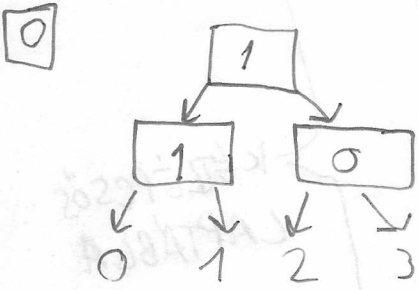


HKA
(utolsó (26). ea)

lemaradt a feladatot
végével.

45. PSEUDO-LRU $\rightarrow 0, 1, 3, 2, 3, 0$!



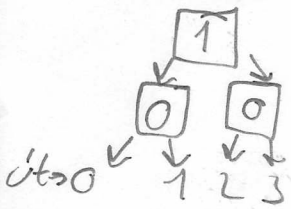
it ?

LRU: 2

1 \rightarrow balra mutat } értékkódok
0 \rightarrow jobbra mutat }

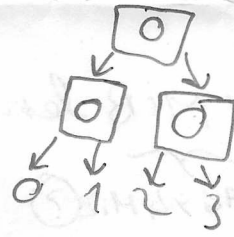
LRU kiválasztásnál fordítva!

1



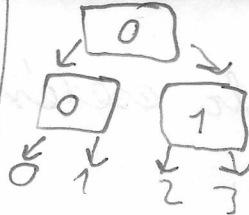
LRU: 2

3



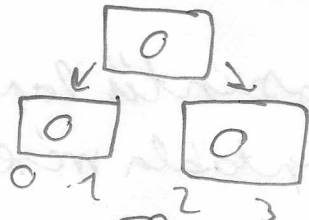
LRU: 0

2



LRU: 0

3



LRU: 0



45. Real-LRU 0 1 3 2 3

0			
0			
0			
0	1	1	1

LRU Blockok:

1, 2, 3

kerül

bármelyik

	0		
	0		
1	0	1	1
0	0	1	1

LRU:

3, 2

1	1	1	0
			0
1	0	1	0
0	0	1	0

LRU: 2

1	1	0	0
1	1	0	1
1	0	0	0
0	0	0	0

LRU 0

1	1	1	0
1	1	0	0
1	0	0	0
0	0	0	0

LRU 0

Rajrok

direkt cache blokkoknál
indexelt leírók

index regiszteres memóriabővítés
egylepéses leírók, kétszempes leírók
segmentálás alapú virtuális tárolás
teljesen leírás, index és olvasás
BSD megújított blokkoknál

40. gyjdf

$$T_{om} = 50 \text{ ns}$$

512 kB, 4 után cache

$$T_c = 10 \text{ ns}$$

$H_r = 90\%$ (fútalati arány)

$$T_{betölt} = 0$$

a) $T_L = ? = 0,9 \cdot 10 + 0,1 \cdot 50 = 9 + 5 = \underline{\underline{14 \text{ ns}}}$

b) $T_{betölt} = 100 \text{ ns}$

$$T_L = 0,9 \cdot 10 + 0,1 \cdot 100 = 9 + 10 = \underline{\underline{19 \text{ ns}}}$$

c) ~~(előtér)~~ 19 ns
vagy, ha a blokk méretét duplájára növeljük

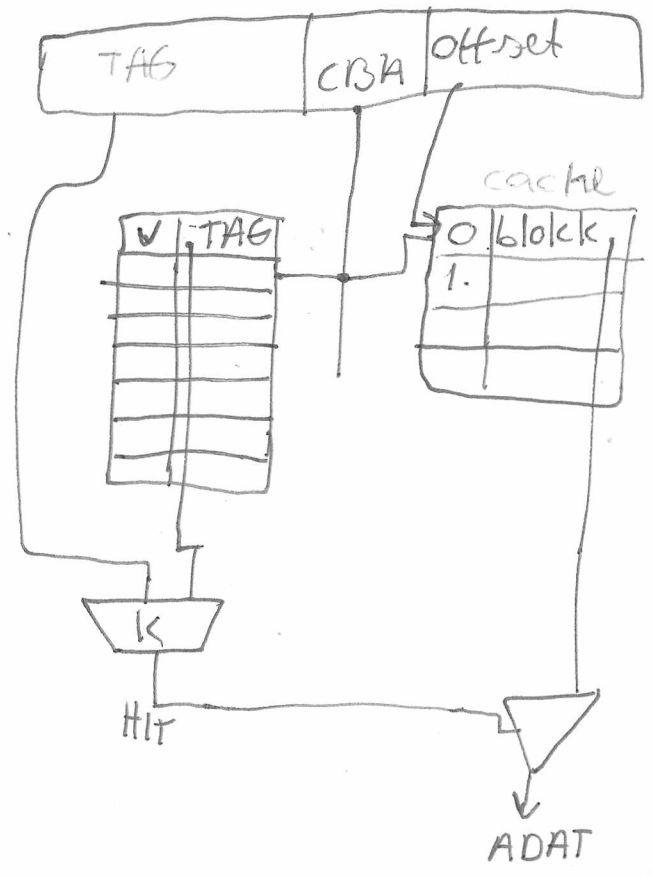
$$0,9 \cdot 10 + 0,1 \cdot 2 \cdot 100 = 9 + 20 = \underline{\underline{29 \text{ ns}}}$$

d) $T_c = 5 \text{ ns}$

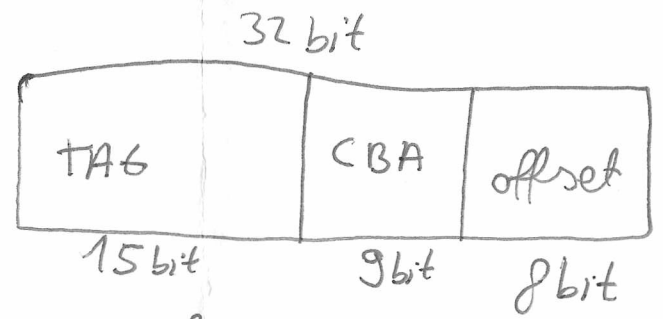
$$0,9 \cdot 5 + 0,1 \cdot 100 = 4,5 + 10 = \underline{\underline{14,5 \text{ ns}}}$$

e) $T_{betölt} = ?$ Hogy $T_L = 9,5 \text{ ns} \Rightarrow$ nem lehet, mert 50 ns -re lenne lehet kijelenni csökkenően, mert $0,9 \cdot T_c + 0,1 \cdot 100 = \text{minimum } 10 \text{ ns}$ \Downarrow

41) Direkt cache



- a) Hány TAG komparátor található a cache?
- b) Hány bites a CBA rezerváló
- c) Hány bites a TAG komparátor



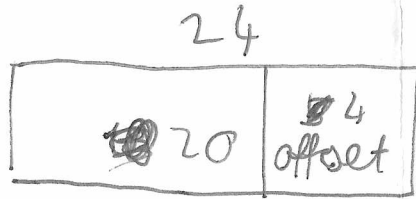
$256 = 2^8 \Rightarrow 8 \text{ bit}$
 $512 = 2^9 \Rightarrow 9 \text{ bit}$

- a) = 1
- b) = 9 bit
- c) = 15 bit

VISSZA 1.
 Gyabari feladatok

44. teljesen associatív cache

a) $\Rightarrow 20$



b) $\Rightarrow 128$ (meg van adva :))

\Downarrow
~~128~~
 $16 = 2^4$

Cache

- behoratali stratégiák: pény sorinti, előrelátó, selektív
- írási stratégiák: write-back, write-through
- blokkoszele stratégiák: direkt selektív esetén WINCS - használatosok
 - LRU
 - FIFO
 - LFU
 - random
- blokkjelölés eljárásai: átlayerelt memóriaműködés (FPM, EDO, BEDO, SDRAM)

- offset: byte
- CBA: blokkos / írásos
- TAG: többi

Real-LRU

- 1) sort 1-esik (alulról feltöltés számolunk)
- 2) sorlapato-áram (balról jobbra számolunk)

Pseudo-LRU

1 → balra mutat } értelem kiadásra való (előretekintés)
0 → jobbra mutat }

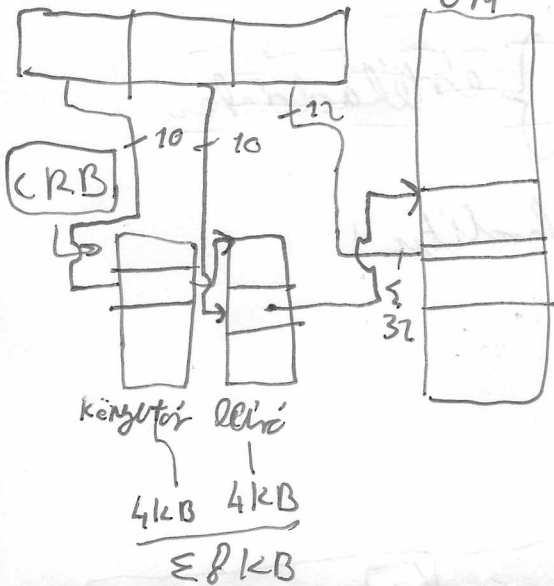
LRU áldorok leírására post fordítás

szekvencia: 0 0 utal 0 1 2 3

Virtuális tárhelyés

segmentumok

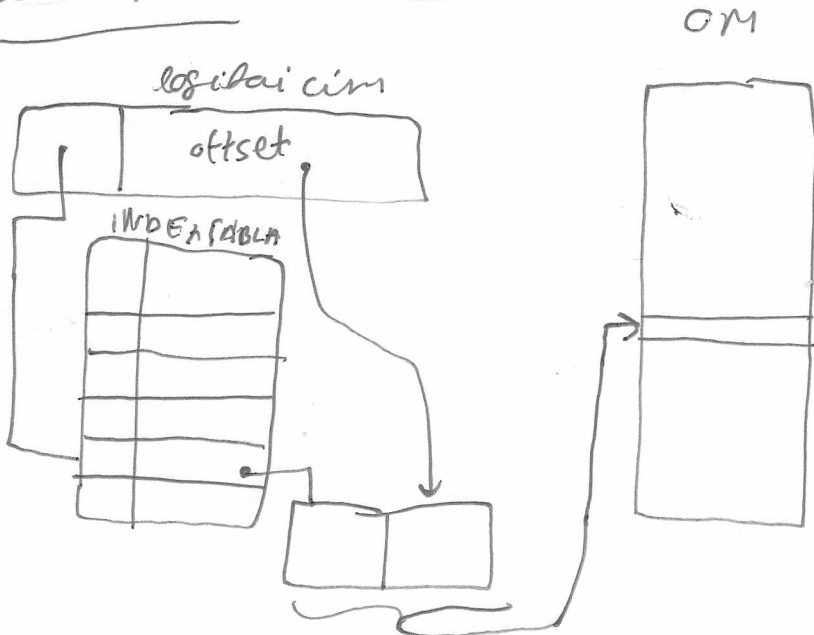
32 bites cím: 4k lapméret → 4 GiB címketű
 → operatív memória



KÉZLÉPCSŐS
LAPTÁBLA

egyszintű laptábla esetén 1 MiB lenne a laptábla mérete.
 VA6 > 4 MiB?

Indexelt lekeperés



logikai cím: 16 bites

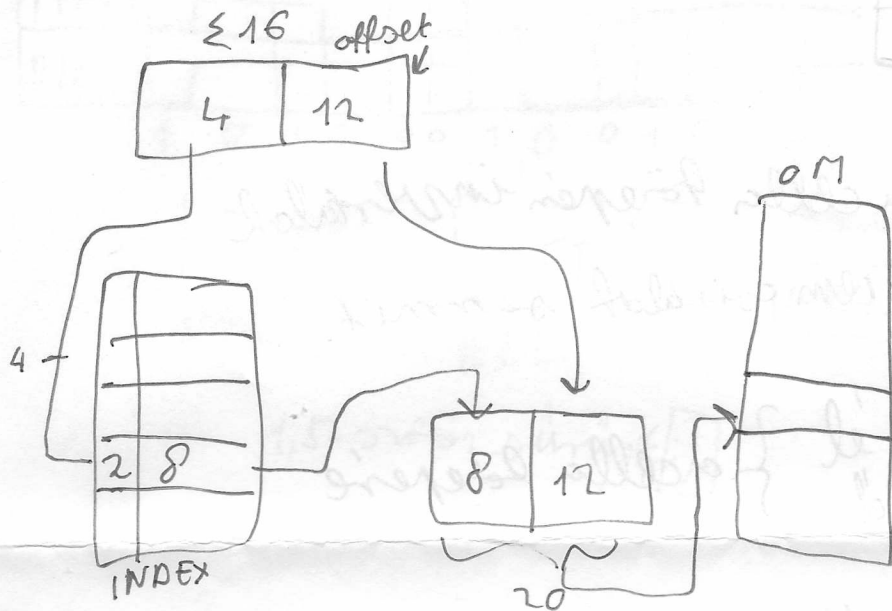
OM = 1 MiB

\Rightarrow 20 cím bit kell hozzá

2 bit veresítésre

Logikai cím felső 4 bitje az index

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



a) INDEX tábla szerű veresítés!

a) 16 db (2^4) 10 bites (2^8) $\Rightarrow 16 \cdot 4 \text{ kibi} = 64 \text{ kibi}$

$10 \text{ bites} = 4 \text{ kibi} = ?$

b) offset: n bites

Mennelevélmez-kódolások

• NRZ: küldeti a bejövő biteket, legegyszerűbb

• NRZI: Ha 1-es jön, a cella köerepén invertálódik
Ha 0-as jön, nem csinál semmit

• PE 1-eset ~~bit~~ felutó él } cella köerepén.
0-ásról lefutó él }

Ha ugyanolyan után ugyanolyan jön, akkor a cellabitaron ^{is} invertálódik.

• FM 0 jön: csak a cellabitaron
1 jön: a cellabitaron és köerepén is } invertálódik.

• MFM: 1 jön: a cella köerepén } van a váltás.
0 jön: a cellabitaron }

Ha 1 után 0 jön, nincs bitarion váltás.
Ha 0 után 1

Memória kapacitás

$$C = \log_2 \left(\frac{K}{B} \right) \text{ (bit/s)}$$

$$K = 2^C \cdot B \text{ (kapacitás)}$$

$$B = \frac{K}{2^C} \text{ (Adatlát.)}$$

1 köerep

0 bitarion

01:

Az MFM létszer jobb (itt) mint az FM

0-0	1	0	0	1	1	0
NRZ	1	0	0	1	1	0
NRZI						
PE	↑	↓	↓	↑	↑	↓
FM						
MFM						

mivel rövidebb időközönként van váltás, annál gyorsabb a kódolás

NRZI: Ha 1-es jön, a cella fölépén invertálal
Ha 0-ás jön, nem csinálol semmit.

PE: 1-eshez feljutás el } a cella fölépése
0-áshoz lefutás el }

Ha ugyanolyan utón
ugyanolyan jön, akkor a cella fölépésén
invertáljuk a jelet.

FM: 0: csak a cella fölépésén
1: a cella fölépésén és lelépésén is invertálal.

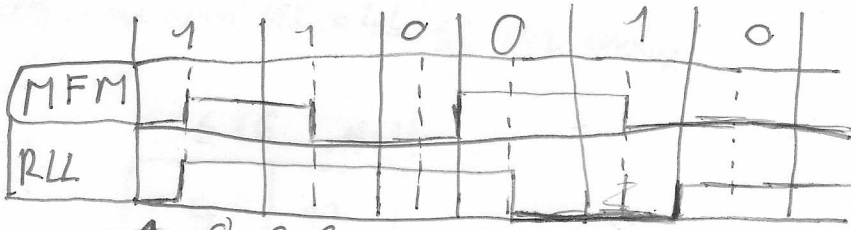
MFM: 1: a cella fölépésén van váltás
0: a cella lelépésén van váltás

Ha utón 0 } nincs a haterón váltás,
0 utón 1 }

RLL kódolás RLL 2.7.

2 db 1-es körült pont

legalább 2-es legfeljebb 7-0 les.



1 0 0 0 0 0 1 0 0 1 0 0

T
MFM

1,5T
RLL

dekódolás
NRZI kód

1,5-szörös sűrűség :)

cache
blokk irán: 2L
cache biltek orp: 4

BEKRD: 12/10
24/50

51-től 5-ös ?

24 25

szerepük vs leírás csatolt
→ kérmilyen OS-t használhat
→ azonos OS-t használóndz ásser.

komm intertek. keris int
reittin
LB FIFO.

Multiprocessor VS multiprogrammrendszerek
→ változó méretű
a tal. paraméter körült
virtualis proc

Kérdés a feladatok virtualis proc és a fizikai proc összerendelés: OS