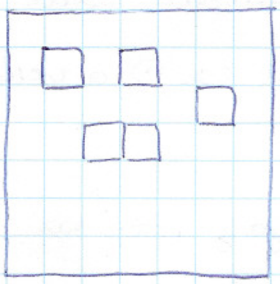
6. előadás X. 17.

Memóriák

ROM RAM

- memória alapja
- legfontosabb tul.: a feszültség kikapcsolásával nem felejti el a tartalmát



Egy maszkon kialakítják azt a bitképet, amit a felhasználó látni szeretne.
(Maszkprogramozott ROM)

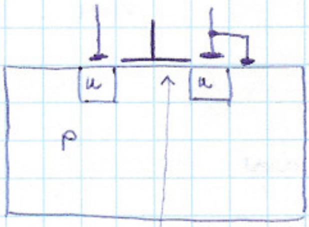
- ez a gyártás nagy darabszám esetén volt kifizetődő.
- gyártunk olyan memóriát, amelyben minden cella tartalma 1-es. Némelyikre nagy áramot kapcsolunk, az a cella tönkremegy, és 0-t fog adni.
- ezt a változtatást a felhasználó tudta megtenni, így lett a PROM. Ezt a felhasználó a helyszínen be tudta égetni. (8 bitből pl. kiégetünk 7 bitet, így kaphatunk 02-t. |0|0|0|0|0|0|1|0|)

Így a költség eloszlott az ilyet rendelők között ⇒ így olcsóbb lett.

- ha változtatni szerettünk a hökchirával kezelt diódát ⇒ nem lehetett (a folyamat irreverzibilis). Kivettük az áramkört, vettünk egy újat, beleégettük az új programot, és lehetett használni megint.

n csatornás „T"-vezérlésű tranzisztor.

technikai felépítése:



ha pozitív feszültséget kapcsolunk, a két elektróda között az egyiket visszatölti, és az elektródához képest lesz ⊕ a töltés.

- A kondenzátorban az elektróda szigetelőanyag, így ha egyszer feltöltöttük, elvileg nem kell többet. (Lassan azért mégis elfolyik az áram.)
  Jött az ötlet, hogy nem kell rombolni. Valahogy a kondiba töltést kell juttatni (nagyobb feszültséggel, mint a töltési feszültség). Ez kb. 12V-tal tehető meg.
  Mivel csak töltésbevitel történt, így ha el tudjuk tüntetni a töltést, újra lehet tölteni. Ez nem rombolt el.
- A kondiban a szigetelőanyagot ionizáljuk, és akkor a töltések megszűnnek. (nem csak a γ-sugárzás, hanem az UV is ionizál) Mindegyik kondiból el lehet vezetni a töltést, és akkor újra programozni lehet.
  Ez volt az (UV)EPROM. → törölhető memória.
- Volt egy oldalág: az OTPEPROM.

  one time programing → 1x programozható

kénytelenek voltak bakelit helyett kerámiából gyártani ezt. Ez is EPROM volt, de nem lehetett törölni, mert nem volt benne ablak.

- A kémikusok hoztak olyan anyagokat, amelyek amorf állapotban 2 állapotban léteznek, így megvan az a két állapot, ami kell nekünk. → EEPROM
  (electrical)

elektronikusan átírhatók az egyik állapotból a másikba. Nem kellett megvilágítani, kivenni, csak árammal elektronikus úton lehetett írni.

(így lehet ROM BIOS-t frissíteni, hogy átírjuk.) → A ROM írható is.

- Ezek az újítások már nagyon drágák (arany áron kaphatók)
- Létrejött a FLASH EEPROM, amiből az EEPROM már lemaradt. → Írható, olvasható, és a feszültség kikapcsolása után is megőrzi a tartalmát.

ROM hátránya:

- a ROM lassabb, a RAM-nál. (azért van, mert a ROM-ot csak rövid ideig használjuk, amíg bebootol a rendszer)
- Az írás nagyságrendekkel lassabb, mint a kiolvasás → így az írás nem igazán javasolt.
- A EEPROM változtatások (írások) száma korlátozott. kb. 1000'000-szor

RAM (random acces memory)

- SRAM (statikus)
- DRAM (dinamikus)

- a magnószalagnál az adatok soros elérésűek, míg máshol közvetlen az adatok elérése

• tetszőleges címek sorrendje

SRAM: • a memória alla tartalma mindig ugyanaz. beleírtuk a tartalmat, amíg mást nem adunk vissza ugyanazt adja vissza

DRAM: • a celláknak tartalmat adunk, és kérdezgetjük, nem felejti el a tartalmát, hosszabb idő után mégis elfelejti

• a dinamikus ram cellája 4 milliszecundum alatt elfelejti a tartalmát.

• nem gyártunk oda kondenzátort, a szigetelő fém még ott marad → a szigetelőn lévő nagyon kicsi áram eltünteti

• a kondenzátort kisütjük a fele értékig. Ha töltött volt, a kisütés után is fent tudta tartani a csatornát → így helyreállítják a töltését, amelyik nem volt töltött, az nem töltődik újra. Gondoskodni kell, hogy a háttérben ezt felfrissítse töltéstartalmat, úgy, hogy folyton kiolvassa.

• kell a memóriatartalom a processzornak is, de a háttérben 4 ms-ként kiolvasódik a memória tartalma.

• 1 GB memóriánk van → 10^9 memóriacellát 4 ms-ként kell kiolvasni

• a memória mátrixos felépítésű. → sor-oszlop címzést alkalmazunk. Megadunk 1 sort, meg 1 oszlopot, amin valójában a metszésben lévő cella

tartalmát aztán kiolvasni.

	D
	R
RAS —	A
CAS —	M

– mondjuk a sort, így a sor minden egyes cellája részletében lesz → így mindegyik frissül.

- a címző félcímzéseket mond → a cél a frissítés. Egyszerre el-kezdjük az összes sort megcímezni → 1 oszlopcímzést adunk hozzá.

DMA feladata: szól, hogy letelt az idő, frissíteni kell.

A DRAM is olyan, mintha statikus lenne, de mégis frissíteni kell, mert a kondenzátorból szivárog az áram.

- előnyös, ha egymást követő címeket gyorsan tudunk olvasni, vagy írni → FPRAM (fast page módú RAM)

  ⇓

  kijelölte a sort, és végig beolvasta a sort (eddig sor-oszlop; sor-oszlop-pal olvastak be), majd utána adott meg oszlopokat.

- megadjuk a sort, majd az oszlopot, és a memóriatartalmat kiolvassuk, majd megint oszlopot, kiolvassuk ... stb.

újítás: EDORAM → fejlett adatkimenet. Nem várakozott a válaszra, megadta a következő címet, ha érkezett a válasz, akkor visszatért, eltárolta valahol, és ment tovább.

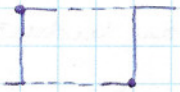
memory interleave: a cellák egy időben címződnek, majd az alacsony helyiérték miatt váltakozik a cellák között.
(kiindulható felezésen, és így a közvetlen egymás után levő cellákat rövidebb idő alatt lehetett elérni)

- SDRAM (synchron data RAM)

  a fast page módot követte, és alkalmazott még egy előre meghatározott ütemet. Minden ütemre jutott egy cella → így gyorsult a kiolvasás.

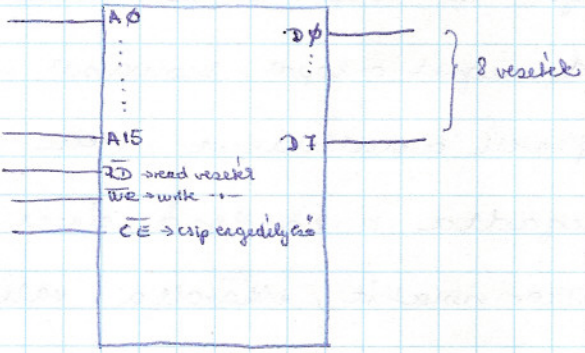
- DDR RAM (double data

  oldalt egy olyan rajzot, amely nem a felfutó élre volt érzékeny, hanem a lefutóra → egymás mellé rajzol → így a „fent" és a „lent"-re is olvasott a memóriából → kétszeresére gyorsult az olvasás.



Címek

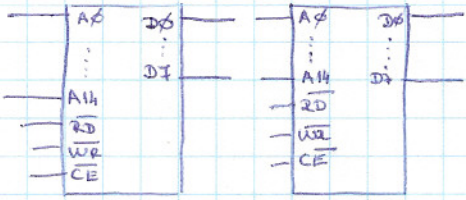
64K statikus RAM



64KB memóriának 16 címbit kell ⇒ 2^{16}-on féle állapot

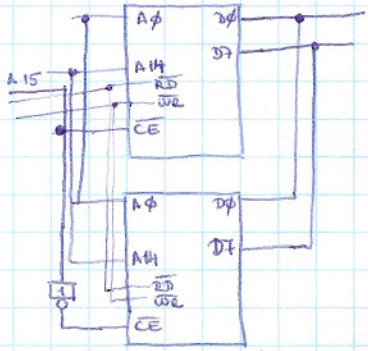
- a read- és write vezeték negálva → mindig 1-et adnak, és ha műveletet végeznek, → 0.
- Chip enable vezeték is negált. Ha alacsony → a memória úgy működik, ahogy kell. Ha magas, → a memória olyan, mintha ott sem lenne (→ nem működik)

egy 64K-s helyett 32K-sból 2db-ot veszünk



úgy tudjuk megoldani, h. 0-32K-ig az egyik, 33-64K-ig a másik működjön. Azt kell, hogy meglássa, hogy melyikben dolgozik, hogy az A15-ösön „0" vagy „1"-es szerepel-e.

Teljesen úgy címzi az első 32K, mint a másikban, csak a címeket nem különböztetjük.



A CPU kiadja, hogy az A15-ről mi az értéke, csak azt nem tudjuk hová tömi.

Ha az egyik memória működik, akkor a másik nem, így az adatvezetékeket is össze lehet tömi, a $\overline{WR}$ és a $\overline{RD}$-eket is, mert mindig az ad adatot, amelyik működik. Kell hozzá egy inverter. (Ha a $\overline{CE}$ 0 ⇒ működik a memória)