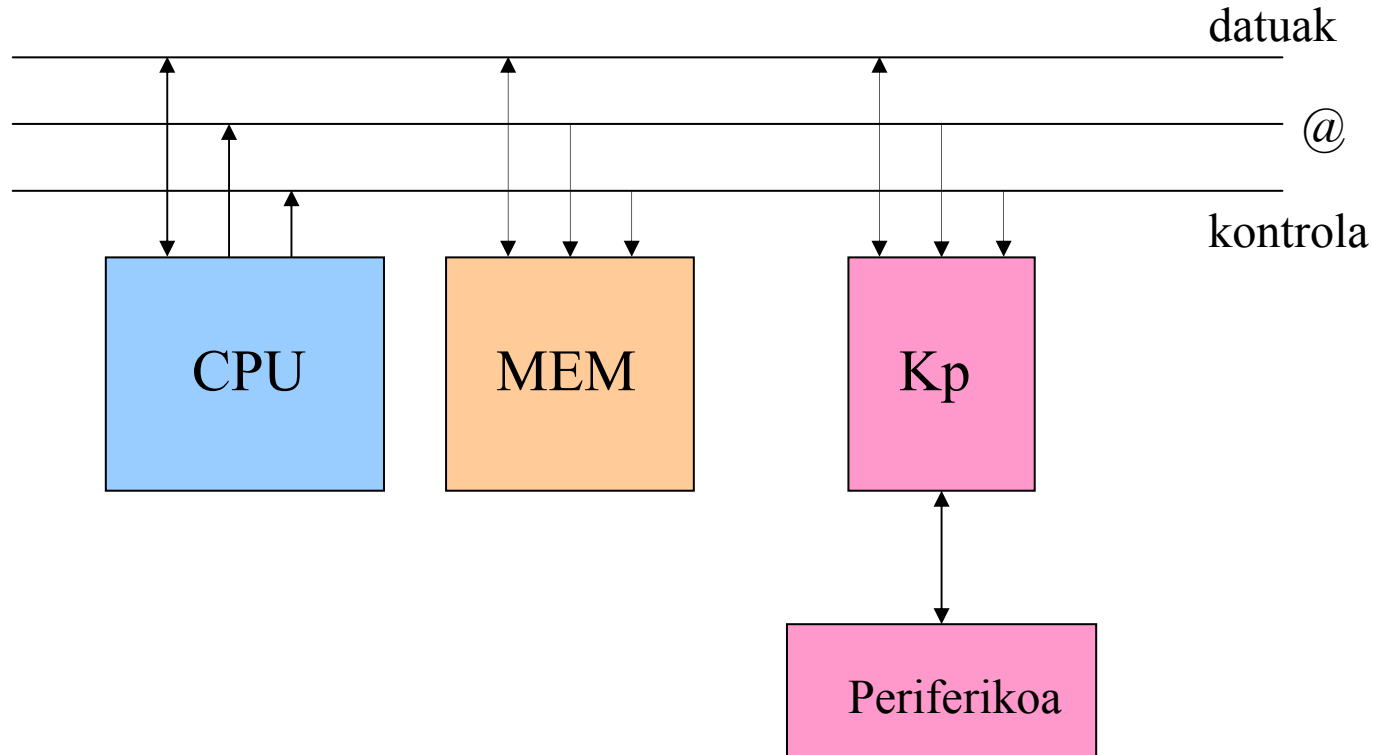
S/I-ko kontroladorearen ikuspegi funtzionala



S/I-ko kontroladorearen erregistroak

- Egoera-erregistroa
 - Egoerari buruzko informazioa: tekla bat sakatu den, diskoan idazketa bukatu den etab.
 - CPU-k irakurtzen du informazio hau
- Kontrol-erregistroa
 - Kontrol-informazioa: zein eragiketa bete behar den (r/w), periferikoaren lana egiteko era etab.
 - CPU-k idazten du informazio hau
- Datu-erregistroa
 - Periferikoa eta CPU-ren artean trukutzen den informazioa

S/I-ko interfasearen eskema



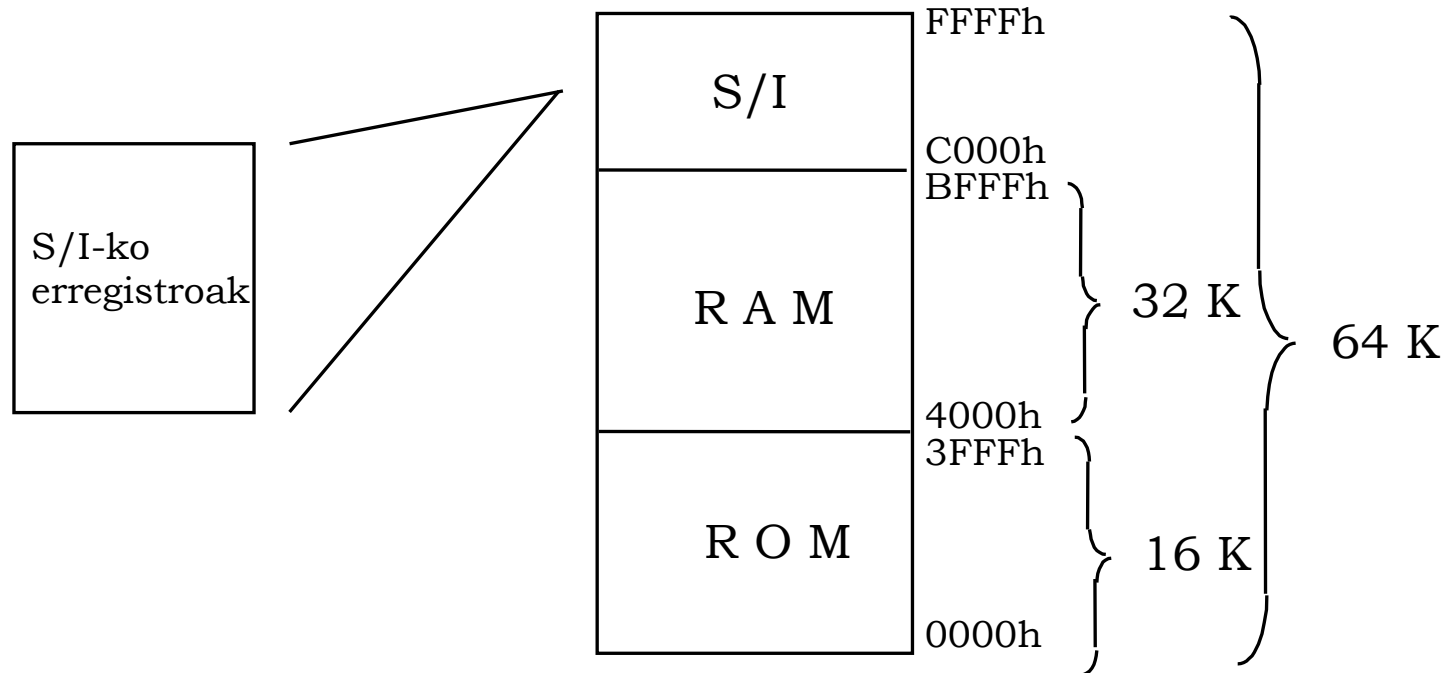
→ Nola atzitzen dira kontroladorearen erregistroak?

- S/I memorian mapeatuta
- S/I memorian ez mapeatuta

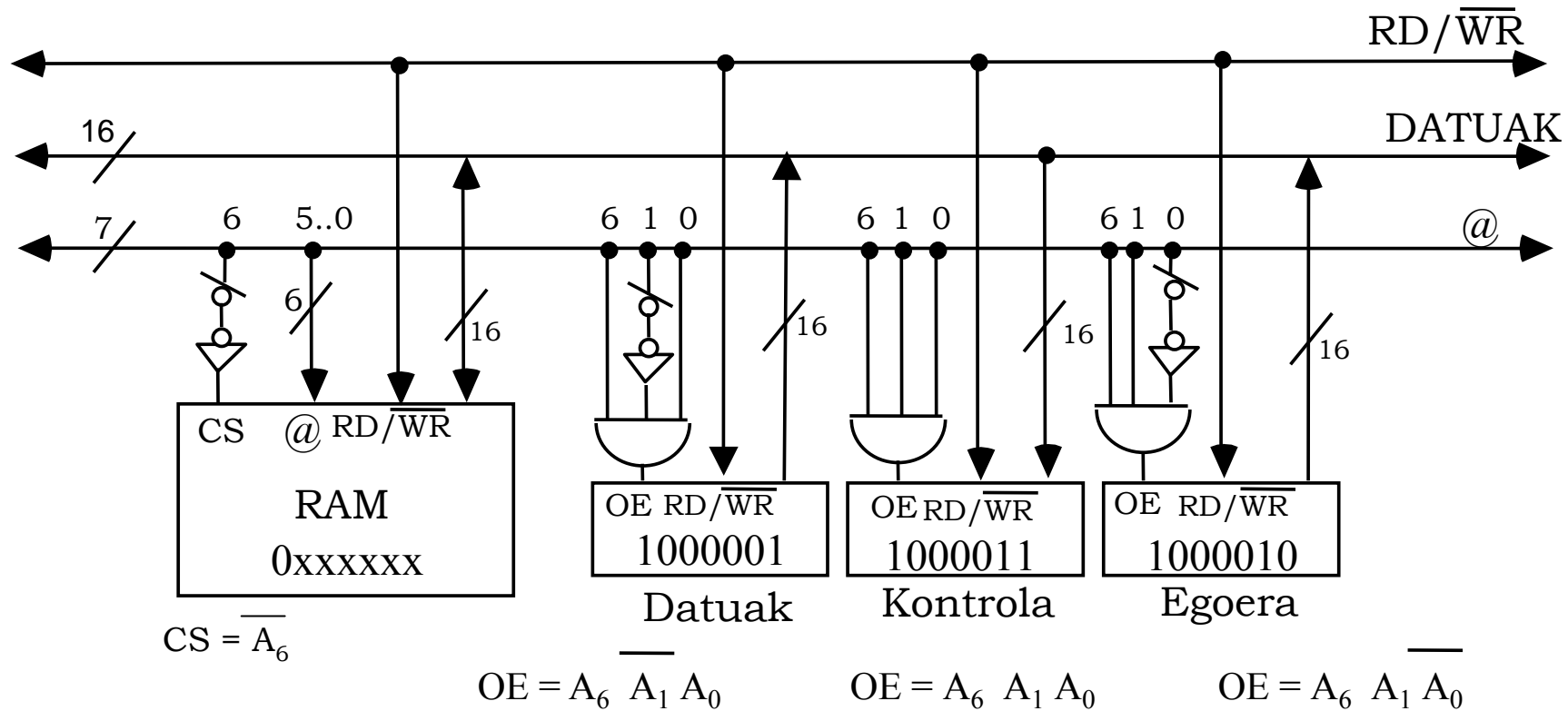
Memoria-mapa

S/I memorian mapeatuta

Memoria-mapa



S/I memoria mapeatuta: adibidea



RAM → 0000000

Datu-erreg. → 1000001

Kontrola-erreg. → 1000011

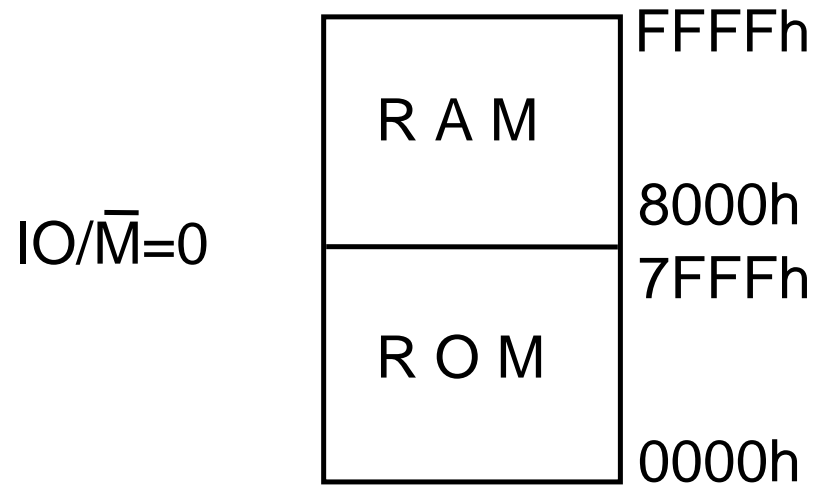
Egoera-erreg. → 1000010

S/I memorian mapeatuta

- Abantailak
 - Erregistroen atzipena: gainontzeko memoriarako erabiltzen diren agindu eta helbideratze-modu berdinak
 - CPU-k ez du desberdindu behar memoriako eta S/I-ko erregistroen atzipenen artean
- Desabantailak
 - Memoriako helbideratze-espazioa txikiagoa da
 - Zaila da kontrolatzen S/I-ko erregistroak ez atzitzea erabiltzailearen programetatik
 - Modu berezia (SE) S/I-ko atzipenearako

Memoria-mapa ↔ S/I-ko mapa

MEMORIA-MAPA



$$\text{RAM} \rightarrow \overline{\text{IO}/\overline{M}} * A_{15}$$

$$\text{ROM} \rightarrow \overline{\text{IO}/\overline{M}} * \overline{A_{15}}$$

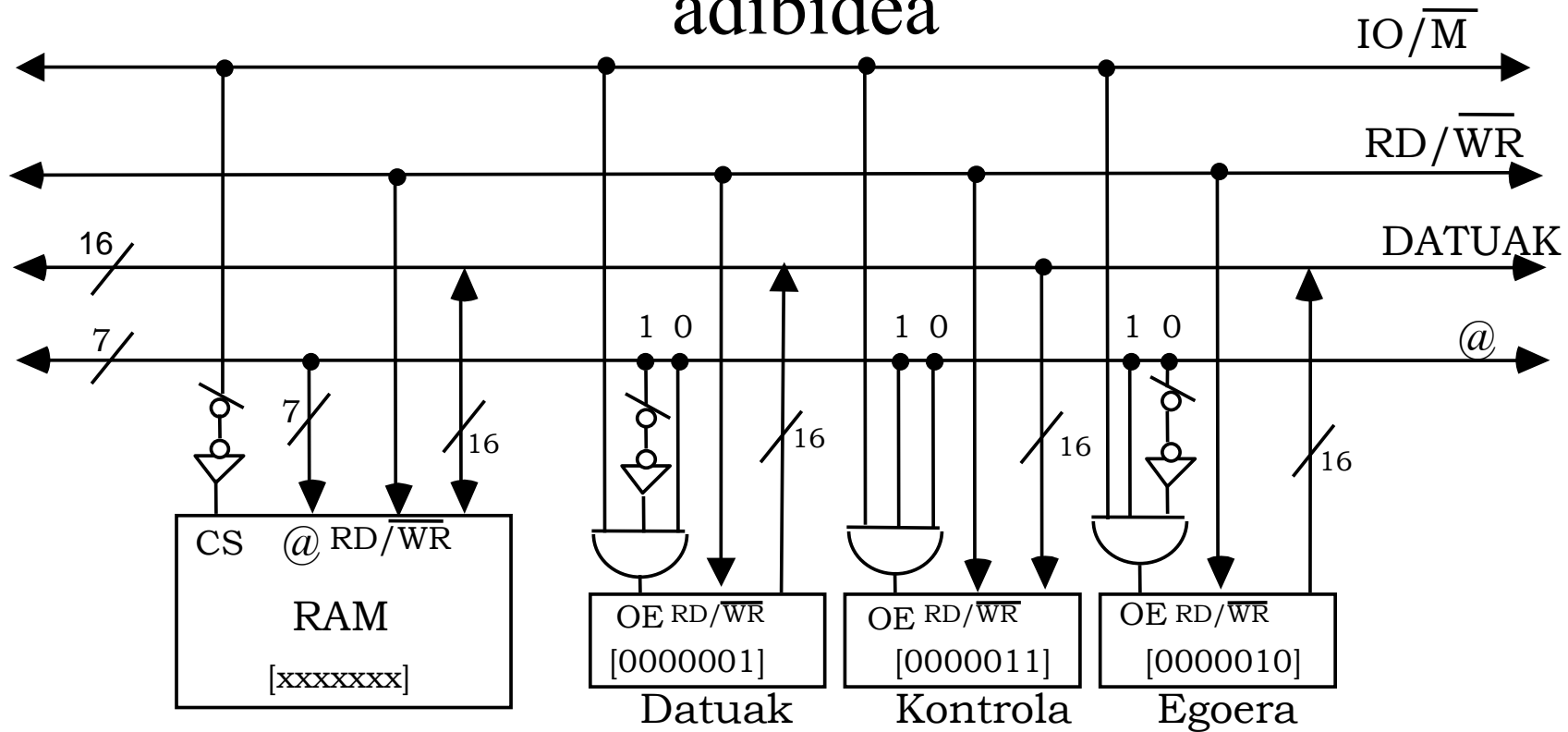
S/I-ko MAPA



$$\text{S/I Erreg.} \rightarrow \text{IO}/\overline{M}$$

S/I bereiztuta edo memorian ez mapeatuta:

adibidea



$$CS = \overline{IO/M}$$

$$OE = IO/M \overline{A_1} \overline{A_0}$$

$$OE = IO/M A_1 A_0$$

$$OE = IO/M A_1 \overline{A_0}$$

RAM → 0000000

Datu-err. → 0000001

Kontrol-err. → 0000011

Egoera-err. → 0000010

in/out aginduak

S/I memorian ez mapeatuta

- Abantailak
 - Ez da memoria-espaziorik galtzen
 - S/I-ko erregistroen atzipena kontrolatzea errazagoa da
→ *in* eta *out* modu berezian soilik (SE)
- Desabantailak
 - Agindu berriak: *in* eta *out*
 - Kontrol-seinale berria busean

[Ariketa_Mapak.pdf]

S/I-ko eragiketen sinkronizazioa

- Nola sinkronizatzen da CPU S/I-ko gailuekin? Nola detektatzen du gailua prest dagoela eragiketa bat hasteko?
- 2 aukera
 - **Inkesta** bidezko sinkronizazioa: CPU-k uneoro aztertzen du kontroladorearen egoera-erregistroa
 - **Etenen** bidezko sinkronizazioa: kontroladoreak adierazten dio CPU-ri prest dagoela eten baten bitartez

Inkesta bidezko S/I

- Egoera bitarteko inkesta → uneoro aztertzen da kontroladorearen egoera-erregistroa
- 2 aukera
 - **Inkesta jarraia (etengabekoa):**

```
while (egoera_erregistroa_irakurri==EZ_PREST) NOP;  
S/I_eragiketak_bete;
```
 - **Inkesta periodikoa (aldizkakoa):**

```
while (egoera_erregistroa_irakurri==EZ_PREST)  
    { beste_eragiketak; }  
S/I_eragiketak_bete;
```

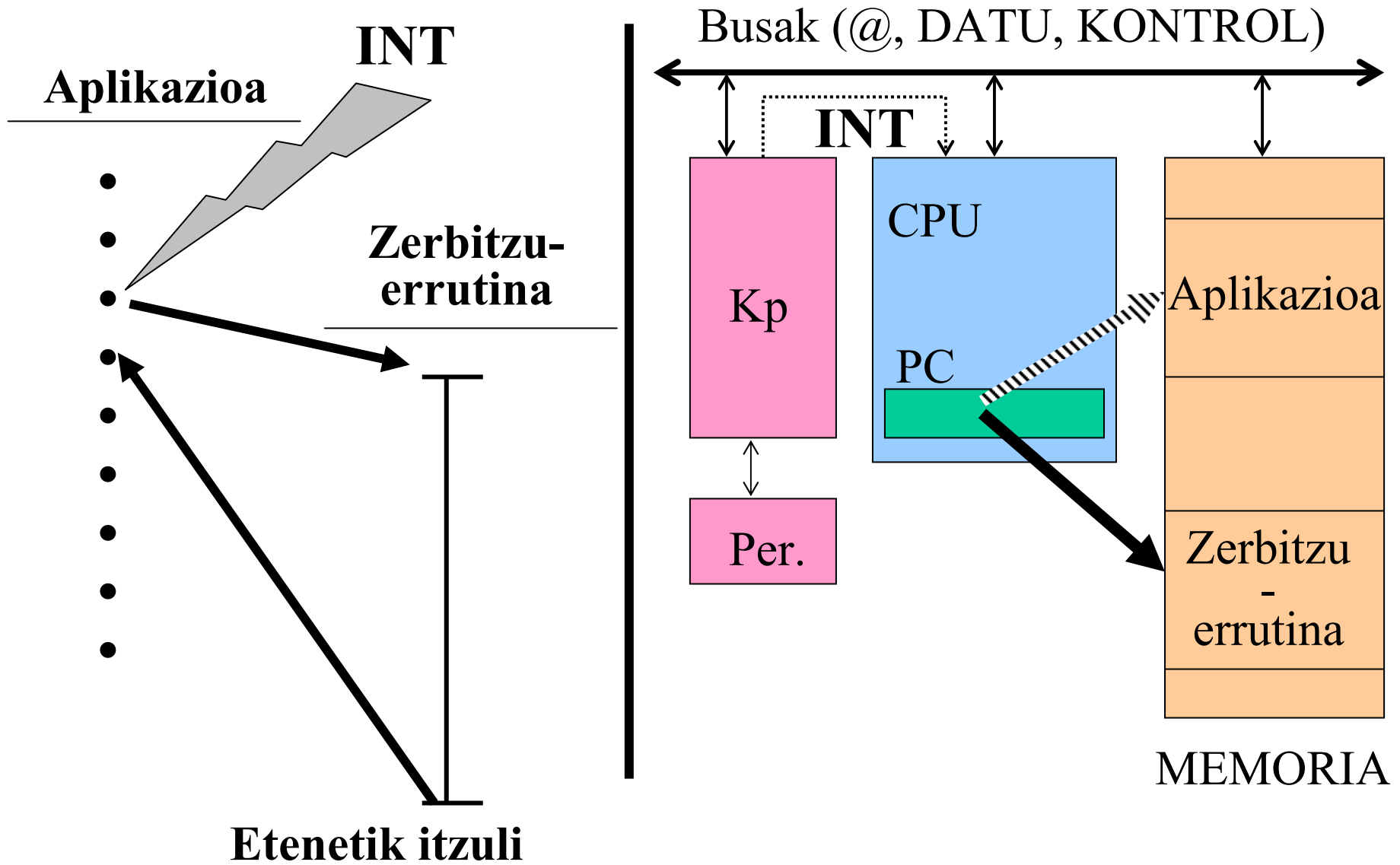
Etenen bidezko S/I

- Etena: prozesadorearen aditasuna behar duen ez-ohiko gertaera
- Gertaera asinkronoak dira: kontroladoreak CPU eteten du transferentzia hasteko prest dagoenean
- Abantaila:
 - CPU-k ez du denborarik galtzen inkesta egiten eta beste lanak egin ditzake
- Desabantaila:
 - CPU konplikatuagoa da (algoritmoa konplexuagoa)

Etenen bidezko S/I: faseak

- Eten-eskaeraren detekzioa
- Etendako programaren **egoera gorde**
- Etenari dagokion **zerbitzu-errutina identifikatu**
(gailuaren araberakoa)
- Zerbitzu-errutina exekutatu
 - **Errutina asinkronoa**: ez du programa nagusiak deitzen
- Etendako programaren **egoera berreskuratu**
(exekuzioa)

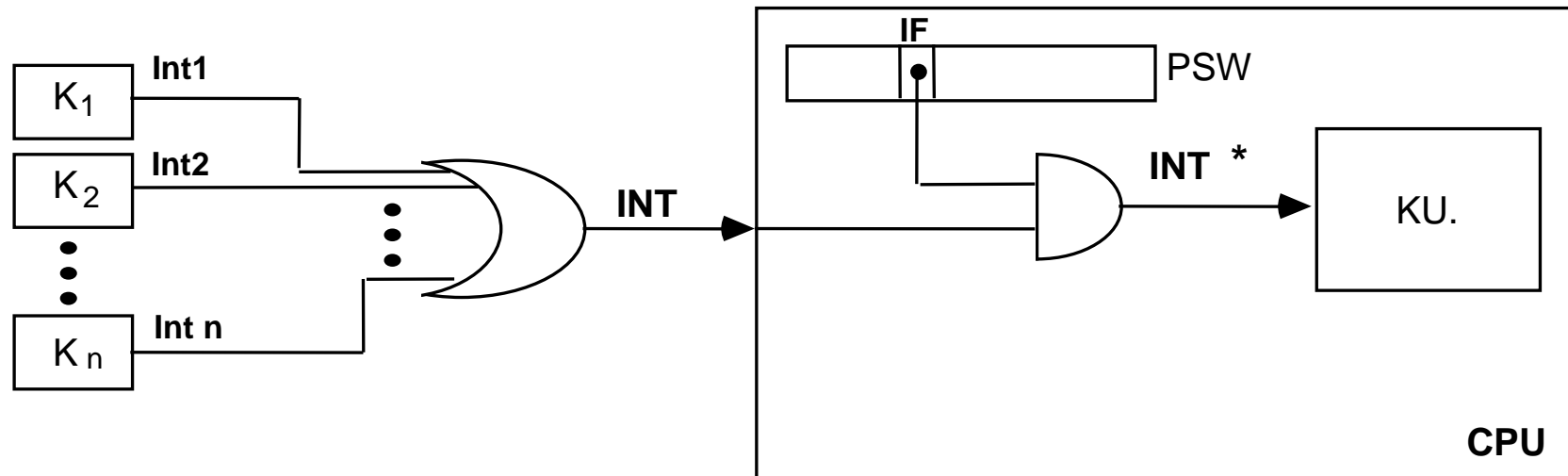
Etenen bidezko S/I



Etenaren detekzioa

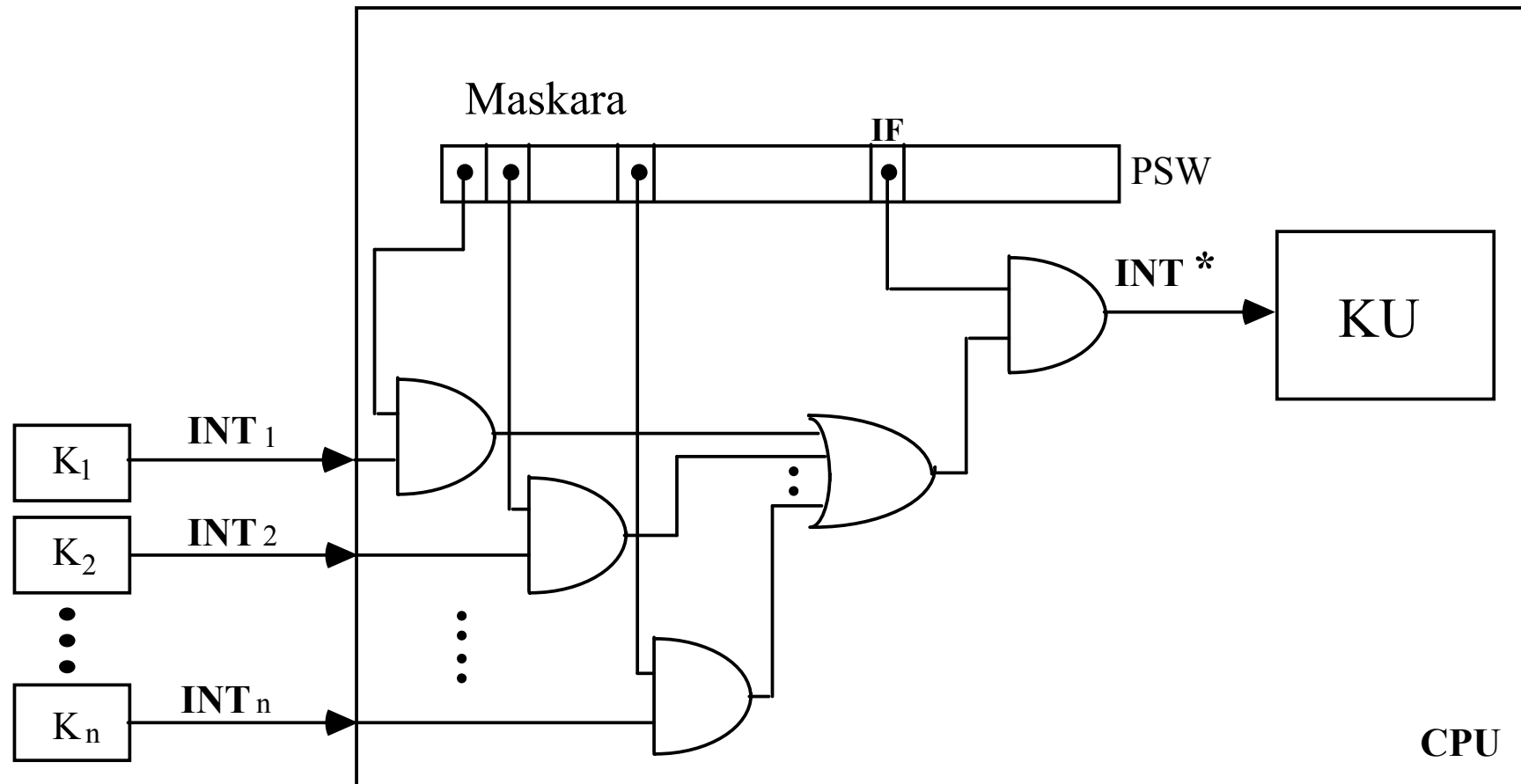
- CPU-k lerro bat edo gehiago ditu gailuek eten-eskaera bidali ahal izateko
 - INT: kontrol-unitatearen sarrera
- CPU-k eten-eskaera seinalearen balioa aztertzen du
 - Aginduaren exekuzio-faseren batean
- Etenaren adierazlea (*Interrupt Flag, IF*):
 - $IF = 0$ → etenak galeraziak/maskaratuak (**CLI** agindua 80x86)
 - $IF = 1$ → etenak baimenduak (**STI** agindua 80x86)
- Maskaratu ezin diren etenak (*Non Maskable Interrupt, NMI*)

Etenen detekzioa



- Eten-adierazlea (*IF*):
 - CPU-k eten bat tratatzerakoan: $IF = 0$
 - Tratamendua bukatzerakoan: $IF = 1$
 - maila bakarreko etenak

Etenen detekzioa eta maskaratzea



Programaren egoera gorde

- CPU-k programaren exekuzioa eteten du etena tratatzerakoan eta etenaren tratamendua bete eta gero berreskuratzen du
- Etendako programari buruzko informazioa gorde behar da:
 - Itzul-helbidea → exekutatu behar den hurrengo aginduaren PC-a
 - Egoera-hitza, PSW → adierazleak etab.
- Zerbitzu-errutinak erabiltzen dituen erregistroen edukia gordetzen du
- Non gordeko da informazio hau? → **PILA**

Etenaren identifikazioa

- arazoak
 - Nola jakin zeinek eten duen?
 - Gailu batek baino gehiagok eteten badu INT seinale bera erabiliz, zein tratatu?
- aukerak
 - **Software** bidezko identifikazioa (adibidez, MC68000)
 - **Hardware** bidezko identifikazioa

Software bidezko identifikazioa

- INT seinalea detektatzerakoan, IF=1 bada, CPU-k **zerbitzu-errutina orokorra** exekutatzen du. Honek **inkesta** bitartez detektatuko du zein gailuk eten duen

```
for (i=0;i<n;i++)
```

```
    if (egoera_err_irakurri (Ki) == PREST)
```

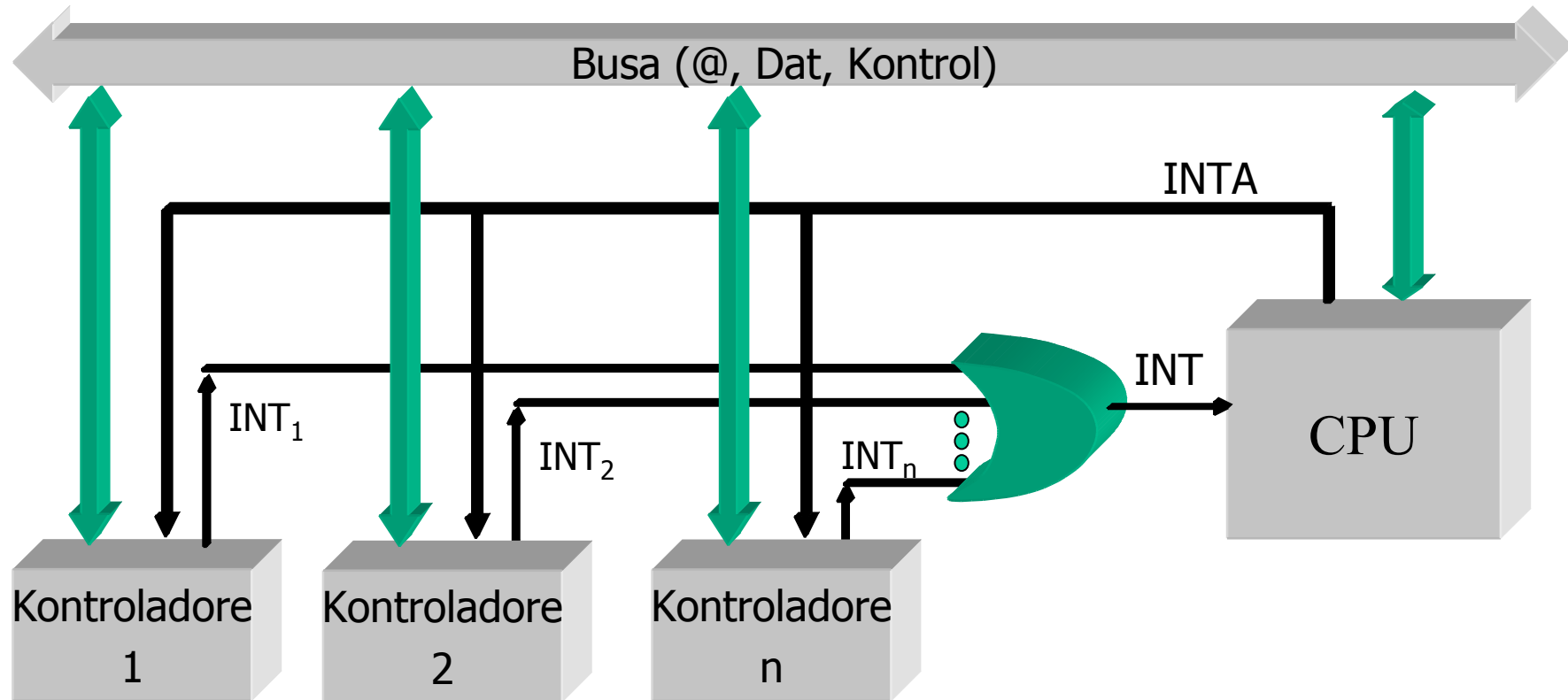
```
        { zerbitzu_errut_Ri(); break; }
```

- Periferikoa identifikatu eta gero, berari dagokion errutina exekutatzen da
- Aldibereko etenak?
 - Inkestaren ordenak gailuen arteko lehentasuna ezartzen du: lehendabizi lehentasun altuena ...

Hardware bidezko identifikazioa

- Gailua bera da identifikatzen dena
- Kontroladoreak CPU-ri bidaltzen dio (hiru aukera):
 - zerbitzu-errutinaren hasierako helbidea, edo
 - agindu baten kodea (jauzi-agindua) CPU-k exekuta dezan.
Horren bitartez zerbitzu-errutinaren exekuzioa abiaraziko da, edo
 - identifikadore bat, zeinen bitartez CPU-k taula bat atzituko duen, bertatik lortuz exekutatu den zerbitzu-errutinaren helbidea (**eten-bektorea**, i80x86)
- CPU-k eten-eskaera onartzen duela adierazteko INTA seinalea aktibatuko dio periferikoaren kontroladoreari

Hardware bidezko identifikazioa



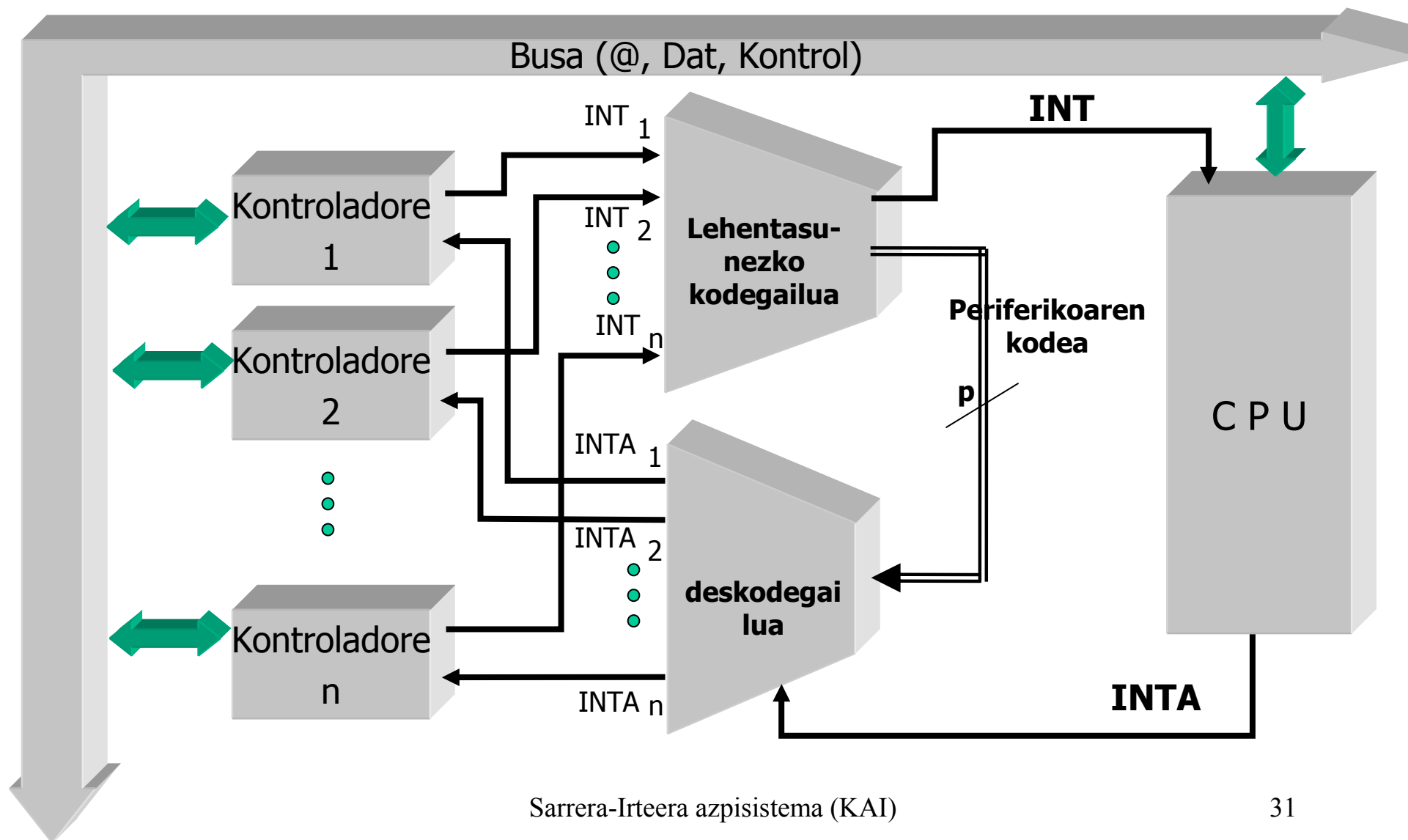
Aldibereko etenak

- Periferiko batek baino gehiagok eteten badu INT lerro berdinetik, nori kasu egin?
- Aukerak

1. Lehentasunezko kodegailua:

- + kodegailuak sortuko ditu, bai INT seinalea, baita lehentasun altuena duen periferikoaren kodea ere
- + deskodegailu batek aurreko kodearen bitartez INTA seinalea bidaliko dio dagokion kontroladoreari

Aldibereko etenen identifikazioa: Lehentasunezko kodegailua



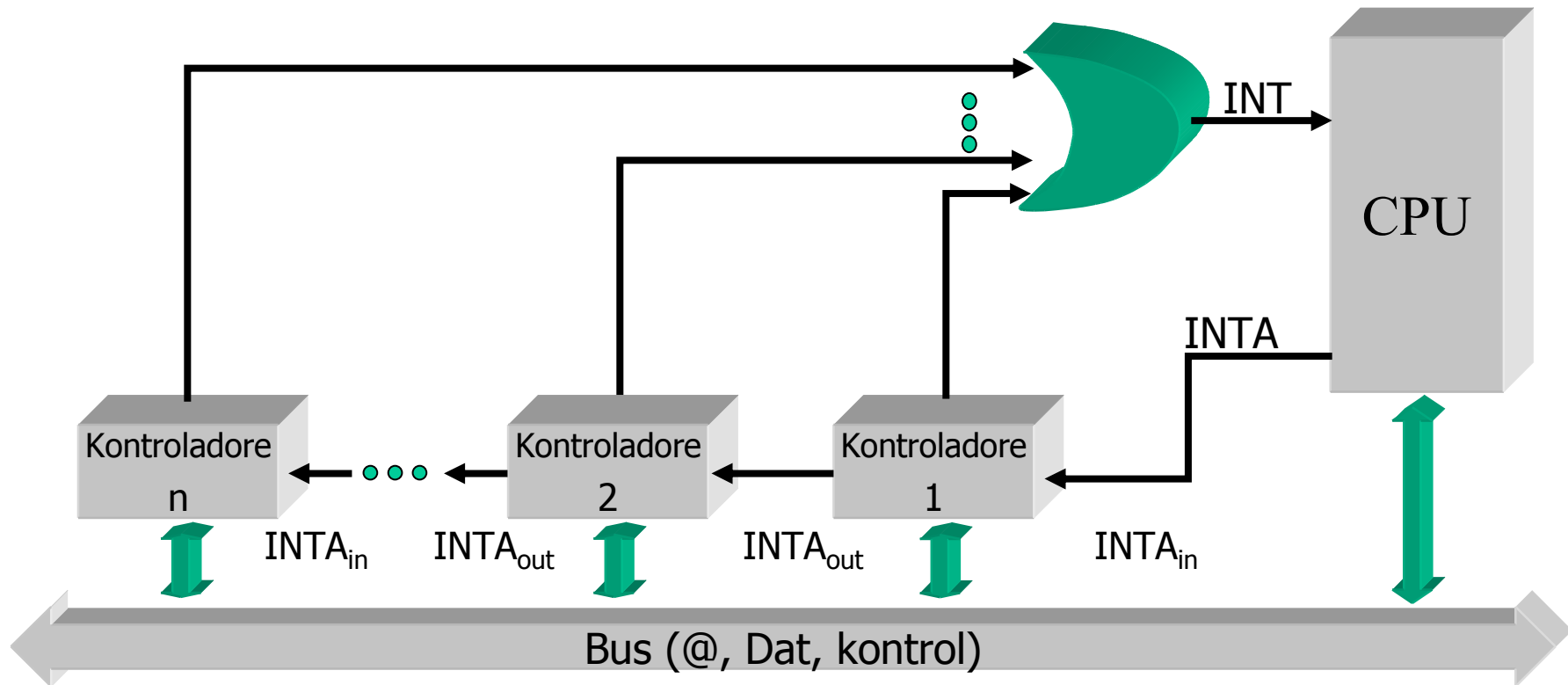
Aldibereko etenak

- aukerak

2. *Daisy-chain* edo margarita-katea:

- + CPU-k INTA seinalea aktibatzen du
- + kontroladoreak INTA seinalea detektatzerakoan
 - + berak ez badu eten-eskaera bidali, INTA seinalea aktibatzen du hurrengo kontroladorearentzat (horrela jarraitzen da eskaera egin duen kontroladorera iritsi arte)
 - + etena bidali badu, ez dio pasako INTA seinalea hurrengo kontroladoreari eta CPU-ri bidaliko dio etena tratatzeko beharrezkoa den informazioa
- + CPU-tik gertuen dauden kontroladoreek lehentasun altuena dute

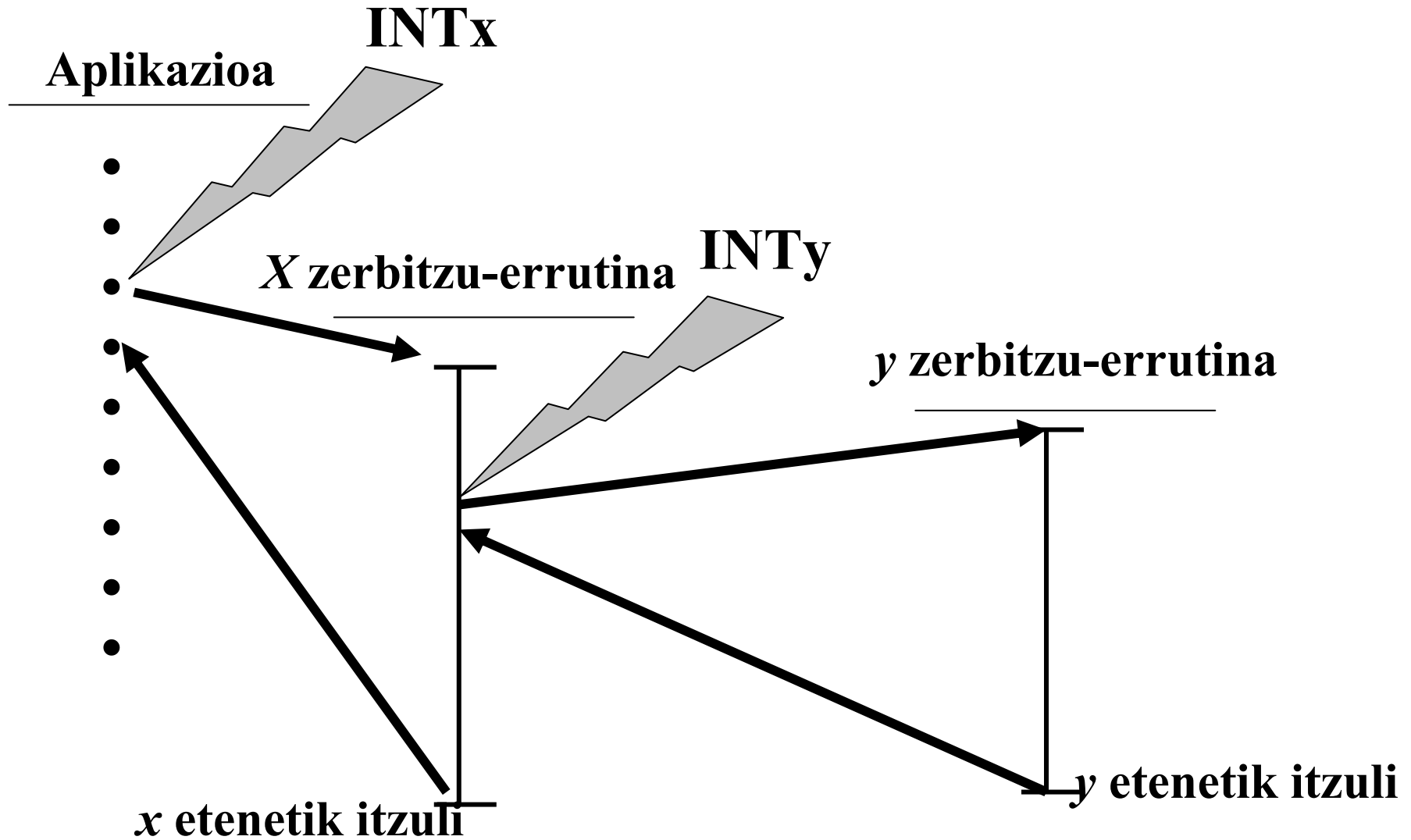
Aldibereko etenen identifikazioa: Daisy-Chain edo margarita-katea



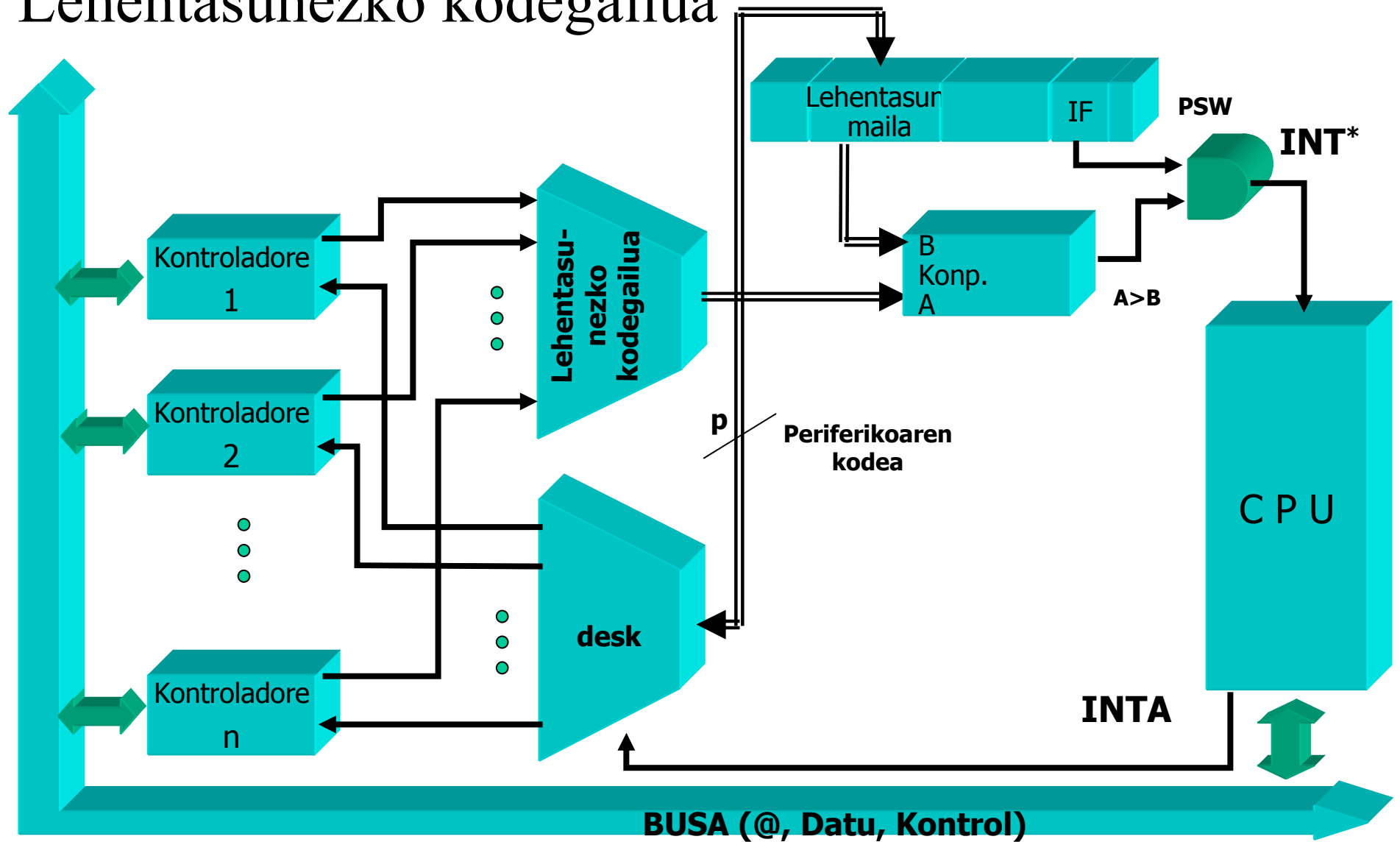
Maila anitzeko etenak

- CPU eten bati dagokion zerbitzu-errutina exekutatzen ari denean beste eten-eskaera bat irits daiteke
 - + exekutatzen ari den zerbitzu-errutina uzten da eten berriari kasu egiteko
 - + IF ez da automatikoki desaktibatzen eten bat tratatzerakoan
- Lehenetasuna:
 - + n lehenetasuna duen periferiko baten etena tratatuko da, etenak baimenduta baldin badaude eta une horretan ez bada ari tratatzen beste eten bat lehenetasun altuagokoa

Maila anitzeko etenak

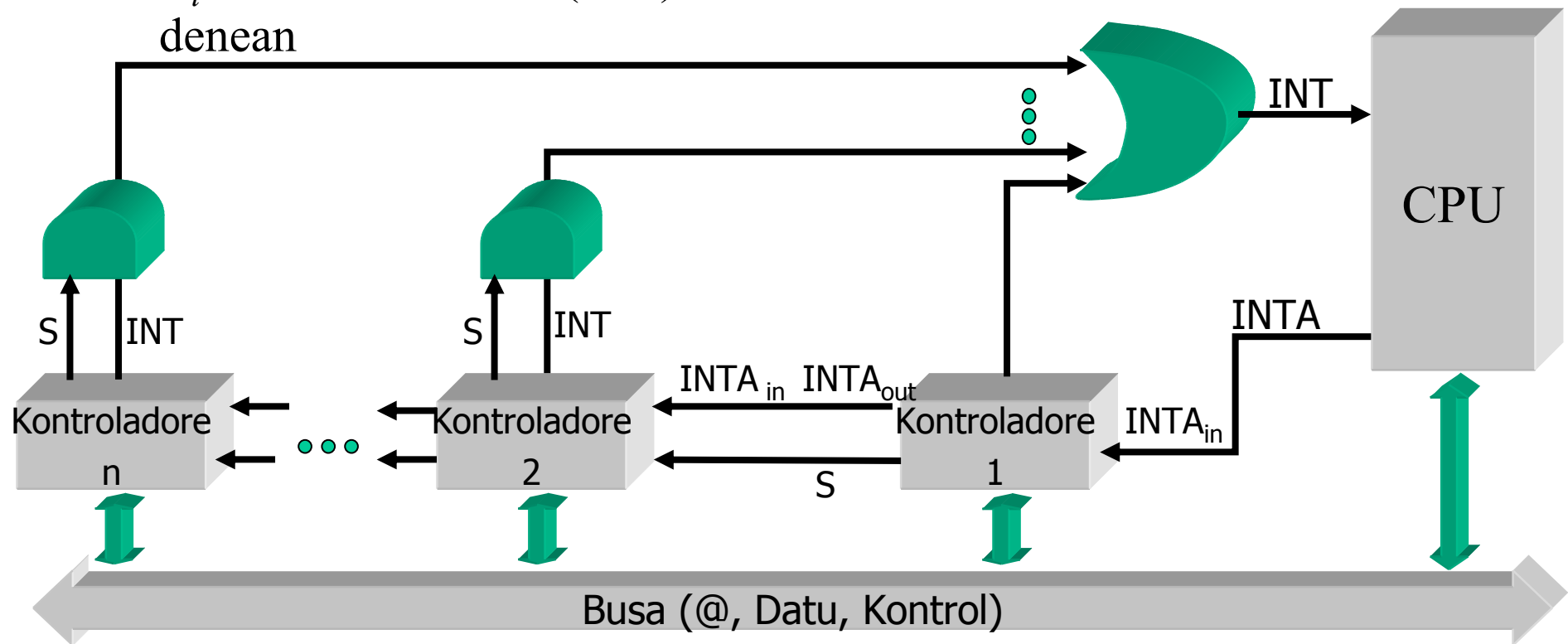


Maila anitzeko etenak: Lehentasunezko kodegailua



Maila anitzeko etenak: *Daisy-Chain* edo margarita-katea

- + K_i -k S seinalea desaktibatzen du ($S=0$) K_i baino lehentasun txikiagoa duten periferikoen maila anitzeko etenak gerta ez daitezzen
- + K_i -k S aktibatzen du ($S=1$) bere etenaren tratamendua bukatzen denean



Eten-kontroladorea

- CPU eta S/I-ko periferikoen kontroladoreen arteko tartekaria da
- Azaldutako guztia kudeatzen duen zirkuitua:
 - + CPU eta periferikoaren kontroladorearen arteko komunikazioa kudeatzen du
 - + lehentasunen arabera zein periferiko zerbitzatu behar den erabakitzen du
 - + maila anitzeko etenak kudeatzen ditu, ...
- Adibidez, i80x86-ko eten-kontroladorea: i8259

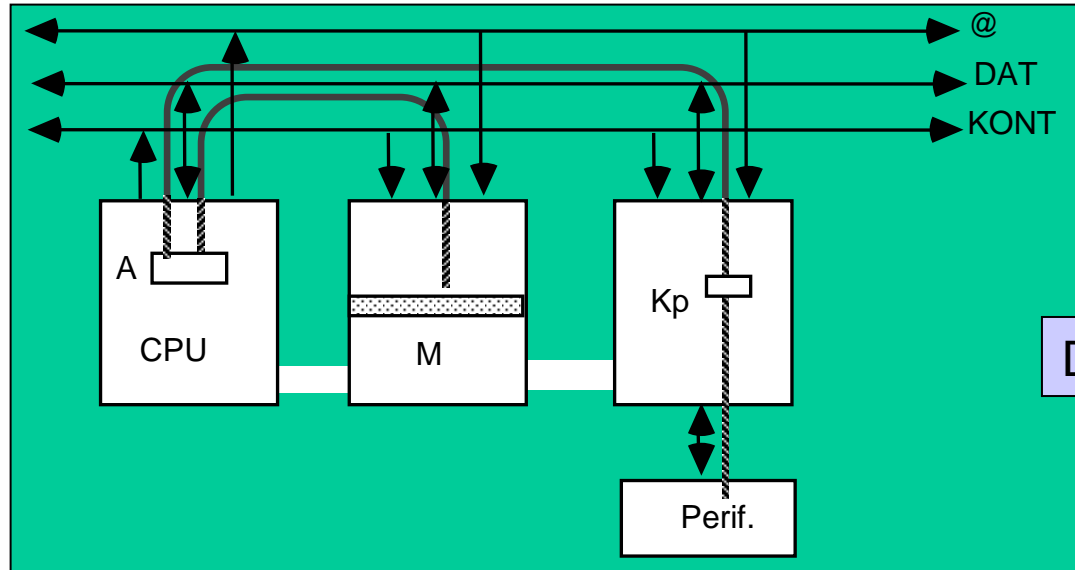
[Laborategiko atala]

DMA bidezko S/I:

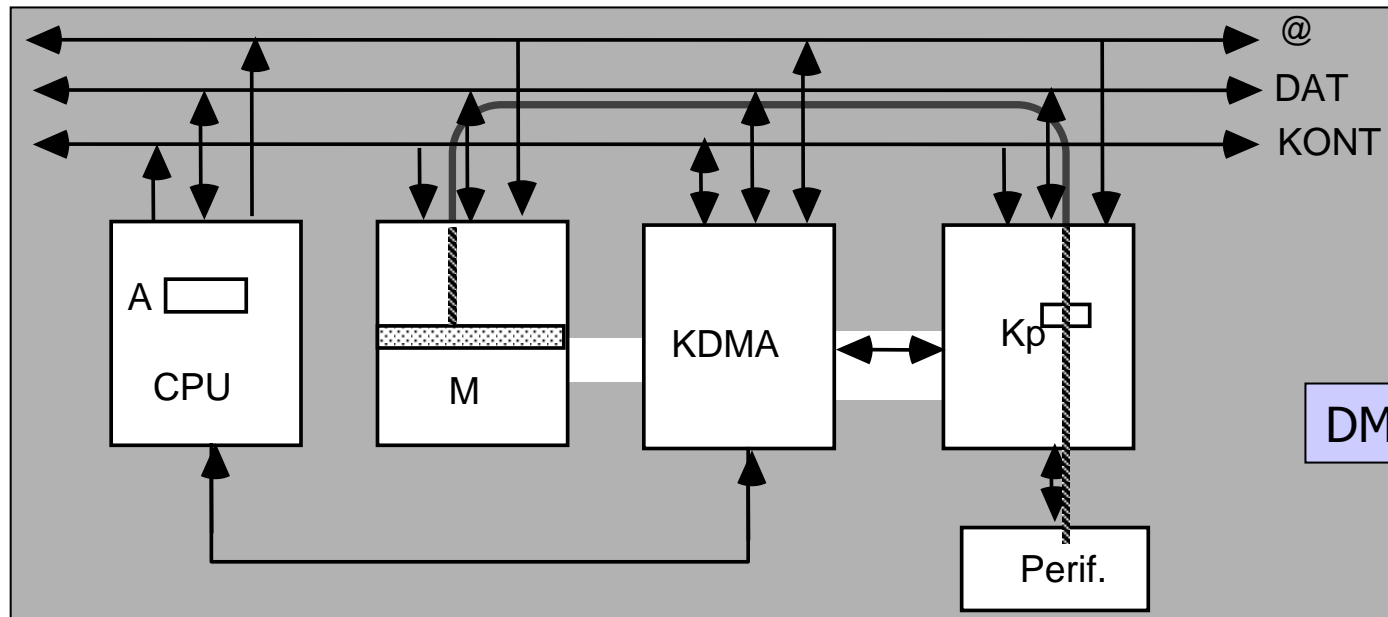
Direct Memory Access / Memoriako zuzeneko atzipena

- Arrazoia:
 - + gailu batetatik (adibidez, disketea) datu-kopuru handien transferentzia memoriako helbide kontsekutiboetara
 - + DMA gabe, CPU etengabe etengo litzateke (adibidez, diskotik transferitu behar den karaktere bakoitzeko)
- Konponbidea: DMA kontroladorea
 - + CPU-k maiz egin beharko lukeen lan sinple eta errepikakor bat ez egiteko erabiltzen den zirkuitu berezia
 - + Prozesadorearen erregistro orokorrak erabili gabe zuzenean memoria atzitu dezakeen zirkuitua

DMA bidezko sarrera/irteera

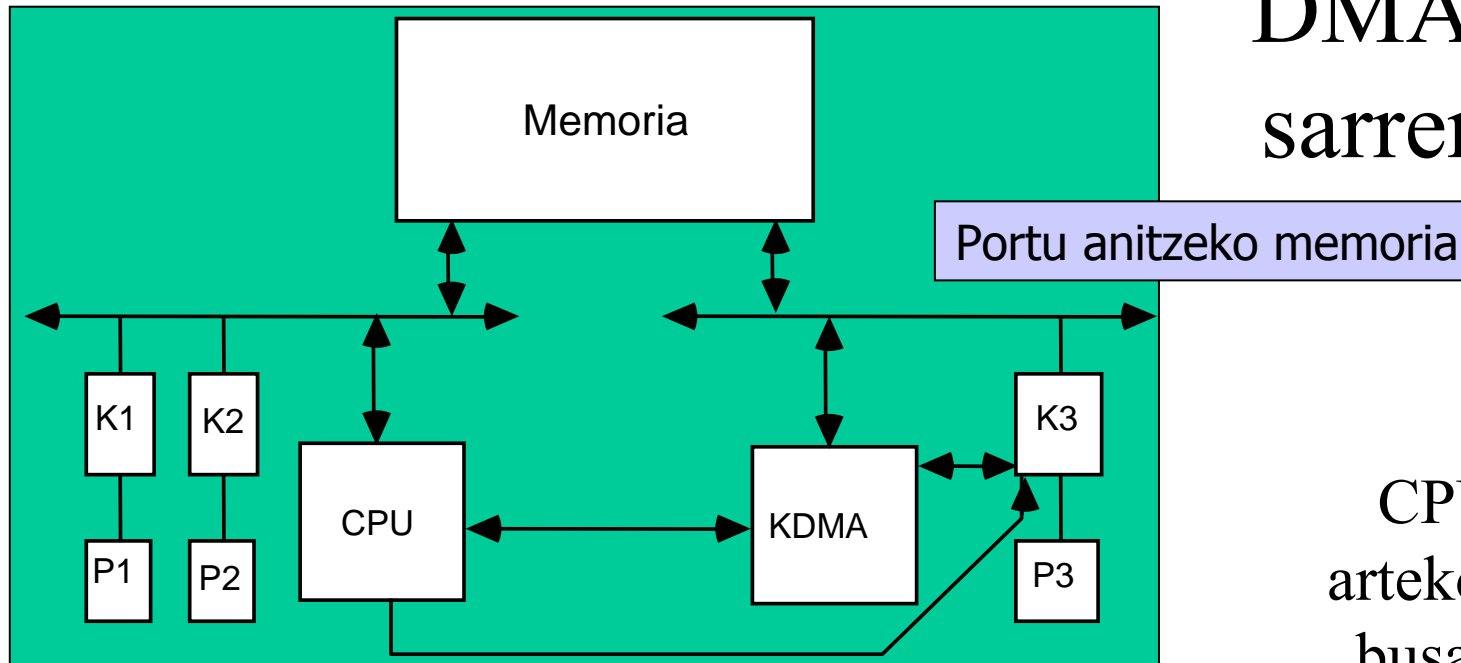


DMA gabe

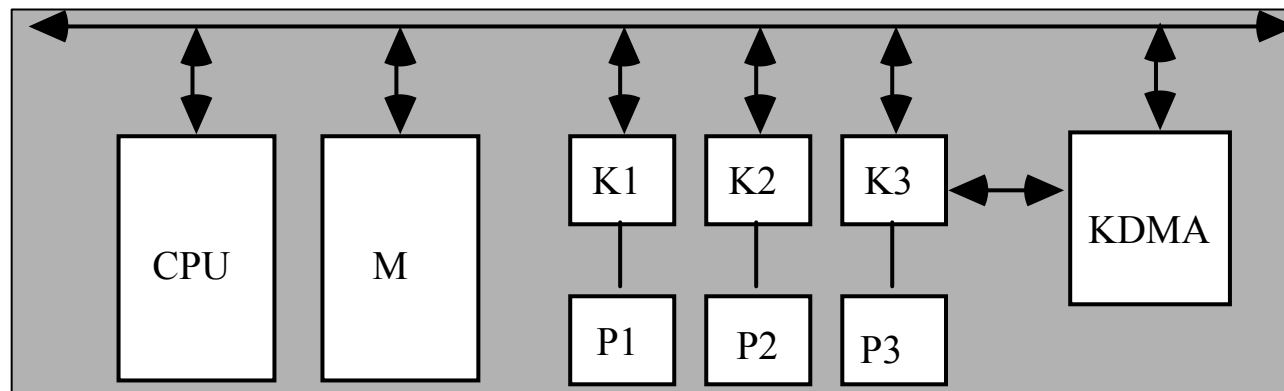


DMA erabiliz

DMA bidezko sarrera/irteera



CPU eta KDMA arteko sinkonizazioa busaren erabileran



Portu bakarreko memoria

DMA kontroladorea

- Helbide-erregistroa: transferitu behar den informazioaren irakurketa/idazketa helbidea
- Datu-kopurua duen erregistroa: transferitzeko geratzen den datu-kopurua gordetzen duen kontagailua
- Kontrol-erregistroa: kontrol-informazioa (irakurketa/idazketa, sinkronizazio modua bukatzean, etab.)
- Egoera-erregistroa: beste edozein kontroladoretan egoera-erregistro batek duen funtzioa

DMA kontroladorea

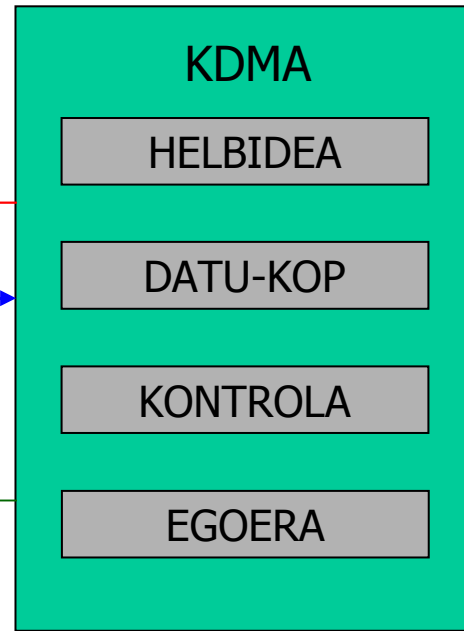
Busaren sinkronizazioa

Eten kontroladorera doan seinalea

BR

BG

IRQ



DMAR

DMAG

Periferikoaren sinkronizazioa

BR *Bus Request*

BUS-a erabiltzeko eskaera

BG *Bus Grant*

BUS eskaera onartzen da

IRQ *Interruption ReQuest*

Eten-eskaera

DMAR *Direct Memory Access Request*

Memoriako zuzeneko atzipenaren eskaera

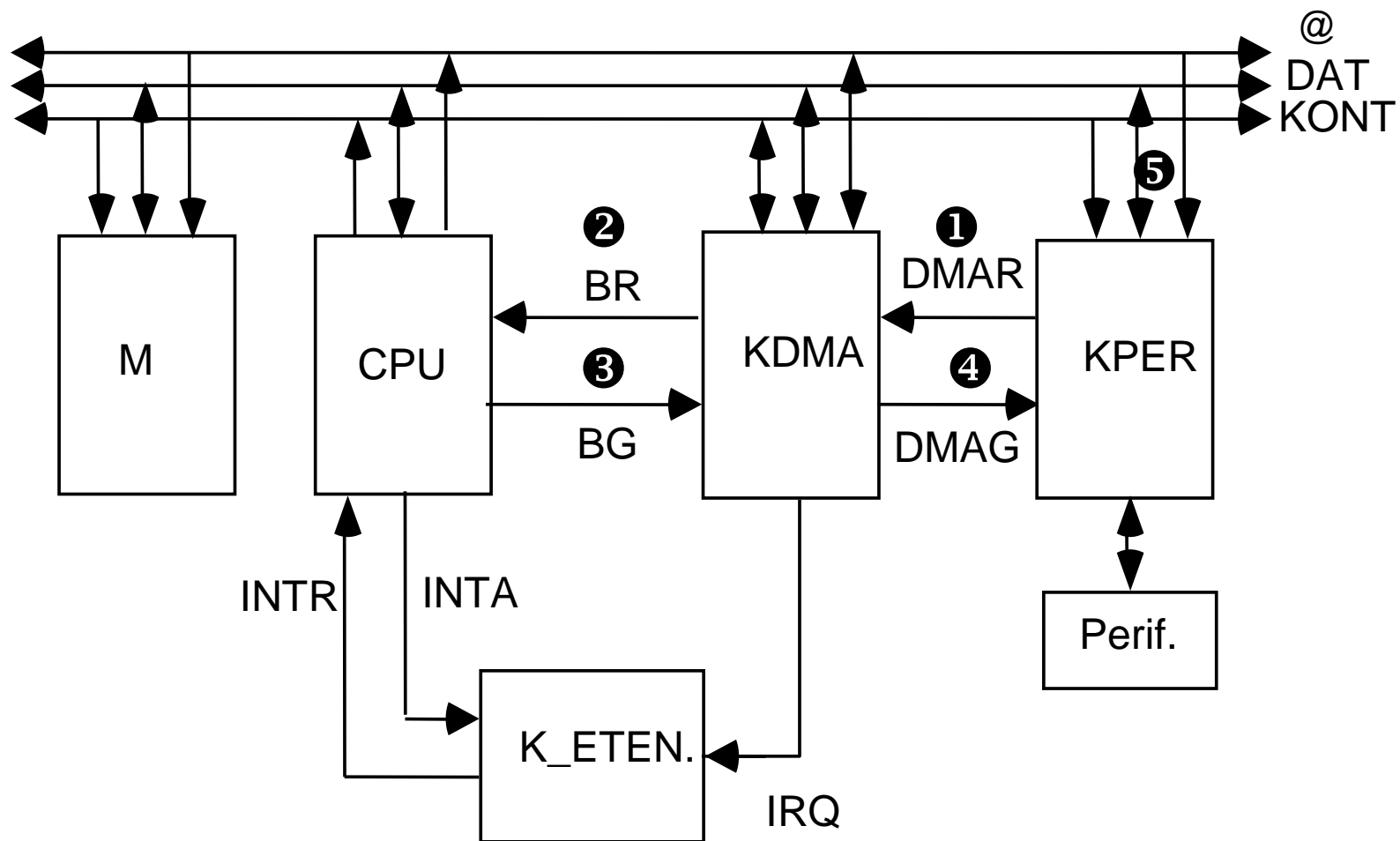
DMAG *Direct Memory AccessGrant*

Memoriako zuzeneko atzipena onartua

DMA bidezko sarrera/irteera: funtzionamendua

- **Aplikazioa (CPU):** periferikoa eta bere DMA kontroladorea programatzen ditu (@, datu-kopurua, irakurketa/idazketa)
- **Periferikoa:** datu bat bidaltzeko edo jasotzeko prest dagoen bakoitzean KDMA-ri adierazten dio
- **KDMA:** datuaren transferentzia egin eta CPU-rekin busaren erabilera sinkronizatzen du
- Datuen transferentzia bukatzean, **KDMA** CPU-rekin **sinkronizatzen** da: inkesta edo etena
 - S/I eragiketa bukatzean bakarrik eteten da CPU

DMA bidezko sarrera/irteera



DMA bidezko sarrera/irteera

Transferentzia (KDMA)

- Kper \rightarrow DMAR (DMA eskaera) ❶
- KDMA \rightarrow BR (Busak erabiltzeko eskaera) ❷
- CPU \rightarrow BG (Busen erabilera onartua) ❸
- KDMA-k datu edo datuen transferentzia egiten du:
 - Helbidea helbide-busean
 - (R/W) seinalea aktibatu
 - DMAG ❹ aktibatu \rightarrow Kper-ek ❺ datu-busa atzitu
- Kper-ek DMAR desaktibatu
- DMA-k BR desaktibatu

DMA bidezko sarrera/irteera

Ziklo-lapurketa bidezko transferentzia

- + KDMA-k noizbehinka CPU-ri ziklo bat lapurtzen dio memoria atzitzeko
- + CPU-ren eraginkortasuna jaisten da periferikoaren abiadura handia bada

Blokekako transferentzia

- + KDMA-k BG seinalea jasotzen duenean, ez du busaren kontrola uzten bloke osoaren transferentzia osatzen duen arte
- + gailu azkarrekin erabiltzeko egokia

DMA bidezko sarrera/irteera

Transferentzia bukatu

- KDMA → CPU sinkronizatu S/I eragiketaren bukaera adierazteko
- **Inkesta jarraiaren** bidez: CPU-k KDMA-ren egoera-erregistroa aztertzen du → ez da interesgarria!!!!
- **Etenen** bidez: KDMA-k CPU eteten du
 - KDMA-ren egoera-erregistroa irakurri
 - Baldin (ez da ondo atera)
 - KDMA berriro programatu
 - Kper berriro programatu
 - Bestela