

**Zalotay Péter**

**DIGITÁLIS TECHNIKA II.**

Elektronikus jegyzet  
BMF Kandó Kálmán Villamosmérnöki Kar

*1. fejezet*

# **A mikrogepek**

**(Mikroszámítógépek)**

**Az 1. fejezet tartalomjegyzéke:**

<b>1. A mikrogép (Mikroszámítógép) .....</b>	<b>4</b>
<b>1.1. A központi vezérlő egység.....</b>	<b>5</b>
<b>1.2. Processzor vezérlés .....</b>	<b>5</b>
<b>1.3. Memóriák.....</b>	<b>5</b>
1.3.1. A program memória( PM ).....	6
1.3.2. Az adatmemória ( DM ) .....	6
<b>1.4. Periféria illesztés .....</b>	<b>6</b>
<b>1.5. A tápegység ( TE ) .....</b>	<b>6</b>
<b>1.6. A busz-rendszer .....</b>	<b>6</b>
<b>1.7. mikroprocesszor .....</b>	<b>7</b>
1.7.1. Utasítás feldolgozó .....	8
1.7.2. Aritmetikai - logikai egység ( ALU ) .....	8
1.7.3. Regiszter - tömb .....	9
1.7.4. Busz illesztés .....	9
1.7.5. Időzítő - vezérlő egység.....	9
1.7.6. A processzor megszakításkezelése .....	9
1.7.7. Memória írás, olvasás.....	10
1.7.8. Periféria írás, olvasás.....	11
<b>1.8. Memóriák, és illesztésük.....</b>	<b>11</b>
1.8.1. A memóriák elvi felépítése.....	12
A tárolás történhet <i>bit</i> -es, vagy <i>bájt</i> -os egységekben .....	12
1.8.2. Csak olvasható (fix) memóriák .....	13
1.8.3. Írható, olvasható memóriák .....	15
1.8.4. Memóriák illesztése.....	16
<b>1.9. Perifériák illesztése a mikrogéphez .....</b>	<b>18</b>
1.9.1. Bemenetek (input) illesztése .....	18
1.9.2. Kimenetek (output) illesztése .....	19
1.9.3. Perifériák címezése .....	19

## 1. A mikrogép (Mikroszámítógép)

### Bevezetés

A tantárgy feladata a **mikroprocesszoros** un. intelligens berendezések - PC-k, ipari irányító készülékek ( PLC-k ), különböző hírközlési rendszerek stb. - működésének megismeréséhez, programozásuk elsajátításához szükséges **alapismeretek megtanítása**. **Jártasságot** kíván nyújtani az **assembly** nyelvű **programozásban**, valamint **készséget** kialakítani a mikrogépek **használatában**.

A felsorolt célkitűzések elérése érdekében foglalkozunk

- a **mikroszámítógépek** általános felépítésével és működésükkel,
- a **mikroprocesszorok** rendszertechnikai szerepével,
- az un. "egy chip-es" számítógép, a **mikrokontroller** alkalmazásával,
- a **perifériák** illesztésével,
- a **processzorközei programozás** (assembly nyelvű) alapjaival,

A mérési gyakorlatok keretében egy általános célú mikrogépen sajátítják el a hallgatók

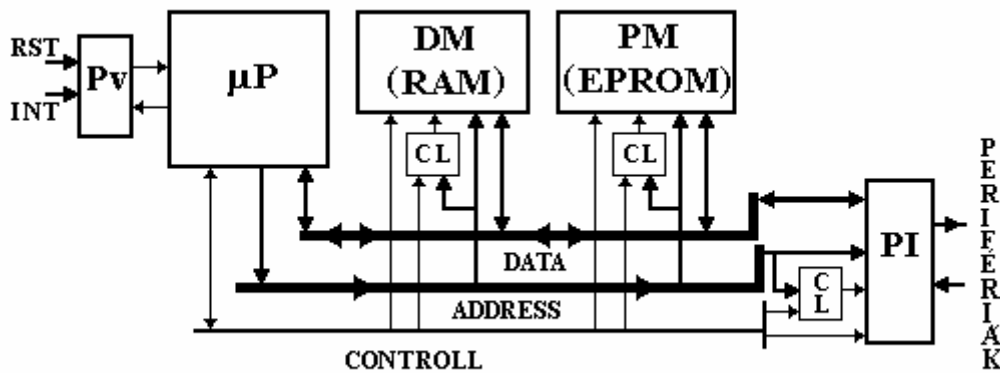
- a **programozás** alapjait,
- a különböző **programszerkezetek** kialakítási módszereit,
- a **megszakítások** használatát,
- a különböző **irányítási** feladatok programozását.

**Mikrogépeknek** (mikroszámítógépek) nevezzük a **mikroprocesszoros**, vagy mikrokontrolleres digitális berendezéseket. A személyi számítógépek ( PC - k ) mellett többek között ilyen készülékek irányítják a különböző ipari gyártósorokat, korszerű gépkocsikat és a szórakoztató elektronika, valamint a háztartási gépek egy nagy csoportját is.

A mikrogépek mindegyikében megtaláljuk

- a **központi vezérlő** egységet ( CPU Central Processor Unit ),
- a **program** memóriát,
- az **adat** memóriát,
- a **periféria illesztő** egységeket,
- és a **tápegységet**.

A mikrogép rendszerteknikai felépítését szemlélteti a 1. ábra.



1. ábra

### 1.1. A központi vezérlő egység

A CPU irányítja a mikrogép program szerinti működését. A működés az ún. **utasítás ciklusok** sorozatából tevődik össze. Egy utasítás ciklusban a CPU

- a programtárból **beolvassa** az azon következő utasítást,
- **végrehajtja** az utasításnak megfelelő elemi műveletet, és
- **előállítja** a mikrogép elemeit **vezérlő** jeleket.

### 1.2. Processzor vezérlés

A mikroprocesszor működését az ún. **processzorvezérlő** jelekkel lehet változtatni.

A működést a beépített, vagy külső **órajel** generátor ütemezi.

További vezérlő jelek:

- a bekapcsolási alapállapotot vezérlő bemenet a **RESET**,
- működést befolyásoló **várakozás** (wait) vezérlő bemenet **READY**,
- **megszakítás-t kérő** bemenet **INT**,
- a megszakítás **elfogadását** visszajelző kimeneti jel **INTA**,
- külső **busz vezérlő**, vagy másik processzor által kiadott ún. **busz kérés**i jelet fogadó bemenet, és a busz kezelését átadó kimeneti jel
- a **RD** (olvasó), **WR** (író), és periféria választó jelek.

### 1.3. Memóriák

A mikrogepek működéséhez feltétlenül szükséges a programot, és a számítások, műveletek operandusait tároló memóriákra. Funkcionálisan ezek külön egységet alkotnak, ugyanakkor fizikailag egy memória is alkothatja ezeket.

### 1.3.1.A program memória( PM )

A mikrogép működését előíró *program utasításainak* megfelelő *műveleti kódokat*, és egyes operandusait tárolja a program memória. A műveletek végrehajtása során e memória területről *csak olvasás* történik. Olyan mikrogépekben amelyek csak egy programot hajtanak végre ( pl. videó-magnó irányítása), a programot ún. fix-memória (ROM, PROM, EPROM, EEPROM) tárolja. A PC - ben futás közben is változik az éppen végrehajtandó program, ezért itt *írható-olvasható* memóriába ( RAM ) tárolódnak az éppen *aktuális* program kódjai, operandusai.

### 1.3.2.Az adatmemória ( DM )

Az adatmemória tárolja a működés során használt *változók*, perifériákról beolvasott, ill. kiviendő információk *aktuális értékeit*. A feladatból adódóan csak írható-olvasható memóriák (statikus és dinamikus RAM, mágneses tárolók stb.) használhatóak ilyen feladatra.

## 1.4. Periféria illesztés

A mikrogép és a külső eszközök *adatforgalma* és *vezérlése* a periféria illesztő egységeken keresztül történik. Mindkettő irányulhat a külső eszköztől a mikrogépbe (bemeneti-, vagy input egység), illetve ellenkező irányba (kimeneti-, vagy output egység ). Az *ember-gép* közvetlen kapcsolat alapeszközei az adat-, és parancsbevitelre szolgáló *klaviatúra*, illetve a különböző *megjelenítő* egységek (képernyő, szám-, és karakter kijelzők stb. ) ahová a mikrogép a műveletek eredményeit írja ki. Amikor a mikrogép egy berendezést irányít ( automatizálás ) az információk és parancsok *gép-gép* közötti illesztő egységeken keresztül történik.

### 1.5. A tápegység ( TE )

A mikrogép feszültség, és áramellátását biztosítja. A leggyakrabban *normál hálózathál* (230 VAC 50 Hz) állítják elő a tápegységek az integrált áramkörök biztos működését szolgáló stabilizált ( $U_{cc}=5\text{ V}$ ), illetve az egyéb célokra használt stabilizálatlan egyenfeszültségeket (12 V, stb.). Különleges mikrogépek tápenergiáját *akkumulátor* is biztosíthatja ( pl. gépkocsik fedélzeti számítógépei, meteorológiai mérésadatgyűjtők stb.)

### 1.6. A busz-rendszer

A *mikroprocesszor* közvetlen környezetét a különböző *memóriák*, programozható *perifériák*, *illesztő*-, és *processzorvezérlő* egységek alkotják.

A környezet elemei közötti adatmozgás, és egyéb műveletek jelei a **BUSZ** vonalakon keresztül történik. Elvi felépítését láthatjuk a 1. ábrán.

A **DATA** (adat) buszon kétirányú adatmozgás lehetséges. A *cél*, vagy a *forrás* mindig a processzor.

Az **ADRESS** (cím) busz vonalain adja ki a processzor azt a címet (programszámláló tartalmát) ahonnan, vagy ahová az adat átvitelre kerül. A címbusz *vonalszáma* (szélessége) határozza meg a csatlakoztatható maximális *memóriakapacitást*. Például ha 16 bites címzés  $2^{16}$  bájt, azaz 64 Kbájt elérését teszi lehetővé.

A vezérlő vonalak, vagy **CONTROL** "busz" változó számú vonala miatt igazában nem nevezhető busznak. Ezeken a vonalakon keresztül *vezérli* a mikroprocesszor a *kiválasztott adatmozgatást*.

### 1.7. mikroprocesszor

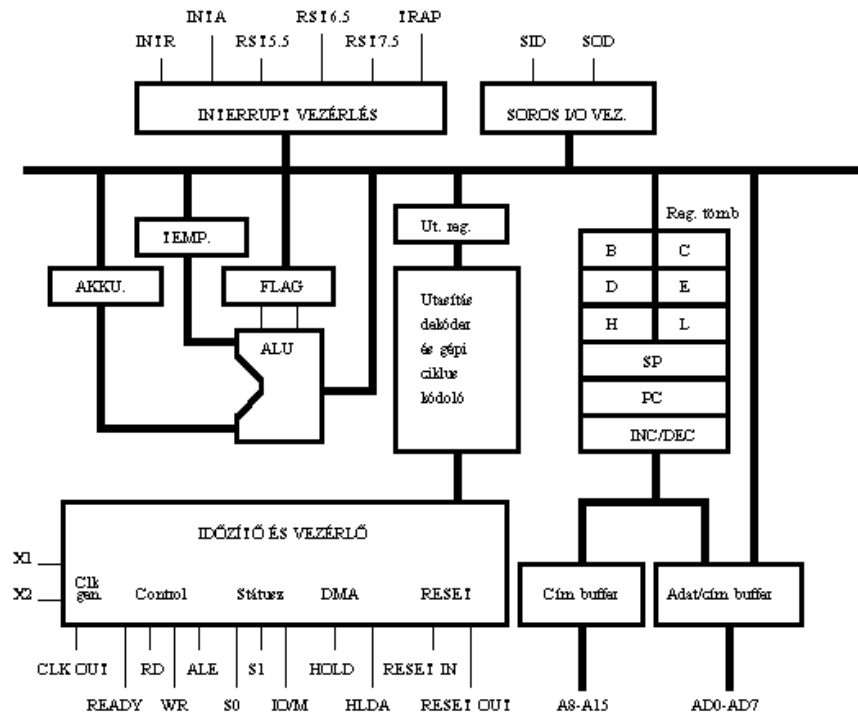
A mikroprocesszor olyan *integrált áramkör*, amely egy digitális számítógép *központi egységének* ( CPU - Central Processor Unit ) *alapvető feladatait* látja el.

Az Intel cég 8080 típusjelű első mikroprocesszora 1973-ban került a piacra. A mai napig számtalan változatot fejlesztettek ki, amelyek közül a legismertebbek a 8085, 8086, 80286, 80386, 80486, 80586 ( pentium ), és a sor folytatódik.

A mikroprocesszorban - gyártótól, és típustól függetlenül - megtalálhatók a

- *utasítás dekóder* és ciklus vezérlő,
- *aritmetikai-logikai* egység ( ALU ),
- *regisztertömb* ( Accumulator, operatív-, és átmeneti tárolók ),
- *Stack* mutató ( Stack-pointer SP ),
- *Program számláló* ( PC ),
- *időzítő* és *vezérlőegység*,
- adat-, és címbusz *meghajtók*.

A továbbiakban az Intel 8085 típusú mikroprocesszor felépítését tárgyaljuk. A mikroprocesszor blokk-sémája látható a 2. ábrán.



2. ábra

### 1.7.1. Utasítás feldolgozó

A soron következő - Fetch ciklusban - beolvasott utasítás egy **regiszterbe** íródik, amelyhez csatlakozó **utasítás dekóder** " értelmezi " az utasítás kódot, vagyis **előállítja** az utasítás végrehajtásához szükséges **belső vezérlőjeleket**.

### 1.7.2. Aritmetikai - logikai egység ( ALU )

A mikroprocesszor az utasításban előírt **aritmetikai** (ÖSSZEADÁS, KIVONÁS), valamint **logikai** (ÉS, VAGY, KIZÁRÓ-VAGY, és TAGADÁS) műveleteket az **ALU** (Arithmetic Logic Unit) műveleti egység végzi. A műveleti egységhez tartoznak az **Accumulátor** (ACC) a **Flag** (jelző), és az átmeneti (temporary) 8 bites **regiszterek**. A legtöbb műveletvégzés előtt az egyik operandust az ACC-be kell vinni, a másik operandusz a regiszter tömbben, vagy a külső memóriában, illetve a programban kell legyen. A művelet eredménye az ACC-be kerül, és egyúttal állíthatja a jelző (flag) regiszterben lévő feltétel biteket. Ezek a bitek

- **Zero** 1 ha az eredmény 0,
- **Carry** 1 ha az eredmény túlcscordul,
- **Sign** 1 ha az eredmény negatív,
- **Parity** 1 ha páros számú egyes van az akkumulátorban,
- **AC** (Auxiliari carry - közbenső átvitel) 1 ha az alsó négy bitről van átvitel.



### 1.7.3. Regiszter - tömb

A processzorban van egy írható, olvasható memória, amely egyrészt bájtos (8 bit), másrészt szavas (16 bit) szervezésű. A **B, C, D, E, H, L** jelölésű hat különálló bájtos regisztert jelent. Ugyanezek regiszterpárként is elérhetők a **BC, DE, HL** jelölésekkel. E regiszterek bájtosan operandusokat, míg a szavas szervezésben operandusokat, vagy címet tárolhatnak. További két regiszterpár a **Stack-pointer ( SP )**, valamint a **Program-számláló ( PC )**. Ezek mindig címet tárolnak. Az SP tárolja a Stack- ( zás ) memória aktuális címét. A PC-ben van a soron következő programbájt címe.

### 1.7.4. Busz illesztés

A mikroprocesszor és a környezete közötti adatforgalom a buszrendszeren keresztül történik. A processzor a **DATA** (adat) buszhoz **kétirányú**, míg az **ADRESS** (cím) buszhoz **egyirányú** (kifelé irányított) buszmeghajtó - **tri-state** - illesztő áramkörökön csatlakozik

A két busz lábkiosztása részben átfedett. Az adatbitek, és a cím alsó 8 bitje azonos lábakon (AD0...AD7) jelenik meg. Míg a cím felső 8 bitje külön lábakon (A8 .. A15) van kivezetve. A közös lábakon kivezetett adat-, és címbitek szétválasztásához egy **külső tároló** (8 bites Latch) szükséges. Ebbe az **ALE** (Address Latch Enable ) jel írja be ciklus elején kiadott címbiteket.

A vezérlő vonalak, vagy **CONTROL** "busz" változó számú vonala miatt igazában nem nevezhető busznak. Ezeken a vonalakon keresztül valósul meg az egész rendszer - pl. adat-cím szétválasztás ( **ALE** ), az írás ( **WR** ), vagy az olvasás ( **RD** ) - vezérlése.

A **processzort vezérlő** jelek biztosítják a működés **időbeli ütemezését** ( **CLOCK**, **READY** ), **alaphelyzetbe** állítását ( **RESET** ), a **busz használatának** átadását másik processzornak, vagy DMA vezérlőnek ( **HOLD**, **HLDA** ), **megszakítás** kérések fogadását ( **INT**, **TRAP**, stb. ), **viSSZajelzést** ( **INTA** ).

### 1.7.5. Időzítő - vezérlő egység

Az **időzítő - vezérlő** egység fogadja a processzort vezérlő jeleket és előállítja a **rendszer** működését **vezérlő** jeleket. A kiadott jelek egy része tri-state illesztésű.

### 1.7.6. A processzor megszakításkezelése

A mikroprocesszorok a **programban** megírt **sorrendben** hajtják végre az utasításokat. Az előre ismert, és jól definiált műveletek sorrendje viszonylag könnyen írható le. Sok esetben a feladat olyan, hogy a mikrogépnek egy **külső esemény** bekövetkeztekor abba

kell hagynia az éppen futó program végrehajtását, és az eseménytől függő *más programot* kell futtatnia. Ezt a folyamatot nevezzük *program-megszakításnak* (interrupt).

Alapvetően az alábbi két megszakítási eljárást alkalmazzák:

- egyik módszernél a megszakítás kérése, és elfogadása után a *külső vezérlő* adja meg azt a *címet*, ahonnan a processzornak hívnia kell az un. *megszakítás szubrutint*. E módszernél a *megszakítást kérő* jel érkezik az INTR bemenetre. A kérés *elfogadása* után a processzor adja az INTA jele(ke)t, amelynek hatására az *adatbuszon* kapja meg a cím bájtoit,
- a másik eljárásnál minden *megszakításkérő* bemenethez (TRAP, RST 5.5, RST 6.5, RST 7.5) egy-egy *adott cím* tartozik, és a processzor innen hívja az aktuális *megszakítás-rutint*.

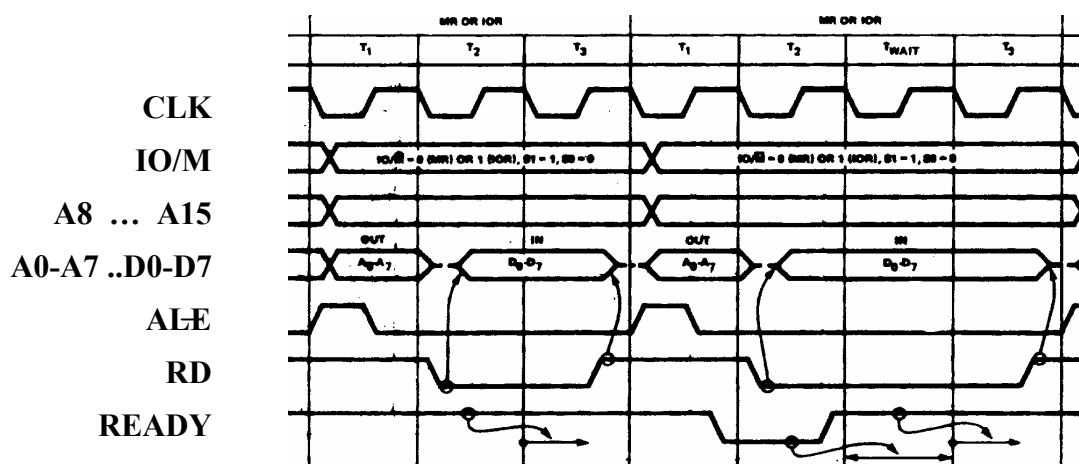
### 1.7.7. Memória írás, olvasás

A *memóriák* funkcionálisan két csoportba sorolhatók. Az egyik a *programot* (PM), a másik pedig az adatokat, és változók értékét tároló un. *adatmemória* (DM).

A processzor a programmemóriából olvassa ki az utasítás-kódokat, illetve a programban foglalt adatokat és címeket. Erre a memóriaterületre írás sohasem történik.

Az adatmemória tartalmát a processzor olvassa is, és írja is. A memória kezelést az IO/M jelű vezérlő vonalának 0 szintjével jelzi.

Egy bájt írása, vagy olvasása egy un. *gépi ciklus* alatt történik, amely az órajel periódusidejével megegyező *ütemekből* ( T1, T2, T3, stb ) áll. A memória, illetve periféria *olvasás* időzítését szemlélteti az 3. ábra.

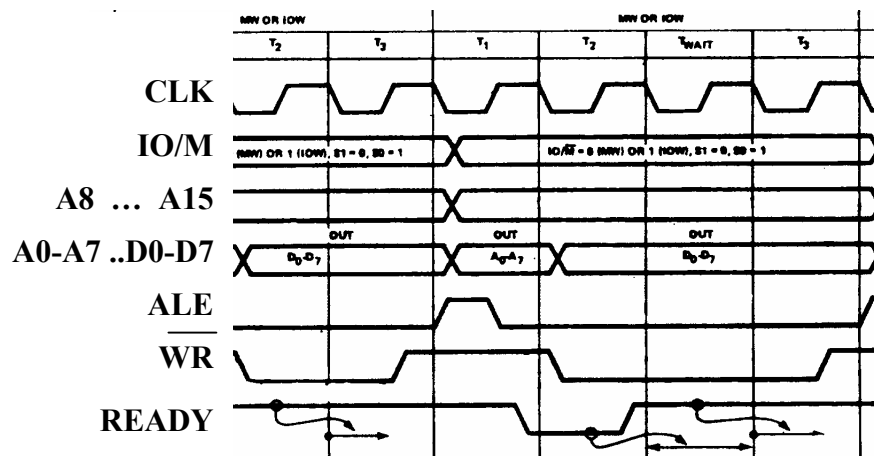


3. ábra

A **T1** (cikluskezdő) ütemben a **címbytek**, az **IO/M** (periféria/memória választó), és az **ALE** (címtároló) jelek kerülnek a megfelelő vonalakra. A **T2** - ben kerülnek a buszra az **adatbytek**, és az **írást** (WR=0), vagy **olvasást** (RD=0) vezérlő jelek. Ugyanekkor vizsgálja meg a processzor a **READY várakozást** kérő bemenetet. Ameddig ez aktív (0 szintű) un. **várakozási ütemet** ( $T_w$ ) következnek. A **T3** ütemben - a vezérlőjel felfutó élénél - történik az adatok **beolvasása** a processzorba (olvasási ciklus), **vagy** a **memóriába** (írási ciklus).

### 1.7.8. Periféria írás, olvasás

A leggyakrabban alkalmazott programozható perifériák az **időztető**, a **megszakítás-vezérlő**, a **kommunikációt** (soros, párhuzamos) végző, és különböző **bites**, vagy **bájtos portok**. A portok rendszeren belüli, vagy más rendszerekhez történő **illesztést** is elláthatják. A memória, illetve periféria **írás** időzítését szemlélteti a 4. ábra.



4. ábra

A processzor **periféria** írásakor, vagy olvasásakor csak **8 bites** címet küld a buszra, és ezt mind a címbusz alsó-, mind pedig a felső 8 bites részén egyformán. Ily módon **256** különböző cím lehetséges. Az IO/M vonal ekkor 1 szintű. Miután az írást a WR, az olvasást pedig a WR jel vezérli, ezért azonos címen egy kimeneti-, és egy bemeneti - 8 bites - periféria érhető el.

A periféria írási-, és olvasási gépi ciklusok időzítése megegyezik a memória elérési gépi ciklusokéval.

### 1.8. Memóriák, és illesztésük

A következőekben röviden áttekintjük a mikrogépekben használt, **félvezető** alapú memóriák

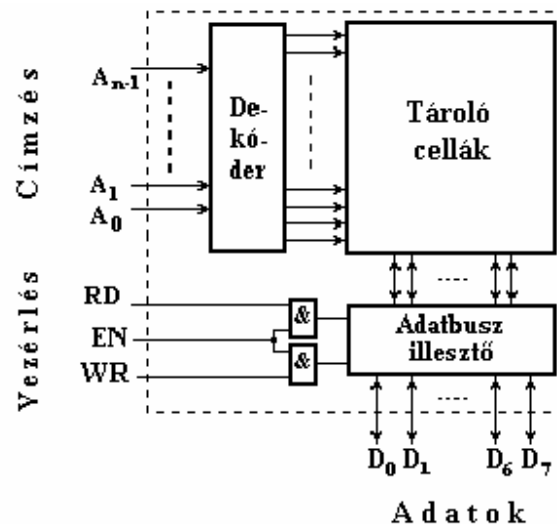
- általános felépítését,

- legfontosabb változatait,
- címzési módját.

A leggyakrabban használt bájt szervezésről beszélünk alapvetően. Az ilyen memóriában tárolt adatok **bájtonként** (8 bit) **olvashatók**, vagy **írhatók**. Nem térünk ki azokra a memóriákra, amelyeknél a tárolt információ egyes bitjei közvetlenül érhetőek el.

### 1.8.1. A memóriák elvi felépítése

Az 5. ábra szemlélteti egy memória (chip) alapvető részeit.



5. ábra

- A **tároló-cellák** -ba kerülnek beírásra a tárolni kívánt információk, **adatok**.  
A tárolás történhet **bit** -es, vagy **bájt** -os egységekben
- A **dekóder** választja ki a - **címek** ( $A_0, A_1, \dots, A_{n-1}$ ) alapján - meghatározott **adatot**.
- Az **adatbusz illesztőn** keresztül jut a kiválasztott helyről az adat a **buszra**, vagy fordítva.

A leírt memória-felépítés elvileg minden változatnál azonos. Lényeges különbség a tárolandó információ (adat) beírási módja között van. Ezek alapján különböztetjük meg a

**csak olvasható** (fix), és az

írható-olvasható tárolókat.

A következőkben áttekintjük e-memória változatok lényeges tulajdonságait.

### 1.8.2. Csak olvasható (fix) memóriák

A **fix** memóriák megnevezés arra utal, hogy a tárolt adatokat **csak olvasni** lehet a mikrogépben futtatott alkalmazói programban. Az adatok beírása csak **programozó** egységgel végezhető.

A beírás történhet

- **maszkolással**, a memória gyártása során, amelyet **ROM** – nak (**Read Only Memori** – csak olvasható memória) nevezünk,
- a **felhasználó** által, de csak **egyszer** programozhatóak a **PROM** – ok (**Programable Read Only Memori** – programozható csak olvasható memória),
- a **felhasználó** által **többször** is programozhatóak az **EPROM** –ok (**Erasable Programable Read Only Memori** – törölhető, és programozható csak olvasható memória), illetve az **EEPROM** –ok (**Electrical Erasable Programable Read Only Memori** – elektromosan törölhető, és programozható csak olvasható memória).

Az **EPROM** –ban tárolt adat **törlése** - a tok tetején lévő quartz ablakon keresztül - meghatározott hullámhosszú **UV sugárral** végezhető („napoztatás”). Az **EEPROM** adatai **elektromos árammal** törölhetőek ki, a készülékből való eltávolítás nélkül.

A következőekben röviden tekintsük át a fix memóriák egyes részegységeinek feladatát, működését.

⇒ **A dekódoló egység**

Feladata, hogy a **címbemenetekre** ( $A_0 \dots A_{n-1}$ ) adott **bináris** kód dekódolásával, **1 az n-**ből kóddá alakításával **kiválasszon** egy **memória-egységet**, bájtot, vagy bitet. A címbemenetek száma **n**, határozza meg a memória **kapacitását**, vagyis hány **egységnyi információ** tárolható benne.

⇒ **A memória cellák**

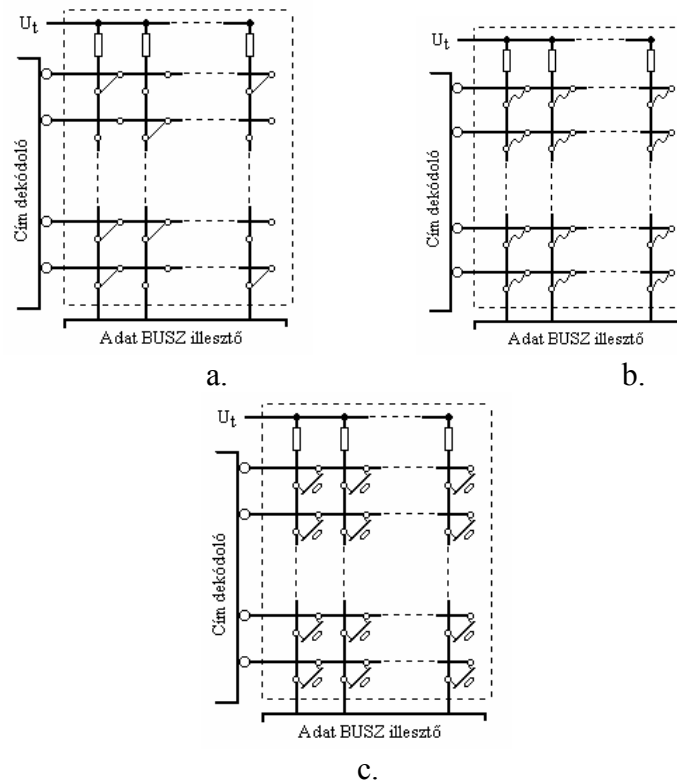
Az ismertett változatok **bájtos-memóriacelláinak** elvi megvalósítását szemléltetjük a 6. ábrán. Mindhárom változatnál **mátrix** elrendezést rajzoltunk. A dekóder a **sorvezetékek** egyikére ad csak **0** szintet. Az **oszlopvezetékek** mindegyike – ellenálláson keresztül – az **1 szintű** tápfeszültségre ( $U_t$ ) csatlakozik. A sor-oszlop keresztpontjait vagy összekötik, vagy nem. A „**rövidzár**”, vagy **hiánya** adja a tárolt információ egy **bitjének az értékét**.

A 6.a.ábrán a **ROM** felépítését láthatjuk. A dekódoló által kiválasztott sornál (bájtnál) azokra az oszlopvezetéseken lesz **0** szint amelyeknél – a gyártás során felvitt - **összekötő**

vezeték *van*, a többi pedig 1 szintű lesz. Az így „*beprogramozott*” bájt jut a *belső adatbuszon* az illesztő egységre. Az *illesztő* egység *kialakítása* – hogy *invertáló*, vagy *nem invertáló* - határozza meg, milyen szintű lesz az engedélyező jel időtartama alatt az adatbusz egyes vezetékeire jutó jel szintje.

A 6.b.ábra a **PROM** felépítését mutatja. Gyárilag mindegyik sor oszlop keresztpontban van összekötő vezeték, vagyis *mindegyik* bit *azonos* értékű. A rövidzárok könnyen olvadó fémből készülnek. A felhasználás előtt egyszer programozható a memória azáltal, hogy *kiolvashatók* a szükséges összekötők. Ily módon a tárolandó információt a felhasználó viheti be a chip-be. *Hibás programozást már nem lehet módosítani!*

A 6.c.ábrán az újra programozható fix-memóriák - **EPROM**, illetve **EEPROM** - kialakítása látható. *Mindegyik* keresztpontban van egy **FET**, amelyeknél a *vezérlő elektróda* (Gate) egy „*sziget*”, amelyekbe bevihetők *elektromos töltések* (elektronok, vagy lyukak), illetve onnan kitörölhetők. A *töltés-bevitel* mindig elektromosan történí, amit „*beégetésnek*” is nevezünk. A szigeteken tárolt töltések *kitörléséhez* szükséges energiát vagy *UV* sugárzással (EPROM), vagy elektromos *áram* révén (EEPROM) visszük be. Azok a tranzisztorok *vezetnek*, amelyeknél a vezérlőelektróda töltött, a többi viszont *szakadás*. A működés további része teljesen megegyezik a ROM, és PROM működésénél leírtakkal.

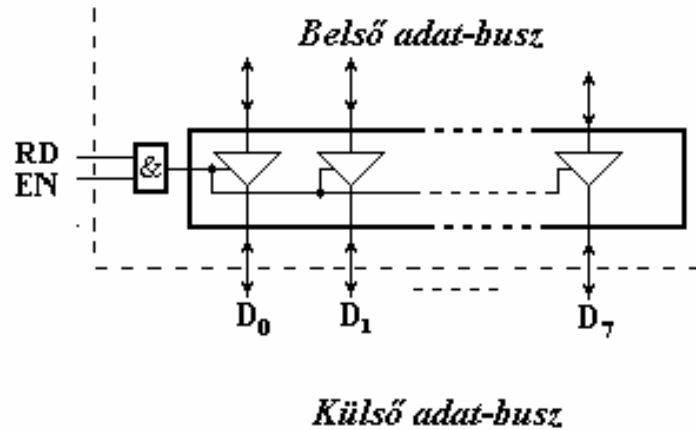


c.

6. ábra

⇒ **A BUSZ illesztő**

Mindegyik memória típusnál a kiválasztott információ **tri-state** kapukon keresztül kerül – **engedélyező jelek** hatására - az **ADAT BUSZ** egyes vezetékeire. Az illisztés egyik változatának elvi kapcsolási rajza látható a 7.ábrán.



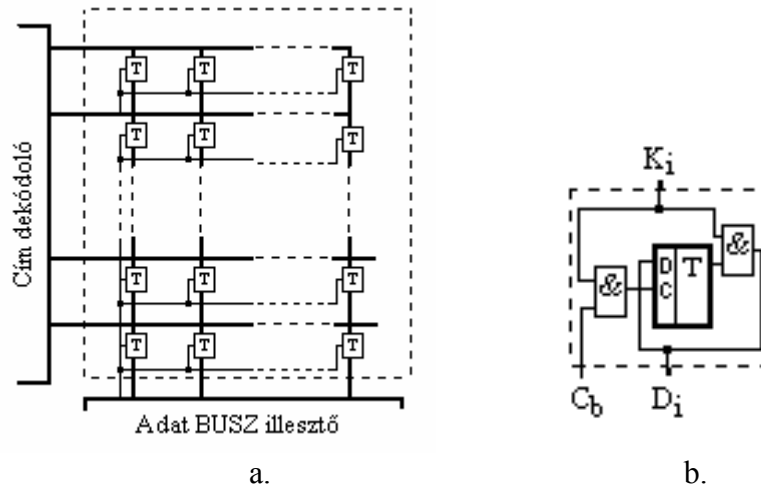
7. ábra

Az ábrán jelölt **tri-state** kapu **nem invertálják** a belső jelet. Létezik olyan megoldás is, amelynél invertálás után kerül a jel a külső adat buszra. Az adatátvitelt két jel együtt engedélyezi. Az EN-el jelölt jel az un. **chip enable CE** (tok engedélyező), vagy másképp **chip-select CS** (tok kiválsztó). A tok olvasásának engedélyezését a **címekből** állítja elő egy **dekódoló logika**. Az olvasó jelet **RD** (Read) a **mikroprocesszor** küldi a vezérlő buszra. A memória-tokoknál ezt az engedélyező bemenetet **OE** (Output Enable) jelöléssel adják meg. A legtöbb memória típusnál mind az olvasó **OE**, mind pedig a tok kiválasztó **CE** (**CS**) bemenetek **aktív szintje a 0**.

### 1.8.3. Írható, olvasható memóriák

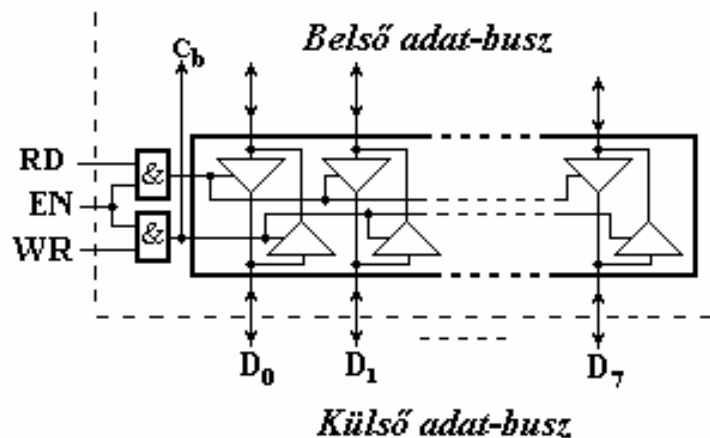
A számítógépen futó alkalmazói programból írható-olvasható memóriákat nevezik **RAM**-nak. A jelölés a **Random Access Memory** („véletlen” hozzáférésű memória) megnevezésből adódik. A véletlen hozzáférés arra utal, hogy a memória egységek teljesen egyformán érhetők el. Találkozhatunk a **RWM** (**Read Write Memory**) elnevezéssel is, amely az írható-olvasható tulajdonságot jelöli.

A 8.a. ábrán a **RAM** tároló celláinak elrendezése, a b. ábrán pedig a **tároló cellák** felépítése látható. Minden bitet egy-egy félvezető alapú **D flip-flop** tárol. Az egyes tárolók **írását**, illetve **olvasását** a dekódoló egység **Ki** jele készíti elő. A flip-flop beírását a **C<sub>b</sub>** billentő jel végzi. Mind az írás, mind pedig az olvasást a külső vezérlőjelek – a busz illesztőn keresztül – hajtják végre.



8. ábra

A 9. ábra szemlélteti egy **RAM** busz-illesztő egységének felépítését. A kiválasztott bájtt bitjeinek **írását** a  $C_b$  jel végzi, amelyet az **EN** engedélyező és a **WR** író jel állít elő, és egyúttal **engedélyezi** a külső adatbusz jeleinek **fogadását**.



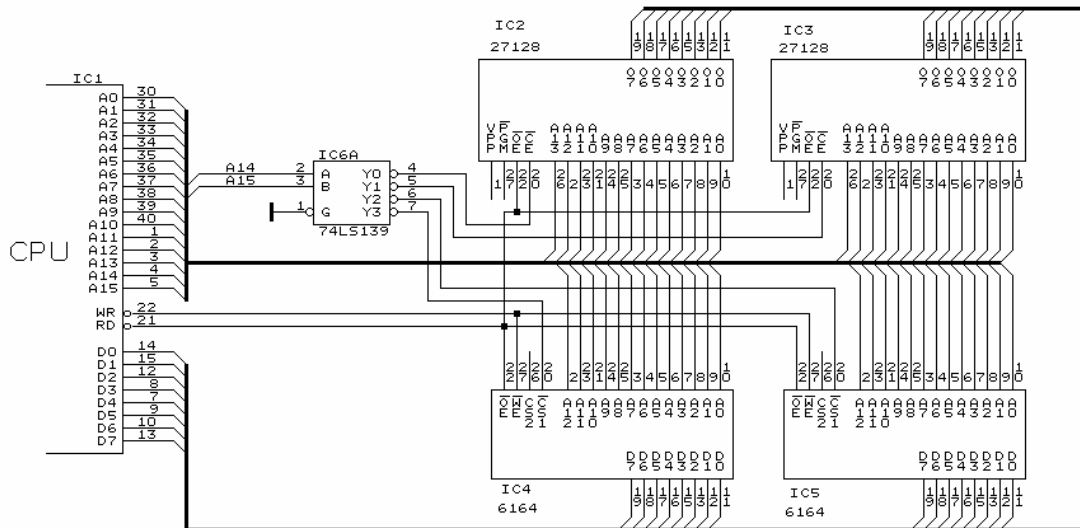
9. ábra

Az **adat olvasása** pedig az **EN** engedélyező, és a **RD** olvasó jelek hatására történik, amikor is a kiválasztott bájtt egyes bitjeinek értékét a **tri-state** kapuk a külső adat-buszra csatolják.

#### 1.8.4. Memóriák illesztése

A **mikrogépekben** különböző **típusú**, illetve **kapacitású** memória-tokok vannak. Ezek **elérése** a CPU felől az eltérő **címük** alapján történik. A 10. ábrán – egy példa segítségével – szemléltetjük a különböző memóriák címzését, illesztésüket a CPU-hoz.





10. ábra

Az ábra szerinti példán a CPU –hoz (IC1) *két* 27128 típusú **EPROM** (IC2, IC3), és *két* 6164 típusú **RAM** (IC4, IC5) illeszkedik. Az egyes memória-tokok *kapacitása címbitek száma* alapján állapítható meg.

Az **EPROM** kapacitása **16 kBájt**, mivel **14** címvezetékekkel ( $A_0 \dots A_{13}$ ) választható egy memória egység a tokon belül.

$$k = 2^{14} = 16384$$

A **RAM** –nak **13** címvezetéke ( $A_0 \dots A_{12}$ ) van, tehát a kapacitása **8 kBájt**.

Az egyes tokok közötti választást a *nem használt* címbitek kombinációi alapján lehet kialakítani. A példában az  $A_{15}$ , és az  $A_{14}$  bitek *négy kombinációját* állítja elő az **IC6** jelű *dekódoló* (SN74LS139). A kapcsolásban használt négy memória-tok *címzési tartományait* foglaltuk össze a következő táblázatban. A *tokon belüli* címzéshez használt címbitekhez írt x jelenti azt, hogy az **0**, vagy **1** értékű is lehet. A **RAM** – nál az  $A_{13}$  bit *nincs felhasználva* sem a tokon belüli, sem a tokok közötti választáshoz, ezért ezt **n** –el jelöltük. Mivel a bit mindkét lehetséges értékénél ugyanaz a tok érhető el, ezért *nem teljes a dekódolás*. Amikor ez nem okoz hibát, az ilyen dekódolás megengedett, mert kevesebb áramkört alkalmazhatunk.

tok	$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$	$A_{11}$	$A_{10}$	$A_9$	$A_8$	$A_7$	$A_6$	$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$	Cím
IC2	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0000H – 3FFFH
IC3	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	4000H – 7FFFH
IC4	1	0	n	x	x	x	x	x	x	x	x	x	x	x	x	x	8000H – 9FFFH A000H – BFFFH
IC5	1	1	n	x	x	x	x	x	x	x	x	x	x	x	x	x	C000H – DFFFH E000H – FFFFH

A teljes dekódolás **hiánya** miatt a RAM-ok minden bájtja **két címen** is hozzáférhető!!

### 1.9. Perifériák illesztése a mikrogephez

A mikrogepek **feladata**, hogy különböző **függetlenváltozók** – bemeneti jelek -értékeitől **függően**, kimeneti jelekkel – **függő változók** - működtessen **készülékeket**.

A bemeneti, és a kimeneti jelek egyaránt lehetnek **digitálisak** (kétértékű), illetve **analógok** (folytonos, kvantált stb.). A tantárgy keretében csak a digitális jelek illesztésével foglalkozunk.

Perifériákat illesztő áramkörök **alapvető** feladatai:

- a csatlakozó periféria adatvonalainak **összekapcsolása** az ADAT busszal,
- **szint-, és teljesítmény** illesztés,
- **potenciál-leválasztás** (esetleges).

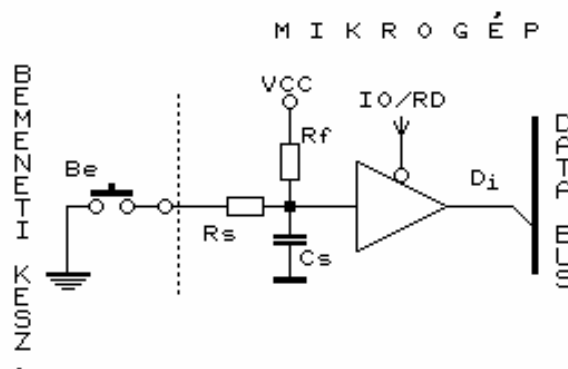
A következő részekben a legegyszerűbb, de alapvető illesztési megoldásokat tekintjük át.

#### 1.9.1. Bemenetek (input) illesztése

A mikrogephez érkező jeleket illesztő áramkörök leggyakoribb megoldása a **bájt** szervezés. Gondoskodni kell a bejövő jelek

- **potenciál** illesztéséről,
- **szűréséről**, és a
- megfelelő időpontban történő **mintavételezéséről**.

A 11. ábra szerinti áramköri kapcsolás egy adat mintavételezésére mutat példát.



11. ábra

Az ábrán a **tri-state** kaput az **IO/RD** jel nyitja, amely akkor 0 szintű, ha a **periféria** (I/O) kezelése **engedélyezett** és **olvasó** jel (RD) van. A bemeneti jelet egy nyomógomb adja, amely csak megnyomáskor szolgáltat zárt áramkört. **Elektronikus** áramkörökben

“szakadás” nem *engedhető* meg. Az *Rf* felhúzó ellenállás biztosítja, hogy a kapu bemenetén állandóan legyen feszültség szint. Amikor nincs lenyomva a *BE* gomb, akkor 1 szintű a bemenet, és ezt is olvassa be a BUSZ –ra a mintavételező jel. A gomb *lenyomott* állapotát 0 szint jelzi. Az *Rs* és *Cs* prell-mentesítő, és szűrő.

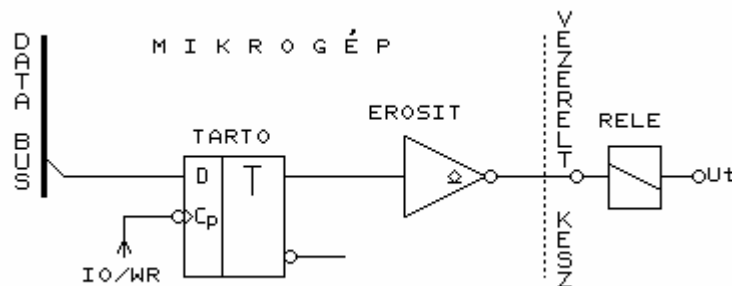
### 1.9.2. Kimenetek (output) illesztése

A mikrogéphez kimenetei a csatlakozó *készülékeket* (kijelzők, relék, stb.) *működtetik*. A vezérelt készülék határozza meg illesztés *feltételeit*, vagyis a

- *szint-átalakítás* jellegét, a
- *teljesítmény-átvitel* mértékét,
- *potenciál illesztés* kívánalmait.

Minden típusú kimeneti illesztő egységben van *tartó* áramkör, mivel a mikrogép a kimeneti jeleket *programciklusonként* csak *egyszer* frissíti. Két jelfrissítés között minden bit értékét tárolni kell!

A kimeneti jel illesztésének egy elvi kapcsolási vázlata látható a 12. ábrán .



12. ábra

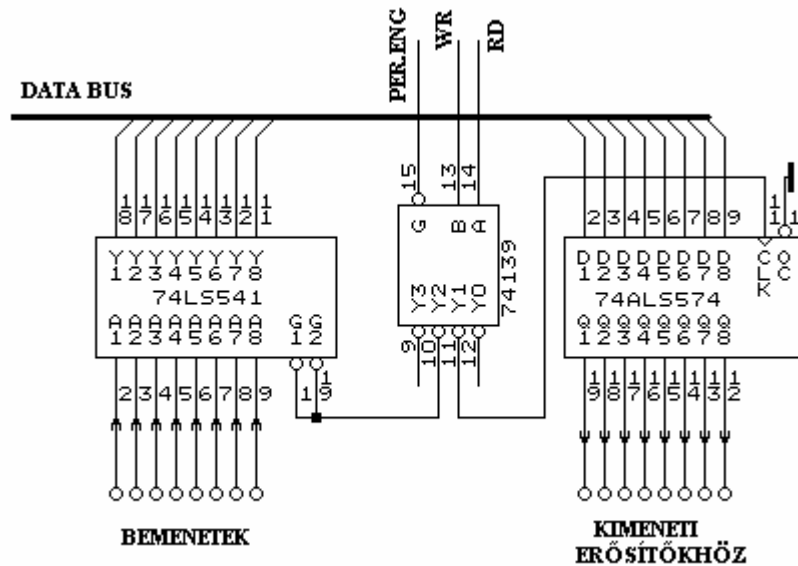
A *tartó* áramkör (D flip-flop) mindig azt az értéket tárolja, amely az *IO/WR* jel 1-0 átmenetekor érvényes adat. A tárolt értéket az *EROSIT* jelű 0 szintre aktív *nyitott-*(open)-*kollektorú* erősítő kapcsolja a kimenethez csatlakozó *RELÉ* tekercséhez. Mivel az erősítő *invertál*, ezért az *1* szintű tárolt értéknél húz *meg* a relé.

### 1.9.3. Perifériák címzése

A perifériákat illesztő áramkörök alapvetően kétféle megoldás szerint *kapcsolódnak* a *CPU környezetébe*. A megoldások:

- a *memóriába* ágyazott, illetve a
- *önálló* perifériaként.

A **mikroprocesszorok** – az **IO** utasításaikkal - külön is kezelhetik a **periféria-elemeket**, ugyanakkor a **RD/WR** vezérlőjelek is használhatók egy-egy diszkrét **regiszter**, vagy **tri-state** kapú vezérléséhez. Az uróbbi megoldást nevezzük **memóriába-ágyazásnak** (memory-mapped). A **mikrokontrollerek** többsége csak **memóriába-ágyazott** perifériakezelést tud vezérelni. A 13.ábrán egy 8 bites bemeneti tri-state kapu (SN74541), és ugyancsak 8 bites tart áramkör (SN74573) illesztése a DATA BUS-hoz..



13. ábra

A következő fejezetekben, illetve a mérési gyakorlatok során több áramköri változat megismerésére is mód nyílik.