

Digitális rendszerek, számítógép architektúrák

2. Zárthelyi

2014-05-16

A

1. Jellemezze a szinkron írási protokolt – mikor mi történik?
2. Rajzolja fel és röviden ismertesse az arbitrációt és fajtáit
3. Definiálja a Mealy automata modellt
4. Vezérlőegység programozható makrocellákkal (fajta, ábrák, jellemző tulajdonságok)
5. Rajzolja fel a XILINX FPGA felépítését (CLB és I/O)
6. Ismertesse a dinamikus memória írás-olvasási folyamatát (idődiagramm)
7. Ismertesse a statikus memória olvasási folyamatát (idődiagramm)
8. Ismertesse, hogy hogyan programozható egy programozható VLSI (rajz+előnyök és hátrányok)
9. Rajzolja fel egy PCI buszos számítógép blokkdiagrammját.
10. Rajzoljon fel egy PCI burst read tranzakció idődiagrammját és a diagrammon mutasson példát a várakozási ciklusra mind a két egység részéről.
11. Jellemezze a szegmentált virtuális memória kezelést. – rajz+jellemző paraméterek tárolása.

3 pontos kérdések 4,5,8

B

1. Jellemezze a szinkron olvasási protokolt- mikor mi történik?
2. Rajzolja le az aszinkron hand shaking folyamatot írás esetén
3. Definiálja a Moor automata modellt (3 halmaz, 2 leképezés, 1 rajz)
4. Mikroprogramozott vezérlőegység feladata, vertikális mikroprogramozás (ábra, előnyök, hátrányok)
5. Vezérlőegység programozható makrocellákkal (fajta, ábrák, jellemző tulajdonságok)
6. Ismertesse, hogy hogyan programozható egy programozható VLSI (rajz+előnyök és hátrányok)
7. Ismertesse a dinamikus memória frissítési folyamatát (idődiagramm)
8. Ismertesse a statikus memória írási folyamatát (idődiagramm)
9. Milyen átviteli technológiát használnak a PCI busz jelei és milyen jelcsoportok találhatók a PCI csatlakozón, továbbá milyen csatlakozó felépítése.
10. Rajzoljon fel egy PCI burst write tranzakció idődiagrammját és a diagrammon mutasson példát a várakozási ciklusra mind a két egység részéről.
11. Jellemezze a lapozásos virtuális memória kezelést – rajz+jellemző paraméterek tárolása

Degörök 2.

1.A Jellemző a szinkron bűsi protokoll!

Közös órajellel működnek a buszon csatolt egységek
Működési időt az órajelciklus határozza meg
Master/Slave

Az órajel a leglassabbhoz igazodik

bűsi ciklus: (CPU $\rightarrow$ Mem.)

- n. Master/Commander átvétele / kiválasztása
- n+1. Átvétel megkezdése, mem. hez adat érkezik
- n+2. mem dönt az adat elfogadásáról / válaszol a Commandernek
- n+3. Slave/Responder nyugtát küld vissza.

1.B a szinkron bűsi protokoll!

bűsi ciklus: (CPU $\leftarrow$ Mem.)

- n. Commander átvétele (CPU busz kezele)
- n+1. Kezele inicializálása (CPU dr. kezelet küld)
- n+2. Válasz a C.-nek a döntésről (mem. interf. dönt)
- n+3. Responder döntése (nyugta)

Mem. kérdése! $\frac{1}{2}$

2.A Arbitráció és fogadás

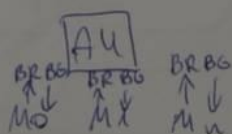
I/O művelet esetén több Master is egyszerre akarja a buszt
Arbitráció adataival vege előtt döntő folyamat, ki lesz a
következő master aki adhat

Master: M Bus Request: BR

Bus Grant: BG

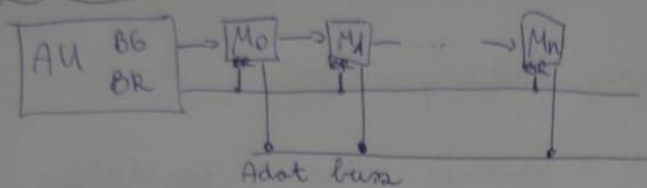
arbitráció
AU

a. Párhuzamos



- leggyorsabb
- drágább is
- FIFO, round rob., prior.

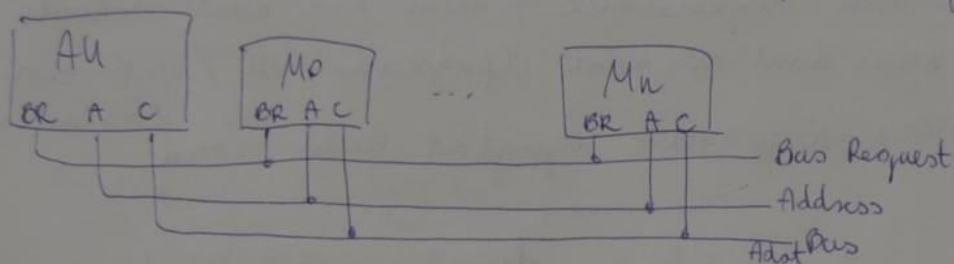
2. b) Sorozat: BG sorozan van kitéve



M_0 legmagasabb prioritású kérdésük meg először kell - e
A végé lehet soha nem

kezül sorra, de bármennyi Master soha kéthet egyből.

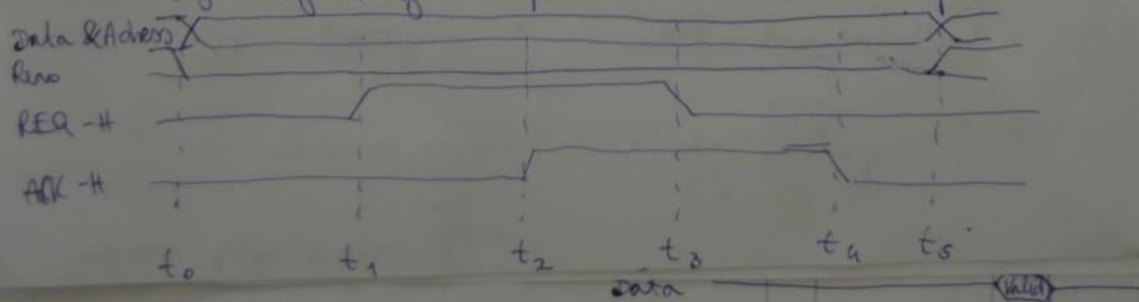
Lekezelés: + Master közös BR buszon igényel
Az AU + kábelben megérkező a legmagasabb prioritású igénylő (FIFO, round robin). Majd a legalsó értékű igénylő



2. b) Azonlapon Hardver Slaking fogalmát írja le és eseten

Utlite: Master or Slave - nel

- t_0 - ban az arbitrációból választott m. megadja a kérését slave alult.
- t_1 - ben m. kérését a request vonalra (+ slave veszi, de csak az t_0 - ban kérését jellel válaszol)
- t_2 - ben nyugtáz slave, le megkapta m - től a adata
- t_3 - m. is megkapja a nyugtázott slave - nel. Siker
- t_4 s ezeléki ut és felrakódítja a request vonalra
- t_5 - ig meg tartja azt m. a req. vonalra



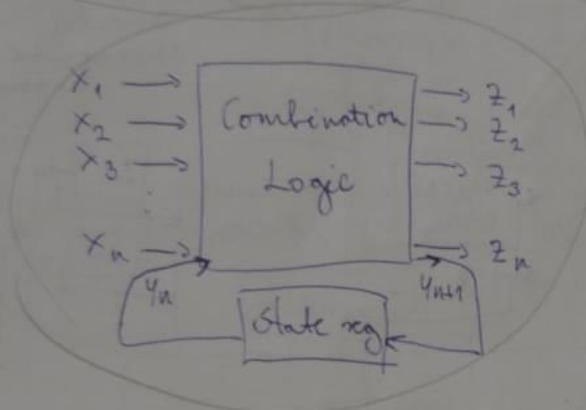
3A Mealy automata modell definíciója

Sorrendi hálózatok alapmodellje

A kimenet nem csak a pillanatnyi bemenetől de a korábbi értékektől is függnek

Bemenet: X Kimenet: Z Állapotok halmara: Y

$$\begin{aligned} \delta(x_n, y_n) &\Rightarrow y_{n+1} && (\text{köv. állapot meghat}) \\ \mu(x_n, y_n) &\Rightarrow z_n && (\text{kimenet meghat.}) \end{aligned}$$



3 halmaz

2 blokk

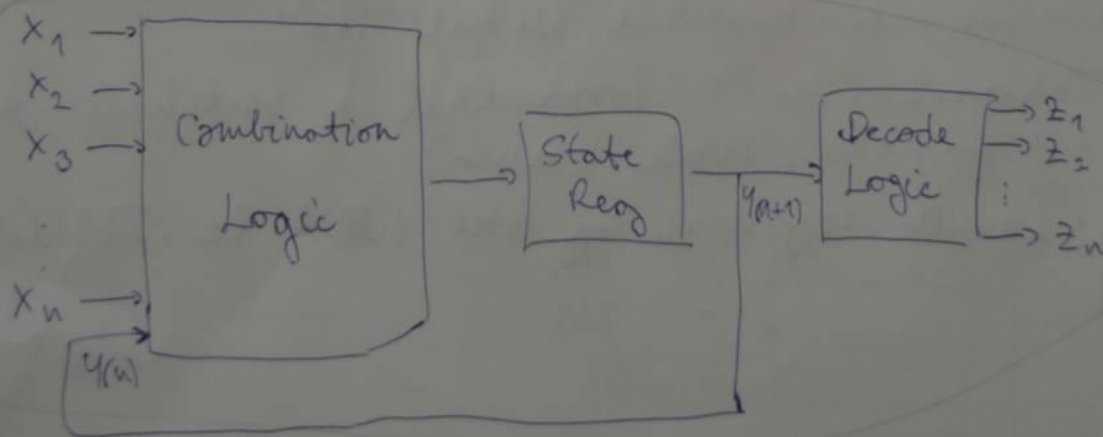
1 rajz

3B. Moore automata modell

Kimenetek közvetlenül csak a pillanatnyi állapotoktól függhet

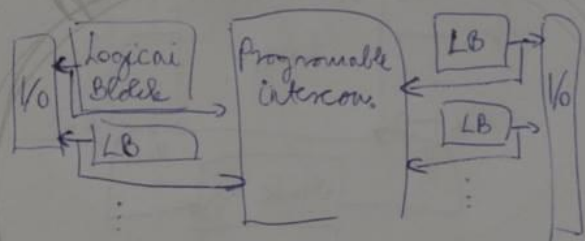
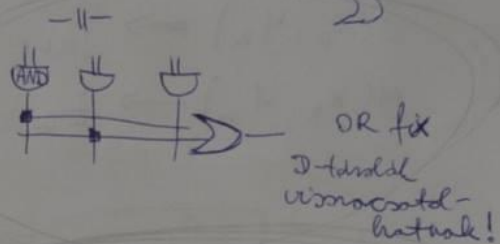
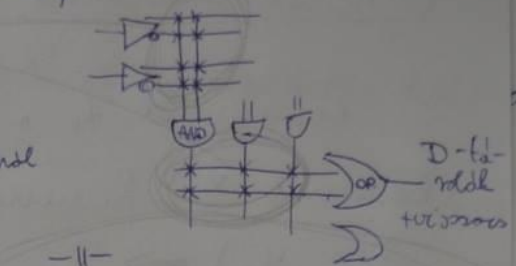
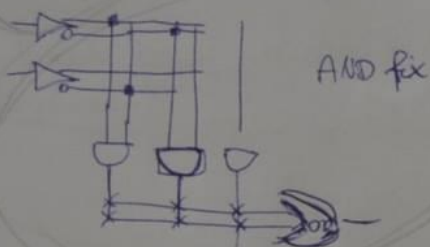
Bemenet: X Kimenet: Z Állapotok: Y

$$\begin{aligned} \delta(x_n, y_n) &\rightarrow y_{n+1} && (\text{köv. állapot meghat}) \\ \mu(y_n) &\rightarrow z_n && (\text{kimenet meghat.}) \end{aligned}$$



2x 4A Vezérlőegység programozható makrocellákkal (fogta, elbra, jellemzők)

- PLA (AND, OR program)
- PAL (AND) gyorsabb a PLAnál
- PROM (OR)
- EPDL, CPLD



LB = PAL / PLA v. Regiszterek
Intercon. = Teljes v. Részleges

4B Mikroprogramozott vezérlő feladata vertikális mikroprog. (elbra, eldny, hátrny)

Feladata: a memóriában lévő gépi kódú utasítások értelmezésének, kivételére bontása és funkcionális egységek vez.

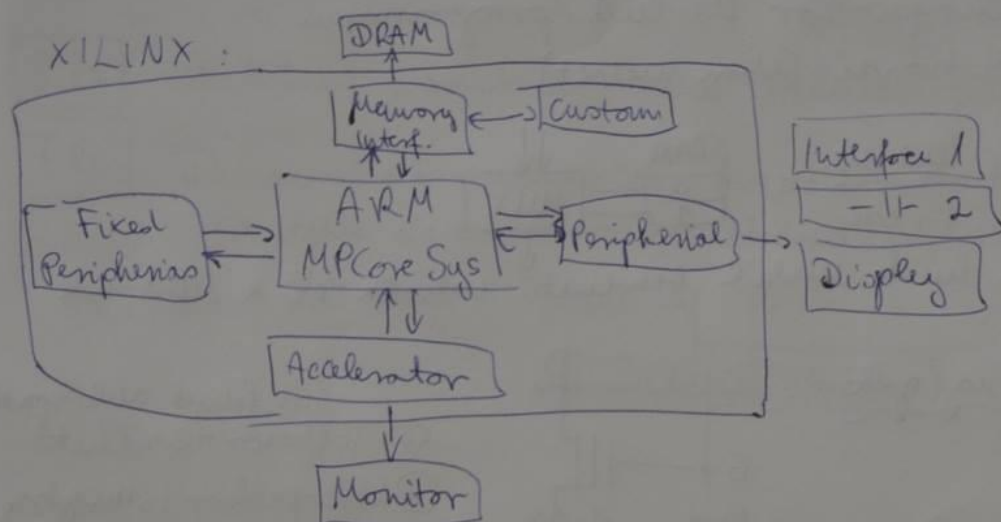
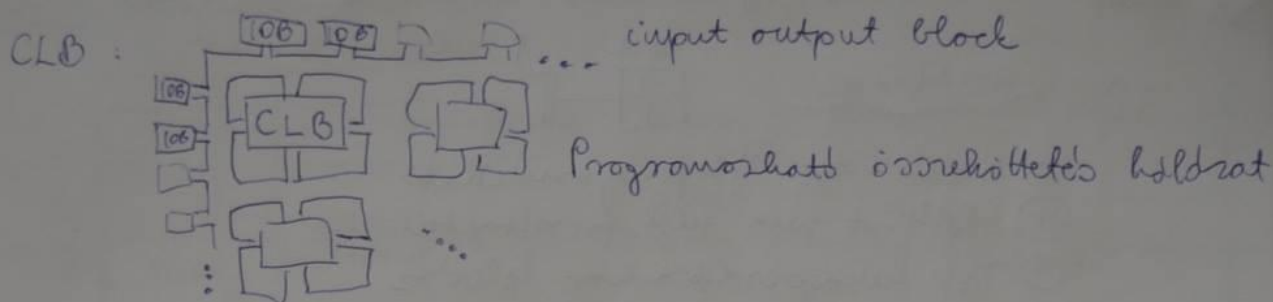
Mikroprogramozott. Az adatutvitel vezérlési pontjait (ROM) memóriából kiolvasott vertikális vagy horizontális mikrokódú utasításokkal állítjuk elő.

Eldny: Takarékos az erőforrással a vertikális mp. (fogasítás, bitek száma)

Hátrny: Lassabb, egyszerre egy bitet állít be, Dehódolással

Albra (?)

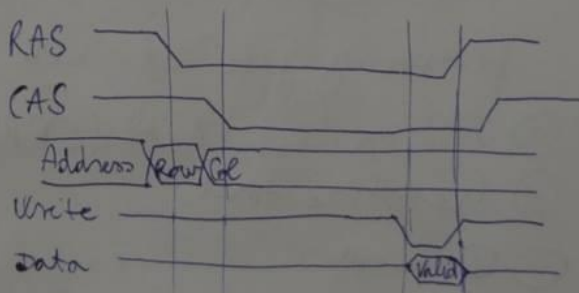
5A. Rajzoljon XILINX FPGA-t (CLB és I/O)



5B. Vezérlésszerű programozható makrocellák:
(forrta, ábra, jellemző)
[PLA, PAL, PROM, CPLD] $\approx$ 4A. ✓

6A. Dinamikus memória írás/olvasás [idődiagram]

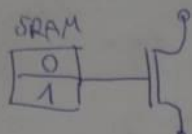
Olvasás : Sor cím : RAS
Oslop.c : CAS



Expansion bus (100 MHz)

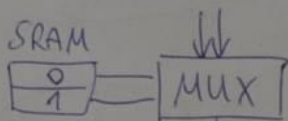
2x [6B] Hogyan programozható egy VLSI (rögzített)

a, SRAM -os:



- ⊕ Végfelhasználó újraprogramozható
- ⊕ SRAM-at nem kell frissíteni
- ⊖ Táp kikapcsolásakor elvész
- ⊖ Behatárolás fel kell programozni
- ⊖ Sdk-transzi (Nagyon nehéz)

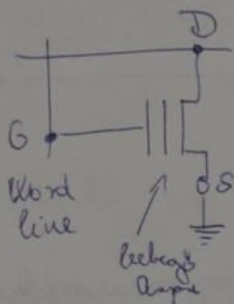
b, Multiplexelés:



SRAM 0 v. 1-ese

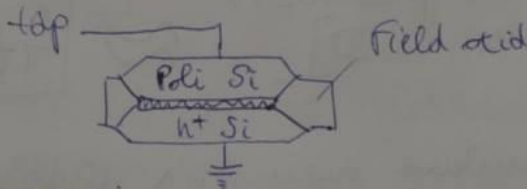
Udvarozta ki a megjelölt bemeneti csatlakozás a MUX-nak

c, Floating gate:



- (UV fénnyel átlátszó)
- ⊕ Töltésérő törölhető
- ⊕ Kikapcsolásakor is megőrzi

d, Antifuse:



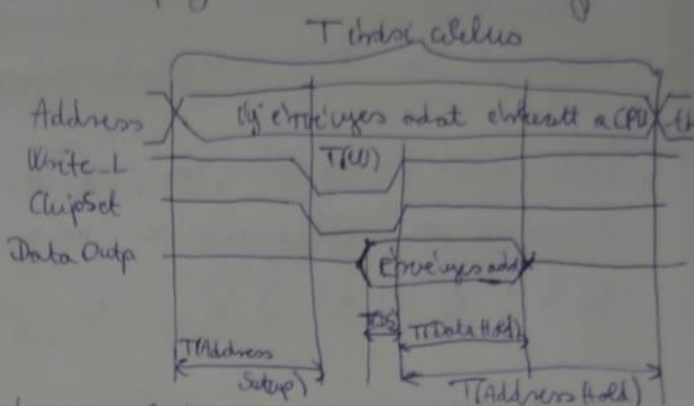
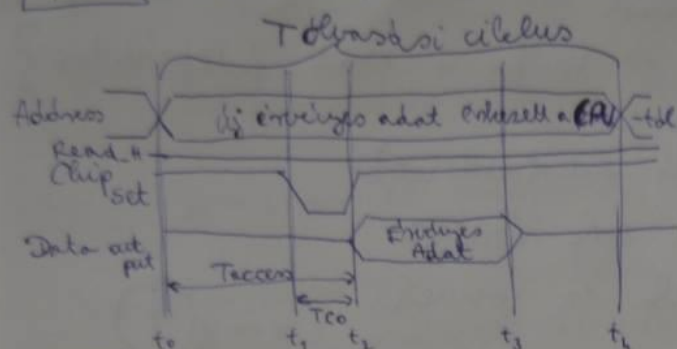
- ⊖ 1-szer lehet egyfelhasználóval programozni
- ⊕ Kicsike
- ⊖ Drága előállítás (Simulátorok)

e, EPROM / EEPROM / flash

Floating Gate-et alk.

- ⊕ UV-vel törölhető (EPROM) Elektron (EEPROM)
- ⊕ Megőrzési idő.
- ⊖ Drága

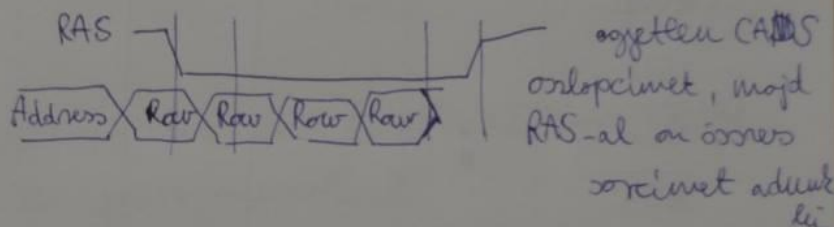
7A Statikus memória olvasási folyamata (idődiagram)



8B Statikus memória írási ciklusa (idődiagram)

7B Dinamikus memória frissítési folyamata (idődiagram)

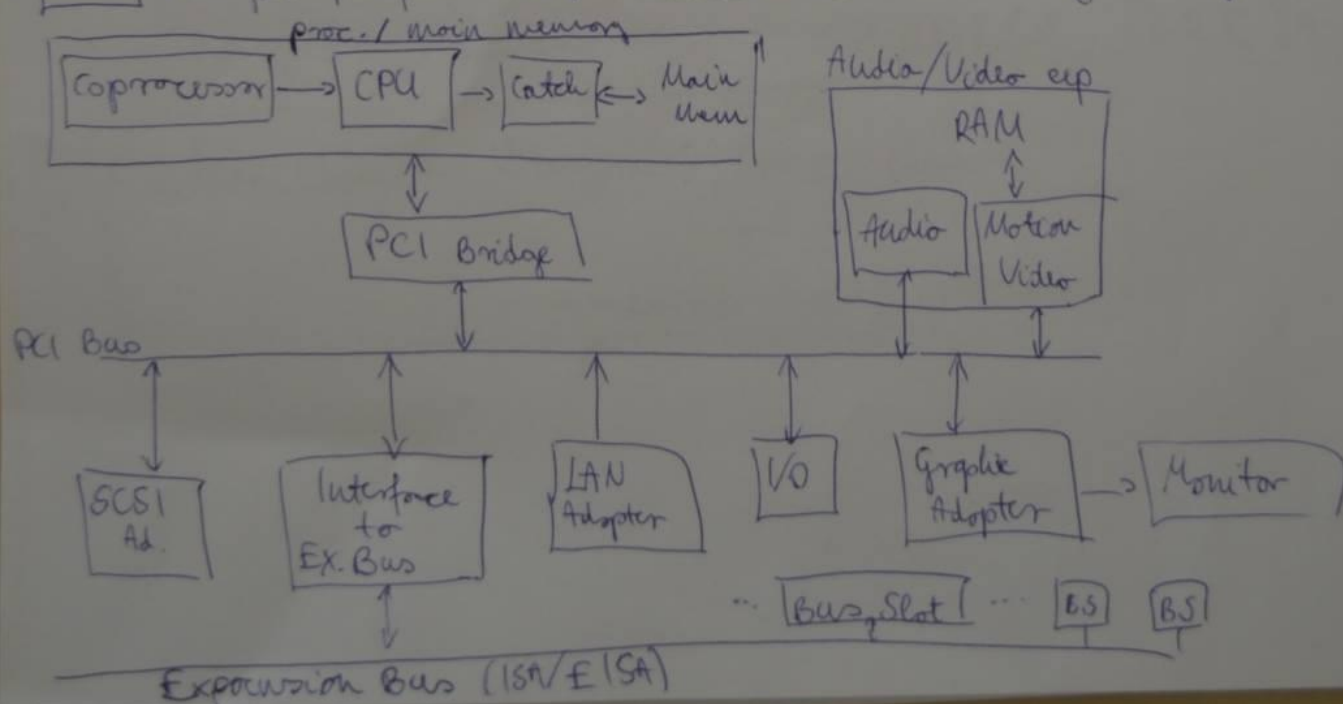
kondíciók között az írást → frissítésig idővel exp. csökken frissíteni kell őket:



8A Hogyan programozható egy program VLSI (rész. e-h)

[SRAM, Multiplexer, Floating Gate Antifuse] ≈ [CB]

9A Rajzolja fel PCI busra PC blokkdiagramját

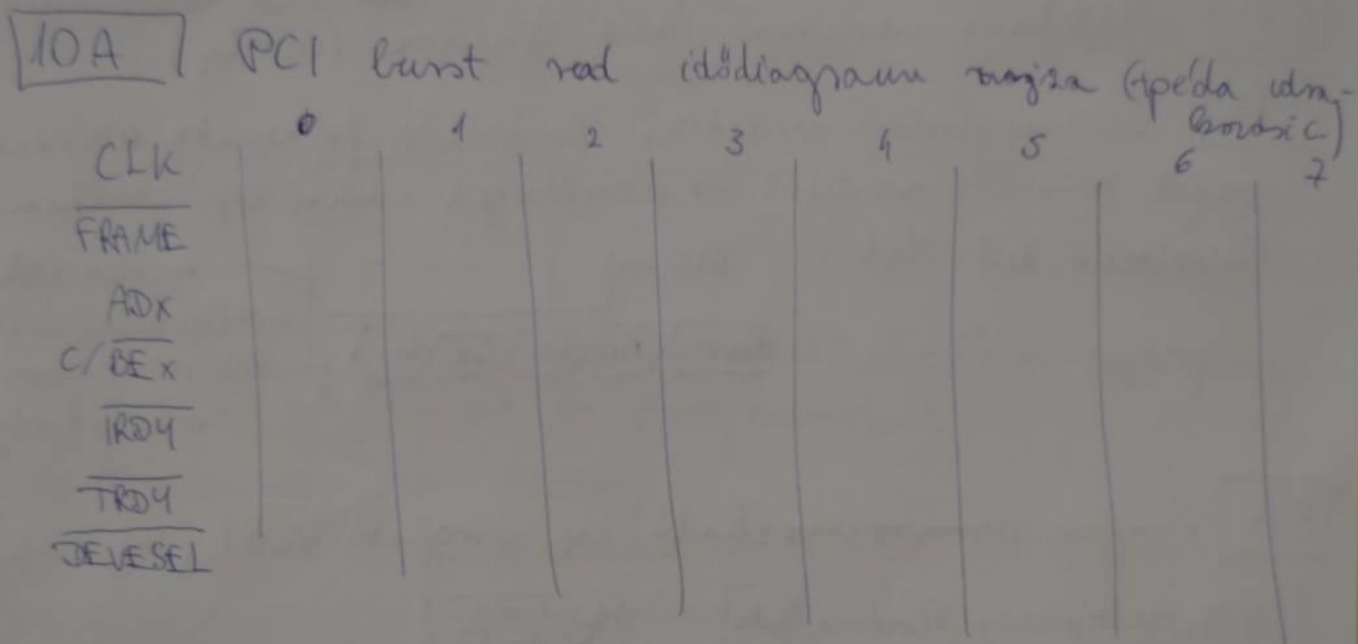


98 ^{híjny} Atviteli tehet-e a PCI busz jélei? ^{használat} Milyen jelcsoporthoz
vannak a PCI csatlakozáson és milyen a csatlakoztatás?

188 lábú csatlakozó (94/94 ddolomkelet)

Zavardóde ellenőrlés ve'gett az GND.

12V; 5V; 3.3V -os jélek is jönnel. ... (?)



10B PCI burst write — " —

11.A. Szegmentált virt. mem. kezelés (rajz + jell)

11.B. Lapazdasos virt. mem. kezelés

Lap: program és memória is fizikailag egyesített részek

⊖ Gondasbigholom kicsi méretű adatot / van osztó
és nagy méretű blokk kell feltérni

$$\text{Fizikai cím} = \text{Lapbázis cím} + (\text{offset})$$

Lap Eltolása

⇒ BELSŐ TÖRÉS.

⊕ Gyors

Szegmentálás: Nagy változó méretű logikai egységekre bontja a memóriát és programokat

1 főregrus: 0. főprógi
1. subrutin
2. csak olvasható
3. do/írható

⊖ Memóriának éves területek alóra mégsem felel egy nagyob regrus be. ⇒ KÜLSŐ TÖRÉDEZÉS.

$$\text{Fizikai cím} = \text{Segmens bázis cím} + \text{Segmens belüli eltolás}$$