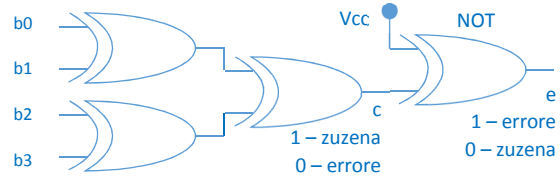


Nombre / Izena: \_\_\_\_\_ Grupo / Taldea: \_\_\_\_\_  
Apellidos / Abizenak: \_\_\_\_\_

### **Galdera teorikoak (4 puntu)**

1.  $b_3b_2b_1b_0$  bit-sekuentzia iristen bada, eta azken bita ( $b_0$ ) parekotasun bakoitiko bit bat bada. Errorea detektatzeko zirkuitu bat proposatu.



2. Hurrengo Hamming hitza jasotzen da:  $b_7b_6b_5b_4b_3b_2b_1 = 0011100$ . Errorea gertatu da transmisioan? Hala balitz, zein da bidalitako jatorrizko datua?

$$c_1 = b_1 \oplus b_3 \oplus b_5 \oplus b_7 = 0$$

$$c_2 = b_2 \oplus b_3 \oplus b_6 \oplus b_7 = 1$$

$$c_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 0 \quad \text{Errorea paritate bit batean (} b_2 \text{), ez datuan: } b_7b_6b_5b_3 = 0011.$$

3. Zer da kontrol hitza?

Datu bidean nahi den eragiketa egiteko behar diren seinaleen multzoa da, kontrol-unitateak (CU) sortutakoak.

4. Zerk bereizten du Von Neumann arkitektura?

Instrukzioetarako eta datuentzako memoria eta bus berak erabiltzen ditu.

5. Zer kontrol-unitate motatan eman daiteke sekuentziario inplizitua eta zertan datza?

Mikroprogramatu batean. Horrek esan nahi du kontrol-memorian mikroinstrukzioak modu sekuentzialean daudela, eta kontrol-hitza bakoitzean ez dela beharrezkoa jarraian exekutatu den mikroinstrukzioaren helbidea adieraztea, hurrengo baina zuzenean.

6. Etendura bat ematen denean, zer da egin behar den lehenengo gauza etenari erantzuteko azpierrutinean?

Egoera gordetzea: dagozkion erregistroak eta kontrol-hitza (edo flagak).

7. Kanalizazio exekuzioan datuen arrisku-arazoa konpontzeko modu bat adierazi.

Softwarea: NOP instrukzioak sartuz.

Hardware 1: arazoa aurretik detektatuz, eta ziklo horretan datu-arriskua badago, ez da pipe-line erabiltzen: burbuila. Ez dira NOP instrukzioak idatzi behar, baina denborari dagokionez, efektua bera da.

Hardware 2: eragiketarako beharrezko emaitza (dagoeneko eskuragarri dagoena baina jatorrizko zirkuituan bideratu gabea) eramango duen bide bat prestatzea, zuzenean egikaritze-unitatearen sarrera egokira.

8. Sarrera/Irteera (I/O) periferikoen arteko datu-transferentzia sinkronizatzeko bi metodoetatik zeinek du fidagarritasun handiagoa, eta zein mekanismo erabil daiteke erantzun ezaren ondoriozko transmisio-erroreak detektatzeko?

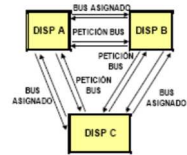
Handshaking.

Erantzun ezaren ondoriozko erroreak detektatzeko Time-out mekanismo bat erabil daiteke: unitate batek kontrol-seinale bat gaitzen duenean denbora-kontu bat hasten da; denbora-tarte baten barruan erantzunik ez badago, errore bat gertatu dela suposatzen da.

Nombre / Izena: \_\_\_\_\_ Grupo / Taldea: \_\_\_\_\_  
Apellidos / Abizenak: \_\_\_\_\_

9. Zertan datza bus artekaritza deszentralizatua, eta artekaritza independentearen estrategia duena?

Artekaritza-metodo deszentralizatuan ez dago artekaritzarik, eta periferikoak beraiek arduratzen dira bus artekaritzaz. independentea denez, gailu bakoitza sisteman dauden gailu adina bus eskatu eta esleitzeko linea ditu, eta guztiak elkarren artean konektatuta daude busa eskatzeko eta esleitzeko.



10. Zertan bereizten da I/O periferikoekin egindako datu-transferentzia maneia, programatuan eta etenen bidez?

Programatuan, prozesadoreak etengabe begiratu behar du ea periferikoak datu bat eskuragarri duten, eta etendurekin, prozesadorea askatuta geratzen da, eta periferikoek adierazten dute datu bat dutela eskuragarri etenen bidez.

11. Sistema batek 2 MB-ko edukierako Memoria Nagusia badu, eta Cache Memoria 64 kB-koa bada eta lerro bakoitzeko 1 kB badu (hitzaren luzera 1 B), zenbat bloketan (NumBlo) banatzen da memoria nagusia? Zenbat lerro ditu (L) cache memoriak?

Egokitzapen metodo guztiz elkarkorra erabiltzen bada, eta cache lerroak 0-tik (L-1)-raino zenbatzen badira, eta blokeak 0-tik (NumBlo-1)-raino. 128 blokeari zer cache lerro legokioke?

$$\text{NumBlo} = 2 \text{ MB} / 1 \text{ kB} = 2 \cdot 2^{10} = 2048 \text{ bloke.}$$

$$L = 64 \text{ kB} / 1 \text{ kB} = 64 \text{ linea.}$$

Egokitzapen guztiz elkarkorrarekin 128 blokea edozein cache linean egon daiteke.

12. Memorian irakurketa 200 atzipenetatik 5 cachean ez dauden datuetara egiten badira. Irakurketak bakarrik kontuan hartuta, zer hobekuntza lortzen da cachea erabiliz, erabili gaberekin konparatuta, batezbesteko memoria atzipen denboran?, baldin eta hobekuntzarik badago. (Datuak: cachera atzipen denbora: 5 ns; memoria nagusira atzipen denbora: 50ns).

$$t_{con\_cache} = (1 - \rho_{miss}) \cdot t_{MC} + \rho_{miss} \cdot t_{MP} = \left(1 - \frac{5}{200}\right) \cdot 5 + \frac{5}{200} \cdot 50 = 6,125 \text{ ns.}$$

$$t_{sin\_cache} = t_{MP} = 50 \text{ ns.}$$

$$\text{Hobekuntza batezbesteko memoria atzipen denboran da: } t_{sin\_cache} - t_{con\_cache} = 50 - 6,125 = 43,875 \text{ ns.}$$

$$\text{Cachearekin denbora atzipena da: } \frac{(50-6,125)}{50} \cdot 100 = 87,5\% \text{ azkarragoa.}$$

#### **Galderak (Azterketaren %40a)**

Ariketekin batezbestekoa egiteko galderen puntuazioaren erdia atera behar da.

Gardera laburra (erantzunen puntuaketa): Zuzena: +1 Okerra: ±0 Zurian: ±0

Test galdera (erantzunen puntuaketa): Zuzena: +1 Okerra: -0,5 Zurian: ±0

#### **Ariketak (Azterketaren 60%a)**

Galderekin batezbestekoa egiteko ariketen puntuazioaren erdia atera behar da.

## Ariketa 1 (2 puntu)

Kontuan hartuta 1., 2., 3. eta 4. irudietako taula, kontrol-unitatea (Control Unit, CU), exekuzio unitatea (Unidade Aritmetiko-Logikoa – Arithmetic-Logic Unit, ALU), PCa kalkulatzeko zirkuitua eta instrukzioen formatua.

Honako instrukzio hauek egin nahi dira:

- 3 erregistroaren edukari 2 erregistroarena kendu eta emaitza 3 erregistroan gorde.
- Emaitza negatiboa bada, -9 posizio jauzi egin.

Esatzen da:

- a) Zein motatako kontrol unitatea da?

**Kableatuta.**

- b) Instrukzioak hamaseitarrean.

**0000101 011 011 010 = 0ADAh.**

**1100001 110 011 111 = C39Fh.**

- c) Lehenengo instrukzioari dagokion kontrol hitza.

**011 011 010 0 010 1 0 1 0 0 0 1.**

- d) 2 erregistroan 2 bat gordeta badago, eta 3 erregistroan 4 bat badago biltegitratuta, eta PCak, jauzi kondizionalaren aurretik, 102Bh balioa badu, zein izango da programa-kontagailuaren hurrengo balioa (Program Counter, PC) offseta instrukzioaren 6 bitez osatuta badago: AD = DR SB? W, X eta Y multiplexoreen irteerak adierazi.

**102Ch;**

**Mux(W) = 0;**

**Mux(X) = 1;**

**Mux(Y) = 102Ch.**

Agindua	Eragiketa kodea	Mnemonikoa	Helbidea	Deskribapena	Egoera bitak
A mugitu	0000000	MOV A	DA, AA	$R[DA] \leftarrow R[AA]^*$	N, Z
Gehitu	0000001	INC	DA, AA	$R[DA] \leftarrow R[AA] + 1^*$	N, Z
Batu	0000010	ADD	DA, AA, BA	$R[DA] \leftarrow R[AA] + R[BA]^*$	N, Z
Kendu	0000101	SUB	DA, AA, BA	$R[DA] \leftarrow R[AA] - R[BA]^*$	N, Z
Murriztu	0000110	DEC	DA, AA	$R[DA] \leftarrow R[AA] - 1^*$	N, Z
AND	0001000	AND	DA, AA, BA	$R[DA] \leftarrow R[AA] \wedge R[BA]^*$	N, Z
OR	0001001	OR	DA, AA, BA	$R[DA] \leftarrow R[AA] \vee R[BA]^*$	N, Z
XOR	0001010	XOR	DA, AA, BA	$R[DA] \leftarrow R[AA] \oplus R[BA]^*$	N, Z
NOT	0001011	NOT	DA, AA	$R[DA] \leftarrow \neg R[AA]^*$	N, Z
B mugitu	0001100	MOV B	DA, BA	$R[DA] \leftarrow R[BA]^*$	
Despl. Eskuin	0001101	SHR	DA, BA	$R[DA] \leftarrow sr R[BA]^*$	
Despl. Ezker	0001110	SHL	DA, BA	$R[DA] \leftarrow sl R[BA]^*$	
Kargatu erag.	1001100	LDI	DA, OP	$R[DA] \leftarrow OP^*$	
Batu eragigaila	1000010	ADI	DA, AA, OP	$R[DA] \leftarrow R[AA] + OP^*$	N, Z
Kargatu	0010000	LD	DA, AA	$R[DA] \leftarrow M[AA]^*$	
Biltegitratu	0100000	ST	AA, BA	$M[AA] \leftarrow R[BA]^*$	
Jauzi zero	1100000	BRZ	AA, AD	if $R[AA] = 0$ $PC \leftarrow PC + AD$ else $PC \leftarrow PC + 1$	N, Z
Jauzi neg.	1100001	BRN	AA, AD	if $R[AA] < 0$ $PC \leftarrow PC + AD$ else $PC \leftarrow PC + 1$	N, Z
Jauzi ez-bald.	1110000	JMP	AA	$PC \leftarrow R[AA]$	

\* Eragiketa haren PC  $\leftarrow PC + 1$  (hurrengo zikloa prestatzeko)

Fig. 1. Instrukzioak.

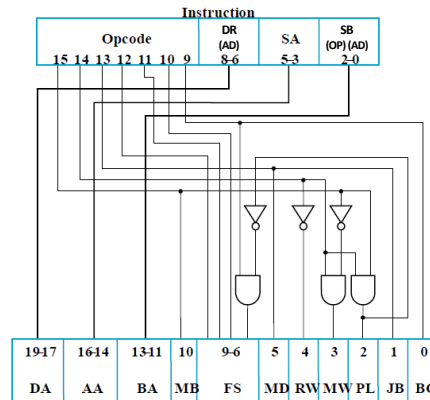


Fig. 2. Kontrol Unitatea.

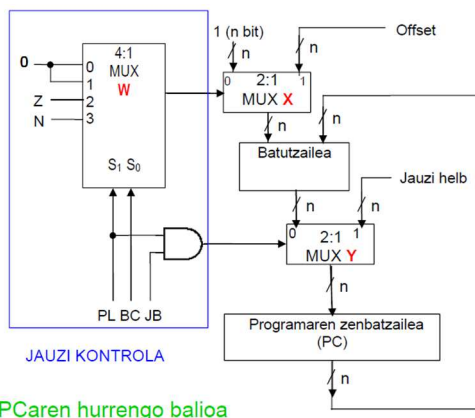


Fig. 3. PCaren kalkulua.

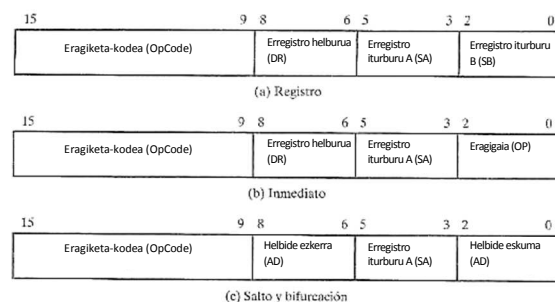


Figura 4. Instrukzioen formatua.

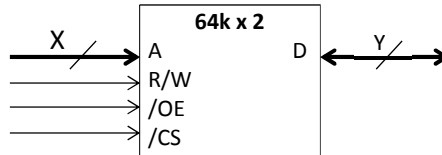
**Ariketa 2 (2 puntu)**

Memoria-zirkuitu integratuak daude, irudiaren modukoak.

- a) Adierazi zenbat linea dituen helbide-busak (X) eta datu-busak (Y).

$$X = \log_2 64k = 16 \text{ linea.}$$

$Y = 2$  linea.

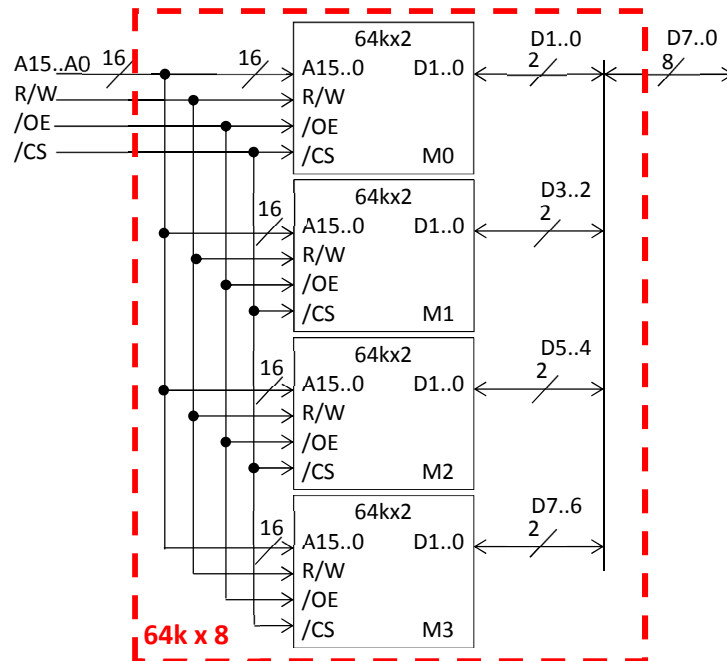


64k x 8-ko memoria bat lortu nahi bada, irudiarena bezalako memoria-zirkuituetan oinarrituta:

- b) Adierazi zenbat memoria-zirkuitu beharko liratekeen.

$$N = \frac{64k}{64k} \cdot \frac{8}{2} = 4 \text{ memoria.}$$

- c) Beharrezko konexio eta zirkuitu integratu guztiak marraztu.



- d) ABh datua 3AF0h posizioan gorde nahi izanez gero, zein memoria-txipetan gordeko litzateke?

Guztietan, 2 bit memoria bakoitzean: ABh = 1010 1011:

10 M3an; 10 M2an; 10 M1ean; 11 M0an.

Nombre / Izena: \_\_\_\_\_ Grupo / Taldea: \_\_\_\_\_  
Apellidos / Abizenak: \_\_\_\_\_

### Ariketa 3 (1 puntu)

Hurrengo irudia kontuan izanda, adierazi zein izango den metagailuaren edukia, eta zein memoria-helbidetara joan behar den datua aurkitzeko, datuak eta helbideak hamartarrean daude:

298	Erag. Kod.   Modua
299	ADRS edo NBR = 500
300	Hurrengo instrukzioa
...	...
400	600
...	...
500	800
...	...
600	200
...	...
700	150
...	...
800	250
...	...
900	350
...	...

Helbideratze modua		Helbide eraginkorra	Acc.
Indexatua	LDA ADRS (R3)	700	150
Berehalakoa	LDA #NBR	299	500
Zeharkako erregistroa	LDA (R2)	400	600
Erlatiboa	LDA \$ADRS	800	250
Zeharkakoa	LDA [ADRS]	800	250
Erregistroa	LDA R2	-	400
Zuzena	LDA ADRS	500	800

PC = 300
R0 = 100
R1 = 300
R2 = 400
R3 = 200

### Ariketa 4 (1 puntu)

Datuen bide bat 4 etapatan banatzen da: A: 8 ns, B: 4 ns, C: 2 ns eta D: 6 ns. Etapa horiek begiztan exekutatzen dira. Kanalizazio bat inplementatzeko, bi erregistro daude, eta bakoitzak 1 ns gehitzen ditu.

- a) Zein da hasierako sistemaren funtzionamendu-maiztasun maximoa?

$$f = \frac{1}{t_A + t_B + t_C + t_D} = \frac{1}{8 + 4 + 2 + 6} = 50 \text{ MHz.}$$

- b) Non jarriko zenituzke erregistroak ahalik eta kanalizaziorik onena egiteko?

A eta D etapen ondoren (edo B eta Dren ondoren).

- c) Zein izango litzateke sistemaren funtzionamenduaren maiztasun maximoa kanalizazioaren ondoren?

$$t_i = \text{Mx}\{t_A + t_{reg}, t_B + t_C + t_D + t_{reg}\} = \text{Mx}\{9ns, 13ns\} = 13ns.$$

$$f_i = \frac{1}{13ns} = 76,92 \text{ MHz.}$$

- d) Zenbat denbora beharko litzateke 3 instrukzio exekutatzeke kanalizazioarekin?

$$t_{3ins} = (n_{instru} + n_{etapa} - 1) \cdot t_i = (3 + 2 - 1) \cdot t_i = 4 \cdot 13ns = 52ns.$$