

KECSKEMÉTI FŐISKOLA  
GÉPIPARI ÉS AUTOMATIZÁLÁSI MŰSZAKI FŐISKOLAI KAR

**Dr. Madarász László**

**Munkafüzet a memóriák és az  
I/O-egységek kezeléséhez**

Kecskemét  
2006

Szerzők:	Dr. Madarász László
Kiadás éve:	2006
Szakmai lektor:	Hóka László
Nyelvhelyességi lektor:	Várkonyiné Stumpf Anikó

Felelős kiadó:	Dr. Ailer Piroska rektor
A kiadást végző kar:	KF GAMF Kar
Példányszám:	100
Terjedelem:	14,5 A/5 ív
Azonossági szám:	KF-GAMFK-H-370

---

A nyomdai munkát végezte:	KF KIK Nyomda
Munkaszám:	2013.189

## Bevezetés

A digitális áramkörök alkalmazásának, a mikroelektronikai tervezésnek egy alapvetően fontos területe a memóriák és a bemeneti/kimeneti egységek ismerete, tervezése. Ezek a témakörök a tanterv változásai közben különböző tantárgynevek alatt jelentek meg, jelenleg a Digitális technika II. tantárgy tematikájában szerepelnek. Ezek a változások magyarázzák meg, miért nincs a jegyzet címében tantárgynév. A témakör azonban minden tantervváltozás ellenére alapvető fontosságú, olyan ismeretanyag, amit mindenképpen meg kell ismernie a mérnök-informatikai szakos hallgatóknak.

Ez az oktatási anyag nem elméleti jegyzet, de nem is példatár. A „munkafüzet” jelleg lényegét, tartalmát tekintve egy távoktatási segédeszköz, de a távoktatásban bevált formai megkötöttségek nélkül. Nem tartalmazza az egyes fejezetek elején az oktatási célt, nem időzik el azon, hogy motiválja Önt a fejezet anyagának elsajátítására, nem sorol fel ellenőrző kérdéseket. Ennek elsősorban a terjedelmi korlát az akadálya, a távoktatási formai jegyeknek megfelelő módon feldolgozva az itt rendelkezésre álló ismeretanyag kb. 4-5 ilyen méretű jegyzetet, füzetet töltene meg. Egyszer talán lesz rá mód, hogy ezen a teljesen kifejtett módon is elkészüljön ez a tananyag, de most nem ez volt a feladat.

A szerző véleménye szerint ez a tömörebb forma is teljes egészében alkalmas az egyéni tanulásra, egy dolog kell hozzá: az, hogy Önben legyen meg az elhatározás: ezt az ismeretrészt megértem, megtanulom. Ha ez hiányzik, tulajdonképpen teljesen mindegy, hogy milyen kialakítású tananyagot vesz a kezébe!

Ez a jegyzet nem előzmények nélküli. A digitális áramkörök alapismereteihez már készült munkafüzet, amelyik évtizedek óta nagyon hasznos segédeszköze a hallgatóságnak. Természetesen idő közben többször át kellett dolgozni, változtak a feladatok, egyes témakörök kimaradtak belőle, mások viszont megjelentek benne. Önök korábban találkoztak azzal a munkafüzetrel, hiszen a logikai függvények egyszerűsítésével, a különféle sorrendi hálózatok tervezésével már megismerkedtek. Elsősorban a levelező tagozatos hallgatók tartják hasznosnak a munkafüzetet, de (ha nem is mindig vallják be) nagyon sokat segít a nappali tagozaton tanulóknak is.

A munkafüzet felépítése tehát már úgyszólván hagyományos. Egy témakörön belül először az ismereteket foglaljuk össze, majd az adott problémakör egy-két jól megválasztott feladatát lehet tanulmányozni, részletes kidolgozással, sok magyarázattal. Ezután olyan feladat következik, ahol a magyarázatok még szerepelnek, de az egyes megoldási lépéseket már Önnek kell megtennie, a megfelelő eredményeket a munkafüzet „üres” részeire be kell írnia, be kell rajzolnia. Ezeknek a feladatoknak a 10. fejezetben meg lehet találni a megoldását. Végül egyes fejezetek végén olyan feladatok is találhatók, amelyeket teljesen önállóan kell megoldania, esetenként külön papíron be is kell adnia a gyakorlatvezető részére.

A munkafüzet azért készült, mert ennek a témakörnek a tanulásakor fokozottan igaz az, amit a digitális technika, a mikroelektronika, a mikrovezérlők tanulása során folyamatosan hangoztatunk: nem elég „megtanulni” a tantárgyat, a módszereket, a lépéseket *meg kell érteni*! Ha megértette, akkor alkalmazni tudja hasonló és többé-kevésbé eltérő problémákra is, mint amilyeneket a mintapéldákban talál. És ez a cél, hogy önállóan meg tudjon oldani újszerű feladatokat is.

Ha ezt az igényünket egyszerűen csak közöltük a hallgatókkal, nem volt sok eredménye. A munkafüzet szinte észrevétlenül juttatja el Önt arra a szintre, amikor már érti a tervezési lépéseket. Természetesen lehet ügyeskedni, de az ügyeskedéssel nem a gyakorlatvezetőt csapják be, hanem önmagukat.

Az említett ügyeskedés egyik módja, hogy egy kitöltött munkafüzetből másolja be a hallgató a sajátjába a megoldásokat (a leglustábbak még a másolást is mással végeztetik el). Ezzel természetesen semmit sem tanul. Egy fokkal jobb hatásfokú az a „szemfülesség”, ha a 10. fejezetből előre beírja, bemásolja az eredményeket a kidolgozásra váró feladatokhoz. Ezzel a módszerrel sem jut el oda, hogy megértene az anyagrészeket.

A javasolt tanulási módszer első lépése: az anyagrész előadásának meghallgatása. Az élő előadás sok olyan részletet megvilágít, ami az írott anyagokból kevésbé olvasható ki. Az előadás után az elméleti jegyzet megfelelő fejezetét kell áttanulmányozni, s így lehet beülni a gyakorlati foglalkozásra. Ott a gyakorlatvezető a hallgatókkal közösen a témakörre vonatkozó feladatokat old meg, a fő lépéseket, fogásokat ott már el lehet lesni. Ezután következik a munkafüzet.

Valószínű, hogy a teljesen kidolgozott feladat a felsorolt „alapozás” után már világos és érthető lesz. Ekkor jön az a feladat, amihez még némi szöveges útmutatót ad a jegyzet. Az útmutatások alapján haladjon előre, az egyes lépéseket oldja meg, írja be és rajzolja be az eredményeket. Ha befejezte a feladatot, akkor ellenőrizze az eredményeit a 10. fejezetből!

A memóriák és I/O-egységek kezelésekor egy-egy feladatot általában több különböző módon is meg lehet oldani. Ezért előfordul, hogy az Ön megoldása helyes, mégis eltér a 10. fejezetben szereplő eredménytől. Ha Ön valóban megértette az anyagrészt, örömmel fogja felfedezni, hogy más módon is el lehet érni ugyanazt a hatást, működést, mint amit az Ön áramköre megvalósít! Az ilyen felfedezések jelzik, ha elértük a célunkat, Ön megértette a feldolgozott anyagrészt.

Munkája során ne feledje, hogy ha kevésnek bizonyulna a munkafüzet, bizonytalanságok maradnának Önben, nem tudná, hogy az Ön megoldása és a „hivatalos” eredmény különbségét mi okozza, akkor nyugodtan fordulhat a gyakorlatvezetőjéhez, a tantárgy előadójához segítségért. Keresse fel őket a konzultációs órákon és beszéljék meg a nehézségeket, a problémákat! De nagyon hasznos az is, ha a hallgatótársával beszél meg ezeket a kérdéseket.

Remélem, hogy ez a jegyzet a memóriák és az I/O-egységek kezelésének megértésében, begyakorlásában legalább olyan hasznosnak bizonyul, mint a digitális áramkörökre vonatkozó korábbi munkafüzet. Ebben természetesen nagy szerepük lesz Önöknek is. Kívánom, hogy közös munkánk eredményeképpen felfedezzék, hogy nem egy megtanulhatatlan tantárggyal állnak szemben, hanem egy izgalmas, logikai feladványokkal teli, érdekes, sok élményt nyújtó szakmai területtel.

Kecskemét, 2006. szeptember

A Szerző

## 1. A memória-áramkörök és az I/O-egységek vezérlése

A digitális áramkörök tervezésekor a felhasznált elemek logikai rajzjeleit használjuk az ábráinkon. A logikai rajzzel segítségével meg tudjuk mutatni, hogy az áramkör hogyan kapcsolódik a környezetéhez, de (különösen a memória-áramkörök esetében) igen sokat elárul a rajzzel az áramkörök adatairól, paramétereiről is. A memória-áramkörök legfontosabb adata a kapacitása, a másik alapvető információ velük kapcsolatban a típus. A típus és a kapacitás ismeretében már meg lehet rajzolni a logikai rajzjelet.

Ha feldolgozta ezt a fejezetet, képes lesz a típus és a kapacitás ismeretében a logikai rajzzel megrajzolására, illetve a rajzzel ismeretében a memória típusának, kapacitásának megadására. Mindezek mellett ismerni fogja a működési módokat, és az üzemmódok vezérlési lehetőségeit is.

A memória-áramkörök kapacitásának két adata van, az egyik a szószám, amit a címbitek számából lehet meghatározni, a másik a szóhosszúság, amit az adatpontok száma alapján lehet megadni (illetve természetesen fordítva is működik a dolog, a szószámból megadható a címbitek száma, a szóhosszúságból pedig az adatpontoké). A kapacitást többnyire egy szorzat formájában adjuk meg, ahol a szószámot szorozzuk a szóhosszúsággal, pl.

1024 x 4 azaz 1 Ki x 4.

Mivel a szószám többnyire nagy érték, gyakran használjuk itt a bináris prefixeket (Ki, Mi). A vezérlőjeleket a memória jellege alapján lehet megadni.

Az adatpontok ( $Q_i$  vagy  $D_i$ ) és a szóhosszúság kapcsolata egyértelmű: az adatpontok száma megegyezik a szóhosszúsággal.

A címbitek számából azért adódik a szószám, mert a memória-áramkörökben mindig teljes a címdekódolás, azaz  $n$  címbemenet esetén az összes lehetséges, azaz  $2^n$  címértéket előállítják és egy-egy rekesz címzésére használják.

Ha a logikai rajzjelen hat címbit látható, azok a következők (többnyire nem írjuk ki minden bit nevét, csak az elsőét és az utolsóét, ugyanígy járunk el az adatbiteknel is):

A0 ... A5.

A hat címbitnek megfelelően a memóriában 64 rekesz van, azaz a szószám 64. A legkisebb című rekesz címe binárisan 000000b, a legmagasabb címűé pedig 111111b. Hexadecimálisan a kezdőcím 00h, a záró pedig 3Fh. A címértékeket szinte mindig hexadecimális számokkal adjuk meg, ezért általában nem is jelöljük a számrendszert, tehát a címhatárok: 00 ... 3F. Mindezt abból az egyetlen adatból állapítottuk meg, hogy a címbitek száma 6.

12 címbit esetében a címbitek megnevezése így alakul: A0 ... A11, a szavak száma 4096 azaz 4 Ki, a címhatárok: 000 ... FFF.

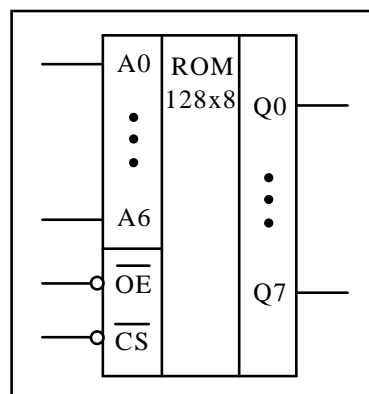
A bináris prefixek 2 hatványait helyettesítik, a leggyakoribbak:

1 Ki =  $2^{10}$  = 1024 ,

1 Mi =  $2^{20}$  = 1048576 .

A vezérlőjelekkel kezelt *csak olvasható memóriákat*, a ROM-, PROM-, EPROM-, EEPROM-áramköröket ebben a jegyzetben többnyire egyszerűen ROM-áramkörként szerepeltetjük, hiszen mindegyiknek azonosak a felhasználási tudnivalói. Az *írható/olvasható RAM*-családból pedig a SRAM-áramköröket fogjuk példánkban felhasználni.

Az **1.1. ábrán** látható egy ROM-áramkör logikai rajza. Minden esetben szerepelnie kell a logikai rajzon a funkciójelnek (ez ebben az esetben a ROM felirat), a bemenőjelek és a kimenőjelek megnevezésének. Az 1.1. ábrán többletet jelent a kapacitás megjelölése. Ha nem írjuk fel, akkor is ismert a kapacitás, hiszen a címbitekből és az adatbitekből meghatározható, viszont a kapcsolási rajzok olvasását megkönnyíti, leegyszerűsíti, ha feltüntetjük.



1.1. ábra

Ennél a memóriánál a címbitek ( $A_0 \dots A_6$ ) száma 7, így a szavak száma  $2^7 = 128$ . Az adatpontok (amik ROM esetén a  $Q_i$  kimenetek) száma nyolc ( $Q_0 \dots Q_7$ ), így adódik a kapacitásra a  $128 \times 8$  kifejezés. A címhatárok hexadecimálisan  $00h \dots 7Fh$ , a bináris  $0000000b \dots 1111111b$  értékeknek megfelelően.

A ROM-áramköröknél két vezérlőjelet használunk, az egyik a kimenet aktivitását engedélyezi, a másik a teljes áramkör működését. A ROM tartalmának kiolvasásához mindkét jelnek aktívnak kell lennie. Mint legtöbbször, ezek a vezérlőjelek is alacsony szinten aktívak. Az 1.1. ábrán a kimenet engedélyezőjele az  $\overline{OE}$ , a teljes áramköré a  $\overline{CS}$ . Ha egyikük vagy mindkettő H szintű, a kimeneti fokozat inaktív, azaz a kimeneti pontok ( $Q_0 \dots Q_7$ ) lebegnek, nagyimpedanciás állapotban vannak. Ha mindkettő L szintű, az éppen megcímezett rekesz tartalma jelenik meg a kimeneti pontokon.

A kimenet engedélyezőjele ( $\overline{OE}$ ) a memória beépítésekor a ROM üzemmódját vezérli, azaz engedélyezi vagy tiltja a kiolvasást, ezért ezt a jelet a processzorok olvasási jeléről működtetjük (pl.  $\overline{RD}$  vagy  $\overline{MEMR}$ ). A  $\overline{CS}$  jelet arra használjuk, hogy a processzor teljes címezési tartományán belül az áramkör címtartományát kijelöljük. Általában egy címdekóder állítja elő a megfelelő címtartományban ezt a vezérlőjelet, ezért ezt a pontot a címdekóderre, esetleg közvetlenül valamelyik címvezetékre kötjük majd rá.

### 1.1. Feladat

Az 1.2. ábrán egy ROM-áramkör logikai rajzjele látható. Állapítsuk meg a kapacitását!

A címbitek száma:

A címbitek száma alapján a rekeszek (szavak) száma:

Bináris prefix segítségével rövidebben is leírható ez az érték:

Az adatpontokból látható a szóhosszúság:

A memória kapacitása tehát:

A 10. fejezetben ellenőrizheti az eredményeit!

### 1.2. Feladat

Ez egy fordított Feladat Most megadjuk egy ROM-áramkör kapacitását, s abból kell meghatározni a különféle paramétereket, majd fel kell rajzolni a logikai rajzjelet.

A ROM kapacitása  $16 \text{ Ki} \times 8$ .

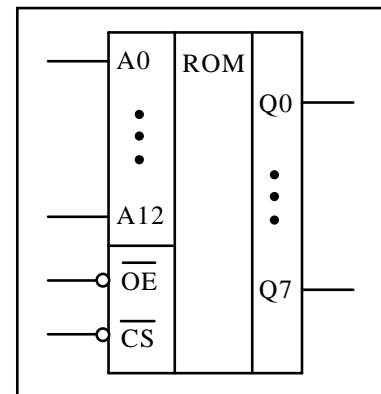
Határozza meg a címbitek számát, megnevezését! Segítünk: mivel a 16 decimális szám  $2^4$ , a Ki pedig  $2^{10}$ , a címbitek száma:

**Fontos megjegyzés!** Ha a számrendszerek használatával és átszámításával problémái vannak, már most alaposan ismételje át ezt a témakört, mert folyamatosan használnunk kell ezeket az ismereteket! Fejben és gyorsan kell átváltani a  $0 \dots 15$  tartományban a számokat decimális, bináris és hexadecimális számrendszerek között és biztosan kell kezelni a Ki, Mi értékeket is!

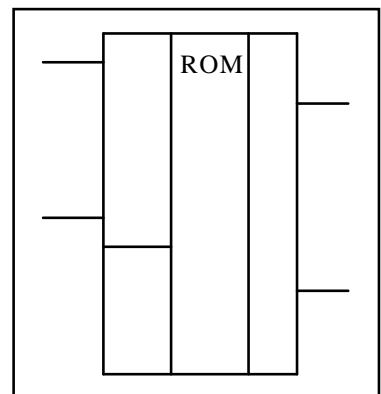
A címbitek megnevezése:

A szóhosszúság, így a kimeneti pontok száma:

A kimeneti pontok megnevezései:



1.2. ábra

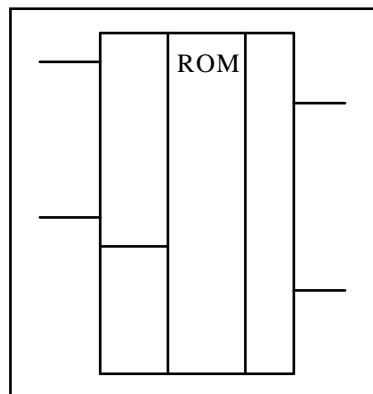


1.3. ábra

Az **1.3. ábrán** megrajzoltuk a ROM-áramkörök logikai rajzjelének fő, állandó elemeit. Egy ilyen ábrába kell majd a megfelelő értékeket beírnia.

A felső bal oldali mezőbe kerülnek a címbitek, az alsó bal oldaliba a vezérlőjelek, a jobb oldaliba a kimeneti pontok megnevezései.

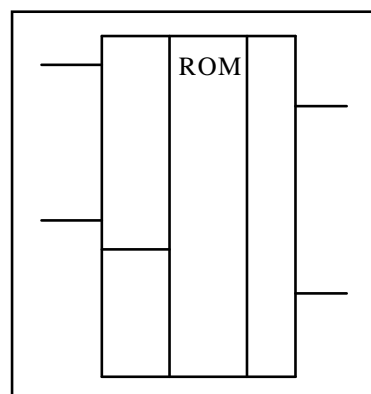
Ezt az első logikai rajzjelet lépésenként alakítjuk ki. A már megismert általános alakzatba az **1.4. ábrán** a vezérlőjeleket írja be! A vezérlőjelek függetlenek a kapacitástól, azaz az 1.4. ábrán látható részleteket minden esetben ugyanígy rajzoljuk, bármekkora is a ROM-áramkör! Ezért ezt az ábrát így is hagyjuk, s a teljes logikai rajzjelet az 1.5. ábrán alakítjuk ki!



1.4. ábra

Az **1.5. ábrába** is írja be a vezérlőjeleket, majd a megfelelő mezőbe a címbemenetek megnevezéseit, végül a kimenetek neveit is! Ahogy szoktuk, a címeknél is és a kimeneteknél is az első és az utolsó megnevezést írjuk ki, közöttük három pont jelzi a többi értéket.

Végül minden jelhez egy-egy rövid csatlakozóvezeték is rajzolunk, a megfelelő oldalra (ezek egy része már szerepel is az ábrán!). Az alacsony aktív szintű vezérlőjeleknél ne maradjanak el az erre utaló kis körök sem!



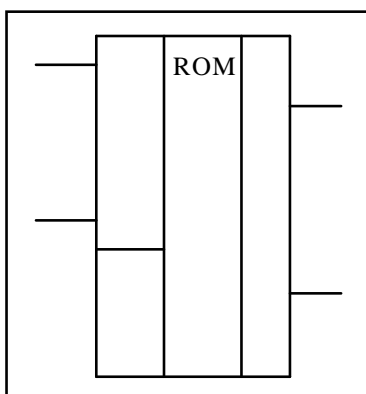
1.5. ábra

A végeredményt megint ellenőrizheti a 10. fejezetben!

### 1.3. Feladat

Most segítség nélkül rajzolja meg egy 4 Ki x 8 kapacitású ROM logikai rajzjelét!

A megoldási lépések természetesen ugyanazok legyenek, mint az előző esetben! A rajzjel alapját megadjuk (**1.6. ábra**), határozza meg a vezérlőjeleket, a címbemenetek számát, megnevezését, a kimenetek számát, megnevezését, és rajzolja meg az áramkört!



1.6. ábra

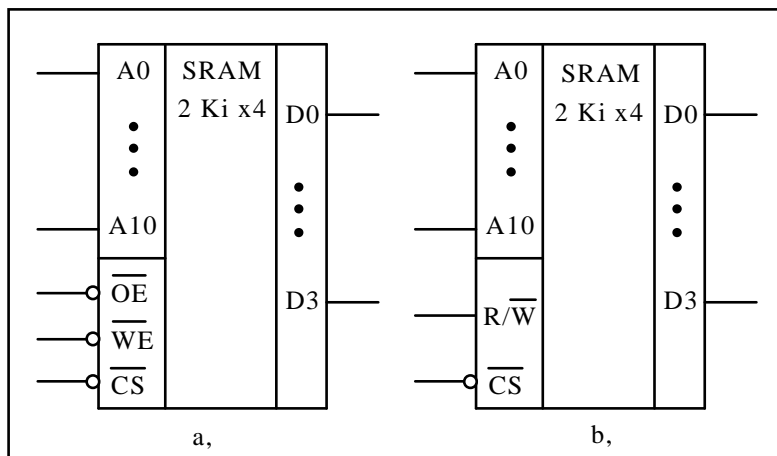
Az eredményt ismét ellenőrizheti a 10. fejezetben.

A logikai rajzjelek tanulmányozásakor esetenként eltérő jelölésekkel is találkozhatunk.

A teljes IC működését engedélyező jel megnevezésére a  $\overline{CS}$  jelölést használjuk majd (Chip Select), de szokásos még a  $\overline{CE}$  (Chip Enable), a  $\overline{DS}$  (Device Select),  $\overline{DE}$  (Device Enable) jelölés is. A felsorolt jelek mind alacsony (L) szinten aktívak, felülhúzás nélkül az aktív szintjük a magas (H) érték. Ha egy memória-áramkörnek több ilyen jele is van, azokat sorszámokkal szokták ellátni. Ilyen esetben gyakori, hogy eltérőek az aktív szintek (pl.  $\overline{DS1}$  és  $\overline{DS2}$ ).

A kimenet működését engedélyező jel az  $\overline{OE}$  (Output Enable). Ha ez a jel alacsony (L) szintű, a kimeneti fokozat aktivizálódik (ha ugyanakkor a  $\overline{CS}$  is L szintű); a H szint a kimeneti fokozat inaktív állapotát jelenti. Ugyanezekkel a jelszintekkel működik az esetenként használt OD jelű vezérlőjel is (Output Disable).

Az írható/olvasható SRAM-memóriák más vezérlőjeleket igényelnek. Az **1.7. ábrán** látható egy 2 Ki x 4 kapacitású SRAM-áramkör. A ROM-elemek vezérlőjele egységes, a SRAM-áramköröket kialakíthatják két vezérlőjelesre is és három vezérlőjelesre is. A kapacitás és a címpontok, adatpontok száma ugyanúgy kapcsolódik egymáshoz, mint a ROM esetében.



1.7. ábra

A címpontok (A0 ... A10) száma 11, ezért az áramkör szószáma 2048, azaz 2 Ki. Az adatpontok, amik a SRAM esetén a  $D_i$  kétirányú pontok ( $D0 \dots D3$ ), négyen vannak, tehát a szóhossz 4 bit. A kapacitás ezért 2 Ki x 4.

A vezérlőjelek egyike azonos a két változatnál, ez a már jól ismert  $\overline{CS}$ , ami a teljes IC működését engedélyezi (L szint) vagy tiltja (H szint). Erre a pontra a SRAM esetén is egy címdekóder, esetleg egy címvezeték kapcsolódik.

A SRAM-áramkör aktív állapotában két üzemmód lehetséges, az olvasás és az írás. A további vezérlőjelek a két üzemmód egyikét jelölik ki. A három vezérlőjeles esetben (a, ábrarészlet) külön jel aktivizálja az olvasást ( $\overline{OE}$ , esetenként  $\overline{RD}$ , azaz Read), és külön jel az írást ( $\overline{WE}$ , Write Enable, vagy  $\overline{WR}$ , azaz Write). A D pontok kimeneti aktivizálása csak úgy valósul meg, ha a  $\overline{CS}$  is és az  $\overline{OE}$  is L szintű. A processzorok olvasást elrendelő vezérlőjele továbbra is az  $\overline{OE}$  bemenetre kapcsolódik (pl.  $\overline{RD}$  vagy  $\overline{MEMR}$ ), a  $\overline{WE}$  a processzortól az írást kiváltó vezérlőjeleket kapja (pl.  $\overline{WR}$  vagy  $\overline{MEMW}$ ).

A b, ábrarészleten látható két vezérlőjeles kivitel esetében egy összevont vezérlőjel választja ki az üzemmódot. Az  $R/\overline{W}$  jel H szintje rendeli el az olvasási működést, ez aktivizálja a kimeneti pontokat, az L szint az írási üzemmódot váltja ki. Ha a két vezérlőjeles SRAM-nál a  $\overline{CS}$  aktív (tehát L) szintű, a kimenet azonnal aktív, amint az  $R/\overline{W}$  H szintű. A processzorok vezérlőjelei közül erre a bemenetre az írást kiváltó jelet szokás rákapcsolni (pl.  $\overline{WR}$  vagy  $\overline{MEMW}$ ), ha tehát a címzés aktivizálja a SRAM-áramkört, a processzor pedig nem rendeli el az írási működést, az IC azonnal olvasódik! Ez egy kellemetlen jelenség, ugyanis az a tény, hogy a processzor pillanatnyilag nem kívánja írni a SRAM-ot, nem jelenti egyértelműen azt, hogy olvasni akarja (lehet, hogy semmilyen működést sem igényel, esetleg I/O elemekkel kíván kommunikálni). A következmény adatütközés, esetleg áramköri károsodás lehet. Ezt a sajátos működést a tervezéskor feltétlenül figyelembe kell vennünk; ha el kívánjuk kerülni, használjunk három vezérlőjeles SRAM-áramköröket!



**1.4. Feladat**

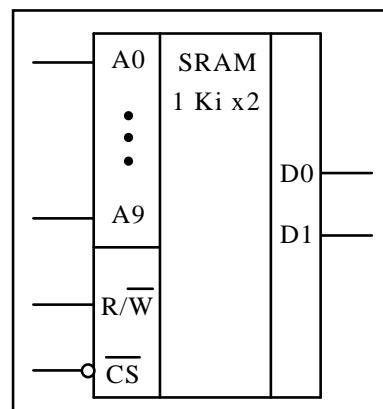
Rajzoljuk meg a két vezérlőjeles SRAM-áramkör logikai rajzjelét, ha a kapacitása  $1 \text{ Ki} \times 2$ . (A ROM-memóriák általában legalább „bájt szélességűek”, azaz 8 bitesek, vagy a 8 bit többszöröse a szóhosszuk. A RAM változatoknál a szóhosszúság gyakran csak 1 bit, de lehet 2 vagy 4 bit is. Természetesen 8 bites, illetve többször 8 bites áramköröket is gyártanak.)

Az  $1 \text{ Ki} = 1024$  szószámból következően a memória címzése 10 bites, a 10 címvezeték:  $A0 \dots A9$ .

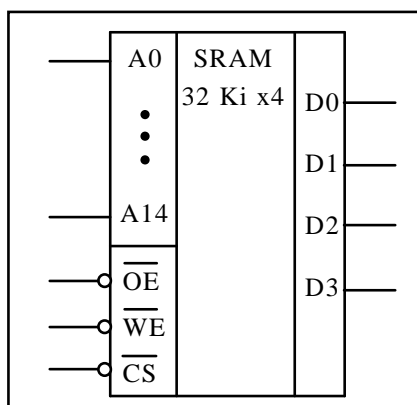
Két adatpont van mindössze, a  $D0$  és a  $D1$ .

Adjuk meg a címhatárokat is:  $000 \dots 3FF$  (a 10 címbit miatt).

A két vezérlőjel a szokásos, így a teljes logikai rajzjel az **1.8. ábra** szerint alakul.



1.8. ábra



1.9. ábra

**1.5. Feladat**

Rajzoljuk meg a három vezérlőjeles  $32 \text{ Ki} \times 4$  kapacitású SRAM logikai rajzjelét!

A  $32 \text{ Ki}$  szószámnak megfelelően 15 darab címvezetékünk lesz ( $2^{15} = 2^5 \times 2^{10} = 32 \text{ Ki}$ ), azaz  $A0 \dots A14$ , a címhatárok:  $0000 \dots 7FFF$ .

A szóhosszúság 4 bit, tehát négy adatpontot kell rajzolnunk ( $D0 \dots D3$ ).

A kívánt felépítés: három vezérlőjeles, tehát a  $\overline{CS}$ , az  $\overline{OE}$  és a  $\overline{WE}$  szerepel.

A teljes logikai rajzjelet az **1.9. ábra** mutatja be.

Következhetnek az Ön által megoldandó feladatok!

**1.6. Feladat**

Rajzolja meg a három vezérlőjeles,  $2 \text{ Ki} \times 2$  kapacitású SRAM-áramkör logikai rajzjelét! A rajzjel alapját az **1.10. ábrán** találja meg, ezt egészítse ki a feladatnak megfelelően!

A szószám  $2 \text{ Ki}$ , ebből következően a címbitok száma:

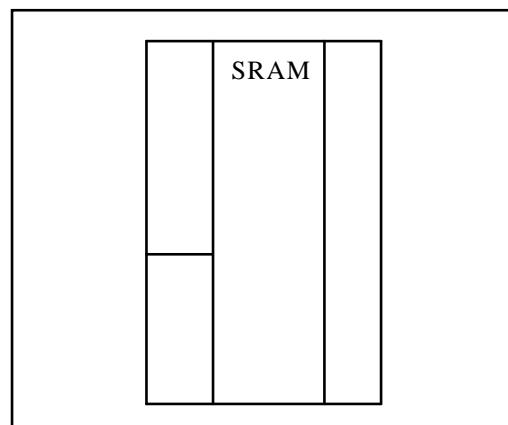
Adja meg a címhatárokat:

A címbitok nevei:

Az adatpontok nevei:

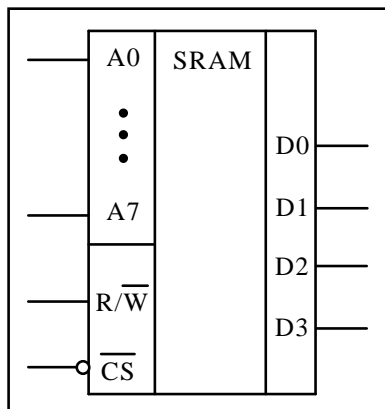
A vezérlőjelek:

Írja be a kapott információkat a rajzjelbe, majd ellenőrizze a 10. fejezetben a megoldását!



1.10. ábra

A címhatárok felírásával kapcsolatban még egy megjegyzést kell tennünk. Mint eddig is láthatta, a kezdőcímet mindig annyi számjeggyel (0-val) adtuk meg, ahány számjeggyű a záró cím volt, akár binárisan, akár hexadecimálisan írtuk fel a címeket. Ezt mindig így szoktuk felírni! Ezért először a záró címet célszerű meghatározni, abból látjuk, hogy a kezdőcímet hány 0-val kell leírunk!



1.11. ábra

### 1.7. Feladat

Az **1.11. ábrán** egy SRAM logikai rajzjele látható. Állapítsa meg a kapacitását! (A gondolatmenete legyen a korábban megismert, minden részletet írjon le!)

Adja meg a címhatárokat is:

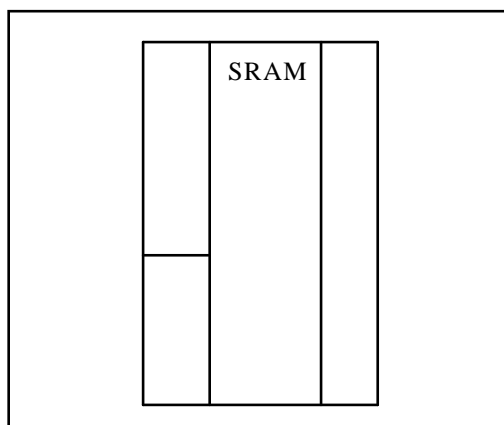
Az eredményét ellenőrizheti a 10. fejezetben.

### 1.8. Feladat

Rajzolja meg a három vezérlőjeles, 8 Ki x 8 kapacitású, három vezérlőjeles SRAM logikai rajzjelét, a **10.12. ábrán** látható alap segítségével!

Itt már nincs segítség, önállóan kell dolgoznia.

Az eredményt azért ellenőrizheti a 10. fejezetben. De valóban csak az eredményt ellenőrizze! Sokkal többet ér, ha első próbálkozásra esetleg hibásan alakítja ki a rajzjelet, azután rájön a hibára és kijavítja, mintha a megoldást bemásolja erre az oldalra...



1.12. ábra

Ráadásként írja ide a címhatárokat:

## 2. Memória bővítése szószámra, szóhosszra

A mikroszámítógépek tervezése során jelentkező igények általában nem elégíthetők ki egyetlen memória IC-vel. Lehetséges, hogy 8 Ki szószámra lenne szükség, a rendelkezésre álló memória viszont csak 2 Ki rekeszes, de ebből több is van. Ilyen esetben a memóriát szószámra bővítjük, nagyobb szószámú memóriaegységet alakítunk ki belőlük.

Más esetben a szószám megfelelő, de a memória IC rövidebb szóhosszúságú a szükségesnél. Pl. 8 Ki x 8 kapacításra van szükség, viszont csak ilyen memória-elemünk van: 8 Ki x 2. Ilyen esetben a szóhossz szerinti bővítés segít, ha van megfelelő számú áramkörünk.

Olyan helyzet is adódhat, hogy egyszerre kell szóhossz és szószám tekintetében is bővítenünk a memóriánkat.

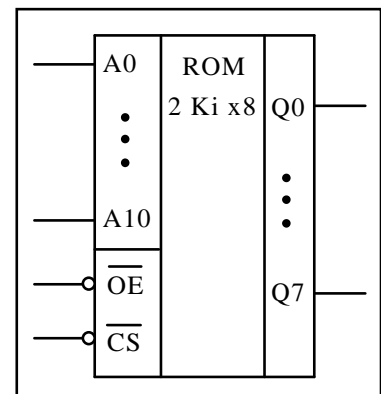
A szóhossz és/vagy szószám szerint bővített memóriát, memóriaegységnek több azonos memóriáramkör alkotja. A teljes memóriaegységnek a csatlakozópontjai ugyanolyanok, mint egy megfelelő kapacitású memória-IC-nek. Ha ROM-ból készítünk bővített egységet, annak is ROM-jellegűek a csatlakozópontjai, ha SRAM-ból, a memóriaegység is SRAM-szerű csatlakozópontokkal rendelkezik.

Ebben a fejezetben csak azonos típusú, kapacitású memóriák összekapcsolása szerepel. A következő fejezetben különféle kapacitású, különböző jellegű memóriák alkotnak majd egy-egy új egységet, amit memóriakártyának nevezünk majd.

### 2.1. Feladat

8 Ki x 8 kapacitású ROM memóriaegységre van szükségünk, a rendelkezésünkre álló áramkörök 2 Ki x 8 jellegűek. A szóhossz megfelel, a szószámot kell bővítenünk. A szükséges 8 Ki szószámot négy darab 2 Ki szószámú memória állítja elő, tehát négy IC-t fogunk felhasználni.

A 2 Ki x 8 kapacitású ROM-áramkör logikai rajzjelét a **2.1. ábrán** találjuk meg. Az előzőek alapján Ön is fel tudta volna rajzolni, 11 cím-bit (A0 ... A10) és nyolc adatkimenet (Q0 ... Q7) szerepel rajta, valamint a ROM IC-k szokásos vezérlőjelei. A címhatárok: 000 ... 7FF.



2.1. ábra

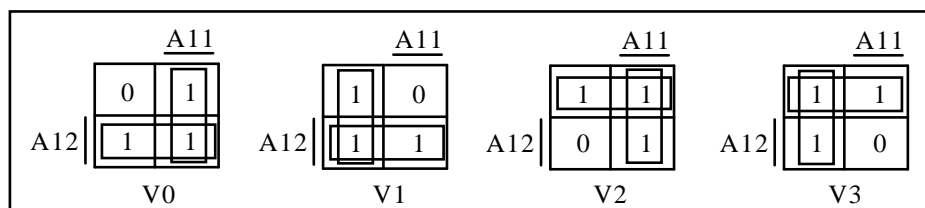
A12	A11	V3	V2	V1	V0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

2.2. ábra

A 8 Ki x 8 kapacitású egységnél a címbitek száma 13, az elnevezésük: A0 ... A12. A címhatárok: 0000 ... 1FFF. Az új címbitek, az A11 és az A12 szerepe az lesz, hogy a 2 Ki x 8 méretű memóriák közül egyet-egyet kijelöljenek. Az egyes memóriák kijelölőjelét V0, V1, V2 és V3 elnevezéssel láthatjuk el, a **2.2. ábra** szerinti minterm tábla mutatja meg az új címbitek és a kijelölőjelek kapcsolatát.

A 2.3. ábrán az egyes kijelölőjelek minterm tábláit különválasztva rajzoltuk meg, onnan kiolvashatók a függvények:

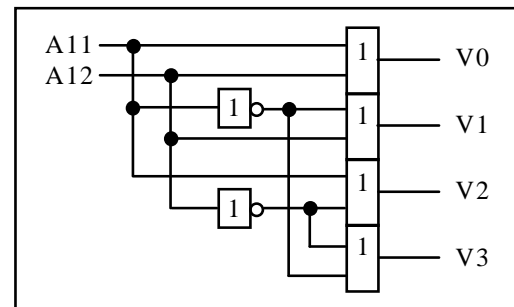
$$V0 = A11 + A12, V1 = \overline{A11} + A12, V2 = A11 + \overline{A12}, V3 = \overline{A11} + \overline{A12}.$$



2.3. ábra

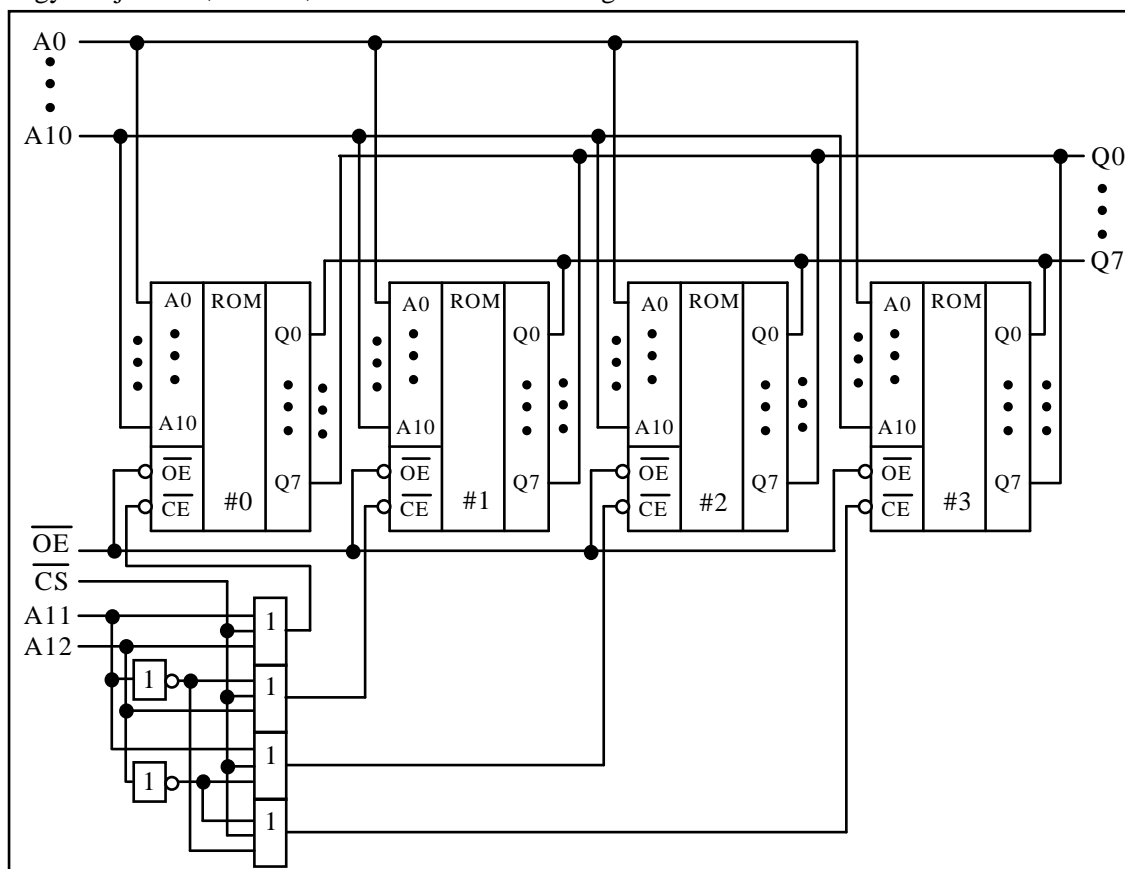
A **2.4. ábrán** látható a kijelölőjeleket előállító kapcsolás, azaz a címdekóder.

A kijelölőjelek aktív szintje L (a minterm táblában 0), mivel a memória-áramköröknél a  $\overline{CS}$  bemenet is alacsony aktív szintű. Ha két címbit (A11, A12) 0 értékéből állítunk elő alacsony aktív szintű kijelölőjelet, a két címvezeték egy VAGY kapura kell vezetni. A többi esetben, ahol egy vagy két címbit 1 értékkel szerepel, a VAGY kapura az ilyen biteket inverterekén át vezetjük rá. Ezt a tapasztalatunkat érdemes megjegyezni, a későbbiekben sokkal gyorsabban tudunk majd kijelölőjeleket előállítani a felhasználásával!



2.4. ábra

A teljes kapcsolást a **2.5. ábra** mutatja be. Látható, hogy az egyes memória-áramkörök saját címbitjei (A0 ... A10) egyszerűen párhuzamosan rá vannak kötve az azonos elnevezésű bemeneti címpontra. A két új címbitet a címdekóder fogadja, ami a négy memória IC egyikét jelöli ki, az A11, A12 kombinációnak megfelelően.



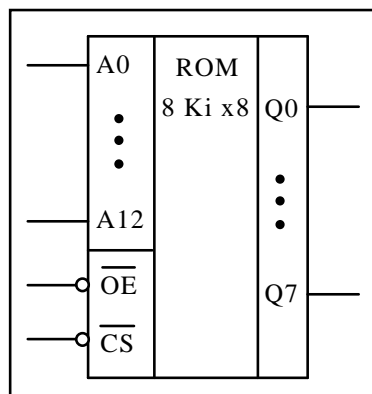
2.5. ábra

Már felírtuk a bővített memóriaegység címtartományát: 0000 ... 1FFF. Az egyes memória-áramkörök ezen belül sorban a következő címtartományokban működnek:

- #0 ROM: 0000 ... 07FF ,
- #1 ROM: 0800 ... 0FFF ,
- #2 ROM: 1000 ... 17FF ,
- #3 ROM: 1800 ... 1FFF .

A 2.5. ábrán egy új elemet is felfedezhetünk, mivel az ott szereplő címdekóder nem teljesen azonos a 2.4. ábrán látottal. Az egyes VAGY kapukhoz egy újabb jel is csatlakozik, ami a memóriaegység bemenő  $\overline{CS}$  jele lett. Mivel a VAGY kapuhoz kötöttük be, a kimeneten akkor lesz L szint, ha ez a jel is L, azaz aktív szintű. Ezáltal a teljes memóriaegységnek is van egy  $\overline{CS}$  bemenete, ami nélkül nem lehetne a mik-

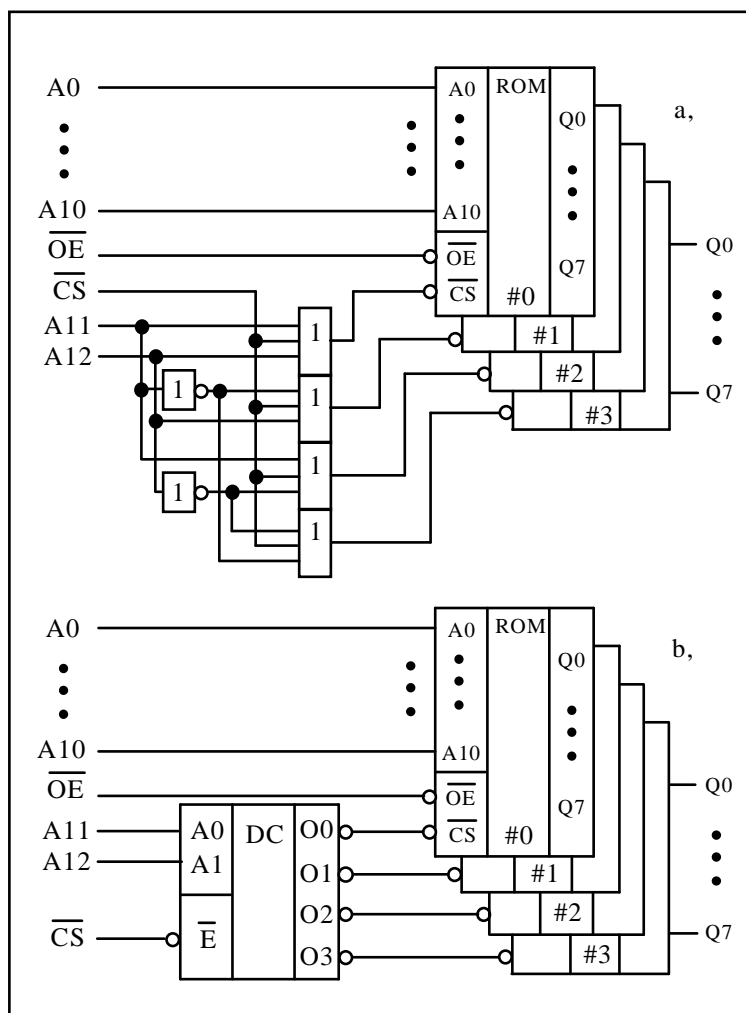
roszámítógépekben kezelni, nem lehetne a processzor címtartományán belül elhelyezni. A bővített memóriaegység így valóban ugyanolyan kívülről, mintha egyetlen, 8 Ki x 8 kapacitású ROM memória lenne (**2.6. ábra**).



2.6. ábra

Ha a kapcsolási rajzon azonos jellegű és azonos kapacitású, azaz teljesen egyforma memória-áramkörök szerepelnek és a csatlakozópontjaik többsége azonosan van bekötve, nem szükséges a 2.5. ábrán látható módon részletesen mindegyiket ábrázolni. Az azonos áramköröket egymás takarásában rajzoljuk ilyenkor, lefelé vagy felfelé annyira toljuk el a rajzelemeket, hogy az eltérően, különböző módon bekötött részleteket ábrázolni lehessen (most a  $\overline{CS}$  pontok az ilyenek csak). Jobbra is eltoljuk egy kicsit a rajzelemeket, a végeredmény a **2.7. ábra** felső részén látható. Az alsó ábrarészleten a címdekódert is egyszerűsítve rajzoltuk meg.

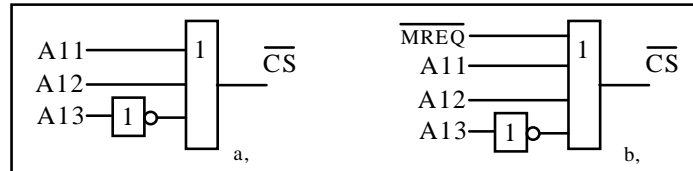
A takarással rajzolt elemeken csak a legfelsőn szerepeltetjük a csatlakozópontok elnevezéseit. A bal oldali csatlakozásokat a legfelső, a jobb oldaliakat a legalsó elemhez kötjük be.



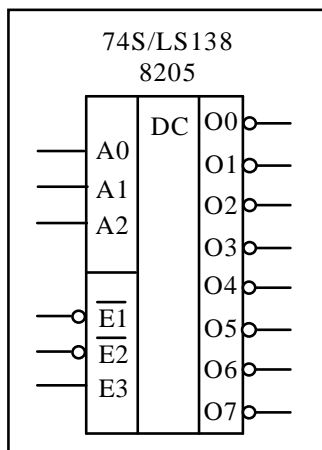
2.7. ábra

Fontos, hogy értse, hogy a 2.5. és a 2.7. ábra ugyanazt az áramkört mutatja be!

Ha a címbitekkel megadott kombinációval kell a memória-áramkört engedélyezni, gyorsan célhoz érünk, ha kapuval oldjuk meg a címkezelést, és a kapu kimenőjelét vezetjük a  $\overline{CS}$  bemenetre. Tegyük fel, hogy egy memória IC akkor működik, ha  $A11 = 0$ ,  $A12 = 0$  és  $A13 = 1$ . Az előzőek alapján egy VAGY kapuval oldható meg a  $\overline{CS}$  jel előállítás, amire az  $A13$ -at inverteren át vezetjük be (2.8. a, ábrarészlet). Ha még egy alacsony szinten aktív vezérlőjelet (pl.  $\overline{MREQ}$ ) is figyelembe kell vennünk (azaz csak akkor érvényesülhet a cím, ha ez a vezérlőjelet is L szintű), azt is egyszerűen bevezetjük a VAGY kapuba (2.8. b, ábrarészlet).



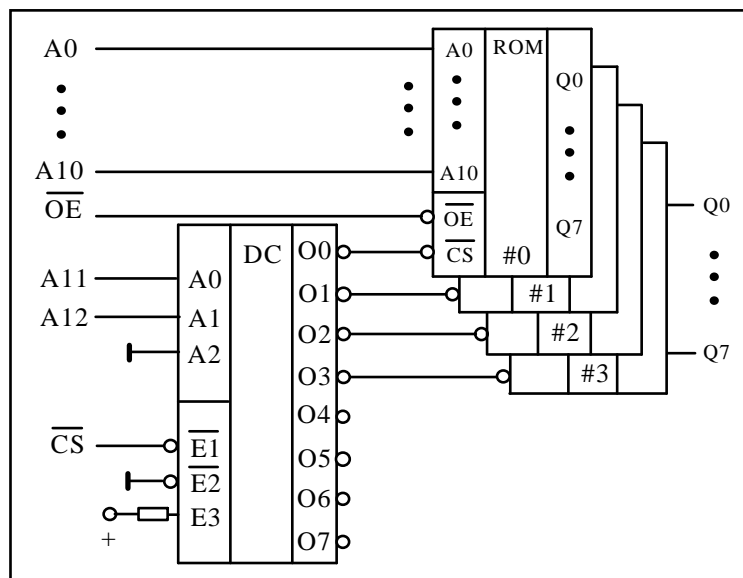
2.8. ábra



2.9. ábra

VAGY kapuba (2.8. b, ábrarészlet).

A 2.7. ábrán egy címdekóder rajzoltunk fel a címdekódolást végrehajtó kapuhálózat helyett. A kereskedelemben készen is kaphatók integrált címdekóderek! Ezek különböző bitszámú bemenetet kódolnak szét, s különböző a vezérlőjel-készletük is. A továbbiakban a 74LS138 (korábbi nevén 8205) típusjelű címdekóder fogjuk használni (2.9. ábra). Ez három bites dekóder, az  $A0$ ,  $A1$  és  $A2$  jelölésű bemeneteire érkező három bitet kódolja szét. A három bit nyolc lehetséges kombinációja a nyolc kimenet ( $O0 \dots O7$ ) egyikét aktivizálja, a kimenetek alacsony aktív szintűek. Amelyik kimenet „sorszámát” az  $A0$ ,  $A1$ ,  $A2$  bemenetekre vezetjük, az a kimenet lesz L szintű, a többi H értékű. A címdekóder csak akkor működ-teti a kimeneteit, ha az engedélyezőjelei (mind a három,  $\overline{E1}$ ,  $\overline{E2}$  és  $E3$ ) aktív szintűek, egyébként minden kimenet H szinten áll. A címdekóder logikai rajzjele a 2.9. ábrán látható. A 2 Ki x 8 kapacitású memóriaegységünk ezzel a címdekóderrel is megtervezhető (2.10. ábra).



2.10. ábra

Mivel csak két címbitet kell dekódolni, a dekóder harmadik bemenetét ( $A2$ ) 0-ra kötöttük. Az engedélyezőjelek egyikét hasznosítjuk  $\overline{CS}$  jelként, a másik kettőt állandó aktív szintre (az  $\overline{E2}$ -t 0-ra, az  $E3$ -at 1-re) kötöttük.

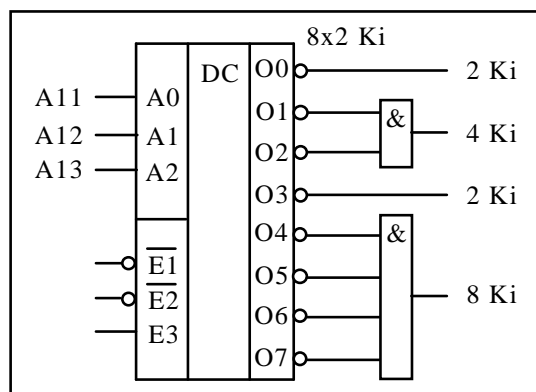
Általában akkor használjuk a címdekóder-t, ha legalább három memória-áramkört kell aktivizálni, az egyszerűbb esetekben kapukkal oldjuk meg a címzést.

A gyári címdekóderek a bemenetek minden lehetséges kombinációjához egy-egy kimenetet rendelnek hozzá, ami azt jelenti, hogy a dekódolt címtartományt 2, 4, 8 vagy 16 egyforma méretű területre osztják fel. Ha eltérő méretű címterületekre van szükségünk, akkor sem kell azonban rögtön elvetnünk a használatukat! Tegyük fel, hogy a következő címterületeket kell kijelölnünk:

2 Ki, 4 Ki, 2 Ki, 8 Ki.

Ha a címdekódere az A11, A12, A13 címvezetéseket kapcsoljuk rá, akkor a kimenetekkel nyolc darab 2 Ki méretű területet lehet kijelölni ( $8 \times 2$  Ki). Ha azt akarjuk, hogy két vagy több szomszédos kimenet hatása egyesüljön egy nagyobb címterületté,  $\overline{\text{ES}}$  kapuval oldható meg a feladat (hiszen ha bármelyik kimenet L szintű, az  $\overline{\text{ES}}$  kapu kimenete is L szintű lesz). A **2.11. ábrán** látható, hogyan lehet a kívánt területek kijelölését megoldani gyári címdekóderrel.

(Ismét felhívjuk a figyelmet arra, hogy ha L aktív kimeneti jelre van szükség és azt több L aktív bemenet egyidejű fennállása idézi elő, akkor VAGY kapuval oldható meg a kialakítás, ha több L aktív jel vagyla- gosan állítja elő az 1 aktív kimeneti jelet, akkor  $\overline{\text{ES}}$  kapuval lehet dolgozni!)



2.11. ábra

## 2.2. Feladat

4 Ki x 4 kapacitású SRAM memóriaegységre van szükségünk, a rendelkezésre álló IC (**2.12. ábra**) kapacitása csak 512 x 4 (azaz  $\frac{1}{2}$  Ki x 4). Ismét szószám szerint kell bővítenünk a memóriát.

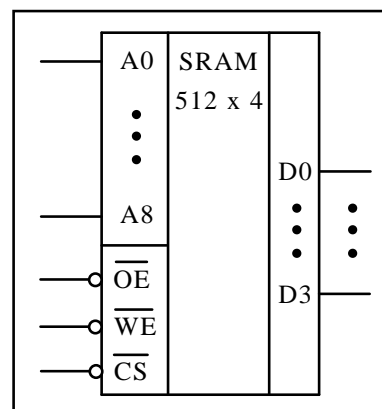
A memória-áramkörünk három vezérlőjeles SRAM-elem. Mivel a szószáma 512, a 4 Ki kapacitáshoz 8 darabra van szükség. Tervezzük meg közösen a memóriaegységet, a címkezelésre címdekódert használjunk fel!

A **2.13. ábrán** a kapcsolási rajz alapját már elkészítettük, szerepel a rajzon a címdekóder és a nyolc SRAM-IC. A memória-áramköröket takarásos megoldással ábrázoltuk. Mivel majdnem minden pontjuk azonosan van bekötve, csak a  $\overline{\text{CS}}$  kezelése tér el, ezt az egyet kellett egyenként ábrázolnunk az eltoltságon.

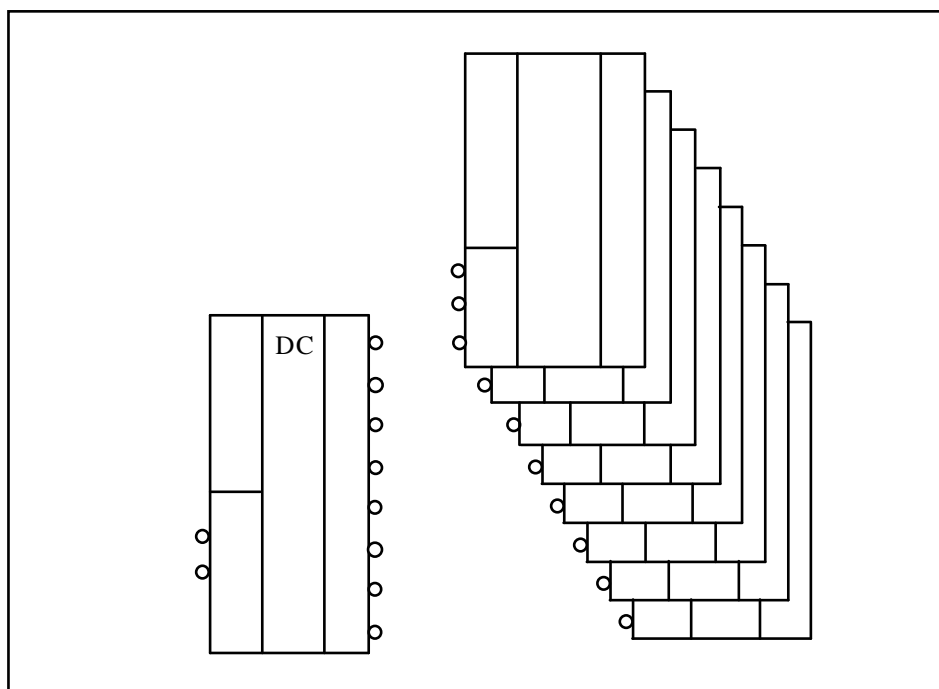
Emlékeztetőül: mit is jelent ez? Azt, hogy mind a nyolc memória IC A0 pontja egymással össze van kötve, hasonló módon a nyolc A1 is és a többi címvezeték is. De össze vannak egymással kötve a D0 pontok is, azután a D1 pontok stb. A vezérlőjelek közül is mind a nyolc áramkör  $\overline{\text{OE}}$  pontja egymással össze van kapcsolva, és szintén összekötöttük a nyolc áramkör  $\overline{\text{WE}}$  pontjait is. Csak a  $\overline{\text{CS}}$  pontok nincsenek összekötve. A nyolc memória-áramkör ezután a kapcsolás többi részéhez összesen 19 + 8 vezetékkel csatlakozik. Számoljon utána!

A címdekóder a már megismert 8205. Mivel nyolc egyforma címterületet kell kijelölni, a három bites címdekóder éppen megfelelő lesz. Az  $\frac{1}{2}$  Ki miatt a memória IC-k saját címbitjei az A0 ... A8 (kilenc darab), a nyolc áramkört az A9, A10, A11 címbitek kombinációi fogják egymástól megkülönböztetni. Összesen tehát a memóriaegység az A0 ... A11 címbiteket használja (12 darab), ami megfelel a teljes kapacitás szószámának, a 4 Ki-nak.

Első lépésként írja be a 2.13. ábrába, az elemekbe a jelek elnevezéseit! A memória IC-knél csak a legfelső elembe kerülnek megnevezések! Ennél az ábránál még egy lépést oldjon meg, a memóriaegység kereténél rajzolja meg a csatlakozópontokat, írja fel az elnevezéseiket! Az „eredő” egység kapacitásából (4 Ki x 4) tudjuk, hány és milyen nevű címbemenet és adatpont szerepel, a SRAM jellegből pedig következnek a vezérlőjelek. A 2.12. ábrán látjuk, hogy három vezérlőjeles megoldást használunk.

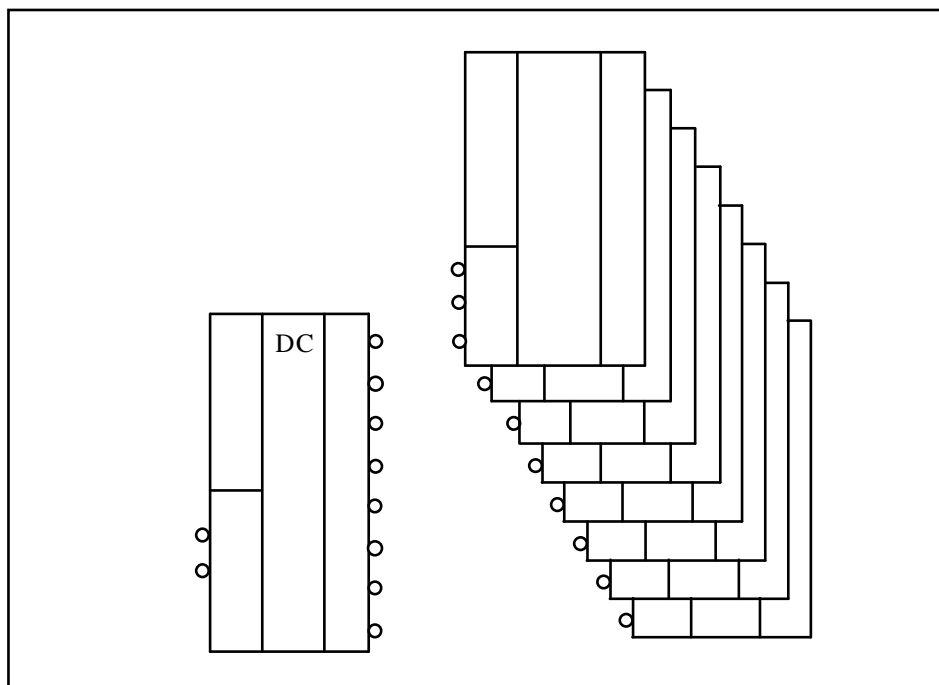


2.12. ábra



2.13. ábra

A 2.13. ábrán szereplő jelöléseket másolja át a 2.14. ábrába is! Azután folytassa a munkát. Kösse be a memória-áramkörök közösített pontjait a memóriaegység csatlakozóihoz. Ezt követheti a címdekóder bemeneteinek bekötése a további címbitekhez, majd a  $\overline{CS}$  külső jel létrehozása. Az utolsó lépés a címdekóder kimeneteinek és az egyes memória-IC-k  $\overline{CS}$  pontjainak az összekötése.



2.14. ábra

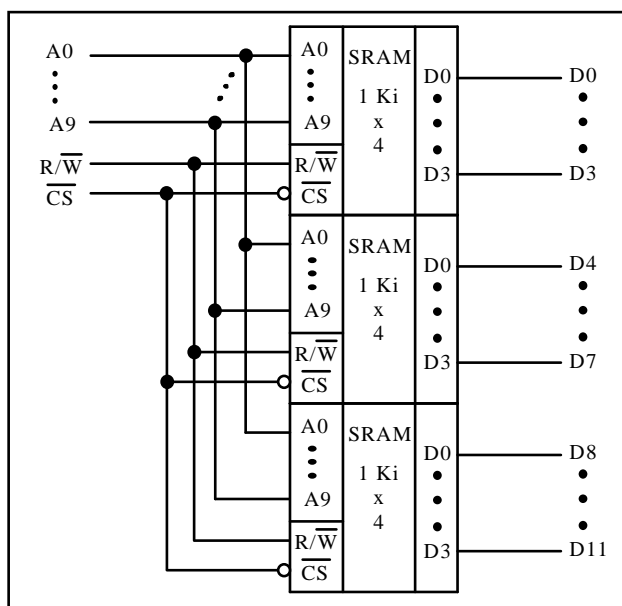
A végeredményt most is ellenőrizheti a 10. fejezetben!



### 2.3. Feladat

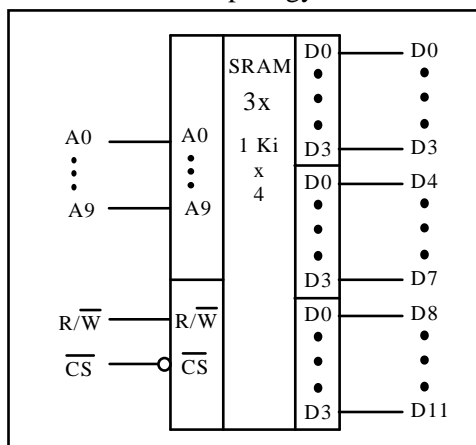
Készítsünk 1 Ki x 12 kapacitású SRAM-modult, 1 ki x 4 SRAM-IC-kből!

A rekeszek száma megfelelő lenne, hiszen 1024 szavasak a memória-áramkörök, de egyenként csak négy bit a szóhosszúságuk. Most tehát szóhossz szerint kell bővítenünk. A teljes szóhossz egy IC rekeszméretének háromszorosa, ezért három áramkörre van szükség. Tulajdonképpen a három áramkör címzését és vezérlését teljesen párhuzamosan kell végeznünk, az „eredő” 12 bites szó első 4 bitjét az első memória IC kezeli, a második négyet a második IC, a harmadik IC pedig a harmadik négybites részt. Az áramkört a **2.15. ábrán** mutatjuk be.



2.15. ábra

Külön nem is rajzoltuk le az egyes memória IC-eket, hanem mindjárt mind a három látható a kapcsolatban. A szóhossz szerinti bővítéskor is van egyszerűsítési lehetőség a rajzoláskor, a **2.15. ábrán** látható kapcsolást a továbbiakban a **2.16. ábrán** szereplő egyszerűbb módszerrel fogjuk rajzolni.



2.16. ábra

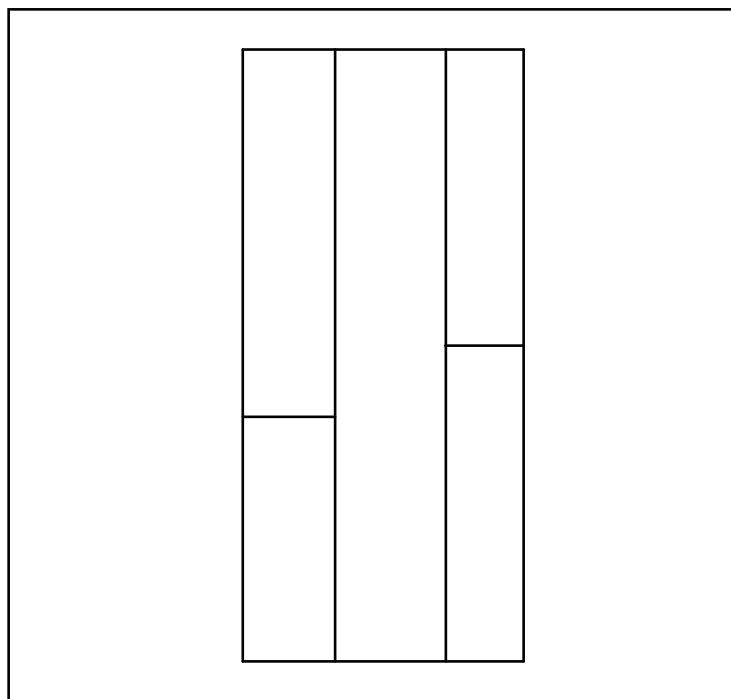
### 2.4. Feladat

Rajzoljon 2 Ki x 16 kapacitású memóriaegységet, amihez 2 Ki x 8 felépítésű ROM-IC-et használhat fel! A szószám ismét megfelelő, tehát itt is szóhosszúság szerint kell bővítenünk. A 2 Ki x 8 méretű memória-áramkörökből két darabra van szükség a 2 Ki x 16 felépítés kialakításához.

Most sem rajzoljuk le külön a két kiindulási ROM-áramkört, hanem mindjárt hozzálátunk a bővített egység felépítéséhez a **2.17. ábrán**. A 2 Ki x 8 méretű ROM címbitjeinek a száma:

A címbitek megnevezése:

Írja be a bemeneti és a kimeneti jelek nevét a 2.17. ábrára! Mivel két áramkör együtt valósítja meg a kívánt szóhosszúságot, az adatpontoknál két mezőt látunk, így kétszer kell beírni a kimenet pontok megnevezéseit is!



2.17. ábra

A végeredményt ellenőrizheti a 10. fejezetben!

### 3. Önálló memóriaegységek, memóriakártyák kialakítása

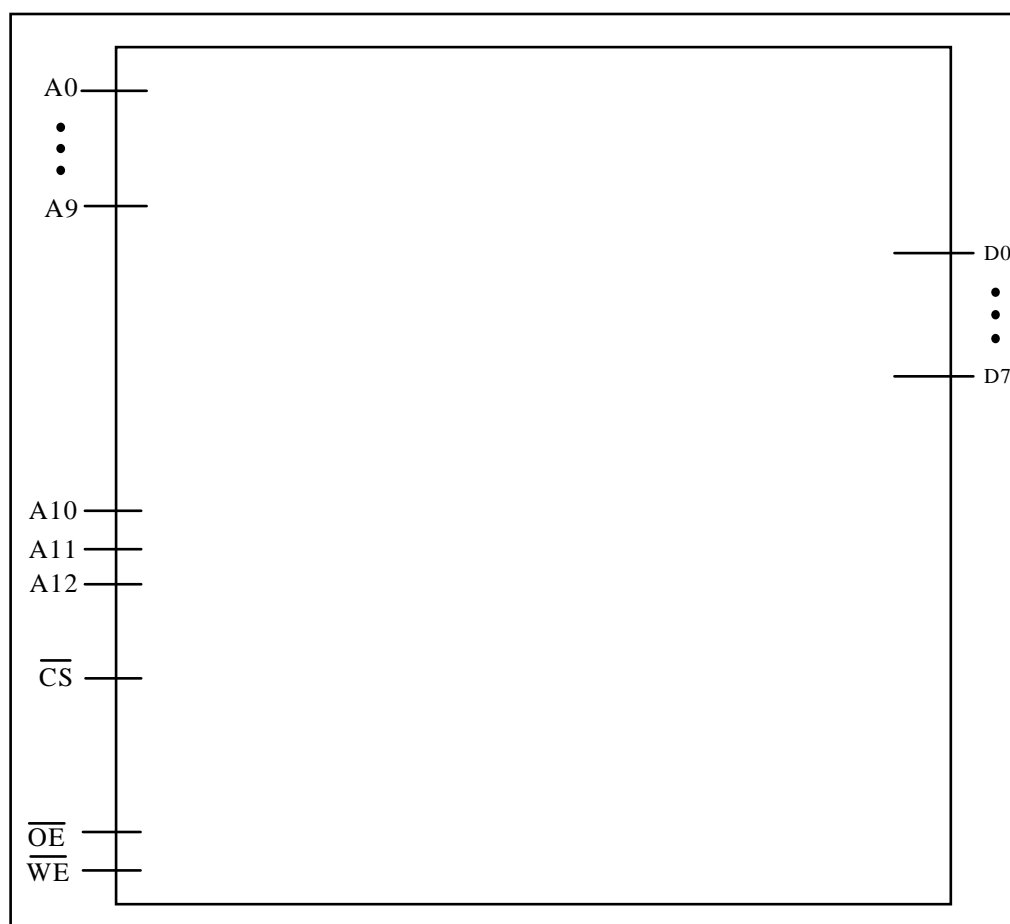
A szóhossz vagy szószám szerint bővített memóriaegységek úgy készültek el, hogy egy memória-áramkör több példányát az igényeknek megfelelően összekapcsoltuk egymással. Az önálló memóriaegység, a memóriakártya is felépülhet több azonos memória IC-ből, de ezek legtöbbször több különféle kapacitású, eltérő jellegű áramkört is tartalmaznak.

A memóriakártyák a mikroszámítógépek önálló egységei, moduláris felépítés esetén ezeket használják fel a memóriaigény kielégítésére. A kártya legfontosabb jellemzője az, hogy milyen kapacitású ROM és milyen kapacitású SRAM memóriát tartalmaz. Az, hogy milyen alkatrészekből, IC-kből áll, már csak a tervezéskor játszik szerepet, ugyanazt a memóriakártyát több különféle IC-készletből is össze lehet állítani.

A tervezéshez a kártya kapacitásértékéből indulunk ki, ezután a rendelkezésre álló memória-áramkörökből kiválasztjuk a felhasználásra kerülő elemeket. Megrajzoljuk a kártya „keretét” a csatlakozópontjaival, majd a kereten belül a memória IC-eket, szükség esetén a címdekódert. Végül elkészítjük az összekötéseket az áramkörök között, illetve a külső pontok és az áramkörök csatlakozói között.

#### 3.1. Feladat

Készítsünk memóriakártyát, ami tartalmaz 5 Ki x 8 ROM- és 2 Ki x 8 SRAM-áramkört! A felhasználható elemek: 1 Ki x 8 ROM és 1 Ki x 8 SRAM, három vezérlőjellel.



3.1. ábra

A kártya adatai alapján rajzoljuk meg a keretét és a csatlakozóit (**3.1. ábra**)!

Az 5 Ki x 8 ROM és 2 Ki x 8 SRAM összesen 7 Ki címtartományt jelent. Ez több, mint a 12 címvezetékkel megoldható 4 Ki, de kevesebb, mint a 13 címvezetékkel kivitelezhető 8 Ki. Nyilvánvalóan 13 címvezeték (A0 ... A12) fogunk használni, az ezekkel kezelhető 8 Ki címterületből 1 Ki kihasználatlan marad. Mivel a kártya megépítéséhez 1 Ki szószámú elemeket használunk, a memória-áramköröknek 10 címbemenete lesz (A0 ... A9), ezért a kártyánál ezt a 10 címpontot egymás közelében helyezzük el. Kissé távolabb a többi (A10, A11, A12) is megrajzolhatjuk, azokat a címdekóder fogja kezelni.

ROM kártyánál a kimeneteket Qi jelöli, SRAM és vegyes kártya esetén Di.

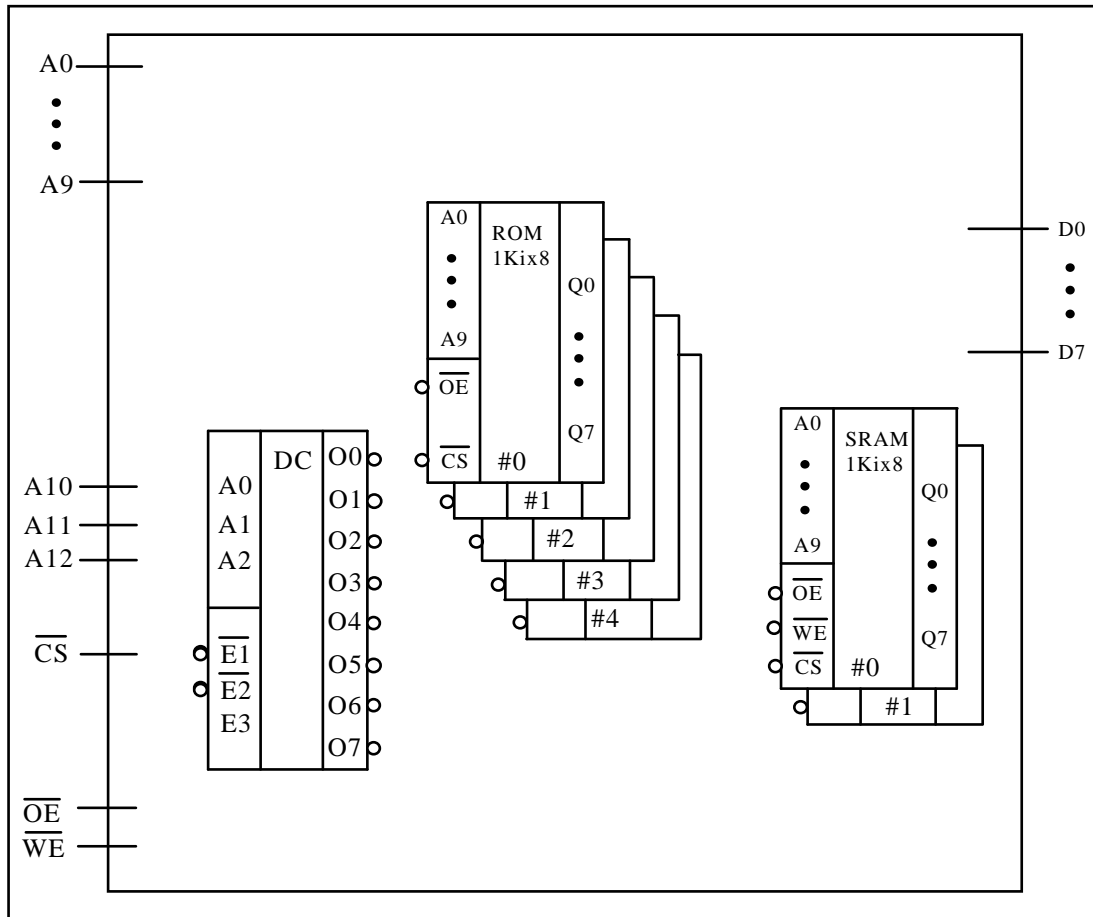
Minden kártyánál meg kell jelennie legalább egy engedélyezőjelnek, pl.  $\overline{CS}$ . Ez teszi lehetővé, hogy a teljes kártyát el lehessen majd helyezni a processzor címterületén belül, tetszőleges helyre.

A vezérlőjeleket a memóriák típusának megfelelően választjuk meg:

- ha a kártya csak ROM elemeket tartalmaz, a  $\overline{CS}$  mellett csak egy  $\overline{OE}$  szükséges,
- ha csak SRAM áramköröket építünk be, a szokásos 2 ( $\overline{CS}$ , R/W) vagy 3 ( $\overline{CS}$ ,  $\overline{OE}$ ,  $\overline{WE}$ ) vezérlőjelet használjuk,
- ha vegyesen szerepelnek a kártyán ROM- és SRAM-áramkörök, mindenképpen lesz  $\overline{CS}$ ,  $\overline{OE}$  vezérlőjel, és további R/W vagy  $\overline{WE}$  is.

A felsorolt megfontolások alapján alakultak ki a 3.1. ábrán látható jelnevezések.

A következő lépésben berajzoljuk a felhasználásra kerülő memória-áramköröket, a szokásos egyszerűsítési megoldásokat is alkalmazva (3.2. ábra).



3.2. ábra

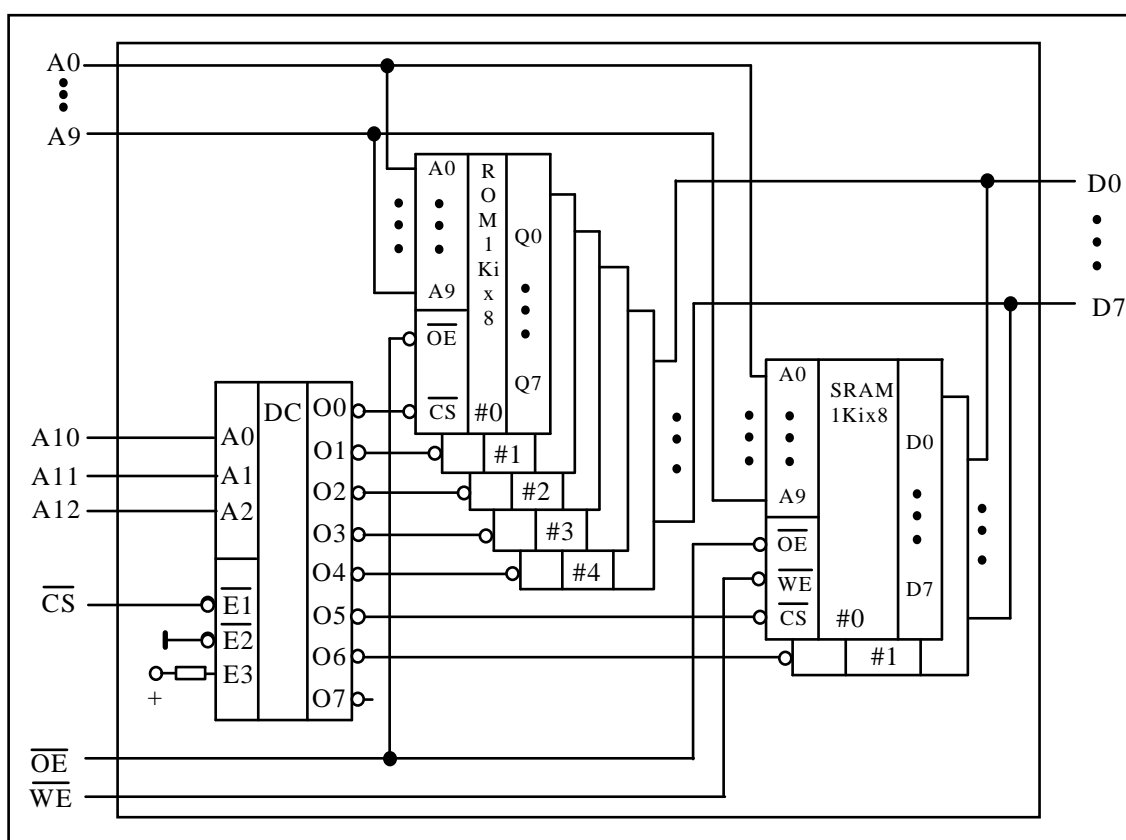
Mivel hét memória-IC alkotja a kártyát, célszerű a címdekóder beépítése.

A memória-áramkörök vezérlőjeleit is a kártya megfelelő csatlakozóihoz kell hozzákötni. A  $\overline{CS}$  pontokat nem használjuk egyelőre, mert azokkal lesznek a kártyán belül kiválasztva az IC-k, a kártya  $\overline{CS}$  pontját a címdekóderrel valósítjuk meg. Az  $\overline{OE}$  pontok mind a kártya  $\overline{OE}$  pontjához kapcsolódnak, a SRAM áramkörök  $\overline{WE}$  pontja a kártya ugyanilyen bemenetéhez.

Mivel minden beépített memória IC szószáma 1 Ki, mindegyiknek A0 ... A9 saját címbemenetei vannak. Ezeket egyszerűen rákötjük a kártya azonos elnevezésű címbevezetésére. Az adatpontokkal ugyanez a helyzet, azokat a kártya megfelelő Di pontjához kell kötni.

A hét darab 1 Ki méretű áramkör közül a címdekóder választ ki egyet, az A10, A11, A12 címbitek alapján, így ezeket a címbemeneteket a dekóderre vezetjük. A kártya specifikálásakor nem volt kikötve, hogy milyen sorrendben szerepeljenek a különféle memória-területek, ezért szabadon választhatunk. Legyenek az alacsonyabb címértékűek a ROM-elemek, a magasabbak a SRAM-ok.

A kártya teljes logikai kapcsolási rajza a **3.3. ábrán** található meg.



3.3. ábra

Ezt a memóriakártyát (5 Ki x 8 ROM, 2 Ki x 8 SRAM) természetesen más memória-áramkör készletből is össze lehetett volna állítani. Néhány további lehetséges elemkészlet:

- 5 db 1 Ki x 8 ROM, 4 db 1 Ki x 4 SRAM,
- 1 db 1 Ki x 8 ROM, 2 db 2 Ki x 8 ROM, 1 db 2 Ki x 8 SRAM,
- 1 db 1 Ki x 8 ROM, 1 db 4 Ki x 8 ROM, 2 db 2 Ki x 4 SRAM,
- 5 db 1 Ki x 8 ROM, 8 db ½ Ki x 4 SRAM.

Nagyszerű gyakorlási lehetőséget jelentenek ezek a különféle összeállítások, a fejezet áttanulmányozása után ezeket is megkísérelheti megtervezni. Ugyanígy a többi, a fejezetben később szereplő kártyát is áttervezheti, más áramköröket felhasználva.

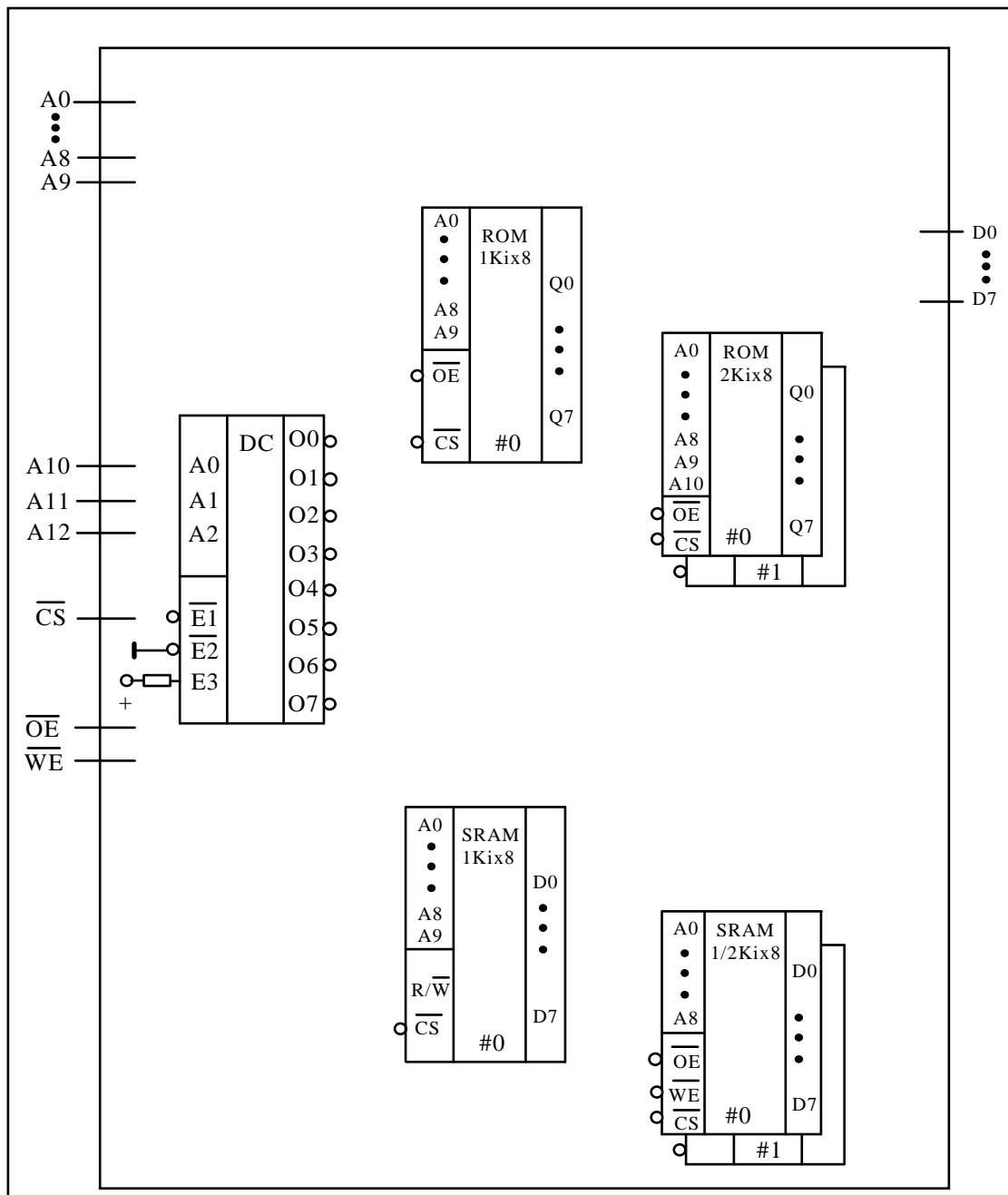
### 3.2. Feladat

Az előző memóriakártyát most új áramkörökkel készítjük el: A rendelkezésre álló IC-k:

1 db 1 Ki x 8 ROM, 2 db 2 Ki x 8 ROM,

1 db 1 Ki x 8 SRAM (2 vezérlőjeles), 2 db 1/2 Ki x 8 SRAM (3 vezérlőjeles).

A kártya csatlakozópontjai változatlanok, de más az áramkörkészlet, amit elhelyezünk benne 3.4. ábrán).



3.4. ábra

A legnagyobb gondot a két 1/2 Ki x 8 méretű áramkör okozza, mivel a dekóderrel nyolc darab 1 Ki méretű címterületet állítunk elő.



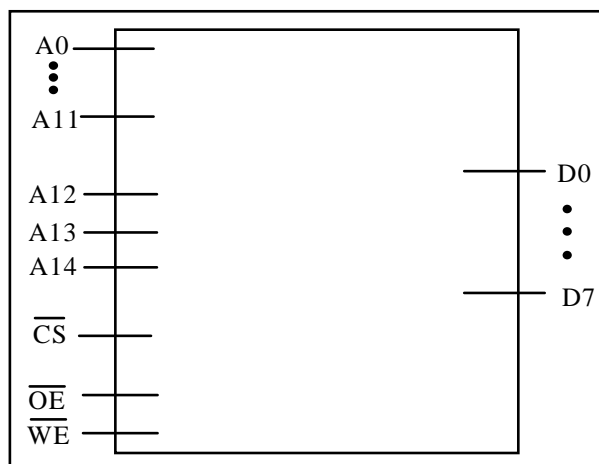
### 3.3. Feladat

Készítsünk memóriakártyát, amely 24 Ki x 8 ROM és 8 Ki x 8 SRAM területet tartalmaz. A felhasználható memória-áramkörök:

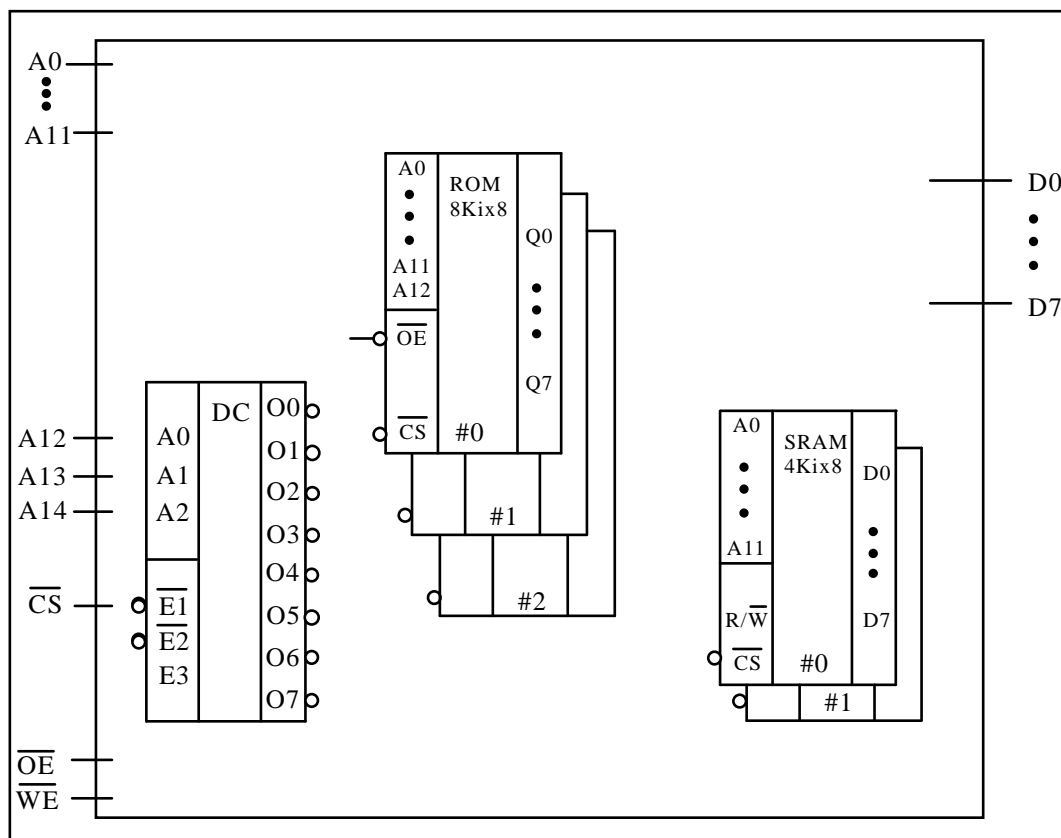
3 darab 8 Ki x 8 ROM, 2 db 4 Ki x 8 SRAM.

A kártya csatlakozópontjait a **3.6. ábra** mutatja be. Összesen 32 Ki címterületet használunk fel, ehhez 15 címbit (A0 ... A14) szükséges. A címdekóderrel 4 Ki széles területeket fogunk kijelölni, az A12, A13, A14 felhasználásával ( $8 \times 4 \text{ Ki} = 32 \text{ Ki}$ ). Ezért az A0 ... A10 címbiteket egymás közelében helyezzük el, hasonlóképpen az A12 ... A14 csatlakozópontjait is.

A 3.7. ábrán már berajzoltuk az áramköröket, az összekötések elkészítése azonban Önre vár!



3.6. ábra



3.7. ábra

Az elkészült kapcsolást ellenőrizheti a 10. fejezetben!

### 3.4. Feladat

Önállóan, beadható módon rajzoljon meg egy memóriakártyát, amely 8 Ki x 8 ROM és 12 Ki x 8 SRAM területet tartalmaz. A rendelkezésre álló áramkörök: 2 darab 4 Ki x 8 ROM, 1 darab 8 Ki x 8 SRAM (3 vezérlőjellel), 1 darab 4 Ki x 8 SRAM (3 vezérlőjellel).

Ha szükséges, az elinduláshoz a 10. fejezetben kap egy kis segítséget!



## 4. Memóriák csatlakoztatása buszrendszerhez, címtérképek

A memória-áramkörök alkalmazásának egyik formája az, ha memóriaegységeket, kártyákat, modulokat készítünk belőlük, s azokat építjük be a mikroszámítógépekbe, de gyakoribb a közvetlen megoldás. Ilyenkor a memória-áramköröket a processzor buszrendszeréről kezeljük.

A buszrendszer az a vezetékcsoport, amely a processzor és a mikroszámítógép egyéb elemei közötti kapcsolatokat lehetővé teszi. A mai processzorok hármas buszrendszerrel működnek, ennek az elemei: az adatbusz, a címbusz és a vezérlőbusz.

Az adatbusz (Data Bus) annyi vezetékből áll, ahány bites a processzor, nyolcbites mikroprocesszorok esetén nyolcból, a jelölésük Di (D0 ... D7). Az adatbusz kétirányú adatforgalmat bonyolít le, hiszen egyes esetekben a processzor küld ki adatokat valamelyik egységhez, máskor a mikroszámítógép valamelyik egyéb eleme küld adatot a processzorhoz. Ha a processzor küld ki adatot az adatbuszon, írásról beszélünk, ha a processzorhoz érkezik adat, azt olvasásnak nevezzük (azaz az írási és olvasási működést mindig a mikroprocesszor oldaláról értelmezzük). A mikroszámítógép minden egysége, eleme, amelyik adatokat fogad vagy küld ki, ezeket az adatokat nyolc vezetéken át kezeli, s ezek a vezetékek mind az adatbusz nyolc vezetékére csatlakoznak. Lényegében a mikroszámítógép egységeinek adatpontjai párhuzamosan kapcsolódnak, s az összekapcsolást megvalósító vezetékcsoportot nevezzük adatbusznak.

A vezérlőbusz (Control Bus) processzoronként változó felépítésű. Az elsődleges feladata az, hogy a vezetékei, a jelei segítségével a processzort tartalmazó központi egység (a CPU, Central Processor Unit) ki tudja jelölni, hogy a mikroszámítógép milyen jellegű egységével (memóriával vagy I/O elemmel) kíván a következő működési egységben, gépi ciklusban kommunikálni, s hogy ez a kommunikáció írási vagy olvasási jellegű-e. A négyféle eseményt (memóriaolvasás, memóriaírás, I/O-olvasás, I/O-írás) az egyes processzorok más-más vezérlőjelekkel jelzik. Egy további fontos működés a megszakítás, ennek elfogadását is a vezérlőbuszon jelzi a CPU. A vezérlőbusz vezetékei közül minden egységhez csak az ott szükségesek csatlakoznak. A felsorolt vezérlőjeleket mind a CPU küldi ki, de egyes vezérlőjelek (pl. a resetelő jel, a megszakítást kérő jel) kívülről haladnak a processzor felé.

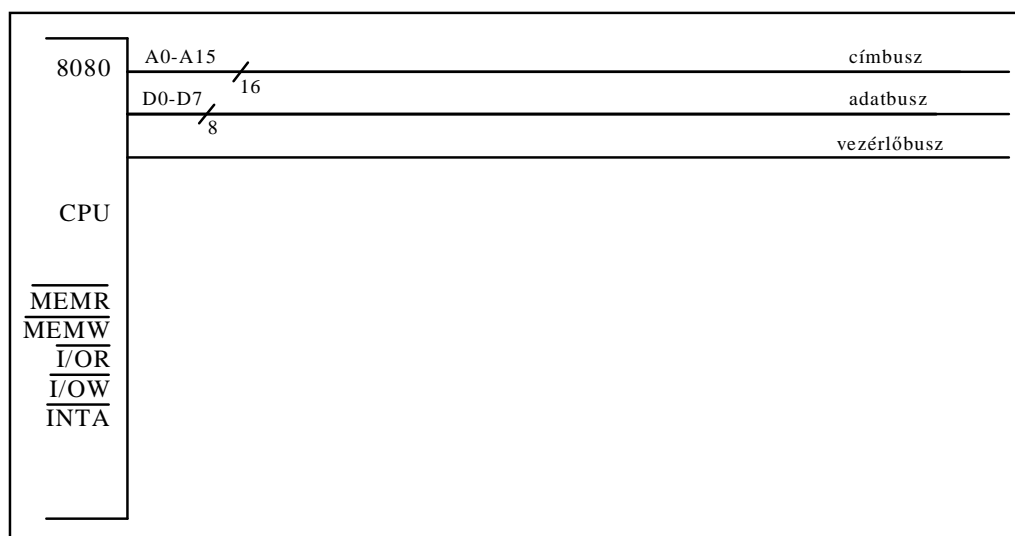
A CPU tehát meg tudja határozni a vezérlőbusz segítségével, hogy egy adott működési egységben, gépi ciklusban pl. memóriából kíván olvasni. Igen ám, de a mikroszámítógépben több memória-áramkör is van, s azok mindegyikében sokezer rekesz található. Ki kell tehát választani az egyik áramkört és azon belül ki kell jelölni az egyik rekeszt. Ezeket a kijelölő, kiválasztó feladatokat látja el a címbusz (Address Bus). Címet a processzorok csak küldenek, nem fogadnak, így a címbusz egyirányú: a CPU felől a többi elemhez haladnak a címjelek.

A címdekóderek a címbusz tartalma alapján választanak ki egy áramkört, s ha azon belül még több működési egység van (pl. egy 1 Ki x 8 kapacitású SRAM áramkörben 1024 rekesz), akkor azok közül közvetlenül a címbitek jelölnék ki egyet. A címbusz vezetékszama, a cím bitszáma határozza meg, hány elemi egységet, pl. memóriarekeszt tud kezelni a CPU, a címbusz vezetékeit Ai jelöli.

A **4.1. ábra** az *Intel 8080 mikroprocesszorának* buszrendszerét mutatja be. Az ábra olyan felépítésű, amelyre a kiegészítő elemeket, memóriákat és I/O egységeket is fel lehet majd rajzolni. A bal oldalon a CPU látható, megjelöltük a mikroprocesszor típust, és segítségképpen felsoroltuk a vezérlőjeleit is. Ezeket a vezérlőjeleket a vezérlőbuszon lehet elérni, onnan kell majd lecsatlakoztatni. A 8080-nál, az Intel rendszerű vezérlőbuszon minden fontosabb üzemmódnak saját külön vezérlőjele van:

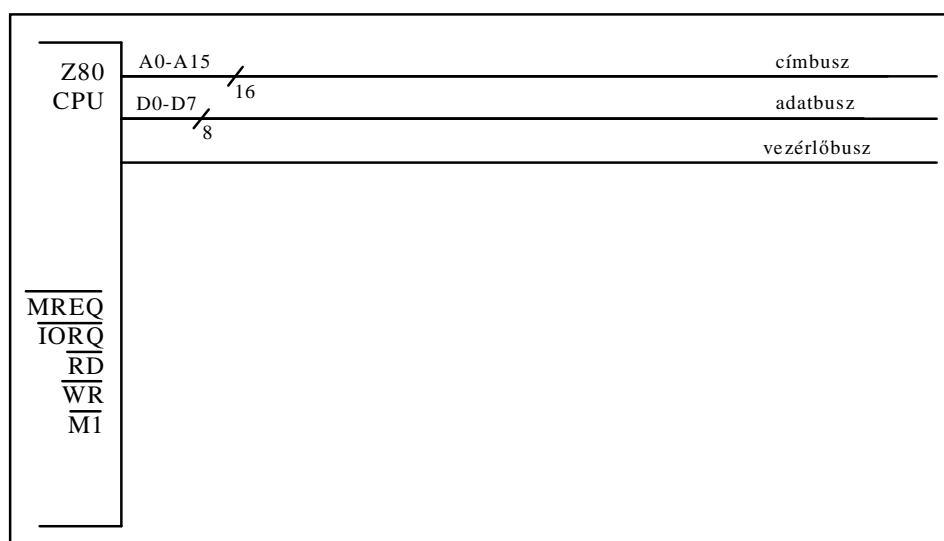
- memóriaolvasáskor a  $\overline{\text{MEMR}}$  aktív, azaz L értékű (Memory Read),
- memóriaíráskor a  $\overline{\text{MEMW}}$  aktív, tehát L szintű (Memory Write),
- I/O-olvasáskor az  $\overline{\text{I/OR}}$  aktív, azaz L szintű (I/O Read),
- I/O-íráskor az  $\overline{\text{I/OW}}$  aktív, tehát L értékű (I/O Write).

Az i8080 egy nyolcbites mikroprocesszor, ennek megfelelően az adatbusz nyolcbites (D0 ... D7). A címbusz 16 bites (A0 ... A15), ami 65 536 elemi egység, azaz 64 Ki kijelölését, címezését biztosítja.



4.1. ábra

Az  $\overline{\text{INTA}}$  vezérlőjelet ebben a jegyzetben nem használjuk (ez az elfogadott megszakítást jelzi). Minden busznak szerepeltetjük a nevét, ferde vonalka mellett a vezetékszámát, és a vezetékek megnevezését is. A következő feladatokban általában az Intel jellegű buszrendszert fogjuk használni. Elsősorban azért, hogy más megoldást is lássunk, néhány kapcsolást egy másik környezetben is bemutatunk, mégpedig a *Zilog Z80 mikroprocesszor* buszrendszere melletti alkalmazásban (4.2. ábra).



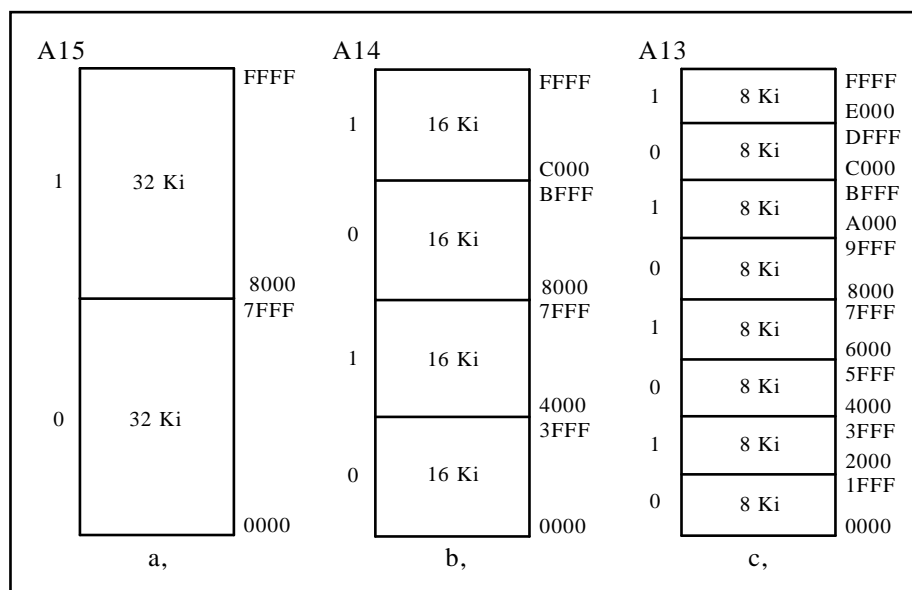
4.2. ábra

A Z80 is nyolcbites mikroprocesszor, ezért itt is nyolcbites az adatbusz (D0 ... D7). A nyolcbites mikroprocesszorokhoz szabványosan 16 bites címbusz tartozik, ezért itt is 16 bites (A0 ... A15) a címbusz. A vezérlőjelek azonban újszerűek, a fő működési módokat két-két jel együttesen határozza meg:

- memóriaolvasáskor a  $\overline{\text{MREQ}}$  és az  $\overline{\text{RD}}$  aktív, azaz L értékű (Memory Require és Read),
- memóriaíráskor a  $\overline{\text{MREQ}}$  és a  $\overline{\text{WR}}$  aktív, tehát L szintű (Memory Require és Write),
- I/O-olvasáskor az  $\overline{\text{IORQ}}$  és az  $\overline{\text{RD}}$  aktív, azaz L szintű (I/O Require és Read),
- I/O-íráskor az  $\overline{\text{IORQ}}$  és a  $\overline{\text{WR}}$  aktív, tehát L értékű (I/O Require és Write).

Az  $\overline{M1}$  vezérlőjelnek a megszakítás elfogadásakor van szerepe, ebben a jegyzetben nem használjuk.

A címbuszról, az egyes címbitek szerepéről még érdemes néhány mondatot szólni. A **4.3. ábrán** mutatunk be néhány fontos összefüggést a címezéssel kapcsolatban.



4.3. ábra

A teljes címtartományt (ami a 16 bites címbusz esetében 64 Ki) egy álló téglalappal szemléltetjük (címtérkép), alul a legkisebb (0000), felül a legnagyobb (FFFF) cím, minden címérték továbbra is hexadecimálisan szerepel. A 64 Ki címtartomány kezdő- és zárócíme tehát: 0000 illetve FFFF.

Ha binárisan felíránk az összes címet 0000000000000000 ... 1111111111111111 között, egyértelműen látszana, hogy a memóriaterületet jelző téglalap (a memóriatérkép) alsó felében az A15 (a baloldali, legnagyobb helyértékű címbit) 0 értékű, a felső felében pedig 1. Ezt mutatja az a, ábrarészlet is. Az A15 tehát a teljes (64 Ki) címtartományon egyszer vált értéket, alul 0, felül 1, ez a két terület egyformán 32 Ki. Mindezt röviden úgy is mondjuk, hogy az A15 32 Ki területeket jelöl ki a címtérképen (kettőt).

Az alacsonyabb helyértékű címbitek többször is értéket váltanak, az A14 például alul 0, majd 1, majd ismét 0, felül ismét 1. Egy-egy ilyen terület nagysága 16 Ki, azaz az A14 által kijelölt címtartományok nagysága 16 Ki (b, ábrarészlet). A c, ábrarészleten pedig azt mutatjuk meg, hogy az A13 a teljes 64 Ki címtartományon belül négyszer 0, négyszer 1 értékű, így nyolcfelé vágja szét a területet, egy-egy sáv tehát 8 Ki méretű – azaz az A13 a címtartományon 8 Ki széles sávokat jelöl ki.

Azt, hogy az A13 által kijelölt sávok 8 Ki méretűek, úgy is ki lehet számolni, hogy az A13 *alatt* 13 címbit van (A0 ... A12), s 13 címbittel 8 Ki területet lehet kezelni!

Ha az A15-öt használjuk majd egy elem működtetésére, egyértelműen 32 Ki területen fog aktivizálódni. Ha a működtetést az A15 0 értéke váltja ki, akkor az alsó 32 Ki-n fog dolgozni az elem, a felső 32 Ki-n akkor lesz kijelölve, ha az A15 = 1 működteti.

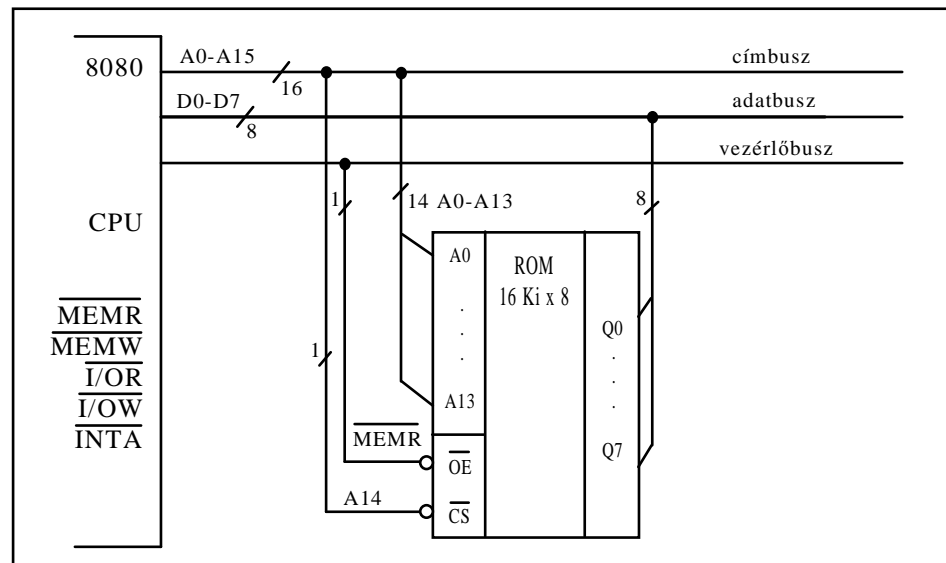
Nem ilyen egyértelmű az A14 használata, hiszen ha az A = 0 aktivizál egy elemet, az két darab 16 Ki méretű területen is működni fog, hasonlóképpen két területen jelölődik ki az egység, ha az A14 = 1 vezérli. A 16 Ki méretű területek közül egyet úgy lehet kiválasztani, ha megadjuk az A14 mellett az A15 értékét is!

A 4.3. ábrán bejelöltük a címtérképen szereplő területek első és utolsó címét is. Javasoljuk, hogy gyakorolja be a 64 Ki terület 32 Ki, 16 Ki, 8 Ki, 4 Ki, 2 Ki, 1 Ki sávokra osztásakor keletkező területek kezdő- és zárócímeit, hogy adott esetben ne kellejen hosszadalmasan számolnia ezeket az értékeket!

Egyelőre azt jegyezze meg pl., hogy ha a 8 Ki-s területeket általában ki akarjuk választani, akkor az A13 címbitét használjuk, ha a nyolc ilyen terület egyikére van szükség, akkor az A13 értékén kívül meg kell adnunk az A14 és az A15 értékét is (ezeket leolvashatjuk a 4.3. ábráról).

#### 4.1. Feladat

Vizsgáljuk meg, milyen címtérületen működik a **4.4. ábrán** látható ROM memória-áramkör! 8080 a processzor, a ROM olvasását a  $\overline{\text{MEMR}}$  vezérlőjel (L szintje) váltja ki. De csak akkor olvasódik a ROM, ha egyúttal a  $\overline{\text{CS}}$  bemenetén is L szint van. Ez pedig egyértelműen attól függ, hogy a címbuszon milyen címérték



4.4. ábra

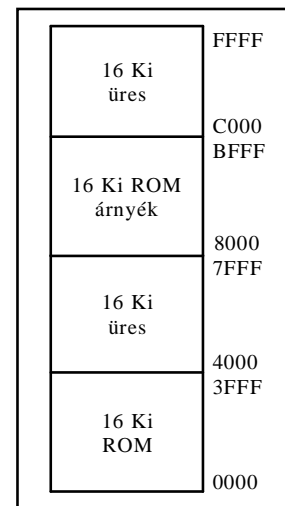
található! Mivel az A14 van erre a pontra kötve, mindannyiszor olvasódik az áramkör, amikor a  $\overline{\text{CS}} = 0$  mellett az A14 = 0. Természetesen ezzel még csak azt mondtuk meg, hogy olvasódik a ROM-áramkör, azt nem, hogy melyik rekesze! Azt pedig az alsó (A0 ... A13) címbitek értéke határozza meg, ezért azokat a címbusz megfelelő vezetékeihez kötjük, az adatpontokat pedig az adatbuszra.

A döntő tehát a ROM működése szempontjából az, hogy a memóriaolvasás gépi ciklus alatt (amikor  $\overline{\text{MEMR}} = \text{L}$ ) milyen értékű az A14. A teljes címtartomány legkisebb címén (0000 hexadecimálisan, binárisan pedig 0000000000000000) az A14 nyilvánvalóan 0 értékű (hiszen itt minden címbit 0 értékű), azaz a 0000 címen a ROM működik! De ha egyesével növelni kezdjük a címértékeket, még jó darabig 0 marad az A14! A 4.3. ábrán látható, hogy 16 Ki területen keresztül 0 az értéke! Eszerint ezen a 16 Ki területen a ROM működik, azaz a 0000 ... 3FFF területen)! Ha tovább növeljük a címeket, az A14 már 1 értékű lesz, ismét 16 Ki területen keresztül, tehát a 4000 .... 7FFF címtérületen nem működik a memória. A következő (8000) címben azonban ismét 0 értékű az A14, és innen további 16 Ki területen keresztül, azaz ismét egy 16 Ki méretű sávot találtunk, ahol a memória olvasódik. Végül a 64 Ki utolsó 16 Ki méretű sávjában az A14 ismét 1 értéket vesz fel, ott a ROM nem működik.

A ROM működését a **4.5. ábrán** címtérképen szemléltetjük. A buszrendszerre egyéb áramkör nem kapcsolódik, ahol a ROM nem aktív, nem olvasódik, ott egyéb elem sem működik, ezért került azokba a területekbe az „üres” felirat.

A 16 Ki ROM áramkör tehát két darab, egyenként 16 Ki területen aktivizálódik, a 0000 ... 3FFF és a 8000 ... BFFF területeken. Ilyen esetben, ha az áramkör többször „megjelenik” a címtérképen, általában a legkisebb című területet tekintjük valódi működési területnek, a többit *árnyéknak*. Ezért került a 8000 ... BFFF területre ez a felirat: 16 Ki ROM árnyék.

A 4.4. ábrán látható áramkört, a 16 Ki szószámú ROM-IC-t éppen ekkora, 16 Ki széles címtérületen aktivizálja (mégpedig két ilyen területen). A tervezés során olyan helyzetet nem állítunk elő, hogy egy memóriát kisebb területen aktivizálunk, mint a saját címtartománya. Az viszont gyakori, hogy szélesebb (kétszeres, négyszeres stb.) címtérületen aktivizálódik egy memória IC. Ilyenkor az aktivizáló területen belül is árnyékok jelennek majd meg. Az, hogy a memóriáknak árnyéka is van, tulajdonképpen semmilyen problémát nem okoz. Ha azonban bővítenénk a memóriát, szükségünk lehet az árnyékkal letakart területre is.

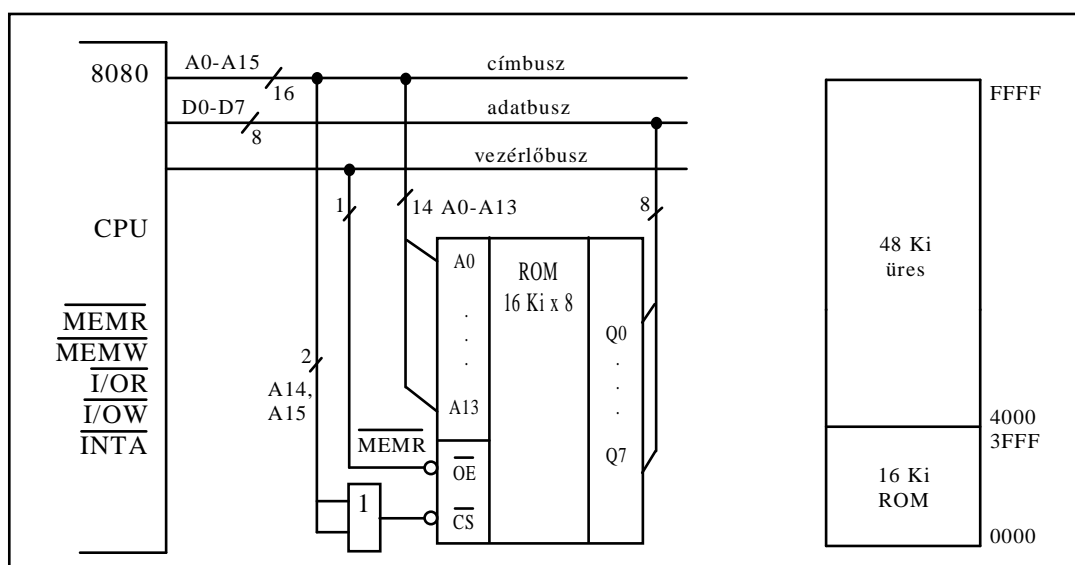


4.5. ábra

Ha egy kapcsolatban a címzés részleges, így a memória árnyékos és az árnyékot meg kívánjuk szüntetni, az áramkört árnyékmentesíteniünk kell. A mi esetünkben vagy a 0000 ... 3FFF címtartományon, vagy a 8000 ... BFFF címtartományon meg kell szüntetnünk a ROM működését. Válasszuk az alsó címtartományt véglegesnek, és a 4.5. ábrán árnyékként jelölt területet szüntessük meg!

A valóságban nem a 8000 ... BFFF területen szüntetjük meg a működést, hanem éppen a 0000 ... 3FFF területen tesszük egyértelművé, ezáltal automatikusan üressé válik az eddigi árnyék területe! A 4.3. ábrán látható, hogy az alsó 16 Ki területen az A15 értéke 0, a felsőnél 1. Az A15 segítségével lehet tehát a két területet megkülönböztetnünk. Eddig azért volt árnyékos a kapcsolásunk, mert az A15 címbitét nem vettük figyelembe a memória kijelölésekor!

Tudjuk, hogy a 0 értékű jeleket VAGY kapuval lehet „összefogni”, ha alacsony aktív szintű jelet kívánunk előállítani, így az áramkörünket csak egy ilyen kapuval kell bővítenünk, arra vezetjük rá az A14 és az A15 címbiteket, a kimenetét pedig a ROM áramkör  $\overline{CS}$  bemenetére (4.6. ábra). Az ábrán látható az új címtérkép is, ahol már nincs árnyék.



4.6. ábra

Ha a VAGY kapura az A15 egy inverteren át érkezik, akkor az alsó 16 Ki-ból tűnik el a ROM, és a 8000 ... BFFF területen fog működni, alatta 32 Ki üres terület lesz, felette 16 Ki.

Az A14 is átvezethető inverteren, így az A15, A14 minden lehetséges kombinációja (00, 01, 10, 11) felhasználható a 16 Ki kiválasztására, a címtérkép többi része pedig üres lesz.

Megjegyezzük, hogy elvileg a 16 Ki szószámú ROM-memória elhelyezhető a címtérképen a 0000 ... 3FFF területen is, a 0001 ... 4000 területen is, a 00032 ... 4002 területen is stb., hiszen ezek mind 16 Ki méretűek. A tervezési gyakorlatban azonban egy memóriát általában olyan területen helyezünk el, amit úgy kapunk, hogy a teljes területet (most 64 Ki) felosztjuk a memóriáinknak megfelelő méretű, egymás melletti területekre, s azokból választunk ki egyet.

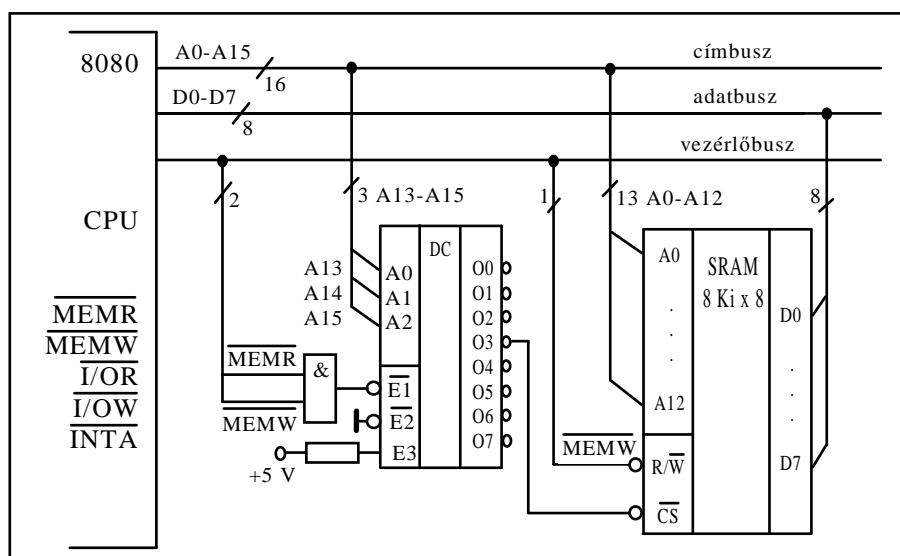
A 4.6. ábrán látható árnyékmentes kapcsolásunkat Z80 mikroprocesszorral is megrajzoljuk. A címbusz és az adatbusz kezelése nem változik, azokat ugyanúgy kell bekötni. A vezérlőjelek azonban módosulnak.

A Z80 esetében memóriaolvasáskor a  $\overline{MREQ}$  és az  $\overline{RD}$  aktív, azaz L értékű. Több módon is megoldható a vezérlőjelek kezelése. A  $\overline{MEMR}$  jel helyére bevezetjük az  $\overline{RD}$  jelet, az A14, A15 címbiteket fogadó VAGY kapu helyére pedig hárombemenetű kaput rajzolunk, és a harmadik bemenetre kerül a  $\overline{MREQ}$  jel. Egy másik lehetőség, hogy a címbiteket kezelő kétbemenetű VAGY kapu megmarad, de egy másik kétbemenetűt rajzolunk a ROM-áramkör  $\overline{OE}$  bemenete elé is.



Ennél az egyetlen memória-áramkörnél a címdekódolást kapuval is egyszerűen meg lehet oldani. Most mégis címdekódot alkalmazunk, mert a címzés kialakításának módszere pontosabban követhető lesz így. A 4.3. ábra a, részletén bemutatott címtérületeket egy 8205 dekóderrel precízen elő lehet állítani, az ábrának megfelelően az A13, A14, A15 címbiteket kell dekódolnunk az áramkörrel. A dekóder kimenetei (O0 ... O7) az ábrán bejelölt sorszámoknak (#0 ... #7) megfelelően jelölik ki az egyes 8 Ki szélességű címsávokat. Ha a 6000 ... 7FFF sávot kívánjuk felhasználni, az O3 kimenetre lesz szükségünk. Ha később át kell helyeznünk a memóriát, vagy másik memóriákat is be kívánunk illeszteni az áramkörbe, a dekóder ebben nagy segítséget nyújt!

Mivel a címdekóder a legmagasabb címbiteket kódolja szét (A13, A14, A15), a dekódolt terület a címtérképen nem ismétlődik meg, nem keletkezik árnyék. A kijelölt terület mérete 8 Ki, a memória száma is 8 Ki, így a kijelölt területen belül sincs árnyék, tehát ez a megvalósítás valóban árnyékmentes! A logikai kapcsolást a 4.9. ábrán láthatjuk.



4.9. ábra

Az áramkör kialakításakor először az ökölszabályoknak megfelelően jártunk el. Felrajzoltuk a CPU-t és a buszrendszert, majd a memória következett. A kapacitásból adódott a címbitek és az adatbitek száma, a jellegből (két vezérlőjeles SRAM-memória) a vezérlőjelek következtek. Bekötöttük a memória adatpontjait az adatbuszra, a saját címbemeneteit (A0 ... A12) a címbusz ugyanilyen elnevezésű vezetékeire. A  $\overline{\text{MEMW}}$  vezérlőjelet kapcsoltuk rá a kétvezetékes SRAM írás/olvasást vezérlő bemenetére, az  $\text{R}/\overline{\text{W}}$  bemenetre. A címdekóderre vezettük az A13, A14, A15 címvezetékeket és az O3 kimenetről engedélyeztük a SRAM működését. Egyetlen érdekes, új részlet látható a kapcsolásban, a címdekóder  $\overline{\text{E1}}$  bemeneténél lévő  $\overline{\text{ÉS}}$  kapu. Ez a kapu azért szükséges, mert a SRAM két vezérlőjeles megoldású, a három vezérlőjelesnél nem lenne rá szükség.

Tegyük fel, hogy memóriaírási ciklusban vagyunk, és a CPU a címbuszra egy 6000 ... 7FFF címtartományba tartozó címet küld ki. Ekkor az adatbuszon lévő adat beíródik a SRAM megcímzett rekeszébe.

Ha a címbuszra ismét a 6000 ... 7FFF tartományba eső cím van, de nem memóriaírási gépi ciklus folyik le, a SRAM minden esetben olvasódní fog, a megcímzett rekeszének a tartalmát az adatbuszra helyezi. Pedig lehet, hogy egy I/O-írási ciklust kezdett éppen a CPU, és ezért a CPU is kiküld egy adatot. Adatütközés, esetleg áramköri meghibásodás következik be! Hogyan lehet elérni, hogy a SRAM csak a memóriaolvasási és memóriaírási ciklusokban működjön? Úgy, ha az engedélyezését ezekhez a ciklusokhoz kötjük! Mivel a 8080 esetében ezt a két ciklust a  $\overline{\text{MEMR}}$  vagy a  $\overline{\text{MEMW}}$  vezérlőjel jelzi, ezek egy  $\overline{\text{ÉS}}$  kapun át engedélyezhetik a címdekódot. Ha egyik jel sem L szintű, az  $\overline{\text{ÉS}}$  kapu kimenete H lesz és letiltja a dekódot. Ezért szerepel ez a kapu a 4.9. ábrán.

Ha kapukkal oldjuk meg a két vezérlőjeles SRAM kijelölését, akkor a címbiteket kezelő kapunál kell a MEMR és a MEMW jelek felhasználásával megoldani, hogy csak akkor működjön az IC, ha e két jel közül valamelyik L szintű. Ilyen megoldás esetében, ha több SRAM áramkört használunk, mindegyiknél külön-külön kell gondoskodni a hamis olvasási működés elkerüléséről, ezért is egyszerűbb a címdekóder használata.

Még néhány szó a címekről. A 4.9. ábrán a SRAM a 6000 ... 7FFF címtartományra került. Mivel ez egy 8 Ki x 8 kapacitású memória, a saját címtartománya 0000 ... 1FFF. Ez a kettősség azt jelenti, hogy ha processzor a 6000 címre ír, a SRAM 0000 saját című rekeszébe fog adatot behelyezni. Ha a CPU a 7AA5 címről olvas, akkor a SRAM-IC 1AA5 című rekeszének tartalmát fogja megkapni.

Fordítva is eljátszhatunk a címekkel. Ha a SRAM 0D39 saját című rekeszébe kívánunk írni, a CPU-nak a 6D39 címet kell a címbuszra helyeznie. Ha a SRAM 1FFF saját című rekeszt kívánjuk kiolvasni, akkor a processzornak a 7FFF címet kell kiküldenie.

A 8080 mikroprocesszor alkalmazása további érdekes „fejtörőre” is alkalmat ad. Tegyük fel, hogy a 4.9. ábra kapcsolásából kihagyjuk a címdekódernél lévő ÉS kaput, az engedélyezőjelet 0 V-ra kötjük. Ha a mikroszámítógépben semmilyen egyéb elem nem csatlakozik a buszrendszerre, nem történik hibás működés, a CPU nyilván csak memóriaírási és memóriaolvasási működéseket fog végezni. Ha a mikrogépet I/O egységekkel is kiegészítjük, akkor már kialakulhat a helytelen működés! Elkerülhetjük ezt, ha az I/O elemeket olyan címtartományon működtetjük, ami nem aktivizálja a SRAM-ot. De ez nem azt jelenti, hogy az I/O címezése nem használhatjuk a 6000 ... 7FFF címeket! A 8080 ugyanis, ahogyan az I/O-elemekkel foglalkozó fejezetekben látni fogjuk, a bemeneteit és a kimeneteit 8 biten címzi, ezt a 8 bitet a címbusz alsó felén küldi ki. A címbusz felső 8 bitje sem lesz kihasználatlan, itt ugyanis ugyanezt a nyolcbites értéket szintén kihelyezi a mikroprocesszor.

A mi esetünkben tehát az I/O működtető gépi ciklusokban 6000 cím nem alakulhat ki, mert ennek nem azonos az alsó és a felső bájtja. Nem jöhet létre 6001, 6002, 6003, 6004, 6005 stb. cím sem. Az első, ami I/O címezéskor is megjelenhet, a 6060. A következő a 6161, majd a 6262 stb. A 6-tal kezdődő ilyen tulajdonságú címek közül a 16. ez lesz: 6F6F. Hasonló módon a 7070, a 7171, ... 7F7F is megfelel a kritériumainknak. Mindebből az következik, hogy a következő I/O címek használatát kellene tiltani:

60 ... 7F .

Egy megépített mikroszámítógép esetében nem elfogadható, hogy az I/O címek egy részének használatát meg kelljen tiltani, ezért célszerű az ÉS kapu segítségével a hamis működéseket megszüntetni. A meghatározott címek tiltása azért sem célravezető, mert egy változó címcsoportról van szó. Ha bővítjük a mikroszámítógép memóriáját és további, két vezérlőjeles SRAM-IC-et kapcsolunk a buszrendszerre, ismét meg kell határoznunk az új, tiltott címeket és ezek használatát sem engedhetjük meg.

### 4.3. Feladat

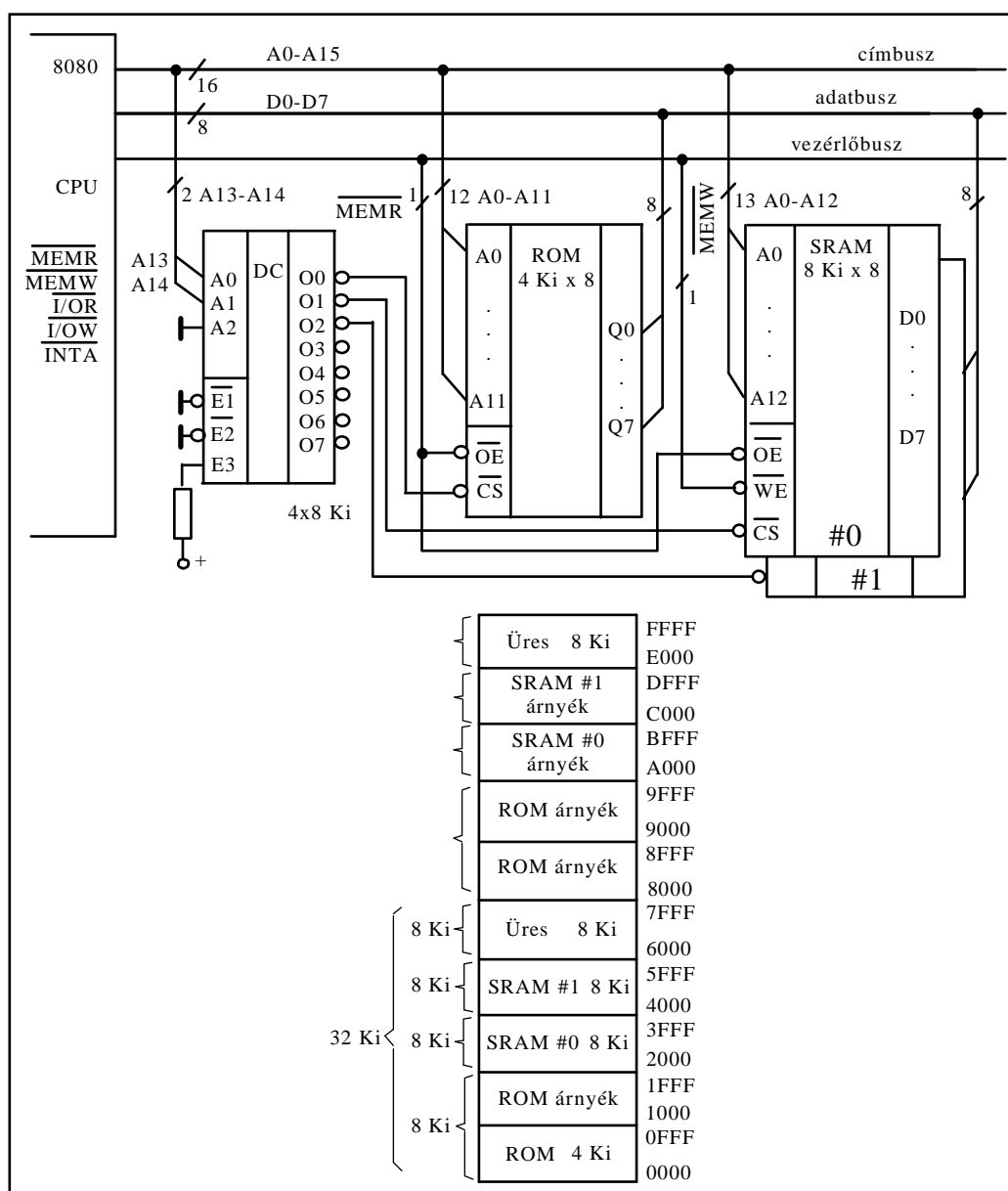
Egy Intel rendszerű mikroszámítógép buszrendszeréhez csatlakoztassunk egy darab 4 Ki x 8 kapacitású ROM-áramkört és két 8 Ki x 8 méretű SRAM-IC-t! A címkezelésre használjunk címdekódert!

A feladat legegyszerűbben úgy oldható meg, hogy a címdekóderrel 8 Ki méretű címtartományokat jelölünk ki. A 64 Ki teljes területen belül a 8 Ki méretű sávokat az A13 jelöli ki. Mivel négy ilyen sávot fogunk felhasználni, az A13 után következő A14 címbit is dekódoljuk, az A13, A14 bitpáros négy kombinációja négy darab 8 Ki méretű területet fog jelezni. Ezek közül egy a ROM-ot, egy-egy további a SRAM elemeket fogja aktivizálni.

Felvetődhet a kérdés, hogy melyik típusú memória legyen az alacsonyabb címterületen. A 8080 (és a Z80) mikroprocesszorok resetelés után a 0000 címről kezdenek utasításokat kiolvasni és értelmezni, ezért ezen a címen ROM-nak kell lennie. Ezért ezeknél a processzoroknál a címtartomány alsó felére helyezzük a ROM-okat, a magasabb címterületekre pedig a SRAM-okat.

Felrajzoljuk a buszrendszert, a jelleg és a kapacitás ismeretében a memória-áramköröket, a címdekódert, a memóriáknál az ökölszabályok alkalmazásával elvégezzük a bekötéseket, a címdekóderre rávezetjük az A13, A14 címbiteket, a kimeneteiről engedélyezzük a memória IC-ket (**4.10. ábra**).





4.10. ábra

A 4.10. ábrán nemcsak a kialakított áramkört mutatjuk be, hanem a címtérképet is. Innen leolvasható, hogy melyik memória-áramkört milyen címtartományon kezeli a mikroprocesszor. A 4 Ki x 8 kapacitású ROM a CPU 0000 ... 0FFF címtartományára került, az egyik 8 Ki x 8 méretű SRAM a 2000 ... 3FFF címekre, a másik a 4000 ... 5FFF területre.

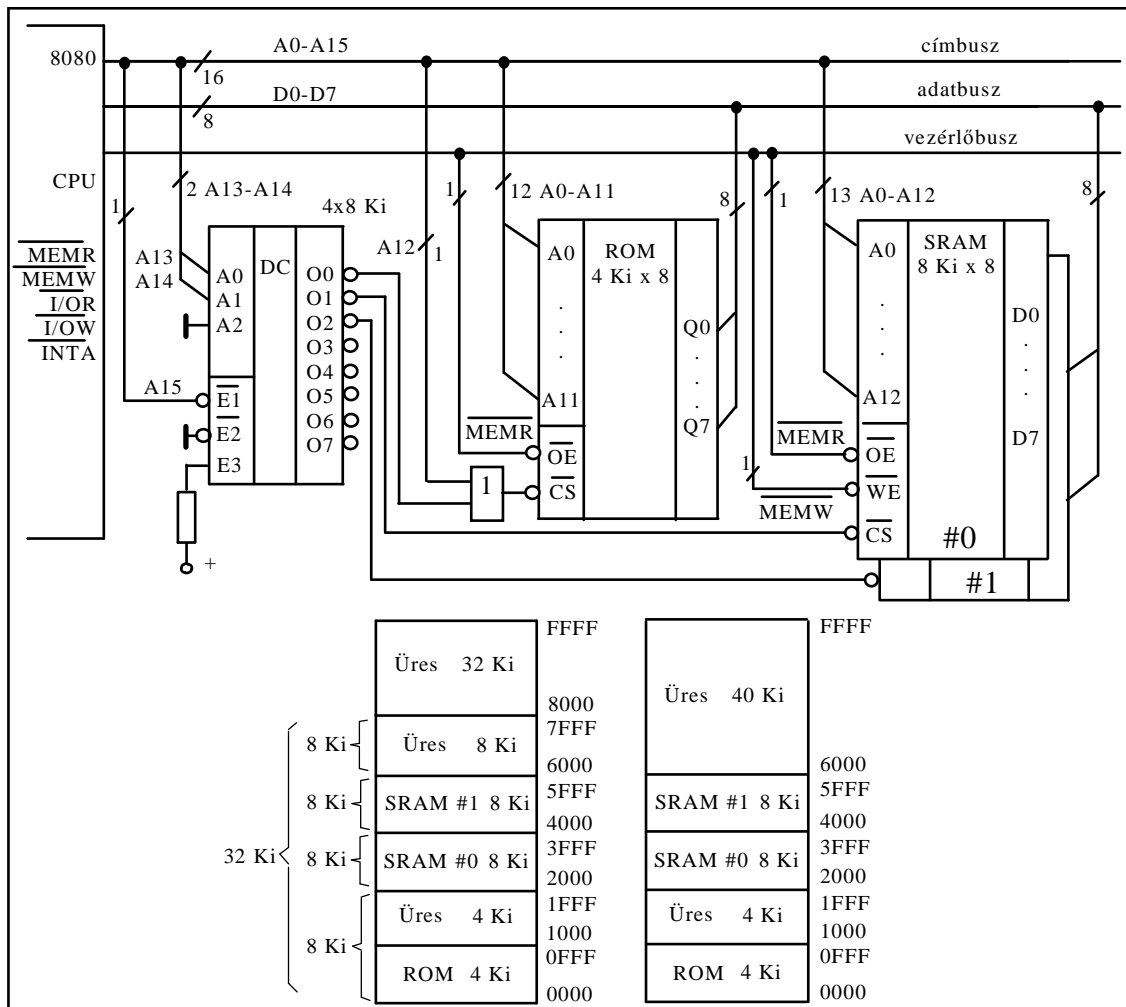
Az is látható azonban a címtérképen, hogy meglehetősen sok árnyék is jelentkezik. Az 1000 ... 1FFF területen a 4 Ki szószámú ROM árnyéka jelenik meg. Ennek az az oka, hogy a 4 Ki x 8 kapacitású ROM-áramkört a dekóder 8 Ki széles címtérületen aktivizálja. Ezen a területen kétszer jelenik meg a ROM, az alsót valódinak, a felsőt árnyéknak tekintjük.

A címdekóder 4 x 8 Ki területet, azaz 32 Ki területet kezel. Mivel az A15 0 értéke is 32 Ki-t jelöl ki és az 1 értéke is, és ezt a címbitet nem használjuk a dekódolásakor, a címdekóderrel kezelt teljes terület megismétlődik a felső 32 Ki sávban. Ott tehát kétszer jelenik meg a ROM árnyéka, és egyszer-egyszer a SRAM-áramköröké.

A kapcsolás árnyékmentesítése úgy történhet meg, hogy megszüntetjük az árnyékok okát. Az 1000 ... 1FFF területen nem jelenik meg a ROM, ha az engedélyezését a 8 Ki széles terület alsó felén engedjük csak hatni, a felsőn nem. A 8 Ki tartományt két 4 Ki méretű sávra osztja az A12, ennek 0 értékével engedélyezhetjük (egy VAGY kapun át) a ROM kijelölését, az 1 értékével nem. Ez a VAGY kapu tehát megszünteti a széles kijelölés miatti árnyékot.

A másik árnyékot okozó hatás az, hogy a dekóder az A15 0 értéke mellett is aktív és az 1 értéke esetén is. Ha csak az A15 = 0 esetben működne, a felső 32 Ki területen nem jelennének meg az árnyékok. Ezt pedig egyszerűen elérhetjük, ha az A15 címbitet rávezetjük a címdekóder egyik alacsony aktív szintű engedélyező bemenetére.

A felsorolt megfontolások alapján készült el a **4. 11. ábrán** látható kapcsolás.



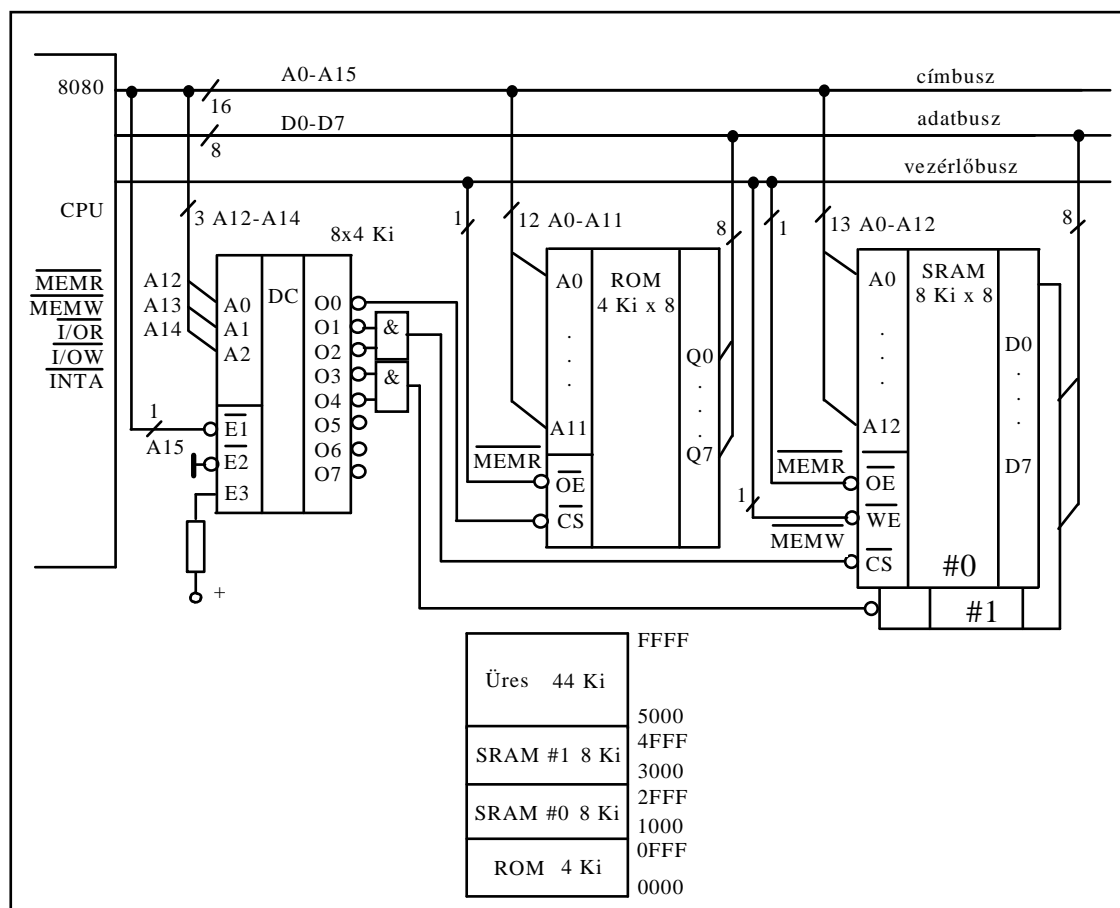
4.11. ábra

Az ábrán a címtérkép is látható. A bal oldali címtérképen látszik, hogy üressé vált az 1000 ... 1FFF (4 Ki méretű) terület és a felső 32 Ki rész. Mivel a felső 32 Ki terület és a közvetlenül alatta lévő 8 Ki (a 6000 ... 7FFF címek között) egyaránt üres, ezeket együttesen, mint 40 Ki üres területet célszerű ábrázolni. Ennek megfelelően rajzoltuk meg a jobb oldali címtérképet.

Pontosan ugyanezt a címtérképet valósíthattuk volna meg úgy is, ha a címdekóder engedélyezőjelét 0 V-on hagyjuk, az A15-öt pedig a címdekóder A2 bemenetére vezettük volna. Ekkor a dekóder 8 x 8 Ki területet állítana elő, ebből az alsó 3 kerülne felhasználásra. Az O0 kimenetnél természetesen továbbra is szükség lenne az A12-t fogadó VAGY kapura.

Mivel az eredeti kapcsolásban (4.10. ábra) az egyik árnyékot az okozta, hogy 8 Ki széles területen aktivizáltunk 4 Ki szószámú memóriát, felvetődhet az az ötlet, hogy a címdekóderrel 4 Ki széles területeket jelöljünk ki. Összesen 20 Ki a feldolgozott terület, azaz  $5 \times 4$  Ki, ezért a három bites dekóder elegendő lesz így is (hiszen nyolc darab 4 Ki területet fog kijelölni).

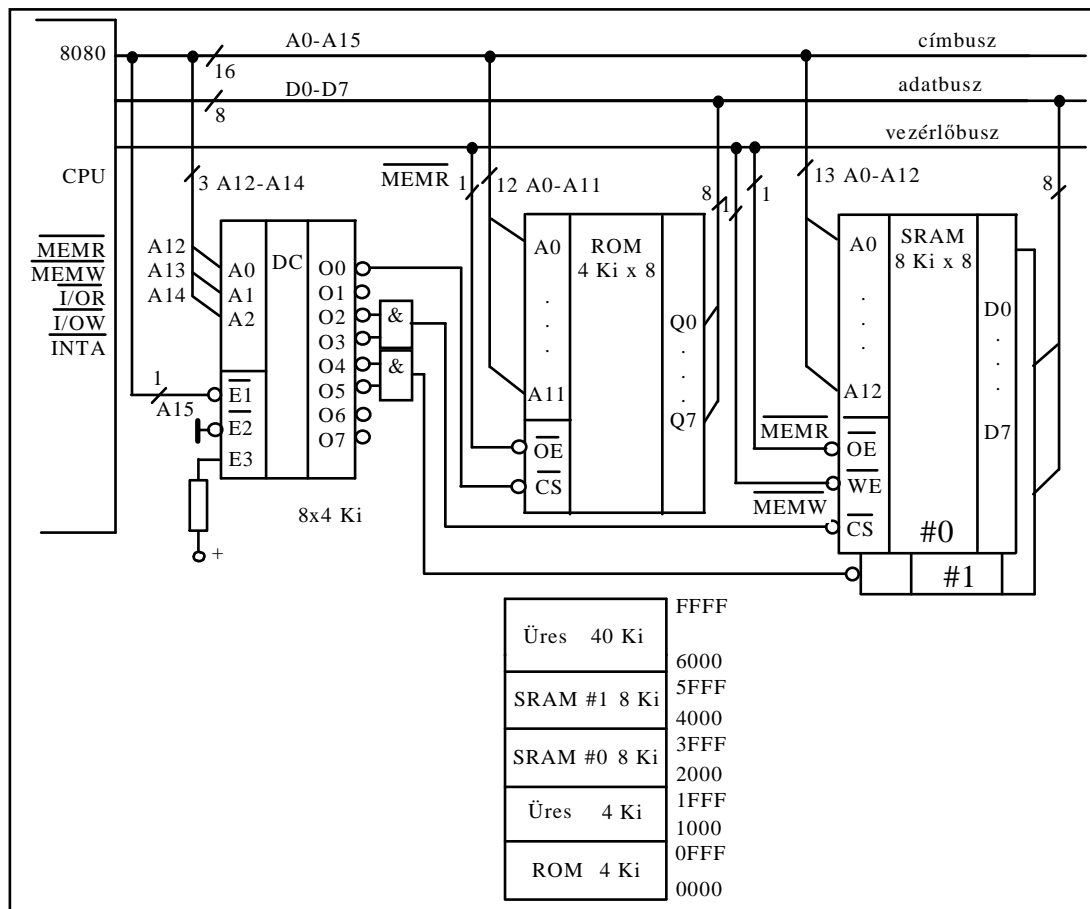
Ahhoz, hogy a címdekóder  $8 \times 4$  Ki kijelölést végezzen, az A12, A13, A14 címbiteket kell dekódolnia. Az A15-öt egy alacsony aktív szintű engedélyező bemenetre vezetjük, így a  $8 \times 4$  Ki, azaz 32 Ki nem fog megismétlődni a címtérkép felső 32 Ki területén. A ROM az O0 kimenetről vezérelhető, a SRAM-okhoz két-két kimenetet  $\overline{\text{ÉS}}$  kapuval egyesítenünk kell (mivel a kimenetek 4 Ki területet aktivizálnak, a SRAM-áramkörök pedig 8 Ki szószámúak). Az áramkört a 4.12. ábrán mutatjuk be.



4.12. ábra

Ennek a megoldásnak egyetlen apró szépséghibája van, ami nem okoz a működésben semmilyen hibát, mégis szót érdemel. Már említettük, hogy a memóriákat úgy szokás elrendezni a 64 Ki teljes címtérületen belül, hogy a memória szószámának megfelelően felosztjuk a területet és egy részterületet ezek közül használunk fel. Ha kapukkal kezeljük a címkijelölést, gyakorlatilag nem is tehetünk másként. Ha címdekódert használunk és azzal a mi memóriánknál kisebb területeket jelölünk ki, az összekapcsoláskor eltolva is elhelyezhető a memóriánk.

A 8 Ki méretű sávok „normál” elhelyezkedése a 4.8. ábrán pontosan megfigyelhető. Látható, hogy a 4.12. ábra kapcsolásában a két SRAM nem ilyen területeken kezelődik. Ha szeretnénk a helyükre tenni ezeket, akkor a címdekóder O1 kivezetését szabadon hagyjuk, s az O2, O3 illetve az O4, O5 segítségével működtetjük a 8 Ki x 8 kapacitású SRAM-IC-eket (4.13. ábra).



4.13. ábra

Általában a fejlesztők igyekeznek azonos típusú áramköröket felhasználni munkájuk során. Feltételezhető tehát, hogy jelenleg 4 Ki x 8 kapacitású ROM- és 8 Ki x 8 méretű SRAM-IC-k állnak rendelkezésre. Ha később a ROM-területet bővíteni szeretnénk, a most felszabadult területre gond nélkül be lehet helyezni egy újabb ROM-áramkört.

Ebben a feladatban a SRAM-áramköröknek három vezérlőjelük volt. Ha két vezérlőjeles SRAM-elemeket használtunk volna fel, a címdekóder elé el kellett volna helyezni a 4.9. ábrán szereplő ÉS kaput.

Zilog buszrendszer esetén a memóriák üzemmódvezérlését az  $\overline{RD}$  és a  $\overline{WR}$  jelekkel oldjuk meg. Ha két vezérlőjeles lenne a két SRAM-áramkör, a címdekóder egyik alacsony aktív szintű engedélyező bemenetére rá kellene vezetni a  $\overline{MREQ}$  jelet, így kerülhetnénk el a hamis olvasási műveleteket. Gyakorlásképpen megrajzolhatja ezt a kapcsolást!

#### 4.4. Feladat

8080 mikroprocesszorra épülő CPU buszrendszerére tervezzünk árnyékmentes memóriát, 2 darab 2 Ki x 8 ROM- és 4 darab 2 Ki x 4 SRAM-áramkört felhasználva! A SRAM-IC-k legyenek három vezérlőjelesek, a ROM-áramkörök címterülete a 0000 címmel kezdődjön!

Mielőtt a rajzoláshoz fogna, elemezzük ki a feladatot!

A 2 Ki x 8 kapacitású ROM-áramkörök (ROM 1 és ROM 2) saját címbitjei: A0 ... A10.

A SRAM-IC-ekből először kettőt-kettőt össze kell kapcsolnunk, hogy így a 2 darab 2 Ki x 4 elemből egy darab 2 Ki x 8 áramkör alakuljon ki. Ilyen „egységet” kettőt tudunk felépíteni, hiszen összesen négy SRAM-áramkörünk van. Az egyik 2 Ki x 8 egységet a SRAM 1 és 2, a másikat a SRAM 3 és 4 alkotja. Az egységek címbitjei szintén ezek: A0 ... A10.

Összesen tehát négy darab 2 Ki széles területet kell a címdekóderrel előállítanunk, a memória áramkörök kijelöléséhez. Ehhez a dekóder bemeneteire az A11, A12 címbiteket vezetjük rá, így az első négy kimenet fog működni (O0 ... O3). Az előírásnak megfelelően a ROM 1 kijelölésére szolgál majd az O0, a ROM 2 működtetője az O1 lesz, a SRAM 1/2 párosé az O2, végül az O3 a SRAM 3/4 kettőst aktivizálja.

Az árnyékmentességet egyik oldalról az biztosítja, hogy minden áramkör szószáma 2 Ki és a dekóder is 2 Ki széles területeket állít elő, másrészt a magasabb címbitek (A13, A14, A15) 0 értékét egy VAGY kapun keresztül az egyik alacsony aktív szintű engedélyező bemenetre fogjuk vezetni, így nem ismétlődik meg ez a 16 Ki a címtérképen.

Itt rajzolja meg a kapcsolási rajzot!

Az eredményt a 10. fejezetben ellenőrizheti.

#### 4.5. Feladat

Ezt a feladatot házi feladatként kell megoldania, ezért az eredményét nem találja meg a 10. fejezetben!

Egy 8080 mikroprocesszorra épülő CPU buszrendszerére tervezzen memóriaelrendezést! A felhasználásra kerülő memória-áramkörök:

1 darab 8 Ki x 8 ROM,  
három darab 2 Ki x 8 SRAM,  
utóbbiak két vezérlőjelek.

A memóriarendszer legyen árnyékmentes, a processzornak megfelelően a 0000 címnél ROM memória helyezkedjen el, a címkezelést címdekóderrel valósítsa meg!

Ha elkészült a kapcsolási rajz, egy másik rajzot is készítsen, ahol a CPU Z80 mikroprocesszorra épül!

Először gyakorlásképpen rajzolja ide a feladatban szereplő IC-eket, a címdekódert, a 8 Ki x 8 ROM-ot és a 2 Ki x 8 SRAM-ot (két vezérlőjellel)!

Most rajzolja ide a memóriarendszer címtérképét!

Itt készítse el a kapcsolási rajzot, egyelőre a 8080-ra épülő CPU kerüljön a rajzra a buszrendszerével, majd gondosan eltervezve helyezze el az áramköri elemeket, az ökölszabályoknak megfelelően készítse el az alap-bekötéseket, végül a címezést és a vezérlést is rajzolja meg!

Itt is rajzolja meg a kapcsolási rajzot, de most a CPU a Z80 mikroprocesszorból legyen kialakítva. Ne feledje, hogy a címzés és az adatkezelés nem változik, csak a vezérlési megoldások!



## 5. I/O-elemek csatlakoztatása buszrendszerhez

Kicsit hosszasan tűnhet ennek a fejezetnek a bevezetése, de érdemes végigolvasni és megérteni, mert csak így lesz érthető az I/O-rendszerek működése! Ahhoz, hogy az I/O-elemek használatát megértse, nagyon fontos, hogy a memória-áramkörök alkalmazásának minden kérdésével tisztában legyen. Ezért ezt a bevezetést a memóriakezelés összefoglalásával célszerű kezdeni.

A memória-áramkörök mindig nagyszámú, általában sok ezer rekeszt tartalmaznak. A rekeszek egyformák, a bitszámukat nevezzük szóhosszúságnak. A nyolcbites processzorok környezetében általában nyolcbites szóhosszúságú (bájt szélességű) memória IC-eket alkalmazunk, ha rövidebb a szóhosszúság, szóhossz irányában bővítenünk kell az áramkört. Ha a szóhosszúság csak 4 bit, két IC-t kell együtt használnunk, a kétbites szóhosszúságúakból négyet. A memória-áramkör a mikroszámítógépben vagy inaktív, azaz nincs kijelölve (ilyenkor általában a tápáram-felvétele is lecsökken, stand-by állapotba kerül), vagy aktív. Az aktív állapotot a CPU váltja ki. Az aktív állapot alapfeltétele, hogy a memória-áramkör általános engedélyezőjele, a  $\overline{CS}$  aktív, azaz L szintű legyen. Aktív állapotban lehet kezdeményezni a memória-áramkör működési módjait. A ROM áramkörök egyetlen működési módja az olvasás, a SRAM áramköröket olvasni vagy írni lehet. A ROM áramköröknél az olvasási üzemmódot kiváltó  $\overline{OE} = L$  jel a kimeneti fokozatot aktív állapotba kapcsolja, a kimeneti pontokon ( $Q_0 \dots Q_7$ ) megjelenik a kiolvasott rekesz tartalma. Ha a ROM nincs kijelölve ( $\overline{CS} = H$ ) vagy nincs kiválasztva az olvasási mód ( $\overline{OE} = H$ ), a kimeneti pontjai nagy impedanciás állapotban vannak, lebegnek.

A SRAM esetében az üzemmódokat két különálló jel is kezelheti (három vezérlőjeles változat), vagy egy összevont (két vezérlőjeles megoldás). A három vezérlőjeles kivitelnél a kimenetet engedélyező jel ( $\overline{OE}$ ) tartozik az olvasási módhoz, ugyanúgy működik, mint a ROM áramkörök esetében. Az írási üzemmódot a  $\overline{WE}$  jellel lehet kezdeményezni. Ha  $\overline{WE} = L$ , akkor az írási üzemmódba kerülünk, a SRAM adatpontjai ( $D_0 \dots D_7$ ) bemeneti jellegűek, az oda vezetett bitértékeket a SRAM befogadja. Ha a  $\overline{WE} = H$ , nincs írási működés, a SRAM adatpontjain esetlegesen megjelenő jelek nem jutnak be az áramkörbe, az adatpontok állapotát ilyen esetben az  $\overline{OE}$  jelértéke határozza meg. Természetesen a SRAM környezetét úgy kell megtervezni, hogy az  $\overline{OE}$  és a  $\overline{WE}$  egyidejűleg ne lehessenek aktív (L) szintűek.

Más a helyzet a két vezérlőjeles SRAM áramkörökkel. Ezek, ha a  $\overline{CS} = L$ , mindenképpen működnek, a működési módot a másik vezérlőjel, a  $R/\overline{W}$  állapota határozza meg. Ha ez a jel L szintű, az írási működés megy végbe, ahogyan fentebb bemutattuk. Ha ez a vezérlőjel H szintű, a kimeneti fokozat aktivizálódik, az adatpontokon megjelenik a megcímezett rekesz tartalma: kiolvasás történik.

A memória-áramkörök címezése két részből áll. A memória IC kapacitásának megfelelő számú saját címbemenettel rendelkezik, pl. a 2 Ki x 8 ROM áramkör a 2048 rekeszének megfelelően 11 címbemenettel ( $A_0 \dots A_{10}$ ). Ennek a memória-áramkörnek így 000 ... 7FF a címtartománya, amit saját címtartománynak fogunk nevezni, mert a mikroszámítógépben a CPU többnyire ettől eltérő címterületen kezeli az áramkört. Ezek a címbemenetek választanak ki a rekeszekből egyet. A saját címbemenetek minden esetben a címbusz azonos elnevezésű vezetékcsoportjára vannak kötve, ennél az áramkörnél tehát a címbusz  $A_0 \dots A_{10}$  vezetékeire.

A mikroszámítógépben kezelt címterület azonban többnyire nagyobb, mint a memória-áramkör saját címterülete. Nyolcbites mikroprocesszorok esetében a CPU címterülete 64 Ki. A 2 Ki széles címtartomány a 64 Ki címterületen éppen 65536 különböző módon helyezhető el. Néhány lehetséges, 2 Ki szélességű tartomány:

0000 ... 07FF,  
0001 ... 0800,  
0002 ... 0801,  
0003 ... 0802 stb.

A 65536 elhelyezési lehetőség közül azokat szokás a 2 Ki szószámú memóriához felhasználni, amelyek a 64 Ki 32 felé való felosztásával keletkeznek ( $32 \times 2 \text{ Ki} = 64 \text{ Ki}$ ). Ezekből néhány:

0000 ... 07FF,  
0800 ... 0FFF,  
1000 ... 17FF,  
1800 ... 1FFF,  
2000 ... 27FF stb.

A memória-áramkör számára a mikroszámítógép címtérképén olyan széles címterületet kell lefoglalni, amekkora a memória IC címtartománya. Ha a példánkban szereplő 2 Ki x 8 ROM memóriát használjuk ebben a mikrogépben, valahogyan meg kell határoznunk, hogy a 64 Ki-n belül melyik 2 Ki méretű címtartományban, melyik 2 Ki szélességű címterületen lehessen elérni, kiolvasni. Ezt a feladatot oldjuk meg a címdekódolással. A címdekódoló kimenőjelét a memória IC általános engedélyező bemenetére, a  $\overline{CS}$  pontra vezetjük.

Egy memória-áramkör esetében tehát a címzés két részből áll. Egyrészt a memória szószámának megfelelő területet címdekódolóval kijelöljük, mégpedig úgy, hogy a lehetséges címterületen belül a kívánt helyre kerüljön, s ezzel a jellel engedélyezzük a memória működését. A saját címbemenetek pedig meghatározzák, hogy a rekeszek közül melyiket kell pillanatnyilag használni.

Most pedig végre rátérünk az I/O-rendszerek működésére. A mikroszámítógépek bemeneti és kimeneti működését soros vagy párhuzamos adatkezeléssel lehet megoldani, mi egyelőre a párhuzamos adatkezelésű elemeket tárgyaljuk. Egy ilyen áramkör a nyolcbites adat minden bitjét egyidejűleg, párhuzamosan tudja kezelni.

Egy I/O-egység is igényelhet több belső címet. Tanulmányaink során találkozunk majd a mikroprocesszorok programozható kiegészítőivel, amelyekben több belső adattároló egység, regiszter is található. Egyes programozható kiegészítőknél címbemeneteket alakítottak ki, azokkal lehet a regiszterek valamelyikét kiválasztani. Ilyen esetben a címterületen annyi címet kell az áramkör számára biztosítani, ahányat az használ.

Ebben a jegyzetben ezekkel az összetett egységekkel nem foglalkozunk. A bemeneti illetve kimeneti működést portokkal fogjuk megvalósítani (amelyekről tudjuk, hogy párhuzamos portok). A port egyetlen nyolcbites adat kivitelére (kimeneti port) vagy befogadására (bemeneti port) szolgál.

A mikroszámítógépben bemeneti port is lehet egynél több, és kimeneti is. A portok közötti választás is címzéssel történik, de mivel egy porton belül további megcímezendő részek nincsenek, egy porthoz egyetlen cím elegendő.

A címzésen kívül vezérlőjelekre is szükség van. Mint már láttuk, az Intel jellegű buszrendszerénél, a 8080 mikroprocesszor alkalmazásakor az I/O-működéseket jelző vezérlőjelek a következők:

- I/O-olvasáskor az  $\overline{I/OR}$  aktív, azaz L szintű,
- I/O-íráskor az  $\overline{I/OW}$  aktív, tehát L értékű.

A Zilog jellegű vezérlőjelekből a Z80 mikroprocesszornál kettő-kettő aktív egyszerre:

- I/O-olvasáskor az  $\overline{IORQ}$  és az  $\overline{RD}$  aktív, azaz L szintű,
- I/O-íráskor az  $\overline{IORQ}$  és a  $\overline{WR}$  aktív, tehát L értékű.

Egy bemeneti/kimeneti-elemnél, röviden I/O-elemnél két kijelölő, működtető jelet célszerű használni. Az egyik fogadja a címdekódolás eredményét, azaz kijelöli az áramkört, a másik a vezérlőbuszról származó működtető jelek eredményeképpen engedélyezi a működést.

A bemeneti port, ha nincs aktivizálva (vagy a címzés nem ennek az elemnek szól, vagy a vezérlőjelek nem aktívak, vagy egyik feltétel sem teljesül), akkor a port kimenetei, amik az adatbuszra csatlakoznak, nagyimpedanciás állapotúak, lebegnek. Ilyenkor ugyanis egyéb elemek használják az adatbuszt, az adatforgalmat nem zavarhatja a bemenőport kimenőértéke. Ha a cím a bemeneti port címe, és a vezérlőjelek aktivizálják a portot, a külvilágból származó adatot átvezeti magán és az adatbuszhoz továbbítja (a kívülről érkező adat számára ebben az időszakban a bemenőport átlátszóvá válik).

A kimenőport, amikor a címzés neki szól és a vezérlőjelek is aktivizálják, az adatbuszról befogadja a CPU által oda helyezett kimeneti adatot. Amikor megszűnik az aktivizálása, a kimeneti pontjain akkor is tovább őrzi ezt a legutolsó kimeneti adatot, mert a kimeneti portban mindig találunk tárolót is.

A bemeneti port és a kimeneti port különféle áramkörökkel megépíthető. Tanulmányaink során megismerkedtünk egy IC-vel, amelyik mindkét feladatot el tudja látni, ez a 74LS412, régebbi nevén 8212. Ez az egyirányú, nyolcbites, háromállapotú kimenetekkel rendelkező, sokoldalúan vezérelhető latch (D tárolósor) az MD jelének 0 V-ra kötésekor bemeneti portként működik, az MD = 1 hatására kimeneti portként (5.1. ábra).

A bemeneti port esetén a 8212 bemenetei a külvilágból fogadják a mikroszámítógéphez érkező nyolcbites bemenőjelet, az IC kimenetei az adatbuszra vannak kötve. Két szabad vezérlőjele van, a  $\overline{DS1}$  és a DS2. A

$\overline{DS1}$  aktív szintje alacsony, a DS2-é magas. A bemeneti port akkor aktív, akkor vezeti át magán a bemenőadatot az adatbuszhoz, ha ez a két jel egyszerre aktív szintű, azaz ha  $\overline{DS1} = L$  és  $DS2 = H$ . A két vezérlő bemenet egyikére a címdekódoló kimenetét vezetjük majd, a másikra a működtető vezérlőjelet

vagy vezérlőjel-kombinációt. A kimeneti portnál a 8212 bemenetei csatlakoznak az adatbuszra, a kimenetein jelenik meg a külvilág számára a kimenő jel, amit folyamatosan, belső tárolójából vezet ide, s csak akkor változik, ha a CPU új kimenetírás művelettel módosítja. Ez a kimenetírás művelet akkor valósul meg, ha a két vezérlőjel aktív szintű. Itt is az egyik vezérlőjellel a címdekóder, a másikkal a CPU vezérlőjeleit szoktuk fogadni. Mindig ügyelnünk kell majd arra, hogy az egyik vezérlőjel (a  $\overline{DS1}$ ) alacsony aktív szintű, a másik (a DS2) magasan aktív.

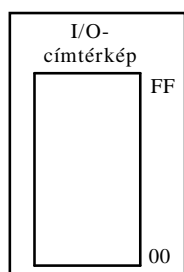
Úgy is használhatók a portok, ha a két szabad vezérlőjelük egyikét állandó aktív szintre kötjük, s a másikra összekapuzzuk a címdekóder és a vezérlőbusz jeleit.

A bemeneti/kimeneti-egységeket, portokat a bemeneti illetve kimeneti utasítások végrehajtása során használja a mikroprocesszor. A 8080 utasításkészletében egy-egy ilyen utasítás van csak. A bemenetet olvasó utasítás:

IN n ,

A kimenetre író utasítás pedig

OUT n .



5.2. ábra

Az utasításokban szereplő  $n$  a bemeneti illetve kimeneti port címe, ami a 8080 (és a Z80) esetében nyolcbites. Eszerint ezeknél a mikroprocesszoroknál az I/O címzés nyolc biten valósul meg, az I/O címek nyolcbitesek. Az I/O működés címhasználata is követhető címtérképen, de ez a nyolcbites címnek megfelelően csak a 00 címtől terjed az FF címig (5.2. ábra). Ezután a memóriát működtető vezérlőjelekkel és 16 bites címekkel kezelt elemeket a 0000 ... FFFF terjedelmű memóriacímtérképen jelenítjük majd meg, az I/O-eszközöket működtető jelekkel aktivizálható elemeket pedig a 00 ... FF címtartományú I/O-címtérképen. Az eddig rajzolt címtérképeink mind memóriacímtérképek voltak!

Az I/O-elemek címkezelésére ugyanazok az elvek vonatkoznak, mint a memóriákéra. A fő különbség az, hogy egy kimeneti vagy bemeneti port egyetlen címet köt le, egyetlen címre van szüksége. Ezt nyolcbites teljes címdekódolással lehet kijelölni. Ha így járunk el, az I/O-címtérképen egyetlen cím lesz, ami a porthoz tartozik.

Ha egyszerűbben oldjuk meg a címdekódolást, nem mind a 8 bitet használjuk fel a dekóderben, egynél több cím is aktivizálja a portot. Ilyen esetben a legkisebb címet ezek közül valódinak tekintjük, a többi címen a port árnyéka jelenik meg. Természetesen a CPU ezek közül bármelyikén képes működtetni a portot.

Ha Zilog rendszerű buszhoz csatlakoztatunk I/O-elemet, a címbusz alsó nyolc bitjéről lehet lecsatlakozni a címdekódoláshoz. Ha azonban 8080 mikroprocesszorra épül a CPU, a nyolcbites I/O cím nemcsak a címbusz alsó felén, az A0 ... A7 címbiteken jelenik meg, hanem egyúttal a felső nyolc vezetéken (A8 ... A15) is! I/O-olvasás vagy I/O-írás gépi ciklusban tehát ugyanaz az érték van az A0 címvezetéken mint az A8-on, szintén megegyezik az A1 és az A9 tartalma stb. A címdekóder tehát akár az A0 ... A7, akár az A8 ... A15 címvezetésekre is csatlakozhat. Így már bizonyára érthetőbb az, hogyan alakultak ki a 4. fejezetben leírt hamis SRAM működési címek.

### 5.1. Feladat

Az **5.3. ábrán** három portot csatlakoztattunk egy 8080 alapú mikroszámítógép buszrendszeréhez. Egy bemeneti és egy kimeneti portot azonos címről, címdekóderrel jelöltünk ki, egy további kimeneti portot pedig közvetlenül, egy címvezetékekkel. Vizsgáljuk meg, hogyan alakul ennél az áramkörnél az I/O működés!

A kimeneti port és a bemeneti port ugyanúgy jelenik meg a kapcsolásban, mint az 5.1. ábrán. A kimeneti port esetében a 8212 bemenetei kapcsolódnak az adatbuszra, a bemeneti portnál a 8212 kimeneti pontjai. A portok jellegét az MD bemenet 0-ra vagy 1-re kötésével határoztuk meg.

A #1 sorszámú bemeneti és kimeneti port azonos címképzéssel működik. Az A0 ... A2 (vagy az A8 ... A10) címbiteket a címbuszról egy címdekóderbe vezettük, ez nyolcfelé szétkódolta a címkombinációkat. Mivel ezek a legkisebb címbitek, egy-egy kimenet egyetlen címet jelent, a címdekóder 8 x 1 méretű címtérületet használ fel. Az O4 kimenőjelet használtuk fel a portok működtetésére, azaz a 04-es címet. A címdekóder kimenetét a 8212 IC-k  $\overline{DSI}$  vezérlőjeleként használtuk fel. Ha további címbiteket nem vennénk figyelembe, a címdekóder által kezelt 8 cím (8 x 1) az I/O-címtérképen 32-szer ismétlődne meg, ennyiszor jelenne meg a #1 bemeneti és kimeneti port árnyéka is. Ezeket az árnyékokat úgy kerültük el, hogy a címdekóder egyik alacsony szinten aktív engedélyező bemenetére egy VAGY kaput helyeztünk, amelybe a magasabb címbiteket (A3 ... A7 vagy A11 ... A15) mind bevezettük.

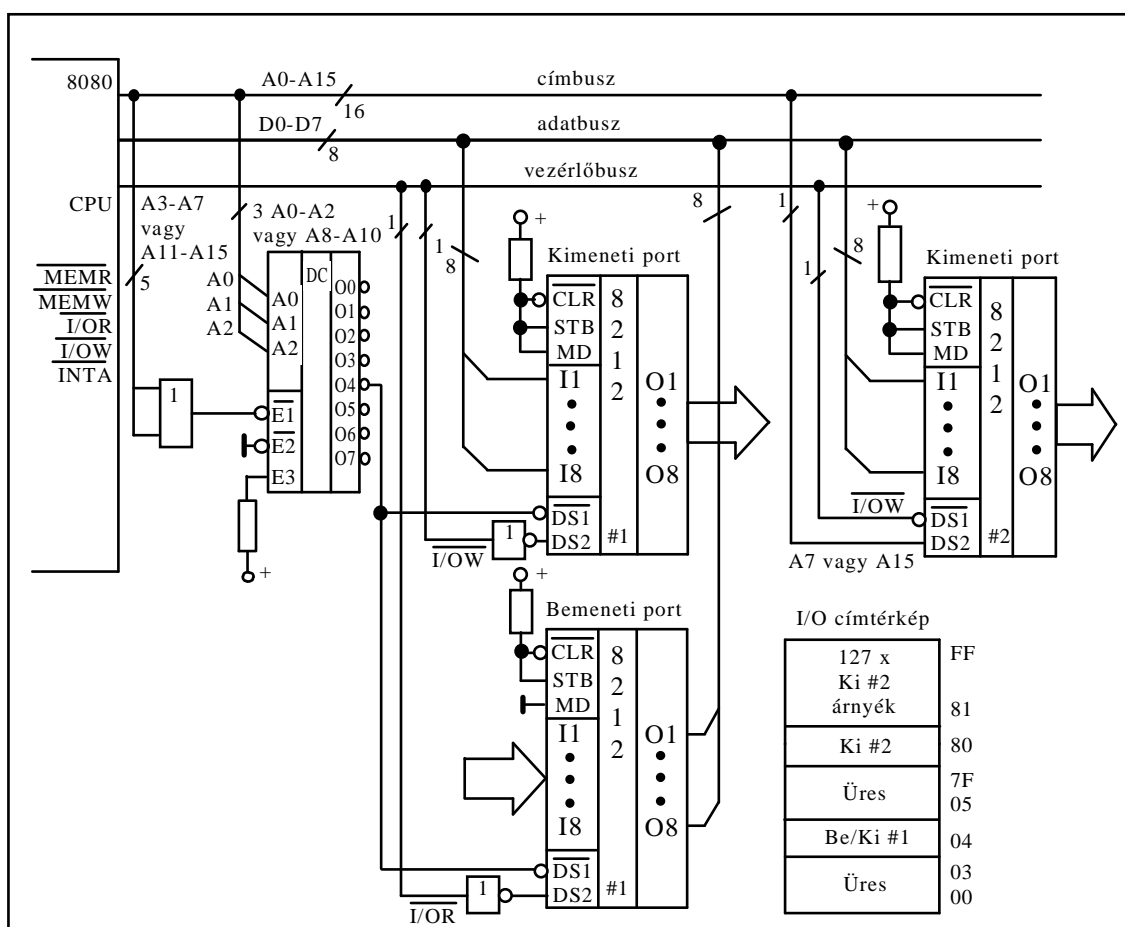
A portokat a vezérlőbuszról az I/O-kezelő vezérlőjelekkel működtetjük, a kimenetét az  $\overline{I/OW}$  vezérlőjellel, a bemenetét az  $\overline{I/OR}$  jellel. Ezeket a jeleket a 8212 áramkörök DS2 bemenetére vezettük rá. Mivel ez a bemenet magas szinten aktív, a vezérlőjelek útjába egy-egy invertert is el kellett helyeznünk.

Más, egyszerűbb címkezeléssel oldottuk meg a másik kimeneti port kijelölését. Itt az A7 címvezeték (vagy az A15) önmagában van felhasználva, azt vezettük rá a 8212 DS2 bemenetére. Mivel ez magas szinten aktív engedélyezőjel, akkor működik majd ez a kimeneti port, ha az A7 (illetve az A15) címvezetéken 1 érték van.

Az A7 (A15) a legmagasabb helyértékű címbit az I/O címben, ez tehát felezi a teljes címezési tartományt. A 00 ... FF címtérület alsó felén (00 ... 7F) 0 értékű, felső felén (80 ... FF) 1 értékű ez a bit. A most alkalmazott kapcsolási megoldásban tehát a #2 kimeneti port a teljes 80 ... FF címtartományban működni fog.

A #2 kimenőportnál a  $\overline{DSI}$  maradt szabadon a működési módot vezérlő jel számára, ezért erre a pontra kötöttük rá az  $\overline{I/OW}$  jelet a vezérlőbuszról. Mivel ez a bemenet alacsony szinten aktív, itt nincs szükség inverterre.

A teljes áramkört az 5.3. ábra mutatja be. Az ábrán megtalálható a címtérkép is, ami ebben az esetben természetesen a nyolcbites címekkel kezelt elemekre vonatkozó I/O-címtérkép, 00 ... FF címhatárokkal. Az alsó négy cím üres, a 04 címen működik a #1 bemeneti és kimeneti port, a következő 124 címhely ismét üres, majd a felső 128-on mindegyiken működik a #2 kimeneti port. A 80 címet tekintjük valódi címének, a többi 127 pedig a #2 kimeneti port árnyéka. A CPU a 80 ... FF címek bármelyikét használhatja a #2 kimeneti portra történő adatkivitelhez.



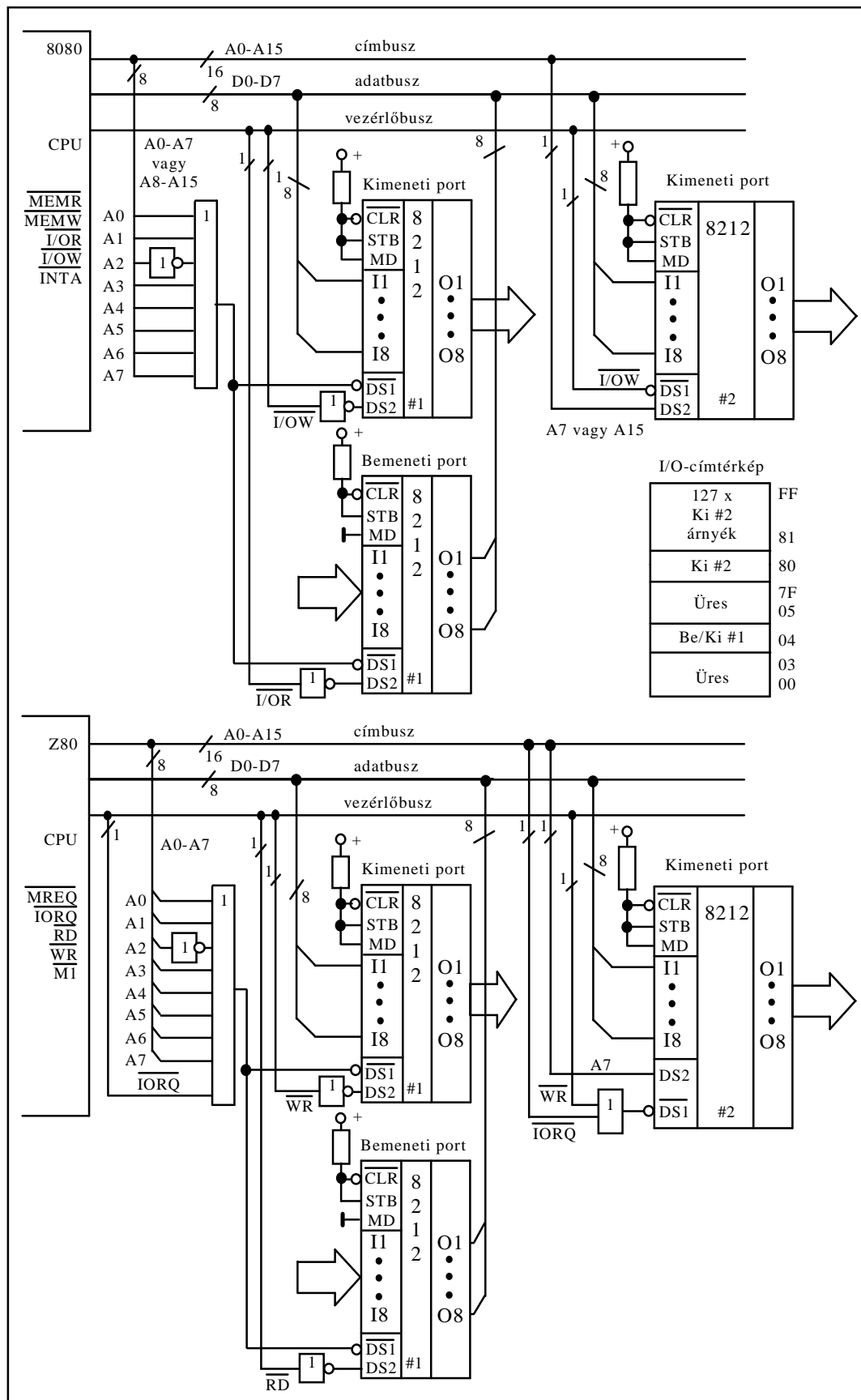
5.3. ábra

Ennek az elrendezésnek természetesen nincs logikus magyarázata, a kétféle címkezelési lehetőséget kívántuk csak bemutatni a segítségével. Ha árnyékmentes I/O működésre van szükség, a címdekóder valamelyik szabad kimenetét kell egy inverteren keresztül rávezetni a #2 kimeneti port DS2 pontjára, az A7 helyére, a kiválasztott címen fog a port működni.

A címdekóder helyett is használhatunk kapukat. Még teljes (árnyékmentes) címkezeléskor is csak nyolcbemenetű kapura van szükség, hiszen az I/O címek nyolcbitesek. Az 5.3. ábrán lévő címdekóder három bitet kezel, a többi 0 értékét egy VAGY kapuval vettük figyelembe. A kiválasztott címben (04) a bitek többsége 0 értékű (a bináris alakja 00000100), ezért egyetlen VAGY kapuval megoldható a fogadásuk, az A2 bitet inverteren át vezetjük az egyik bemenetére. A 0 értékek VAGY kapuval történő összegyűjtése a kimeneten alacsony aktív szintű működtető jelet eredményez, tehát közvetlenül rávezethető a portok  $\overline{DS1}$  bemenetére. Természetesen az I/O-címtérkép nem változott meg, hiszen a címzési rendszeren nem módosítottunk, csak más áramköri egységekkel valósítottuk meg. Az átalakított kapcsolási rajz látható az 5.4. ábra felső felén.

Az 5.4. ábra alsó felén ugyanez a megoldás szerepel, de a CPU egy Z80 mikroprocesszorra épül. A címzési rendszer semmit sem változik a processzorok cseréje miatt, de a vezérlőjelek újszerűek.

Minden működési fázist két egyidejűleg aktív vezérlőjel jelez a Z80 esetében, ahogyan ezt korábban bemutattuk. Az írási illetve olvasási jelleget a  $\overline{WR}$  illetve az  $\overline{RD}$  jel közvetíti, azt pedig, hogy I/O-elemet kíván írni vagy olvasni a mikroprocesszor, az  $\overline{IORQ} = L$  jelzi. A #1 portoknál az  $\overline{I/OW}$  jel helyére vezetjük a  $\overline{WR}$  jelet, az  $\overline{I/OR}$  jel helyére az  $\overline{RD}$  jelet, majd a címkezelő VAGY kapu egy kilencedik bemenetéhez kötjük az  $\overline{IORQ}$  jelet. Ez a megoldás szerepel az 5.4. ábra alsó felén.



5.4. ábra

A #2 kimeneti portnál más megoldást kellett választanunk, mert itt eddig nem szerepelt VAGY kapu. A  $\overline{WR}$  jelet itt VAGY kapun át vezettük a 8212 áramkörökhöz, s ennek a másik bemenetéhez kötöttük az  $\overline{IORQ}$  jelet.

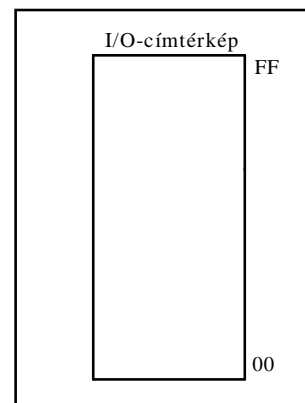
Az Intel és a Zilog jellegű buszrendszer használatakor arra is ügyelni kell, hogy az Intel 8080 az I/O-egység címét a címbusz alsó és felső bájtján egyaránt kiküldi, a Z80 azonban nem. Az 5.4. ábra alsó felén tehát ezért nem szerepelnek vagyilagosan a címbusz felső felének jelei!

## 5.2. Feladat

Most Ön fog megrajzolni egy I/O-rendszert! A 8080 mikroprocesszorra épülő CPU buszrendszerére csatlakoztasson egy kimeneti és egy bemeneti portot úgy, hogy mindkettő az I/O-címtérképen, az E0 címen működjön, árnyékmentesen! A címkiválasztást kapuval oldja meg!

Mint az 5.1. példában is láthatta, egy bemeneti és egy kimeneti port elhelyezkedhet azonos címen, hiszen egyidejűleg nem lehet aktivizálni ezeket. A működtető feltételeik egyidejűleg semmilyen körülmények között nem alakulnak ki! Ugyanakkor az ilyen megoldás egyszerűsíti a címezést, a kivitelezett áramköröknél jelentős megtakarítást jelent.

Először az I/O-címtérképen mutassa meg a portok elhelyezkedését! Az I/O-címtérképek alapját az **5.5. ábrán látja**, ezt felhasználva rajzolja meg az Ön rendszerének I/O-címtérképét, benne a bemeneti és a kimeneti port elhelyezkedését!



5.5. ábra

Ha kapuval oldjuk meg a címkiválasztást, célszerű a címet binárisan felírni! A feladatban a portok címe E0, azaz kettes számrendszerben:

11100000.

Ha a címbitek elnevezéseit is felírjuk a bitek fölé, máris látható, hogyan kell a címbiteket fogadó VAGY kapunál eljárunk, azaz melyik címbitet lehet közvetlenül rávezetni a VAGY kapura, melyiknél kell inverter:

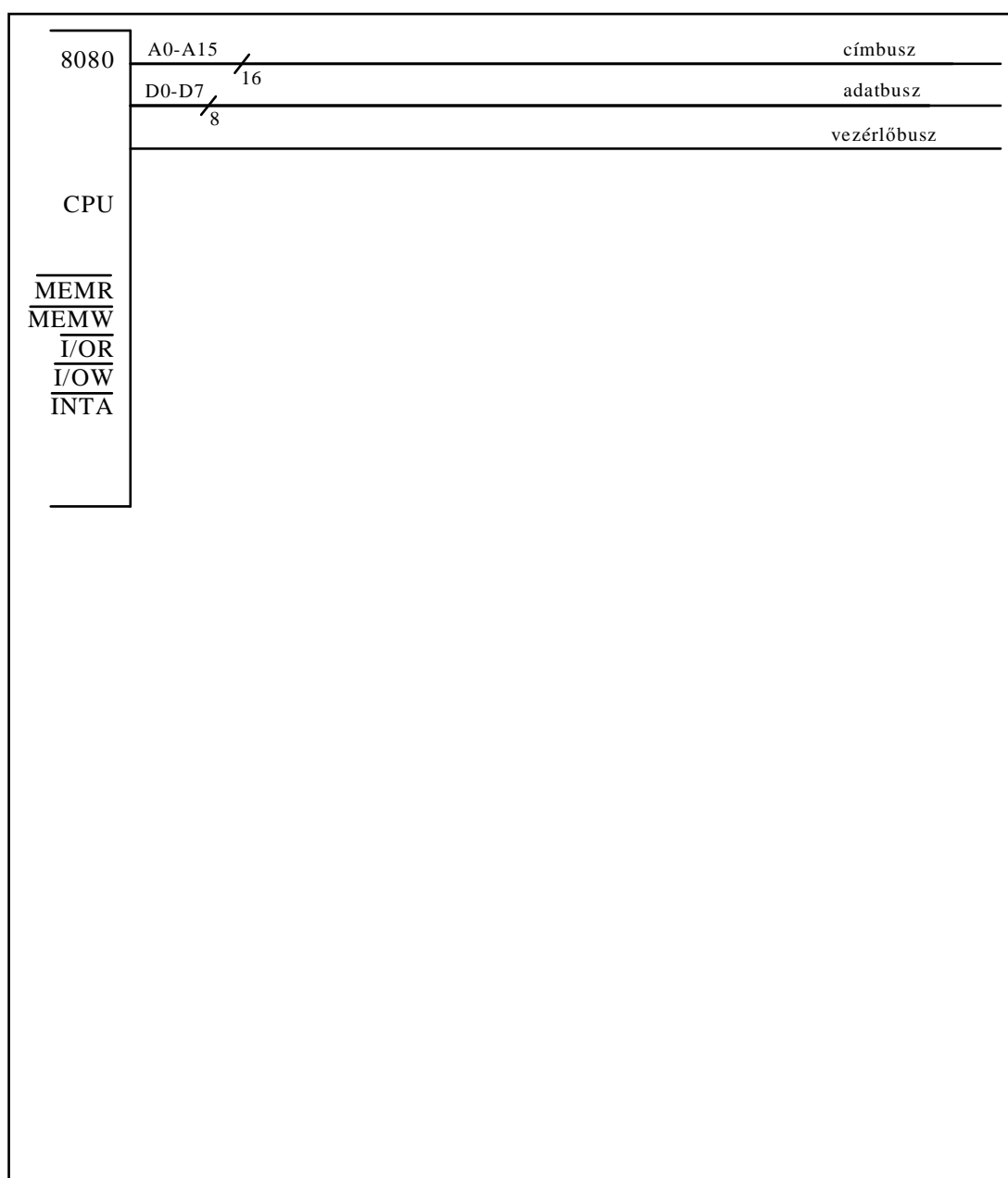
A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	0	0	0	0	0

Írja fel ide, melyik címvezetékkel kell inverteren át fogadnia!

A kapcsolási rajzhoz előre megrajzoltuk a CPU-t és a buszokat (**5.6. ábra**). Először a két portot rajzolja be, az ábra jobb oldalára, a kettőt egymás alá. Talán már „puskázás” nélkül is sikerül a 8212-ből kialakított kimeneti és bemeneti port ábrázolása. Ha mégsem, nyugodtan lapozzon előre és keresse meg ezeket a rajzokat! De ha fejből dolgozik, akkor is ellenőrizze a munkáját az 5.1. ábra alapján!

A következő elem a címeket kezelő VAGY kapu legyen, a portoktól balra elhelyezve. Az előző oldalon meghatározta, melyik címbitekkel kell inverteren át fogadnia, a többi közvetlenül bekötheti.

Harmadik lépésként kösse össze a címdekódoló VAGY kapu kimenetét és a portokat, majd a vezérlőbuszról vezesse le a megfelelő vezérlőjeleket és azokat is csatlakoztassa. A  $\overline{DS1}$  és DS2 bemenetek használatától függ, hogy azok előtt szükséges-e inverter. Kösse a címdekódoló VAGY kapu kimenetét a  $\overline{DS1}$  pontokra, s a CPU vezérlőjeleit a DS2 bemenetekre! A kész kapcsolást ellenőrizheti a 10. fejezetben.



5.6. ábra



### 5.3. Feladat

Ismét egy házi feladat! Először 8080-hoz, majd Z80-hoz rajzolja meg azt az I/O-rendszert, melyben négy darab bemeneti port és négy darab kimeneti port működik, árnyékmentesen! A címkezelést címdekóderrel oldja meg! Az I/O-címtérkép következő címeit használja fel:

FC, FD, FE, FF .

Elsőként készüljön el az I/O-címtérkép!

Rajzolja meg a kapcsolást 8080 mikroprocesszorra épülő CPU-val!

Most rajzolja meg a kapcsolást Z80 mikroprocesszorra épülő CPU használatával!

## 6. A négyféle tervezési lehetőség: memóriába ágyazott memória és I/O, I/O-ba ágyazott memória és I/O

Az eddigiekben természetesnek tartottuk, hogy a memória-áramköröket a CPU memóriavezérlő jeleivel kezeltük, s azok a memória-címtérképen jelentek meg, az I/O-elemek pedig, melyeket az I/O-jelek aktivizáltak, az I/O-címtérképen szerepeltek. A továbbiakban a memória-címtérképen megjelenő memóriákat *memóriába ágyazott memóriának* nevezzük, az I/O-címtérképen ábrázolható I/O-elemeket pedig *I/O-ba ágyazott I/O-nak*. Azért használjuk majd ezeket az elnevezéseket, mert további lehetőségek is adódnak a memóriák és az I/O-elemek kezelésére!

Miért jelentek meg az eddig beépített memória-áramkörök a memória-címtérképen? Azért, mert a memóriát működtető vezérlőjelekkel aktivizáltuk őket! Ha egy áramkört a CPU memóriát működtető vezérlőjeleivel aktivizálunk, akkor az áramkört memóriába ágyasztuk! Ez egyrészt azt jelenti, hogy az áramkör a memória-címtérképen fog megjelenni, másrészt azt, hogy a memóriákhoz kialakított teljes 16 bites címbusz használhatjuk a címezésükre a nyolcbites mikroprocesszoroknál (nagyobb szóhosszúságú processzorok esetében a címbusz is szélesebb, több vezetékből áll). Ezért rajzoljuk a memória-címtérképet 64 Ki terjedelműre, a 0000 kezdőcímtől az FFFF záró címig, s ezért jelennek meg ezek az áramköri elemek ezen a címtérképen.

A memória-működtető vezérlőjeleit a mikroszámítógép a memóriát kezelő működési egységeiben, gépi ciklusaiiban használja, aktivizálja. Ezek a gépi ciklusok a következők:

- M1 gépi ciklus,
- memóriaolvasás gépi ciklus,
- memóriaírás gépi ciklus,
- veremolvasás gépi ciklus,
- veremírás gépi ciklus.

Az M1 gépi ciklus az utasítások opkódjának beolvasására szolgál. A 8080 minden utasításának egybájtos az opkódja, tehát egy utasítás végrehajtása során egy M1 gépi ciklus van. A Z80 utasításai egy részének két, sőt három bájt méretű az opkódja, ezért itt több M1 gépi ciklus is következhet egymás után. Az M1 gépi ciklus automatikusan zajlik le, a CPU (a felhasználótól, a programtól függetlenül) a programmemóriát a teljes 16 bites címbuszon címzi és közben memória-működtető vezérlőjeleket küld ki.

Az általános memóriaolvasás gépi ciklus szolgál egyrészt a hosszabb utasítások opkód utáni bájtjainak a beolvasására, valamint a memóriareferenciás utasítások végrehajtása közben szükséges memóriatartalom-kiolvasások elvégzésére. Memóriareferenciásnak nevezzük azokat az utasításokat, amelyek végrehajtása során (az utasítás beolvasását követően) a CPU a memória valamelyik rekeszét olvassa vagy írja.

A 8080 esetén pl. a MOV A,M egy egyszerű memóriareferenciás utasítás. Az a feladata, hogy egy memóriarekesz tartalmát (amit egyébként a mikroprocesszor H és L belső regiszterének tartalma címez meg) az akkumulátorba helyezze. Ez tehát egy memóriareferenciás utasítás, mivel a végrehajtása közben memóriatartalmat kell olvasni. Az utasítás végrehajtása két gépi ciklusból áll. Az első egy M1 gépi ciklus, beolvasódik az opkód. A második egy memóriaolvasás gépi ciklus, a címbuszra kikerül a H,L regiszterpár tartalma, a vezérlőbuszon aktív (azaz L) szintre vált a MEMR vezérlőjel, a megcímzett memóriarekesz tartalmát az adatbuszról befogadja a CPU és az A (akkumulátor) regiszterbe helyezi. A mikroprocesszoroknak igen gazdag választéka van memóriareferenciás utasításokból.

A memóriaírás gépi ciklus csak a memóriareferenciás utasítások végrehajtása közben fordul elő, ha a végrehajtáshoz az szükséges, hogy a CPU adatot írjon ki valamelyik memóriarekeszbe. Az előzővel ellentétes hatású a MOV M,A utasítás, amelyik az akkumulátor (A) tartalmát a memória egyik (a H és L regiszterek tartalmával megcímzett) rekeszébe helyezi. Az utasítás végrehajtása most is két gépi ciklusból áll. Az első itt is egy M1 gépi ciklus, ekkor beolvasódik az opkód. A második egy memóriaírás gépi ciklus, a címbuszra kikerül a H,L regiszterpár tartalma, a vezérlőbuszon aktívvá válik a MEMW vezérlőjel, s az adatbuszra az akkumulátor tartalma.

A buszok ilyen állapotának hatására a memória a megcímezett rekeszébe befogadja az adatbusz tartalmát, azaz az akkumulátor tartalma beíródik a H,L tartalmával kiválasztott memóriarekeszbe. A mikroprocesszorok utasításkészletében nagyszámú olyan memóriareferenciás utasítást találunk, amelyek végrehajtásakor a memóriába ír a CPU.

A verem a memóriának egy logikailag elkülönített része, amit a mikroprocesszor egy külön címezéssel, a 16 bites veremmutatóval (SP) tud kezelni. A verembe utasításokkal is lehet írni, és utasításokkal is olvasható a tartalma, de a legfontosabb felhasználása automatikusan történik. A mikroprocesszor, ha szubrutinra lép, a visszatérési címet előzőleg a verembe helyezi. Amikor a szubrutin végéhez ér, a főprogramba úgy tér vissza, hogy a címet a veremből olvassa ki. Veremírásakor az SP tartalma kerül a címbuszra, a vezérlőbuszon aktivizálódik a  $\overline{\text{MEMW}}$  jel. Veremolvasáskor is az SP tartalma van a címbuszra, a vezérlőbuszon viszont a  $\overline{\text{MEMR}}$  jel aktív.

Az elmondottak alapján kiegészíthetjük a vezérlőjelekre vonatkozó összefoglalónkat!

A 8080-nál, az Intel rendszerű vezérlőbuszon a memóriát igénylő gépi ciklusokban:

- M1, memóriaolvasás, veremolvasás esetén a  $\overline{\text{MEMR}}$  aktív,
- memóriaírás, veremírás esetén a  $\overline{\text{MEMW}}$  aktív.

Tudjuk, hogy a Z80 mikroprocesszor más vezérlőjeleket használ. A memóriát működtető gépi ciklusok itt így alakulnak:

- M1, memóriaolvasás, veremolvasás esetén a  $\overline{\text{MREQ}}$  és az  $\overline{\text{RD}}$  aktív,
- memóriaírás, veremírás esetében a  $\overline{\text{MREQ}}$  és a  $\overline{\text{WR}}$  aktív.

Sok esetben az M1, a memóriaolvasás és a veremolvasás gépi ciklusokat egyszerűsítve, összefoglalva memóriaolvasási ciklusoknak nevezzük, mivel a buszrendszerre csatlakozó külső elemek szempontjából a különbségtételnek nincs jelentősége. Hasonló módon a memóriaírás és a veremírás gépi ciklusokat is szokás összefoglalva memóriaírási ciklusoknak nevezni.

Mindezek a gépi ciklusok a teljes 16 bites címbuszt használják, az aktivizálódó áramkörök így a 0000 ... FFFF címhatárokkal rendelkező, 64 Ki méretű memória-címtérképen jelennek meg. Az így kezelt áramköri elemeket tekintjük tehát memóriába ágyazottaknak.

A másik lehetőség az I/O-ba ágyazás. Ha az áramköri elem az I/O-címtérképen jelenik meg, I/O-ba ágyazásról beszélünk. Hogyan jön létre ez a fajta működés?

A mikroprocesszoroknál többnyire találunk I/O-olvasási és I/O-írási gépi ciklusokat is (vannak mikroprocesszorok, amelyeknek nincsenek ilyen gépi ciklusaik, a 8080 és a Z80 rendelkezik ezekkel). Kizárólag az I/O-referenciás utasítások végrehajtása közben jelentkezik ezek a gépi ciklusok. Ha az utasítás azt igényli, hogy egy I/O-eszköztől bemenőadatot olvasson a processzor, akkor I/O-olvasási gépi ciklust hajt végre, ha egy I/O-eszközhöz kimenőadatot kell küldeni, az utasítás végrehajtásakor egy I/O-írás gépi ciklus kerül sorra. A processzorok utasításkészletében az I/O-referenciás utasítások készlete többnyire szegényes, a 8080 mikroprocesszornak mindössze két ilyen utasítása van, az IN n és az OUT n. Az n mindkét esetben az I/O-eszköz nyolcbites címe. Legyen ez pl. 2C (mivel cím, ez is hexadecimális érték természetesen).

Az IN 2C utasítás végrehajtásakor az első gépi ciklus M1, beolvassa az utasítás opkódját a CPU. Az utasítás azonban kétbájtos, a második bájttal a cím (ebben az esetben 2C). Ezt egy memóriaolvasás gépi ciklusban fogadja a processzor. Az utasítás harmadik gépi ciklusa egy I/O-olvasás gépi ciklus. Ebben a ciklusban a címbuszra (a 8080-nál megismert módon) az I/O-cím kettőzve jelenik meg, azaz a címérték 2C2C, a vezérlőbuszon pedig aktivizálódik az  $\overline{\text{I/OR}}$  jel. E jel és a cím alapján működésbe lép a megfelelő bemeneti eszköz és a külvilágból érkező bemenőadatot az adatbuszra helyezi, ahonnan a mikroprocesszor beolvassa azt.

Az OUT 2C utasítás végrehajtásakor az első két gépi ciklus (egy M1 és egy memóriaolvasás) az előzőekkel azonos módon folyik le. Az utasítás harmadik gépi ciklusa most egy I/O-írás gépi ciklus. Ebben a ciklusban is a címbuszra a 2C2C jelenik meg, a vezérlőbuszon pedig aktivizálódik az  $\overline{\text{I/OW}}$  jel. A megcímezett kimeneti elem befogadja az adatbuszról a kimenőadatot.

A 8080 mikroprocesszor I/O-elemeket működtető gépi ciklusaiban:

– I/O-olvasáskor az  $\overline{I/OR}$  aktív,

– I/O-íráskor az  $\overline{I/OW}$  aktív.

Ismételjük meg a Z80 I/O-működtető vezérlőjeleit is:

– I/O-olvasáskor az  $\overline{IORQ}$  és az  $\overline{RD}$  aktív,

– I/O-íráskor az  $\overline{IORQ}$  és a  $\overline{WR}$  aktív.

Az utóbb felsorolt gépi ciklusok, melyek I/O-működést kezdeményeznek, a 16 bites címbusznak csak az alsó felét használják, mivel az I/O címek nyolcbitesek, az aktivizálódó áramkörök így a 00 ... FF címhatárokkal rendelkező, 256 címértékű I/O-címtérképen jelennek meg. Az így kezelt áramköri elemeket tekintjük tehát I/O-ba ágyazottaknak.

Áttekintettük tehát a két, általunk ebben a jegyzetben kezelt mikroprocesszor esetére a memóriába ágyazás és az I/O-ba ágyazás lehetséges eseteit. Ez azonban a memória- és I/O-kezelésnek csak az egyik oldala. Arról ugyanis az egyes eseteknél nem szóltunk, hogy az aktivizálódó vezérlőjelek milyen áramköri elemet működtetnek. A természetes esetek azok, ha a memóriát működtető vezérlőjelekkel memória-áramkört aktivizálunk, azaz a memóriába ágyazott memória, valamint az I/O-ba ágyazott I/O, azaz az I/O-t működtető jelekkel történő I/O-aktivizálás.

Felhasználhatjuk azonban a memóriát működtető jeleket I/O-elemek vezérlésére is, így jutunk a *memóriába ágyazott I/O-áramkörökhöz*. Ha pedig az I/O-t működtető jelekkel memória-áramkört aktivizálunk, az *I/O-ba ágyazott memória* esetéhez jutunk.

A memóriába ágyazott memória minden mikroszámítógépben megjelenik, mivel az utasítások beolvasása és a veremkezelés megkívánja ezt. A memóriába ágyazott I/O előnye, hogy a bemeneti és kimeneti adatokat nemcsak egyszerűen beolvasni vagy kiírni lehet, hanem a sokoldalú memóriareferenciás utasításokkal is lehet kezelni. A memóriába ágyazott I/O egyik nehézsége a címkezelés. Mivel egy port egyetlen címet igényel, 16 bites teljes dekódolással lehet csak árnyékmentesen kezelni, ami bonyolult címdekódot jelent. Ha egyszerűbb címdekódolást alkalmaznak, akkor egy helyett több címet rendelnek az áramkörhöz, így a port nagyszámú árnyékkal jelenik meg a memória-címtérképen. A memóriába ágyazott összes elem a memória-címtérképen jelenik meg, címezésük a teljes 16 bites címbusszal valósul meg.

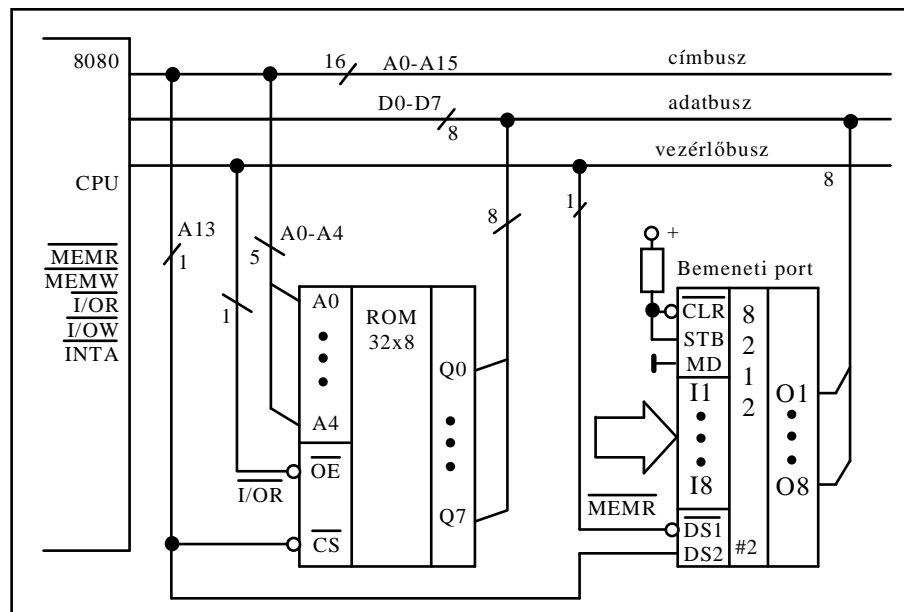
Az I/O-ba ágyazott I/O a mikroprocesszorok I/O-referenciás utasításaival kezelhető bemeneteket és kimeneteket jelent. Az I/O-ba ágyazott memóriákat is csak ezekkel az utasításokkal lehet elérni, ezért kiegészítő lehetőségként, pl. azonosító kódok tárolójaként szokták csak használni ezt a módszert. Mivel az I/O címezési lehetőség szűkös, ilyen módon csak szerény kapacitású memória-áramköröket lehet a mikroszámítógépbe beépíteni. Az I/O-ba ágyazott minden elem az I/O-címtérképen jelenik meg, címezésük alapvetően a címbusz alsó nyolc bitjével valósul meg.

Nem mindig áll rendelkezésre a négyféle bekötési mód. A Motorola mikroprocesszorainak pl. nincsenek I/O-referenciás utasításai, nincsenek I/O-vezérlő jelei. Ezért ezek mellett a processzorok mellett minden áramkör, akár memória, akár I/O-egység, memóriába ágyazottan használható csak.

A felsorolt esetek alapján látható, hogy a mikroprocesszor, a CPU csak a buszrendszert látja, az arra kapcsolódó áramköröket nem! Ha egy 8080 az IN n utasítás hatására egy I/O-egységről olvasni kíván, kiküldi a címét a címbuszon, alacsony szintre állítja az  $\overline{I/OR}$  jelet. De az már a mikroszámítógép tervezőjétől függ, hogy ez a buszállapot milyen áramköri egységet aktivizál, valóban egy bemeneti perifériát, esetleg egy kis kapacitású ROM- vagy SRAM-memóriát!

### 6.1. Feladat

A **6.1. ábrán** két áramkör csatlakozik egy 8080 alapú CPU buszrendszerére, egy 32 x 8 kapacitású ROM és egy bemeneti port. Vizsgáljuk meg, melyik elem hova van ágyazva, milyen címeken működteti azokat a CPU, vannak-e árnyékaik vagy árnyékmentes a kialakítás. Az eredményeink alapján rajzoljuk meg a megfelelő címtérképeket – a memóriába ágyazott elemeket a memória-címtérképen tüntessük fel, az I/O-ba ágyazottakat az I/O-címtérképen. Végül árnyékmentesítsük a kapcsolást!



6.1. ábra

Azt elég hamar fel lehet fedezni, hogy az eddigi két természetes beágyazás helyett itt a két különlegesen látjuk, az I/O-ba ágyazott memóriát és a memóriába ágyazott I/O-t. A címjelölés igen egyszerű, az A13 címvezeték választja ki mindkét áramkört. Vegyük részletesebben szemügyre a ROM-IC bekötését!

A7	A6	A5	I/O-címterkép	
1	1	1	Üres	FF E0 DF
1	1	0	ROM árnyék	C0 BF
1	0	1	Üres	A0 9F
1	0	0	ROM árnyék	80 7F
0	1	1	Üres	60 5F
0	1	0	ROM árnyék	40 3F
0	0	1	Üres	20 1F
0	0	0	ROM 32	00

6.2. ábra

A 32 x 8 kapacitásnak megfelelően 5 címbemenete van az áramkörnek (A0 ... A4), ezek a címbusz azonos elnevezésű pontjaihoz csatlakoznak, az adatpontok az adatbuszhoz (az ökölszabályoknak megfelelően). Mivel I/O-ba ágyazott memóriaként használjuk, az  $\overline{OE}$  pontra az  $\overline{I/OR}$  jelet vezettük.

Hogyan alakul a címkiválasztás? A ROM szószáma 32, az I/O-címterképen 256 címérték helyezkedik el. Az első feladatunk annak a megállapítása, hogy az I/O címekből melyik bit kezeli a ROM-ot! Ugyanis az eredeti I/O cím, mint tudjuk, csak nyolcbites (A0 ... A7). Mivel a 8080 ezt a nyolcbites címet a felső nyolc címbuszvezetéken is kiküldi, az alsó és a felső nyolc vezetékét megfeleltethetjük egymásnak:

$$\begin{matrix} A8 & A9 & A10 & A11 & A12 & A13 & A14 & A15 \\ A0 & A1 & A2 & A3 & A4 & A5 & A6 & A7 \end{matrix}$$

A megfeleltetésből látható, hogy az A13 címvezetéken ugyanaz az érték van I/O-működés esetén, mint az A5-ön. A ROM címzését tehát úgy kezelhetjük, mintha az A5 címbuszvezeték lenne rávezetve a  $\overline{CS}$  pontjára.

Az A5 az I/O-címterképen 32 címelem szélességű területeket jelöl ki, mégpedig 8-at ( $8 \times 32 = 256$ ). Mivel közvetlenül csatlakozik a  $\overline{CS}$  pontra, akkor működik a ROM, ha az A5 (azaz az A13) 0 értékű. A **6.2. ábrán** rajzoltuk meg az I/O-címterképet, feltüntettük az A5 értékeit is és a magasabb címbitek értékeit is, bejelöltük, melyik tartományokban működik a ROM, a legkisebb ilyen tartományt (00 ... 1F) valódi címzési területnek kezelve, a másik hármat árnyékként feltüntetve. A címterképen a fennmaradó területek, az a négy darab, 32 szélességű terület, ahol A5 (azaz A13) 1 értékű, üres bejegyzést kap, ott nem működik semmilyen áramkör.

A 32 x 8 kapacitású ROM-áramkör tehát I/O-ba van ágyazva. Címzése részleges, ezért árnyékokkal jelenik meg az I/O-címtérképen, a valódi elérhetősége: 00 ... 1F.

A 15	A 14	A 13	Memória- címtérkép	
1	1	1	8192 x port árnyék	FFFF E000 DFFF
1	1	0	8 Ki üres	C000 BFFF
1	0	1	8192 x port árnyék	A000 9FFF
1	0	0	8 Ki üres	8000 7FFF
0	1	1	8192 x port árnyék	6000 5FFF
0	1	0	8 Ki üres	4000 3FFF
0	0	1	8191 x port árnyék	2001
			Bemeneti port	2000
0	0	0	8 Ki üres	1FFF 0000

6.3. ábra

Most térjünk rá a bemeneti portra! A port a 8212 áramkörből a szokásos módon van kialakítva. Mivel a DS1 pontjára a MEMR vezérlőjel van rávezetve a vezérlőbuszról, a kimeneti port memóriába van ágyazva! A DS2 pontra kapcsolódik az A13, tehát ez oldja meg a 64 Ki címterületen való elhelyezést.

A memória-címtérképet az A13 8 Ki szélességű területekre bontja, méghozzá nyolcra (8 x 8 Ki = 64 Ki). Mivel az A13 közvetlenül csatlakozik a DS2 bemenetre, a bemeneti port akkor működik, ha az A13 = 1, azaz az ilyen tulajdonságú, 8 Ki széles címterületeken. Az első ilyen terület a 2000 ... 3FFF címhatárok között található. A bemeneti port valódi címe ezért a 2000, s itt rögtön árnyékok jelennek meg a 2001 ... 3FFF címtartományon, összesen 8191 darab. A feljebb lévő három darab 8 Ki széles sávon ismét a bemeneti port árnyékait látjuk, egy-egy sávban 8192 cím van, ezért 8192 árnyékot kellett a memória-címtérképre beírni. Azokon a 8 Ki széles területeken, ahol az A13 = 0, nem működik semmilyen áramkör, ezért került azokba az üres bejegyzés. A memória-címtérképet a **6.3. ábra** mutatja be. A memória-címtérképen az A13 és a magasabb (A14, A15) címbitek értékeit is feltüntettük.

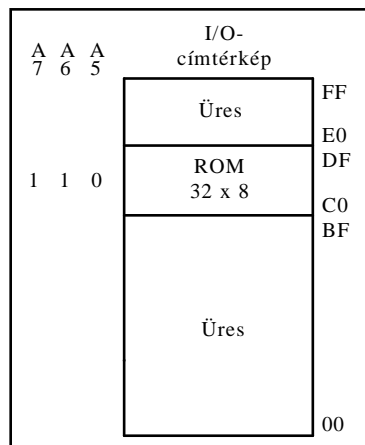
Hogyan lehet használni most a ROM-ot illetve a bemeneti portot? A ROM olvasásához az IN n utasítást kell a programban elhelyezni, címként megadva annak a rekesznek a címét, amelyiknek a tartalmára szükségünk van. Az IN 0A nyilvánvalóan a ROM 0A saját című rekeszének tartalmát fogja az adatbuszon megkapni, de ugyanezt a rekeszt kiolvashatjuk az IN 4A, az IN 8A vagy az IN CA utasításokkal is. Memóriaolvasással a ROM tartalmát most nem tudjuk elérni.

A bemeneti portot viszont úgy kezelhetjük, mint egy memóriarekeszt. A MOV A,M utasítás, ha a H,L regiszterpárban pl. a 2000 címet helyeztük el, az akkumulátorba helyezi a bemeneti porton éppen beérkező adatot. Ugyanez történik, ha a H,L tartalma 2001 vagy 3FFF vagy pl. FFFF. Pontosan így működik az IN n utasítás is, a megcímezett bemeneti porton lévő adatot az akkumulátorba teszi. De most sok más lehetőségünk is van! Ha pl. a MOV C,M utasítást használjuk, a bemenőport adata a C regiszterbe kerül! És tetszőleges másik memóriareferenciás utasítást is használhatunk, a bemeneti port valamelyik lehetséges címét beépítve, így közvetlenül különféle logikai és aritmetikai műveleteket lehet végezni a bemenő adattal, tetszőleges belső regiszterbe lehet helyezni. Mindezeket a műveleteket az IN n utasítás elvégzését követő további egy vagy két utasítással lehet csak megoldani I/O-ba ágyazott port esetén.

Következzék az árnyékmentesítés!

Az I/O-címterképen a 32 x 8 kapacitású ROM áramkör helyezkedik el. Igaz, hogy a címkijelölésére felhasznált A5 (A13) éppen 32 címhelynyi területet jelöl ki az I/O-címterképen, de nem egyet! Az A5 (A13) ugyanis az I/O-címterképen 4-szer lesz 0, 4-szer lesz 1 értékű, miközben a címek a 00 értéktől végigfutnak az FF értékig. Mivel a ROM-áramkört az A5 (A13) 0 értéke jelöli ki, a 6.2. ábra szerint összesen négyszer jelent meg az I/O-címterképen. A 6.2. ábrán az A5 (A13) értékének változásai mellett feltüntettük az A6 és az A7 (A14 és A15) címbitek értékeit is! Látható, hogy a négy címterület, ahol a ROM aktív, az A6 és az A7 segítségével megkülönböztethetők egymástól! A legalsón az A7A6 = 00, a következőnél A7A6 = 01, a harmadiknál A7A6 = 10, a legfelsőnél pedig A7A6 = 11.

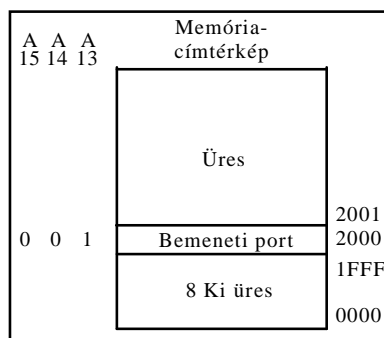
Válasszuk az utóbbi, legfelső területet azok közül, ahol eredetileg működött a ROM, és az árnyékmentes megoldásnál csak itt jelenjen meg! Az árnyékmentes kapcsolás I/O-címterképe tehát a **6.4. ábra** szerinti legyen!



6.4. ábra

A kapcsolatban a címkijelölést úgy kell elvégeznünk, hogy mindhárom címbitet figyelembe kell vennünk! Ha a VAGY kapus vezérlést használjuk (ami a 0 értékeket tudja összefogni), az 1 értékű címbiteket inverteren kell átvezetni. A ROM  $\overline{CS}$  bemenetéhez ezért egy VAGY kapu kerül, ami az A5 (A13) bitet közvetlenül fogadja, az A6 és az A7 (azaz az A14 és az A15) címbiteket pedig inverteren keresztül.

Hasonlóan lehet megoldani az árnyékmentesítést a bemeneti portnál is. A különbség csak annyi, hogy egy porthoz egyetlen címet kell rendelni, ezért az árnyékmentes címkiválasztás 16 bites címdekódolást kíván. Legyen a bemenőport címe 2000, a memória-címterkép pedig a 6.5. ábra szerinti.



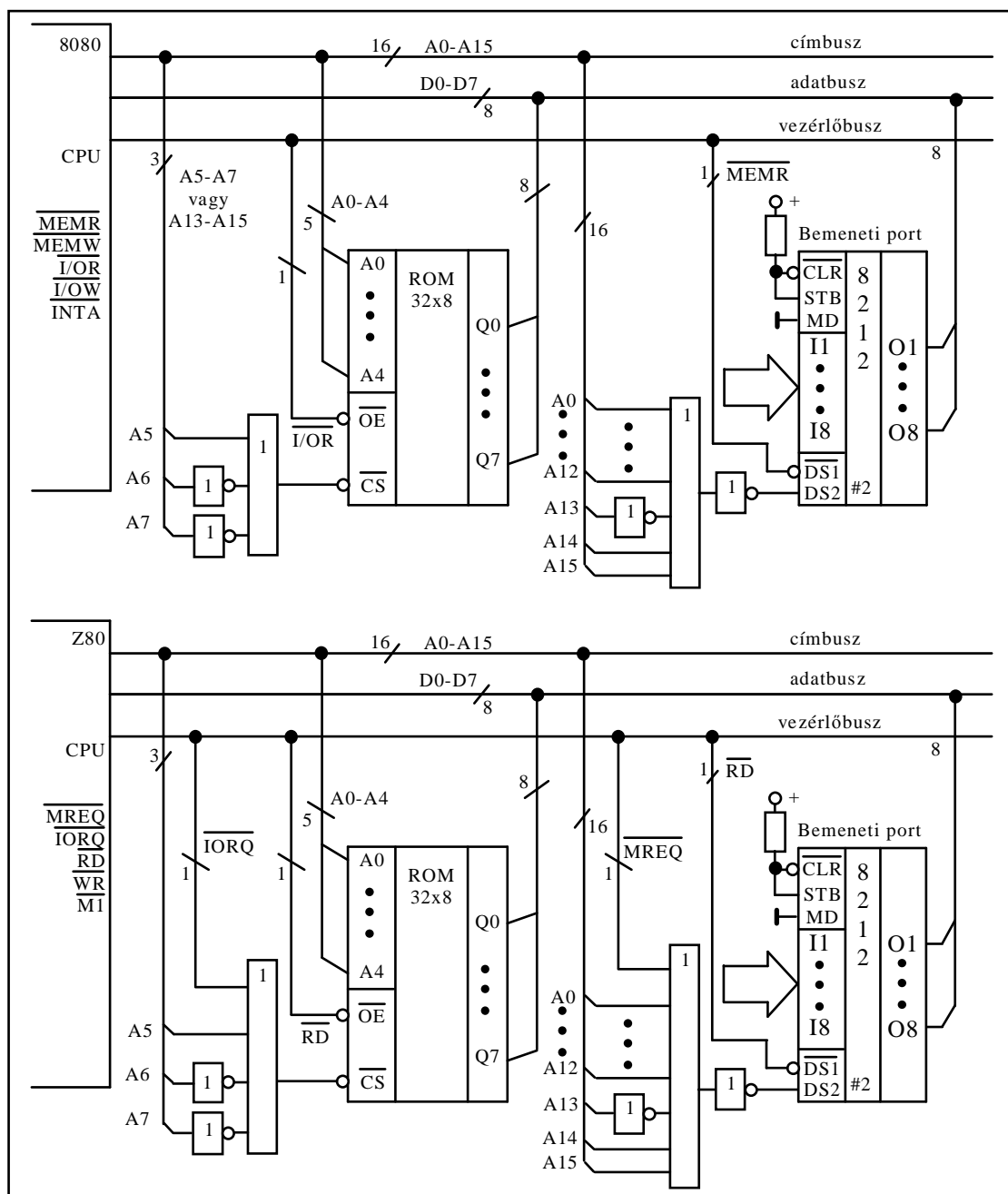
6.5. ábra

A bináris alakból leolvasható az egyes címbitek értéke:

0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0



A címbitek közül az A13 értéke 1, a többi 0. A címkijelölésre használt VAGY kapura tehát az A13-at kell inverteren keresztül bevezetni, a többit pedig közvetlenül köthetjük a kapura.



6.6. ábra

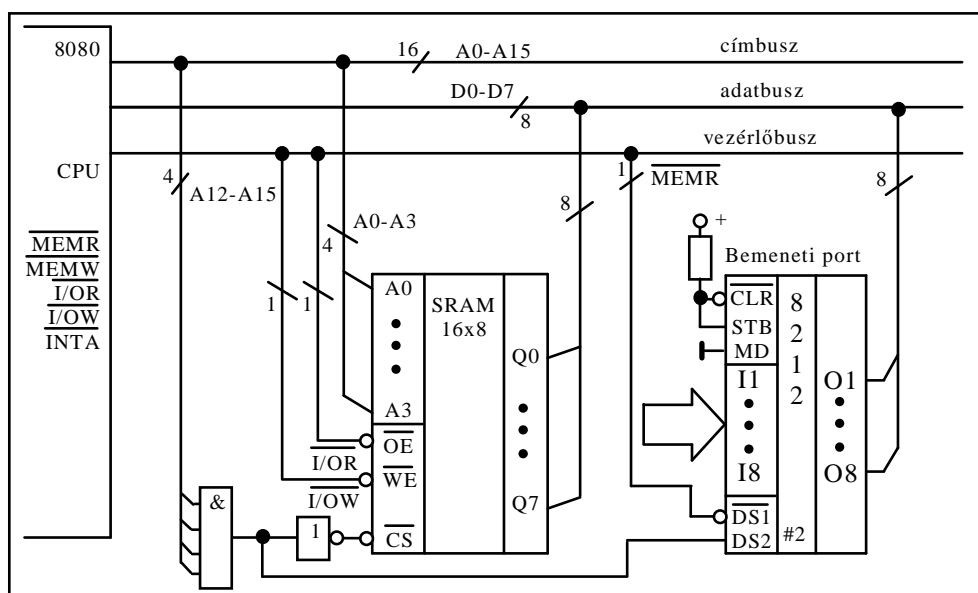
Az árnyékmentes kapcsolás a **6.6. ábra** felső részén látható, a címkijelölő áramkörök a megbeszéltek szerint készültek el.

Az ábra alsó részén olyan CPU mellé helyeztük ugyanezt az árnyékmentes memória- és I/O-rendszert, amelyik Z80 mikroprocesszorra épül. Szokás szerint a memória-áramkör saját címezése, az adatpontok csatlakoztatása, a címkijelölés változatlan, kivéve azt, hogy a Z80 esetében az I/O-elemek címezésére csak az A0 ... A7 bitek használhatók, az A8 ... A15 vezetékek nem. A vezérlőjeleket át kellett alakítani az új processzornak megfelelően.

Az üzemmódvezérlés az I/O-ba ágyazott ROM és a memóriába ágyazott port esetében is az  $\overline{RD}$  jellel valósul meg, hiszen a ROM működtetése is és a bemeneti port beolvasása is ezzel a vezérlőjellel aktivizálódik. Vagy ennek a jelnek az útjába kell egy-egy új VAGY kapu, amelyik a memória vagy az I/O kijelölésére szolgáló vezérlőjelet fogadja, vagy a már meglévő, a címszél szereplő VAGY kapuhoz kell az új jelet vezetni. A 6.6. ábra alsó részén az utóbbi megoldást választottuk. A ROM előtti VAGY kapura (mivel ez I/O-ba van ágyazva) az  $\overline{IORQ}$  jelet vezettük rá, a memóriába ágyazott port előtti VAGY kapura pedig a  $\overline{MREQ}$  jelet.

## 6.2. Feladat

Határozza meg, hogy a **6.7. ábrán** látható kapcsolásban melyik áramköri elem hova van ágyazva, milyen címeken érhető el, milyen jellegű (teljes vagy részleges) a címszél, s mindezek alapján rajzolja meg a mikroszámítógép felépítésének megfelelő címtérképeket!



6.7. ábra

Először ismerkedjen meg a buszrendszerre csatlakozó elemekkel, ellenőrizze a saját címbitek és az adatbitek bekötését, a port irányának helyességét!

Ezután vizsgálja meg, melyik hova van ágyazva!

A bemeneti port ágyazása:

A SRAM ágyazása:

Ezután a címkijelölést nézze meg! Az I/O cím nyolc bitje, mint tudja, a magasabb nyolc címvezetéken megismétlődik. Ennek alapján állapítsa meg, az I/O címben melyik bitek felelnek meg az  $\overline{ES}$  kapura vezetett négy bitnek? Ennek alapján számolja ki, melyik címterületet jelöli ki a kapu! Ügyeljen az  $\overline{ES}$  kaput fogadó bemenet aktív szintjére is!

Hasonlóan határozza meg, hogy a memória-címtérképen hol, mekkora, területet határoz meg az  $\overline{ES}$  kapu! Most is ügyeljen az  $\overline{ES}$  kaput fogadó bemenet aktív szintjére!

A memória kapacitását vesse össze annak a címterületnek a méretével, ahol aktivizálva van, állapítsa meg, ismétlődik-e a kijelölt területen a SRAM, lesz-e itt árnyéka?

Hasonló módon állapítsa meg, hogy a bemenőport mekkora címtérületen van aktivizálva, hány címet igényel a kezelése, s ezek alapján van-e a kijelölt címtartományon belül árnyék?

Az utolsó lépés annak megállapítása, hogy a magasabb címbitek figyelmen kívül hagyása eredményez-e további árnyékokat, akár a memória címtérületén, akár az I/O címtérületén. Mindezek alapján rajzolja meg a címtérképet! Az eredményét a 10. fejezetben ellenőrizheti!

### **6.3. Feladat**

Ismét olyan feladat következik, amit segítség nélkül kell kidolgoznia! Előbb 8080 mikroprocesszorra épülő CPU-hoz, majd Z80-as rendszerhez tervezze meg a buszokra csatlakozó memória- és I/O-rendszert az alábbiak szerint! Két darab, 16 x 4 kapacitású SRAM-memóriát kell I/O-ba ágyazottan, árnyékmentesen elhelyezni, a lehető legmagasabb címtartományon. Azután tervezzen egy bemeneti és egy kimeneti portot, memóriába ágyazottan, teljes dekódolással (azaz árnyékmentesen), a 8000 címre!

Először az I/O-ba ágyazott memóriával foglalkozzon!

Rajzolja meg a memória logikai rajzjelét!

Rajzolja meg az I/O-címtérképet, a feladatnak megfelelően helyezze el a memória-áramköröket!

A címkijelölést kapuval oldja meg, írja ide, milyen kapura lesz szükség, melyik címbiteket kell rávezetni, hol szükséges inverter!

Rajzolja ide a bemeneti és a kimeneti port logikai jelét, ha 8212-vel valósítja meg azokat!

Rajzolja meg a memória-címtérképet, a feladatnak megfelelően!

Az eddigieket felhasználva rajzolja fel a 8080-ra épülő CPU-t és buszrendszerét, majd a memória-áramköröket és a portokat!

Rajzolja meg az előző áramkört olyan CPU felhasználásával, amelyben Z80 mikroprocesszor működik!



## 7. Logikai kapcsolások tervezése címtérképek alapján

A 7. és a 8. fejezet tulajdonképpen az eddig tanultak összegzése. Ebben a fejezetben címtérképek alapján kell megrajzolnunk a mikroszámítógép memóriarendszerét és/vagy I/O-rendszerét. Ilyen jellegű részfeladataink eddig is voltak, egyszer-egyszer előzőleg is a címtérképből kiindulva rajzoltunk meg egy áramkört részletet. Most a címtérképek olvasásának lépéseit összegezzük, majd a kiolvasott információk módszeres felhasználását mutatjuk be.

Mindenekelőtt a feladat szövegezéséből olvassuk ki a mikroprocesszor típusát. Ha a feladat erre nem tér ki, tetszés szerinti típust választhatunk. Az elkészülő logikai kapcsolási rajz keretét, a CPU-t és a buszrendszert a processzor típusa határozza meg.

Ezt követően vegyük szemügyre a kiindulásként kapott címtérképeket. Lehet, hogy egyet, lehet, hogy kettőt kapunk meg. Ha mindkét címtérképet megadja a feladat, a kettő közül egyszerre csak egyet kell foglalkoznunk, gyakorlatilag a két címtérképet egymástól függetlenül lehet kezelni. Ha a címtérképen nem szerepel a memória-áramkörök szószáma, a megadott címértékek alapján határozzuk meg és írjuk be az értékeket. Ha a címtérképre beírták a szószámokat, de nem adták meg a címhatárokat, akkor pedig a címeket számoljuk ki, és írjuk fel az egyes címterületek kezdő- és végpontjához. Mindenesetre a következő vizsgálódásokat úgy kezdjük el, hogy a címtérképre legyenek beírva az egyes címterületek nagyságát jelző értékek is, és legyenek feltüntetve a kezdő- és a zárócímek is. A címértékek elsősorban akkor hasznosak, ha a cím kiválasztást kapukkal kívánjuk megoldani, de a címdekóder kezelésében is segítenek.

Vegyük tehát vizsgálat alá az egyik címtérképet. Az első, amit meg kell állapítanunk, hogy memória vagy I/O-címtérképről van-e szó. Ha megnevezték a címtérképet, természetesen nincs gondunk. Ha nem szerepel a címtérkép neve, akkor a címhatárok, címértékek segítenek az azonosításban, hiszen a tárgyalt mikroprocesszorok esetében a memória-címtérképen 16 bitesek (hexadecimálisan négyjegyűek) a címek, az I/O-címtérképen a nyolcbites címeket kétjegyű hexadecimális számok jelzik. Ha a memória-címtérkép van előttünk, a rajta ábrázolt elemek memóriába vannak ágyazva. Az I/O-címtérkép azokat az elemeket tartalmazza, amelyek I/O-ba vannak ágyazva.

A következő lépésben a címtérképen szereplő áramkörökkel ismerkedjünk meg! A memória-áramköröknél olvassuk le a kapacitást, majd vizsgáljuk meg, hogy a szószámnak megfelelő szélességű címtartománnyal van-e kijelölve vagy annál szélesebbel (utóbbi helyzetre az utal, ha a memória mellett azonnal az árnyéka is megjelenik a címtérképen)! Ha több memória is szerepel, mindet vizsgáljuk meg. Az eredményeink alapján meg tudjuk mondani, hogy címdekóderrel megoldható-e a Feladat. Akkor használhatunk címdekódert, ha a kijelölt címtartományok azonos méretűek és legalább 3–4 szerepel a kapcsolásban. Egyetlen áramkör kedvéért nem szoktunk címdekóder IC-t beépíteni, azt inkább kapuval kezeljük.

Ha címdekóder használata mellett döntöttünk, a kijelölésre kerülő címtartományok mérete és száma alapján meg tudjuk mondani, hogy melyik a legkisebb kezelt címbit és hogy hányat kell dekódolnunk.

Ezután azt nézzük meg a címtérképen, hogy a címdekóderrel működtetett elemek összessége megjelenik-e a címtérképen ismételt, mint árnyék. Ha nem, akkor a magasabb, a címdekóderrel nem kezelt címbiteket is fel kell dolgozni, a címdekódert ezek segítségével a címtérkép meghatározott területére kell helyezni.

Végül, ha SRAM memória is szerepel a feladatban, tisztázni kell a vezérlési megoldását. Ha a feladat nem köti ki a típust, célszerű a három vezérlőjeles megoldását választani. Ha két vezérlőjeles SRAM IC-t építünk be, akkor a címkijelölésnél (a kapunál vagy a dekódernél) meg kell oldani, hogy csak memória-kezelő gépi ciklusokban működjenek a memória-áramkörök!

Az I/O-elemek esetében azt kell szem előtt tartanunk, hogy egy portnak egyetlen címre van szüksége. Sőt, egy ilyen címen akár egy bemeneti és egy kimeneti port is elhelyezhető. Ha a címtérképen egy cím van a porthoz rendelve, akkor memóriába ágyazás esetén 16 bites, I/O-ba ágyazás esetén nyolcbites teljes címdekódolásra van szükség. Ha szélesebb területen jelenik meg a port, akkor valószínűleg címdekóder szerepel a kapcsolásban, s annak egy kimenete működteti ezeket az áramköröket.

A kiértékelő lépések után következhet az áramkör megrajzolása. A címdekóder általában a CPU közébe szoktuk helyezni. Az azonosan kezelt elemeket egymástól kis távolságra rajzoljuk meg, azaz a memóriába ágyazott áramköröket egy csoportba, s egy másik csoportba az I/O-ba ágyazott IC-ket.

A bekötések elkészítésekor első lépésben az ökölszabályok segítenek bennünket. Minden adatpontot az adatbuszra kapcsolunk, a memória-áramkörök saját címpontjait a címbusz megfelelő vezetékére.

A címkezelés megrajzolása után a kijelölő jeleket rávezetjük a memória IC-k  $\overline{CS}$  pontjára, a portok egyik engedélyező bemenetére (itt ügyelni kell az aktív szintre, szükség esetén invertert is használunk kell). Az üzemmódokat vezérlő jelek kiválasztásakor kell figyelniük arra, hogy az áramkör memóriába vagy I/O-ba van-e ágyazva.

### 7.1. Feladat

A **7.1. ábrán** látható memória-címterkép alapján tervezzük meg egy Intel 8080 mikroprocesszorra épülő mikroszámítógép memóriarendszerét!

A feladat egyértelműen kijelöli a mikroprocesszor típusát, tehát a logikai kapcsolást egy 8080-ra épülő CPU buszrendszerére kell rárajzolnunk. Csak egy memória-címterképet kaptunk, azaz minden áramkörünk memóriába lesz ágyazva. Nincsenek viszont megjelölve az egyes címterületek határcímei. Ezt kell először pótolnunk!

Azt tudjuk, hogy az 1 Ki címterület saját címhatárai 000 – 3FF, tehát az alsó területnél a kezdőcím (0000) után felírhatjuk a zárócímet is: 03FF. A következő, 2 Ki terület saját címhatárai 000 – 7FF, de most a kezdőcím 0400. A legegyszerűbb, ha a számoláskor 1 Ki címterületekkel lépkedünk feljebb. A 0400-nál 1 Ki-val feljebb lévő cím 07FF, a következő (0800) címnél 1 Ki-val magasabban lévő cím pedig 0BFF.

Memória-címterkép	
Üres	FFFF
	1800
2 Ki SRAM	17FF
	1000
1 Ki üres	0FFF
	0C00
2 Ki ROM2	0BFF
	0400
1 Ki ROM1	03FF
	0000

7.2. ábra

Memória-címterkép	
Üres	
2 Ki SRAM	
1 Ki üres	
2 Ki ROM2	
1 Ki ROM1	

7.1. ábra

Innen ismét 1 Ki-val kell felfelé számolnunk, a kezdőcím 0C00, a záró pedig 0FFF. A 2 Ki SRAM címterületénél a kezdőcím 1000, a záró pedig 2 Ki-val több, azaz 17FF. Az üres terület az 1800 címmel kezdődik, a teljes memória-címterkép zárócíme pedig FFFF. A címterkéket is feltüntető memória-címterkép a **7.2. ábrán** látható.

Vizsgáljuk meg, milyen áramköröket tartalmaz a címterkép! Csak memória-elemet látunk, ez tehát egy memória-rendszer. Két különböző kapacitású ROM-memória és egy SRAM alkotja az áramkör-készletet. A szószámok: 1 Ki, 2 Ki és ismét 2 Ki. A 2 Ki szószámú ROM nincs a „helyén”, hiszen alatta csak 1 Ki terület van, a 2 Ki szabályos helye 1 Ki-val feljebb kezdődne (0800 címnél). Ez azt sejteti, hogy 1 Ki széles sávokat nyújtó címdekóderrel oldották meg a feladatot, a 2 Ki szószámú memóriák két kimenetet összekapcsolva kaptak meg, így kerülhetett a ROM az 1 Ki-k számára normál kezdőcímre.

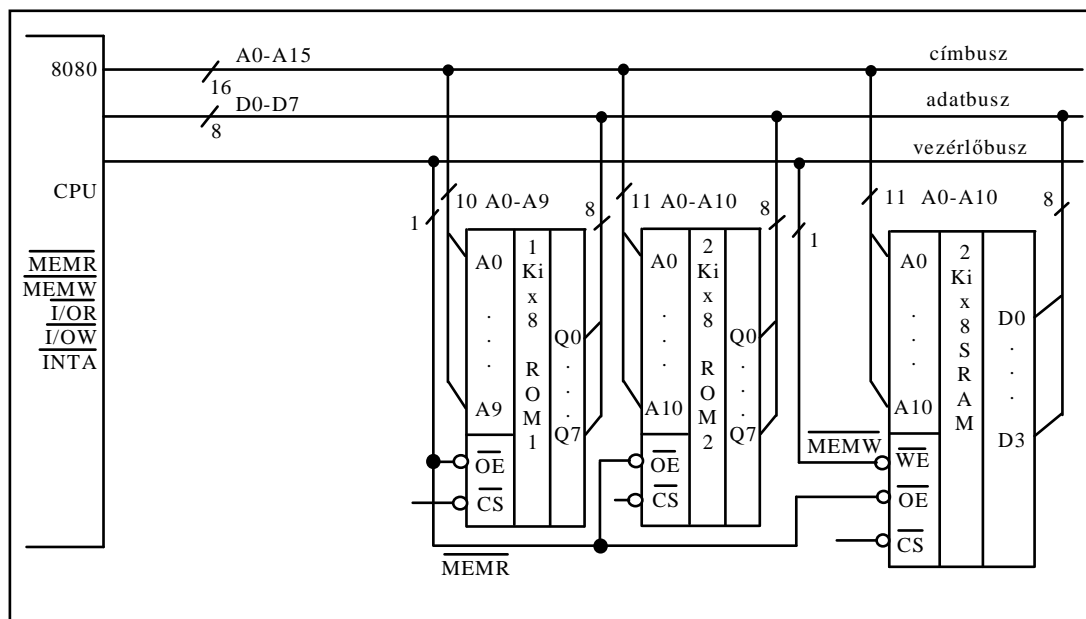
A teljes, kikódolt és felhasznált címterület  $1\text{ Ki} + 2\text{ Ki} + 1\text{ Ki} + 2\text{ Ki} = 6\text{ Ki}$ .

Mivel a dekóder 1 Ki széles területeket jelöl ki, ez hat kimenetet jelent. Ehhez két címbemenet kevés, ezért három címbemenetet fog a dekóder szétkódolni, a nyolc kimenete közül a két felső kihasználatlan marad. Mivel a kimenetek által kijelölt terület nagysága 1 Ki (annak tíz belső címvezetéke az A0 ... A9), a legkisebb dekódolt címbit az A10.

Az eddigiekből következik, hogy a címdekóder az A10, A11, A12 címbiteket fogja szétkódolni, így állítja elő a  $8 \times 1\text{ Ki}$  címterületet. A címdekóder által kezelt teljes címterület 8 Ki, azaz 0000 ... 1FFF. A megadott memória-címterkép szerint az áramkörök a továbbiakban nem jelennek meg, a többi terület üres, ez azt jelenti, hogy a magasabb címbitek (A13, A14, A15) megfelelő értékével a címdekóder egyértelműen el kell helyezni a 64 Ki-n belül. Az A13, A14, A15 címbiteknek nyolcféle kombinációja van, így a 8 Ki kezelt terület 8 különféle helyre lenne helyezhető. A feladat legalulra, a 0000 címtől kezdődően helyezte el a memóriákat, ott mindhárom magasabb címbit 0 értékű. Ezért a címbitek 0 értékét kell összefogni egy kapuval, erre a célra a VAGY kapu a megfelelő, ez akkor ad a kimenetén 0-t, ha mindhárom bemenete 0. A címdekóder egyik alacsony szinten aktív engedélyezőjelét állítja majd elő a VAGY kapu.



A következő lépés az lehet, hogy a memória-áramköröket berajzoljuk a buszrendszer mellé, bekötjük az ökölszabályoknak megfelelően az adatpontjaikat, a saját címpontjaikat és az üzemmódvezérlő jeleiket. A memória IC-k logikai rajzjelét a típusuk és a kapacitásuk alapján rajzoljuk meg. Mivel a feladatban nincs megkötve a SRAM típusa, legyen az egyszerűbben kezelhető három vezérlőjeles változat! A két ROM-áramkör sajnos nem azonos (az egyik 1 Ki x 8, a másik 2 Ki x 8 kapacitású), ezért nem használhatjuk az egyszerűsítő ábrázolási módot, nem lehet a két ROM-ot egymás takarásában megrajzolni. A három memória IC-t a **7.3. ábra** szerint lehet a buszrendszerhez kapcsolni. A  $\overline{CS}$  pontok egyelőre bekötetlenek, hiszen ott történik majd meg a címkijelölés, a címdekóder kimeneteinek felhasználása. Ezek a kijelölőjelek fogják a memória-címtérkép meghatározott címtartományaira helyezni az egyes memória-áramköröket.



7.3. ábra

Az ábra készítésekor igyekezzünk, hogy a címdekóder számára megfelelő méretű hely maradjon. A címdekóderrel tudjuk, hogy az A10, A11, A12 címbiteket kell kezelnie, s hogy az egyik, alacsony aktív szintű engedélyező bemenetére VAGY kapun át a továbbiakat (A13, A14, A15) kell bevezetnünk.

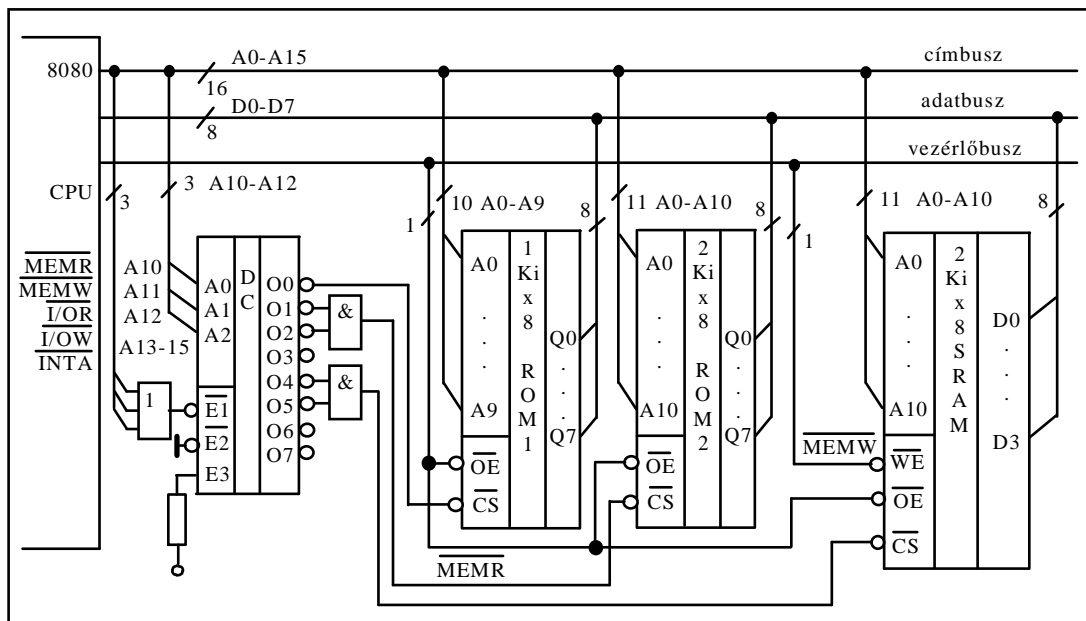
A kimenetei így 1 Ki széles területeket állítanak elő. A legelső vezetjük a ROM1-hez, a következő kettőt összefogjuk egy ÉS kapuval és így mehetnek a ROM2-höz. Az O3 szabadon marad, az O4 és az O5 ismét ÉS kapun át a SRAM kijelölésére szolgálnak.

Vegye észre, hogy ha egy memória-áramkör aktivizálása a címdekóder összefogott kimeneteivel valósul meg, akkor annak a memória IC-nek van olyan saját címbemenete, amit a címdekóder is kezel! Ha két kimenetet egyesítünk, egy ilyen címbit van, ha négyet, akkor kettő. A mi feladatunk megoldásánál két kimenetet fogtunk egybe az ÉS kapukkal. A memóriák szószáma 2 Ki, ezért a saját címbitjeik: A0 ... A10. És valóban, a címdekóder az A10 bitet dekódolja!

Azt is érdemes megfigyelni, mi a különbség a „helyén” lévő 2 Ki tartomány és a „rossz” helyen lévő címkijelölése között. A SRAM az O4 és az O5 kimenetekről kapja a vezérlését, az A12, A11, A10 címbit-hármas ezeknél 100 illetve 101 értékű, tehát csak az A10-ben különböznek egymástól. A ROM2, amelyik nem a 2 Ki szélességű területek egyikére került, a következő két címbit-hármasnak megfelelő kimeneteken (A1 és A2) van: 001 és 010. Bár ezeknél is különböző az A10, de a másik két bit sem azonos értékű!

A kapcsolás kialakításához három vezérlőjeles SRAM-ot választottunk. Ha nem így tettünk volna, a címdekóder egyik alacsony aktív szintű engedélyezőjéhez egy ÉS kaput kellett volna rajzolnunk, s azzal fogadni a  $\overline{MEMR}$  és a  $\overline{MEMRW}$  jeleket.

A teljes áramkört, beleértve a címdekódert is, a **7.4. ábra** mutatja be.



7.4. ábra

7

## 2. Feladat

A 7.5. ábrán egy I/O-címterképet látunk. Tervezzük meg a 8080-ra épülő mikroszámítógép I/O-rendszerét a címterképnek megfelelően! Az elkészült áramkör alapján rajzoljuk meg a Z80-ra épülő változatot is! Az I/O-ba ágyazott SRAM legyen három vezérlőjeles! A címterképen nem szerepelnek a címértékek, ezért azzal kezdjük a munkát, hogy ezek kiszámoljuk!

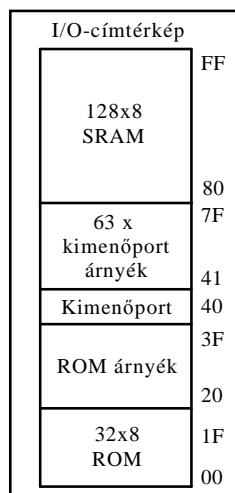
Az alsó 32 címérték (ROM áramkör) közül az első a 00, az utolsó az 1F. (Ez ugyanis öt bit, és öt biten lehet 0-tól 31-ig elszámolni.) A ROM árnyéka újabb 32 címhelyet foglal el, tehát a kezdőérték itt 20, a záró pedig 3F.

A kimenőport is az I/O-ba van ágyazva, a helyét ezután egyértelműen a 40-es cím jelöli ki. Felette 63 további címen is ez a port aktivizálódik, azaz a 41-es címtől egészen 7F-ig.

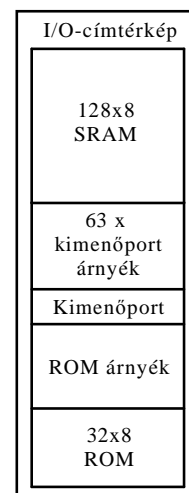
Így jutottunk el az I/O-ba ágyazott SRAM területéhez, ami láthatóan a 256 címhelyet tartalmazó I/O-címterkép teljes felső felét elfoglalja. Az alsó címhatár helyesen alakult, ez a 80, a felső pedig az FF (a címterkép legmagasabb címe). A címhatárokat is bemutató címterképet a 7.6. ábrán láthatjuk.

Vizsgáljuk meg a címterületek méretét! A ROM 32 címértéket foglal el, felette az árnyéka ugyanennyit. Ez a rajzrészlet már megmutatja, hogy címdekóderrel van megoldva a címkijelölés, a dekóder 64 szélességű címsávokat jelöl ki, ezért lesz a 432 x 8 kapacitású ROM-nak rögtön itt, a valódi címterülete mellett árnyéka. A kimeneti port elhelyezkedése megerősíti az előbbi megállapításunkat. A következő 64 szélességű dekóder-kimenet működteti a kimeneti portot, ami a 64 címhelyes tartományban nyilvánvalóan 63 árnyékkal jelenik meg. A címdekóder következő két kimenete együttesen jelöl ki 128 címhelyet, így fogja kiszolgálni az SRAM-ot.

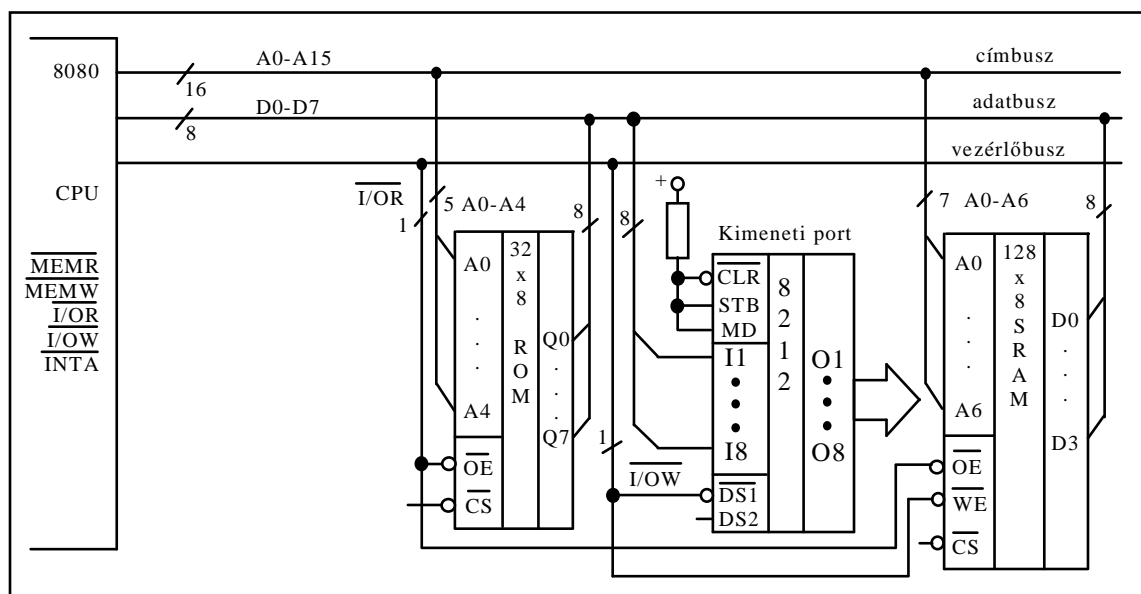
Összesen tehát 4 darab 64 címhelynyi területet jelöl ki a címdekóder, ehhez két címvezetékkel kell dekódolnia, az A6 és az A7 jelzésűeket. Mivel az I/O címezésére további címbitek nem is szolgálnak, nincs további árnyékképződésre lehetőség, a címdekóderre egyéb címbiteket nem kell rávezetni. Rajzoljuk meg először a buszrendszert, a memória-áramkört és a kimeneti portot (7.7. ábra)!



7.6. ábra



7.5. ábra



7.7. ábra

A buszrendszert és az áramköröket az ökölszabályok szerint össze is kötöttük, így az adatpontokat az adatbuszra, a memória saját címbemeneteit a címbusz azonos elnevezésű vezetékcsoportjára kapcsoltuk és megrajzoltuk az I/O-ba ágyazásnak megfelelő vezérlőjeleket is.

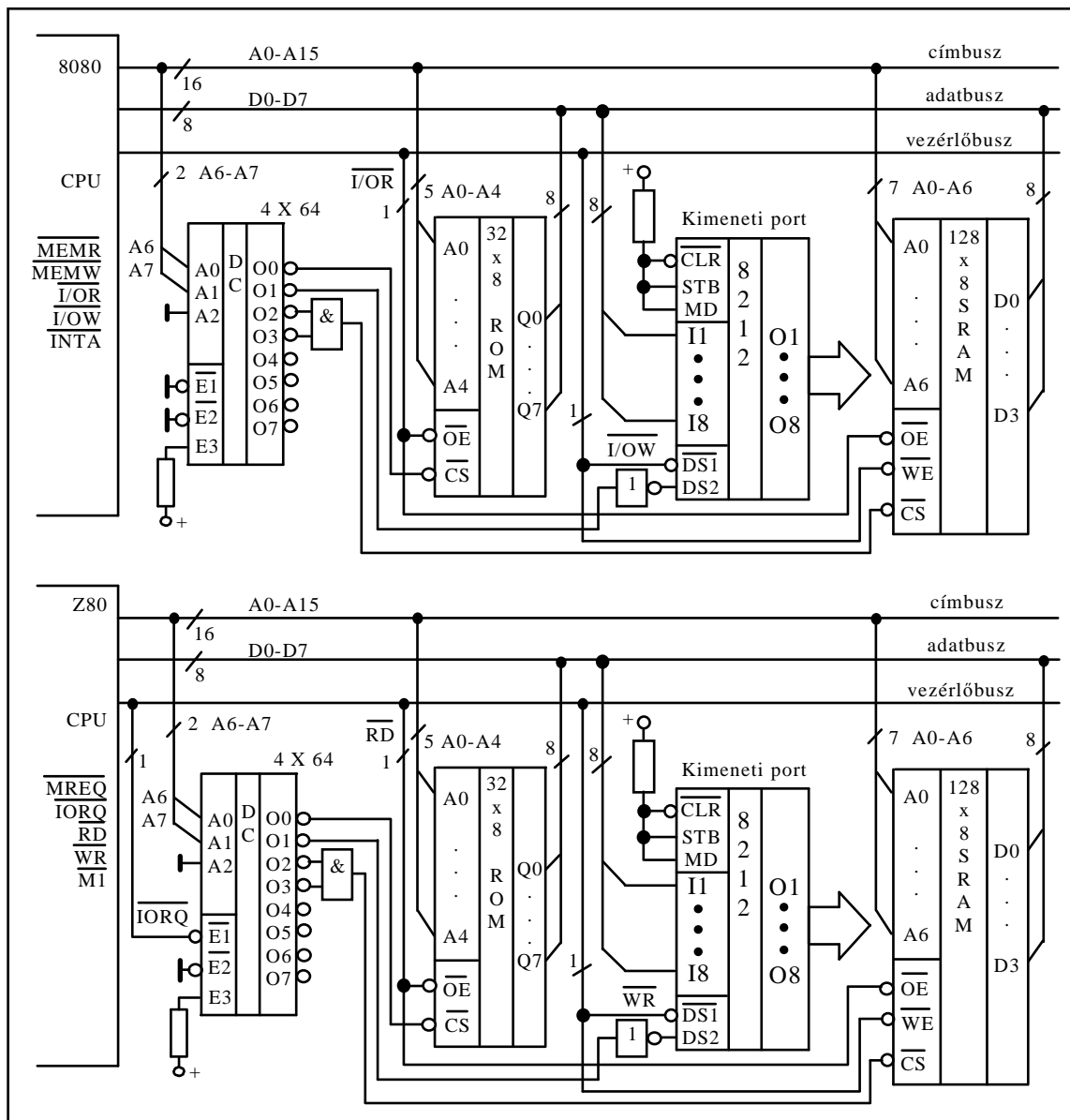
Természetesen a három IC más sorrendben is megjelenhetne az ábrán. Arra ügyeljünk elsősorban, hogy a címdekódernek elegendő hely maradjon, s az a CPU mellé kerülhessen!

A címdekóderrel még érdemes néhány szót szólni. A címtérkép elemzése alapján úgy határoztunk, hogy 4 x 64 jellegű címdekódolást fogunk alkalmazni, a 128 x 8 kapacitású SRAM-hoz ezért majd két kimenetet össze kell fogni, a másik két elemet közvetlenül lehet kezelni.

Ha szemügyre vesszük még egyszer a címtérképet (a 7.5. vagy a 7.6. ábrán), látható rajtuk, hogy más megoldást is választhatnánk. A címdekóderrel fele akkora címtérületeket, azaz 32 címértéket tartalmazó sávokat is kijelölhetnénk! Összesen nyolc ilyen tartomány van az I/O-címtérképen, az A5, A6 és A7 cím-bitek szét kódolásával lehet ezeket kijelölni. A 128 szószámú SRAM kijelöléséhez így négy kimenetet kellene összefogni. A ROM azonban éppen 32 címhelyes elem! Egy dekóderkimenet pontosan elég lenne számára. Csak azért kellene két kimenetet ÉS kapuval összefogni és a ROM-hoz vezetni, hogy a címtérkép szerinti árnyék megjelenjen. Így tehát azért építenénk be többletáramkört, hogy árnyékos legyen a kapcsolás. A kimeneti port is csak 32 címhelyet foglalna le, itt is egy újabb ÉS kapuval érhetnénk el a címtérkép szerinti 64 előfordulást. Azt kell tehát mondanunk, hogy logikailag helyes lenne a 8 x 32 felbontású címdekóder alkalmazása, a címtérkép megfelelné a feladatban előírtak, de nem lenne ésszerű, mert felesleges elemeket tartalmazna. Ezért a helyes megoldás az eredeti, 4 x 64 jellegű dekódolás!

A 7.8. ábra felső felén látható a teljes kapcsolás, amit az elmondottak alapján rajzoltunk meg a 8080 alapú CPU-hoz. Alatta a Z80-ra épülő mikroszámítógép látható, ugyanezzel az I/O-rendszerrel. Ez a kapcsolási rajz megint úgy készült, hogy a memóriák és az I/O-egység adatpontjainak és a saját címpontoknak, valamint a címkezelésre szolgáló részleteknek a bekötését megtartottuk, a vezérlőjeleket kellett átalakítanunk. Az  $\overline{I/OR}$  jel helyére vezetjük az  $\overline{RD}$ -et, az  $\overline{I/OW}$  helyett a  $\overline{WR}$ -t használjuk. Ezeket a jeleket egy-egy VAGY kapun át is bevezethetnénk az áramkörökbe, s e VAGY kapuk másik bemenetére kellene az I/O-ba ágyazást megvalósító  $\overline{IORQ}$  jelet rákötnünk. Egyszerűbb azonban, ha ezt a jelet a címdekóderhez vezetjük, egy alacsony aktív szintű engedélyező bemenetre. Ezt a megoldást választottuk a 7.8. ábrán látható Z80-as kapcsolás készítésekor.

Mivel három vezérlőjeles SRAM áramkört használtunk, nem kellett a hamis olvasásokkal foglalkozni. Mit kellett volna tennünk, ha két vezérlőjeles ez az IC?



7.8. ábra

A 8080 mikroprocesszorra épülő áramkörnél a címdekóderhez kellene rajzolnunk egy  $\overline{\text{ÉS}}$  kaput, de itt nem a  $\overline{\text{MEMR}}$  és a  $\overline{\text{MEMRW}}$  jeleket kellene rávezetni, hanem az  $\overline{\text{I/OR}}$  és az  $\overline{\text{I/OW}}$  jelpárost, hiszen most akkor működhetnek az áramköreink, ha ezen jelek egyike alacsony szintű, azaz aktív. Ha az  $\overline{\text{ÉS}}$  kaput nem építenénk be, memóriaolvasási és memóriaírási gépi ciklusokban a SRAM mindig olvasódna, ha a címbuszon olyan cím fordulna elő, aminek az alsó bájtja a 80 ... FF tartományba esne!

A Z80 mikroprocesszor esetében is a címdekóder vezérlése lenne a megoldás, ami a 7.8. ábra alsó részén már meg is történt! Ebben az elrendezésben tehát nyugodtan kicserélhető a három vezérlőjeles SRAM egy két vezérlőjeles változatra, nem fordulhat elő hamis olvasás, hiszen a címdekóder már eddig is csak I/O jellegű működésekkor aktivizálódott!

A 7.7. és a 7.8. ábrán nem tüntettük fel, de a 8080 mikroprocesszorra épülő mikroszámítógépben az A0 ... A7 címbitek, mivel I/O-címzésről van szó, minden esetben lecserélhetők az A8 ... A15 címbitekkel (mint tudjuk, a Z80 esetében ez a csere nem engedhető meg!)

### 7.3. Feladat

A **7.9. ábrán** látható memória-címtérkép alapján tervezze meg egy 8080 alapú mikroszámítógép memóriarendszerét! A SRAM legyen három vezérlőjeles!

Memória-címtérkép	
Üres	FFFF
	5000
8 Ki SRAM	4FFF
	3000
ROM2 árnyék	2FFF
	2800
2 Ki ROM2	27FF
	2000
4 Ki üres	1FFF
	1000
4 Ki ROM1	0FFF
	0000

7.10. ábra

Mivel csak memória-címtérképet kapott, minden elem memóriába van ágyazva. Segítségképpen még közösen ki-számoljuk az egyes memóriaterületek határcímeit, a **7.10. ábrán** ezeket is feltüntettük.

Vizsgálja meg az ábrán szereplő memória-áramköröket, s a kapacitásuk alapján határozza el, milyen tartományokat állítson elő a címdekóder! Ezután a címdekóderrel szétkódolt címbiteket kell kiválasztania. Mivel a felső címterületen nem ismétlődnek meg az elemek, a címdekóderre a magasabb címbiteket is rá kell vezetni!

Rajzolja meg a CPU-t és a buszrendszerét, majd a memória-áramköröket! Kösse be az adatpontokat, a saját címpontokat és az üzemmódvezérlő jeleket! Ezt követően rajzolja meg a címdekódert, vezesse rá a dekódolásra kerülő címbiteket, majd a magasabb címbiteket is. Végezetül kösse össze a címdekóder kimeneteit és a memória IC-k engedélyező bemeneteit! Az eredményt ellenőrizheti a 10. fejezetben!

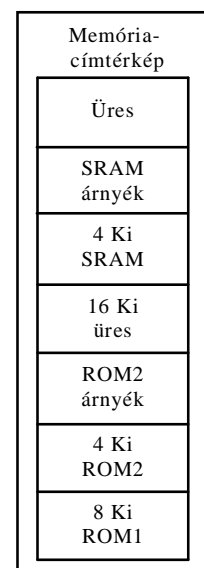
Memória-címtérkép	
Üres	
8 Ki SRAM	
ROM2 árnyék	
2 Ki ROM2	
4 Ki üres	
4 Ki ROM1	

7.9. ábra

#### 7.4. Feladat

A **7.11. ábrán** látható memória-címtérkép alapján tervezze meg egy 8080 alapú mikroszámítógép memóriarendszerét! A SRAM legyen három vezérlőjeles!

Mivel csak memória-címtérképet kapott, minden elem memóriába van ágyazva. A tervezéshez nem adunk segítséget, az előzőekben részletezett lépéseken haladjon végig, majd rajzolja meg a kapcsolást! Az eredményt ellenőrizheti a 10. fejezetben!



7.11. ábra

**7.5. Feladat**

A házi feladata az, hogy a **7.12. ábrán** látható I/O-címtérkép alapján tervezze meg egy 8080 alapú mikroszámítógép I/O-rendszerét, majd dolgozza át Z80-ra is!

Mindenekelőtt a kapcsolásban szereplő elemeket kell ismernie, ezért rajzolja ide a portokat (8212-vel), a ROM-ot és a címdekódot!

I/O-címtérkép	
ROM árnyék	FF E0 DF C0 BF
32x8 ROM	
Üres	80 7F
Bemenőport és kimenőport árnyékok	41 40
Bemeneti és kimeneti port	3F
Üres	00

7.12. ábra

A címtérkép elemzése alapján állapítsa meg, mekkora területeket fog a címdekóderrel kijelölni, és ennek alapján melyik címbiteket kell majd dekódolnia!

Rajzolja meg a kapcsolást 8080 alapú mikroszámítógépre!

Készítse el a kapcsolást Z80 mikroprocesszorra is!



## 8. Kész mikroszámítógép memória- és I/O-rendszerének elemzése

Ez a fejezet, hasonlóan az előzőhöz, az eddig szerzett ismeretek összefoglalását, ismétlését szolgálja, de közben új szempontrendszert is ad a mikroszámítógépek felépítésének értékeléséhez. A kész mikroszámítógép kapcsolási rajzát elemezve elsősorban a memóriarendszert és az I/O-rendszert értékeljük majd. A célunk az egyes áramkörtípusok elemzési megoldásának kiértékelése, a címtérképek elkészítése, az árnyékosság elemzése, igény esetén az árnyékmentesítés elvégzése.

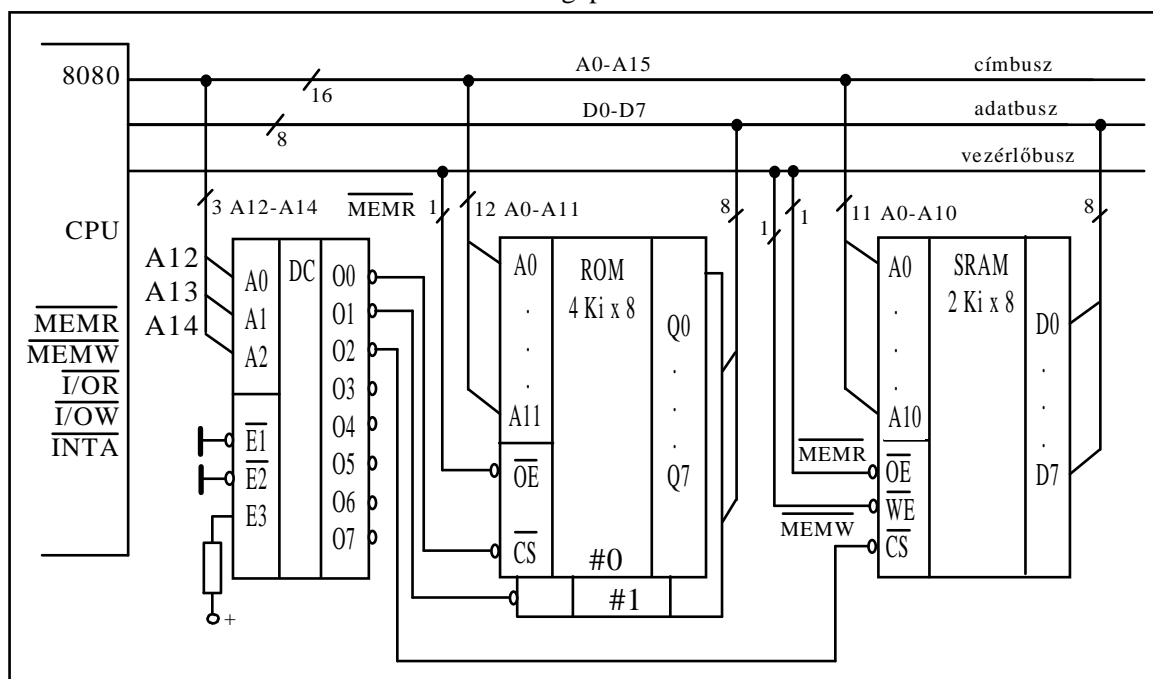
Mivel mindazt felhasználjuk eközben, amit eddig tanultunk, a mikroszámítógépek elemzését kiemelt fontosságú feladatcsoportként kezeljük. Természetesen az itt szereplő néhány feladat csak bevezetést jelenthet ebbe az elemző munkába, ezért elsősorban ne a konkrét kapcsolások sajátosságaira figyeljen majd, hanem az elemzés módszerére, gondolatmenetére!

A mikroszámítógépekben több ROM és több RAM memória-áramkör és egy vagy több port is lehet. Egy ilyen összetett memória- és I/O-rendszer esetében célszerű logikus sorrendben haladva megállapítani, hogy vannak-e árnyékok, s ha szükséges, az árnyékmentesítéshez át kell tervezni a memóriák, I/O-elemek kezelését. A munkánk lépései tehát a következők:

- a memória- és I/O-kezelés elemzése,
- a címtérképek felrajzolása,
- az árnyékosság vizsgálata, az árnyékok és a valódi terület bejelölése,
- az árnyékmentesítés megoldásának megtervezése, a kapcsolás átalakítása,
- az árnyékmentes címtérkép megrajzolása.

### 8.1. Feladat

Elemezzük ki a 8.1. ábrán látható mikroszámítógép memória- és I/O-rendszerét!



8.1. ábra

Első áramkörünkben csak memóriákat látunk, így az I/O-rendszerrel, I/O-címtérképpel nem kell foglalkoznunk.

Rajzoljuk meg a memória címtérképét, jelöljük be a memóriák címtartományait, az esetleges árnyékokkal együtt! A továbbiakban végezzük el a memóriarendszer árnyékmentesítését! Az ábrázolt memória-rendszer, mint látható, memóriába van ágyazva.

Első lépésként határozzuk meg, milyen címtérületeket jelöl ki a címdekóder! Mivel a legkisebb dekódolt címbit az A12, a dekódolt területek mérete 4 Ki. A dekóder három címbitét fogad (A12 – A14), így a dekódolt területek száma 8. Célszerű, ha az ábrára is ráírja, hogy 8 x 4 Ki. Most pedig határozzuk meg az első négy kimenet által kezelt címtartomány kezdő- és zárócímét:

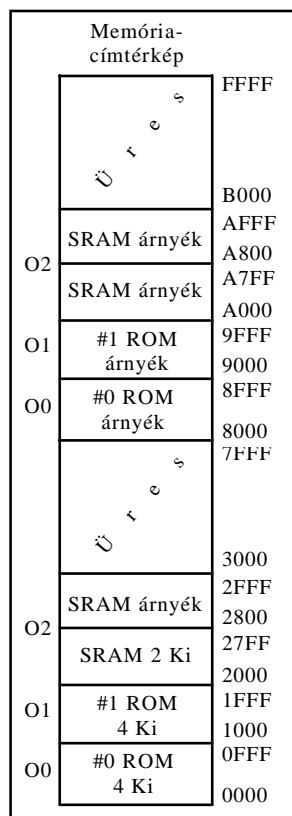
O0 kimenet (4 Ki): 0000h – 0FFFh ,  
 O1 kimenet (4 Ki): 1000h – 1FFFh ,  
 O2 kimenet: (4 Ki) 2000h – 2FFFh ,  
 O3 kimenet: (4 Ki) 3000h – 3FFFh .

(A címértékeket a kezelt bitekből és az adott kimenet számából is meghatározhatja, de célszerűbb, ha az előző fejezetekben említett módon begyakorolja legalább a 2 Ki, 4 Ki, 8 Ki méretű egymást követő területek határoló címértékeinek felírását!)

Most vegyük szemügyre a memória-áramköröket! Két egyforma ROM szerepel, A0 – A11 címbitekkel, a szószámuk tehát 4 Ki (az ábrán fel is van tüntetve). A SRAM címbitjei: A0 – A10, a szószáma ezért 2 Ki (ez is szerepel az ábrán). Ha a kapott kapcsolási rajzon a memória-áramkörök kapacitása nem volt megadva, ennél a lépésnél írjuk fel a kiszámolt értékeket!

Az eddigiek alapján mit gondol, lesznek-e árnyékok?

A címdekóder első két kimenete tehát egy-egy 4 Ki méretű memóriát aktivizál, maga a kimenet is 4 Ki



8.2. ábra

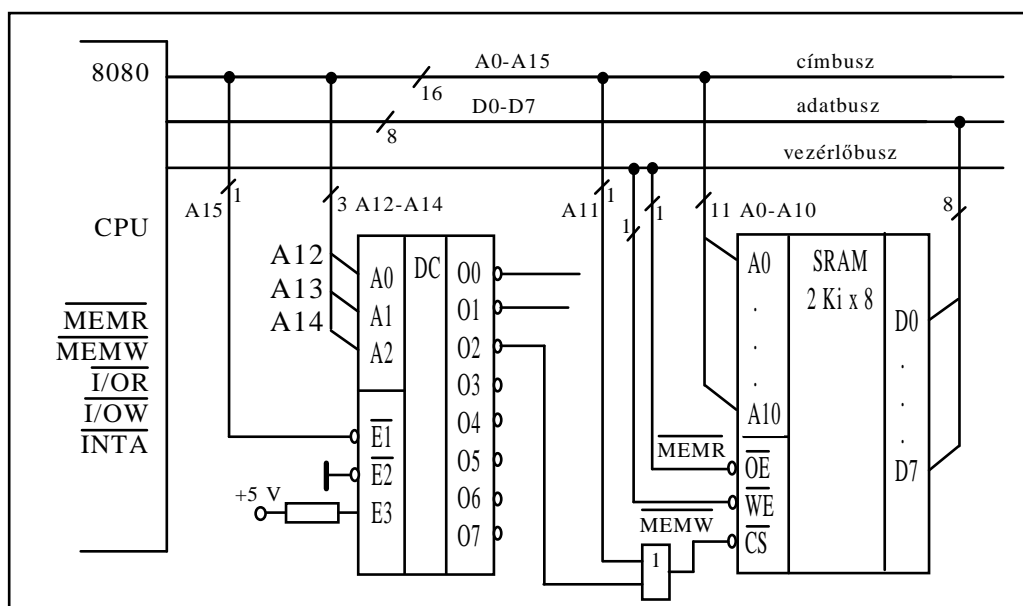
területet jelöl ki, itt a címtérképen a 4 Ki területen az illető ROM (#0 illetve #1) jelenik majd meg, a két legalsó, 4 Ki méretű területen. Az O2 kimenet is 4 Ki területet jelöl ki, de oda egy 2 Ki kapacitású SRAM-memória csatlakozik, az tehát egyszer valódi, egyszer pedig árnyék jelleggel fog szerepelni a címtérképen, alulról a harmadik 4 Ki méretű részleten belül. (Az előbbi kérdésre válaszként Ön is így képzelte el az árnyékok megjelenését? Ha nem, az most még nem baj, hiszen még csak gyakoroljuk az elemzést. Mindenesetre azt láthatja, hogy a válaszhoz azt kell tisztáznia, hogy mekkorák a memória-áramkörök és mekkora területen vannak működtetve, aktivizálva.)

A címdekóder által kezelt összes terület  $8 \times 4 \text{ Ki} = 32 \text{ Ki}$ , azaz a teljes címtérület fele. A teljes címtérület két felét az A15 0 illetve 1 értéke különbözteti meg, de ezt a címbitét a mi kapcsolásunk nem kezeli. Ezek szerint a felső 32 Ki-ban ismét megjelenik először a két ROM, mint árnyék, majd kétszer a SRAM, szintén árnyékként.

A 8.2. ábrán látható a memória-címtérkép, amit az előző gondolatok alapján rajzolhattunk meg. Láthatóak rajta a valódi memória-területek és az árnyékok. A címtérkép mellé az egyes részterületek kezdő- és zárócímét is felírtuk, felhasználva az előzőekben elkészített címérték-felsorolást is.

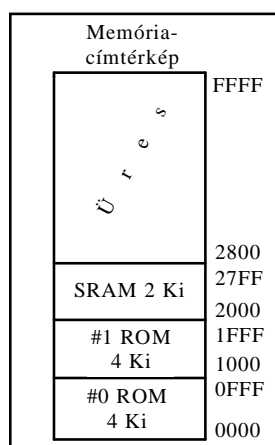
A címtérkép és a kapcsolási rajz megfeleltetését segíti az is, hogy az egyes címtartományok mellett a bal oldalon azt is felírtuk, hogy azokat a címdekóder melyik kimenete jelöli ki. Az O2 jelzést két címtartomány közös határvonalára írtuk, mert ez a kimenet a két területet (2000 ... 27FF és 2800 ... 2FFF) együtt, azaz a 2000 ... 2FFF (4 Ki méretű) címsávot aktivizálja. A memóriarendszer címezése tehát részleges, a címtérképen árnyékokat látunk.

Tegyük fel, hogy bővíteni szeretnénk a mikroszámítógép memória-rendszerét, ezért el kell tüntetnünk az árnyékokat! A 64 Ki teljes memóriaterületen kétszer megjelenő memóriarendszer ellen a címdekóderre, egy L aktív szintű engedélyező bemenetre rávezethetjük az A15 címvezetékét, így a felső 32 Ki területen a címdekóder nem ad ki működtető jelet, a felső árnyékok megszűnnek. A ROM-ok címjelölését nem kell megváltoztatni. A 4 Ki területen aktivizált 2 Ki méretű SRAM-memóriát a 4 Ki-n belül csak akkor engedjük működni, ha az A11 (mely az alsó és a felső 2 Ki területet különbözteti meg) címvezetéken 0 érték van, ha ezen a címvezetéken 1 áll, nem engedélyezzük. Ezt a SRAM engedélyező bemenete elé helyezett VAGY kapuval lehet elérni, melyre az O2 és az A11 jelek csatlakoznak. A kapcsolás módosítását a **8.3. ábrán** láthatjuk, a változatlanul kezelt két ROM-áramkört az ábrán nem ismételjük meg.



8.3. ábra

A módosult, árnyékmentessé vált memória-címtérképet a **8.4. ábra** mutatja be.

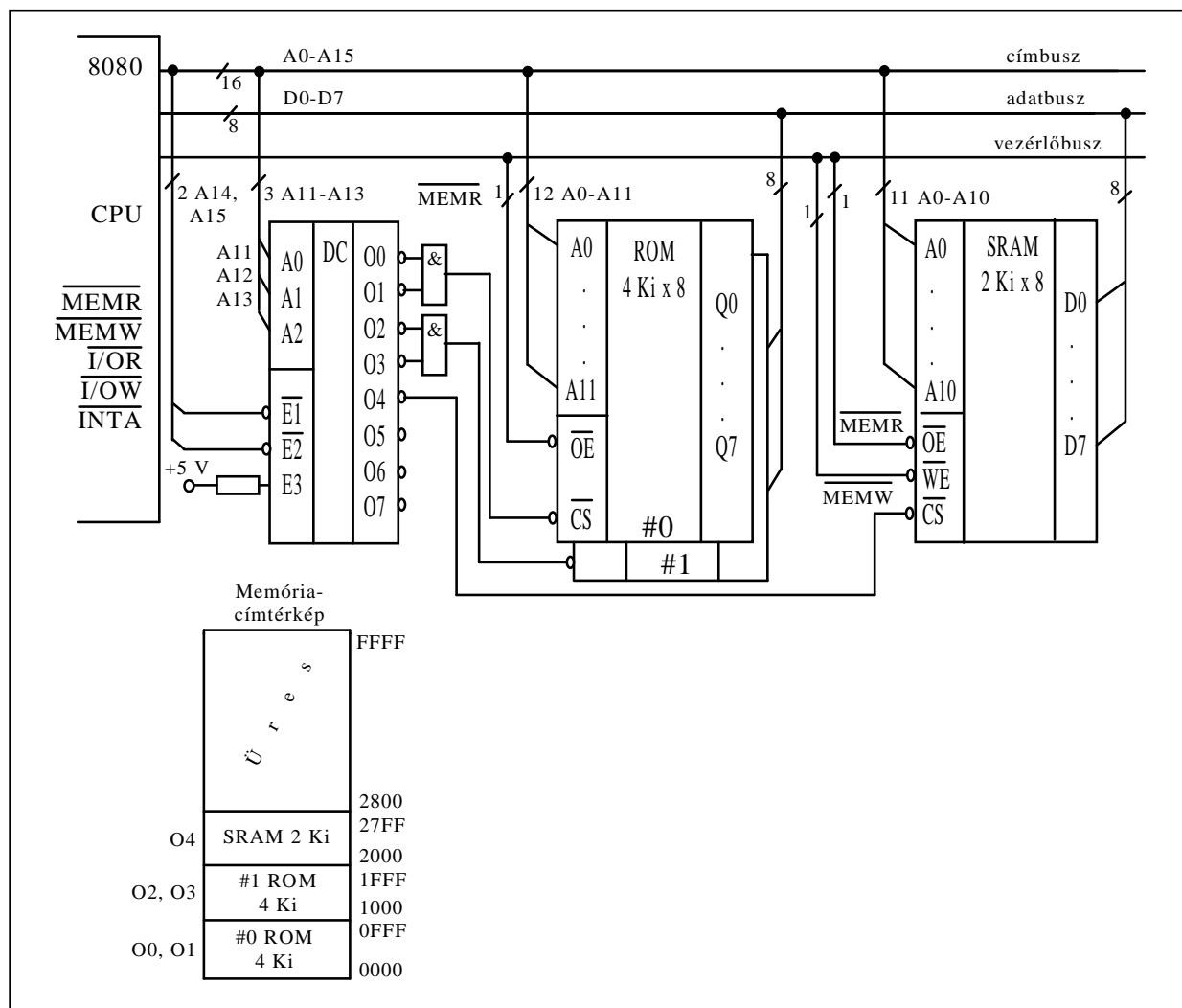


8.4. ábra

Ezzel az árnyékmentesítés megtörtént, a mikroszámítógép címtérképéről minden árnyékot eltüntetünk. Ha bővíteni kívánjuk a memóriarendszert, ennek ezután már nincs akadálya. A SRAM-IC-nél alkalmazott kapus árnyékmentesítés azonban nem a legkedvezőbb megoldás. Ha pl. egyetlen újabb 2 Ki x 8 kapacitású SRAM-áramkörrel bővítjük a kapcsolást, azt ismét kapuzottan kellene árnyékmentesíteni.

Mivel a kapus árnyékmentesítés nem a legmegfelelőbb megoldás, másikat is célszerű keresnünk! Az lehet az alternatív megoldás, hogy a címdekóder alakítjuk át úgy, hogy 8 x 4 Ki helyett 8 x 2 Ki területet jelöljön ki! Ehhez a legkisebb dekódolt címbit az A12 helyett az A11 lesz, azaz a dekóderre az A11 – A13 címbitek érkeznek. A teljes dekódolt terület most 16 Ki, ez négyszer jelenik meg a 64 Ki memória-címtérképen. Ha ebből a négyből továbbra is csak a legalsót kívánjuk használni (az ismétlődések, azaz az árnyékok elkerülésére), akkor a címdekóder L aktív szintű vezérlőbemeneteire rá kell kapcsolnunk az A15 mellett az A14 címvezetékét is.

Mivel a két ROM kapacitása 4 Ki, két-két címdekóder-kimenetet (O0 és O1 illetve O2 és O3) ÉS kapuval egyesítenünk kell, majd az O4, immár árnyék nélkül vezérelheti a SRAM-áramkört. A kapcsolást a **8.5. ábrán** lehet megtekinteni, ahol az új memória-címtérkép is megtalálható.



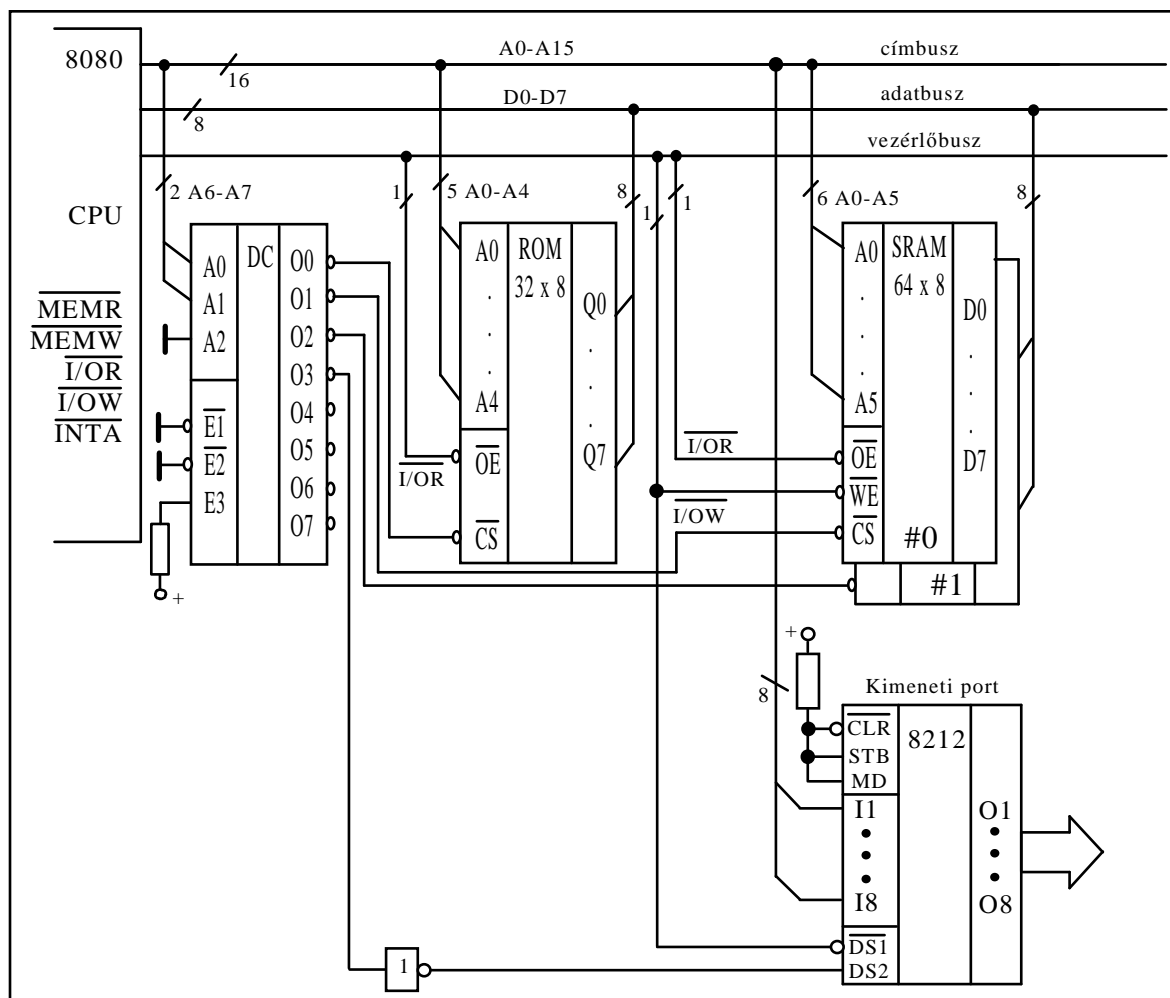
8.5. ábra

A címtérkép bal oldalán ismét látható, hogy melyik címtérületet a címdekóder melyik kimenete aktivizálja. A későbbi munkánkat, az áramkörök azonosítást jól segítik ezek a jelzések, ezért célszerű megszokni, hogy szerepeltessük azokat. A megoldásunkban az is látható, hogy a ROM-áramkörök kijelölésére két-két kimenet összekapcsolva került felhasználásra.

## 8.2. Feladat

Árnyékmentesítse a **8.6. ábrán** látható mikroszámítógépet!

Ha a feladat csak árnyékmentesítést kér, akkor is először elemezni kell a memória- és I/O-rendszert, el kell készíteni a címtérképeket, azután lehet az árnyékmentesítéshez hozzáfogni!



8.6. ábra

A kapcsolási rajzon látható áramkörök mind I/O-ba vannak ágyazva (ezt a működési módokat kezelő vezérlőjelekről lehetett leolvasni). Vizsgálja meg a memória-áramköröket, a címpontok alapján határozza meg a méretüket, és ellenőrizze le az ábrába beírt kapacitásértékeket!

Most vegye szemügyre a címdekódert, állapítsa meg, mekkora területeket jelölnek ki a kimenetek, olvassa le a kijelölt területek számát is, és ezt az információt is írja rá a kapcsolási rajzra!

Mielőtt tovább menne, írja fel a címdekóder nyolc kimenete által kezelt címtérület méretét, kezdő- és zárócímét (vegye figyelembe, hogy a dekóder A2 bemenete L szintre van kötve)!

- O0:
- O1:
- O2:
- O3:

Fogalmazza meg, hogy a ROM mekkora területen van aktivizálva, így ezen területen belül van vagy nincs árnyéka! Írja le az eredményét:

Tegye meg ugyanezt a két SRAM-áramkörrel kapcsolatban is!

Elemezze ki a kimeneti port címkiválasztási megoldását, írja le, árnyékmentesen vagy árnyékosan jelenik meg a címtérképen!

Most azt kell megnéznie, vannak-e olyan magasabb címbitek, amelyeket a kapcsolás nem vesz figyelembe! Melyik a legmagasabb feldolgozott címbit, melyik a legmagasabb működő címbit? A vizsgálata eredményét ismét írja be ide:

Mindezek alapján már tudja, hol helyezkednek el az áramkörök az I/O-címtérképen, s hogy hol jelenik meg árnyék – rajzolja meg a címtérképet! (Ne feledkezzen meg arról, hogy ez most I/O címtérkép, a teljes címtartomány csak 00 – FF!)

A memóriák esetében az árnyékot úgy szüntesse meg, hogy a címdekóderrel kisebb területeket jelöljön ki! Mekkora lesznek ezek a területek?

Most melyik címbiteket kell rávezetnie a címdekóderre?

A kimenőport esetében egészítse ki a címdekóder kimenetét VAGY kapu segítségével, így árnyékmentesítsen! Melyik címbiteket kell a VAGY kapuhoz vezetnie?

Lesz-e most külön kezelendő magasabb címbit?

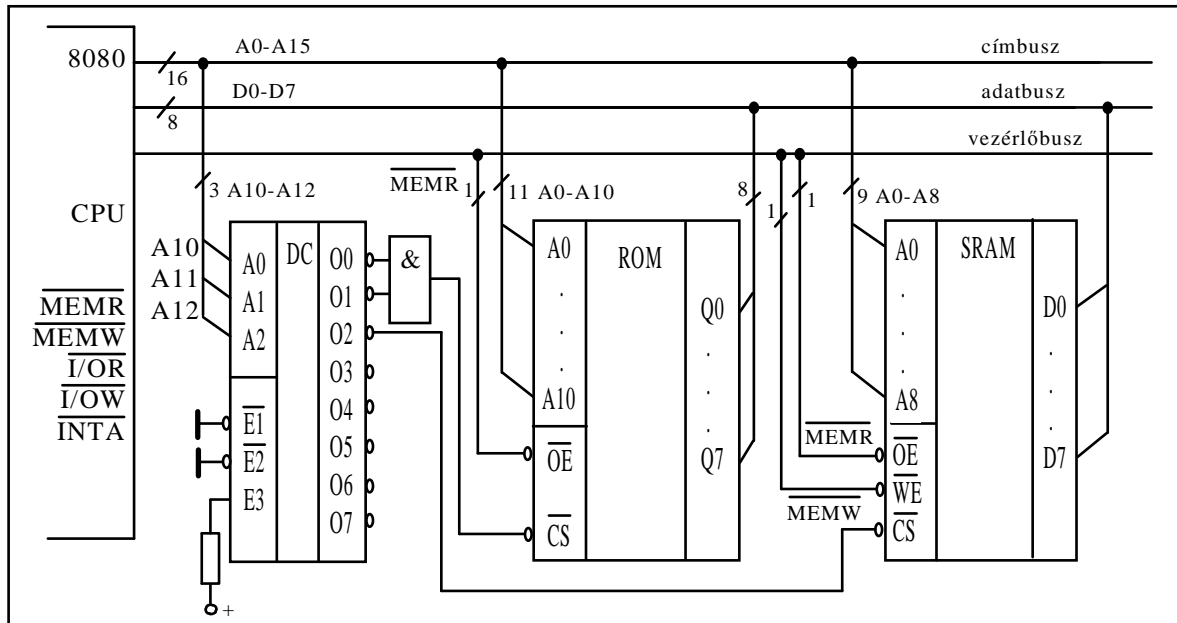
Mindezek alapján meg lehet rajzolni az árnyékmentes I/O-címtérképet:

Végül pedig készítse el a kapcsolási rajzot az árnyékmentes esetre!

Miután elkészült, a részeredményeket és a rajzokat ellenőrizheti a 10. fejezetben!. Mivel a kapcsolási rajz több változatban is elkészíthető, a gyakorlaton beszélje meg az oktatójával, hogy az eredményei helyesek-e!

### 8.3. Feladat

Ha gondosan áttanulmányozta a kidolgozott példát, készítse el a házi feladatot is! Vegye szemügyre a **8.7. ábrán** látható kapcsolást, elemezze ki, majd végezze el az árnyékmentesítést!



8.7. ábra

Itt elemezheti ki a címrendszert:

Rajzolja meg a címtérképet, nevezze is meg!



Tervezze meg az árnyékmentesítést!

Rajzolja meg az árnyékmentes címtérképet és az új kapcsolási rajzot!

A mikroszámítógépben előfordulhatnak olyan címtartományok (tiltott címtartományok), melyeket a gép használatakor el kell kerülni. Ebben a fejezetben megtudjuk, milyen esetekben találkozhatunk tiltott címtartományokkal, s arra is kitérünk, hogy milyen következményekkel jár e tartományok használata. A tiltott címtartomány a memória-címterület vagy az I/O-címterület olyan része, sávja, területe, értéke, amit a szoftverek futása közben nem szabad használni. Ezeken a területeken, mint majd látjuk, egyidejűleg több áramköri elem is aktivizálódna.

A tiltott címtartományon lévő elemek lehetnek kimeneti jellegűek. Ezek írási ciklusban működnek, a következmény az lesz, hogy egyidejűleg a különféle elemek mindegyikébe bekerül a kiírt információ, bár a CPU azt csak az egyiknek szánta. Komolyabb a baj bemeneti, olvasási jellegű ciklusokban. Ha több elem aktivizálódik, a kimeneteik egyszerre az adatbuszra vezetik az értékeiket. Ezek természetesen különbözhetnek. Az egyik következmény az, hogy a buszról nem lehet leolvasni az értékeket, az adatútközés következtében valamilyen vegyes értéksor áll elő.

A nagyobb probléma azonban az, hogy a kimeneti fokozatok egymást károsíthatják. Tegyük fel, hogy az egyidejűleg aktív kimenetek ellenüteműek és azt is, hogy az adatbusz D0 vezetékére az egyik működő elem 0 értéket továbbít, a másik 1-t. Ez azt jelenti, hogy a kimeneti fokozatok egymást rövidre zárják, a H szintet kiküldő áramköri elem kimenetére a másik L szintet vezet. A H szintet előállító kimeneti tranzisztor a D0 buszvezetékét a pozitív tápfeszültségre kapcsolja, az L szintet a másik kimenet tranzisztora ugyanezt a vezetéket a GND-re. A két tranzisztor tehát a tápfeszültséget a GND felé vezeti ekkor, azaz rajtuk keresztül teljes tápfeszültség-rövidzár jön létre. Előfordulhat, hogy mindkét kimeneti fokozatban azonnal tönkremegy a tranzisztor, többé a két áramkör nem működőképes, de ha az egyik kimenet nagyobb áramú, a másik kisebb, akkor nagy valószínűséggel a kisebb áramú tranzisztor fog pillanatszerűen tönkremenni.

A tiltott címtartományokkal foglalkozva felmerül a kérdés: lehetnek-e a mikroszámítógépben azonos címtartományokon kezelt különféle elemek, vagy sem.

Ebben a jegyzetben a mikroszámítógépekben memóriába illetve I/O-ba ágyazottan négyféle elemet alkalmazunk, ROM-, SRAM-memóriákat, bemenőportokat és kimenőportokat. Akkora bakt a tervezéskor már a kezdők sem követnek el, hogy két vagy több azonos elemet, pl. két ROM-memóriát azonos címtartományra helyeznének, azonos címtartományban aktivizálnának, ez nyilvánvalóan megengedhetetlen. De a különféle áramkörök már megjelenhetnek azonos címtartományokon, elsősorban az árnyékos szervezés esetén. Minden esetben meg kell tiltanunk az ilyen címterületek használatát? A következőkben a különféle párosításokat fogjuk szemügyre venni.

A ROM-áramkörök csak olvashatóak, a SRAM-áramkörök írhatóak és olvashatóak. Ha közös címtartományon vannak és olvasnánk valamelyiket, mindkettő egyszerre működne és a kimeneteken kialakulna a konfliktushelyzet, ezek tehát azonos címen nem működhetnek. Ha mégis azonos címre kerülnek, azt a címtartományt nem szabad használni. A ROM-áramkör és a bemeneti egység sem kerülhet azonos címre, olvasáskor mindkettő az adatbuszra helyezné az adatát és létrejönne az ütközés.

Más a helyzet a ROM-memória és a kimeneti port azonos címre kerülésekor. Mivel a kimeneti egységet a CPU csak írja, a ROM-ot pedig csak olvassa, nem jön létre problémás állapot, ezek lehetnek azonos címeken.

A SRAM-áramkörök sem bemeneti sem kimeneti egységekkel nem lehetnek közös címeken, mivel a SRAM írható is, olvasható is. Írásakor a kimeneti elemmel lépne fel egyidejű adatátírás, olvasáskor a bemeneti egységgel adatútközés jön létre.

A kimeneti és a bemeneti portok viszont ismét elhelyezhetők közös címeken, hiszen az egyiket csak írja, a másikat csak olvassa a CPU (ilyen megoldással már a jegyzetben is találkoztunk)! A közös cím használatának az az előnye, hogy közös címdekódert lehet alkalmazni.

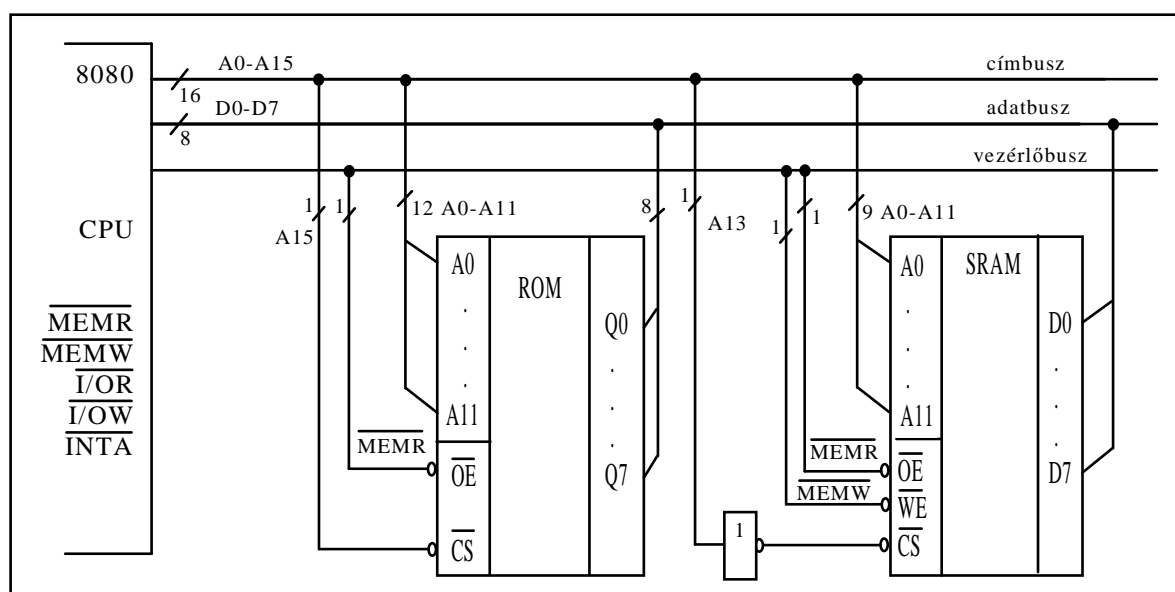
A mikroszámítógépekben tehát, ha feltételezhető a tiltott címtartományok előfordulása, azt kell megvizsgálni, hogy a meg nem engedett párosítások valamelyike előfordul-e. Ha találunk ilyet, meg kell állapítani azt a címtartományt, ahol ez a megengedhetetlen együttes működés kialakul, s azt kell tiltott címtartományként kezelni (a programok futásakor azokat a címeket nem szabad felhasználni, kezelni, kiadni). E munka segítésére foglaljuk össze egy táblázatban a lehetséges párosításokat és az értékelésüket!

Az azonos címen elérhető elemek	A helyzet értékelése
ROM és RAM	Tiltott
ROM és bemeneti port	Tiltott
ROM és kimeneti port	Megengedett
RAM és bemenet	Tiltott
RAM és kimenet	Tiltott
Bemenőport és kimenőport	Megengedett

Most, miután megismertük a tiltott címtartományok fogalmát is, egy mikroszámítógép memóriarendszerének elemzése alapján megkereshetjük ezeket a címtartományokat. Ha ezt megtettük, azután dönthetünk a továbbiakról. Az egyik lehetőség az, hogy a mikroszámítógép felhasználójának megtiltjuk e címtartományok használatát, a másik megoldás pedig a memóriarendszer átalakítása olyan módon, hogy ne jelenjen meg benne tiltott tartományok.

#### 8.4. Feladat

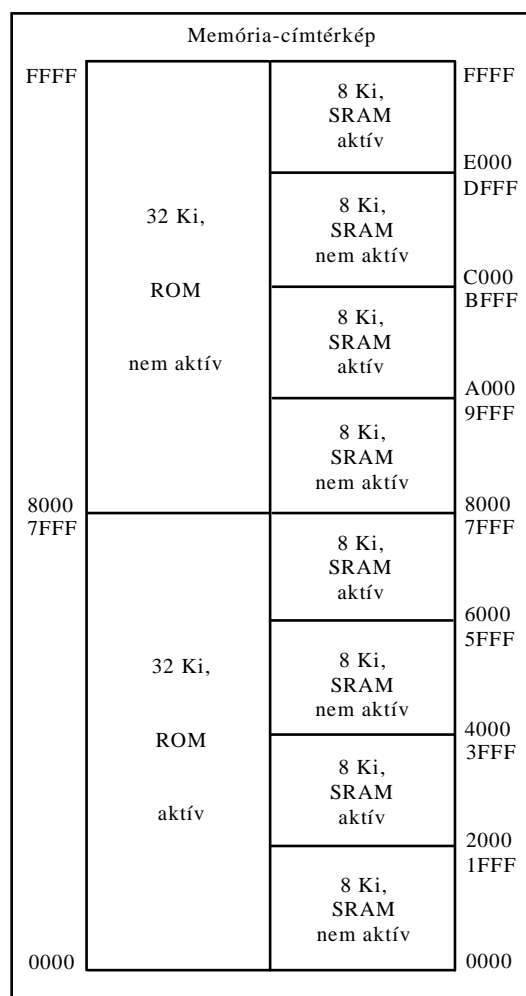
A **8.8. ábrán** mutatjuk be a mikroszámítógépet, amelyiken a hozzáértő szem bizony rögtön felfedezheti a tiltott címtartományokat! Miből lehet erre következtetni? Abból, hogy két memória-áramkör szerepel a mikroszámítógépben, és mind a kettő igen nagyvonalúan van csak címdekódolva. Következésképpen mindkettő számos árnyékkal jelenik majd meg a címtérképen, s így számítani lehet arra, hogy lesznek egymást fedő aktív területeik. Márpedig a SRAM és a ROM nem működhet azonos címtartományon, mint az előzőekben ezt megállapítottuk.



8.8. ábra

Írja be a memóriákba a kapacitásukat a címbemeneteknek megfelelően! Az áramkörök vezérlőjelei alapján látható, hogy a memóriák memóriába ágyazottan működnek. Mivel egymást fedő területekre számítunk, a címtérképet újszerűen ábrázoljuk. Az egyes elemeket, memória-áramköröket külön-külön mutatjuk be egy-egy 64 Ki méretű, teljes memória-címtérkép méretű téglalapon. A valódi címtérkép természetesen ezeknek a részleteknek az egyesítéséből áll.

A ROM-memóriát az A15 működteti, mégpedig a 0 értékével. Ez a címbit a 64 Ki címtérületet felezi, a ROM az alsó félen lesz aktív, a felsőn inaktív. A SRAM is egyetlen címbittel vezérlődik, az A13-mal. Az A13 a címtérületen 8 Ki széles területeket jelöl ki, s mivel inverteren keresztül kezeli a SRAM-ot, a memória az A13 = 1 értékeknél fog működni. Mindezeket a **8.9. ábra** mutatja be.



8.9. ábra

A ROM kapacitása 4 Ki x 8, a 8.9. ábra szerint a kijelölése 32 Ki területen történik. A SRAM kapacitása is 4 Ki x 8, ezt az áramkört 8 Ki széles tartományokban aktivizálja az áramkörünk. Mindkét memória-áramkör kapacitása kisebb tehát, mint az a címterület, ahol aktivizálódna, így árnyékokkal jelennek meg az aktivizáló területeken belül. A SRAM esetében az aktivizáló terület ismétlődik, természetesen ott is árnyékként jelenik meg a memória. Mindezeknek megfelelően rajzoljuk meg a memória és az árnyékok előfordulását (**8.10. ábra**)!

Ha a 8.10. ábrát vizsgáljuk, látható, hogy nem volt felesleges az az óvatosság, hogy az egyes áramköröket külön-külön ábráztuk, egymás melletti 64 Ki területet jelképező téglalapokon. Azonos méretűek a memória-elemek, és több esetben azonos címtartományokon szerepelnek. Ha egyetlen téglalapon rajzoltuk volna meg a memória-címtérképet, nem biztos, hogy minden esetet egyértelműen ábrázolni tudtunk volna!

Még kevésbé lenne követhető a címtérkép, ha különféle méretű áramköri elemeket, esetleg memória-áramköröket és portokat tartalmazna a mikroszámítógépünk. Ezért tehát ha a memóriarendszer és/vagy az I/O-rendszer felépítése olyan, hogy azonos címtartományon többféle áramköri elem is aktivizálódik, mindig ezt az ábrázolási megoldást kell alkalmazni!

A ROM és a SRAM legkisebb címértékű előfordulását ezen a címtérképen valódi címterületnek adtuk meg, de ez még nem végleges, hiszen a tiltott címtartományok miatt előfordulhat, hogy az így kiválasztott címterületet nem is szabad használnunk!

Memória-címtérkép			
FFFF	ROM  nem  aktív	SRAM árnyék	FFFF
		SRAM árnyék	F000
		SRAM árnyék	EFFF
		SRAM nem aktív	E000
		SRAM árnyék	DFFF
		SRAM árnyék	C000
	ROM árnyék	SRAM árnyék	BFFF
		SRAM árnyék	B000
		SRAM árnyék	AFFF
		SRAM árnyék	A000
		SRAM nem aktív	9FFF
		SRAM nem aktív	8000
8000	ROM árnyék	SRAM árnyék	7FFF
7FFF	ROM árnyék	SRAM árnyék	7000
7000	ROM árnyék	SRAM árnyék	6FFF
6FFF	ROM árnyék	SRAM árnyék	6000
6000	ROM árnyék	SRAM árnyék	5FFF
5FFF	ROM árnyék	SRAM nem aktív	4000
5000	ROM árnyék		
4FFF	ROM árnyék	SRAM árnyék	3FFF
4000	ROM árnyék	SRAM árnyék	3000
3FFF	ROM árnyék	SRAM	2FFF
3000	ROM árnyék	SRAM	2000
2FFF	ROM árnyék	SRAM	1FFF
2000	ROM árnyék	SRAM	1000
1FFF	ROM árnyék	SRAM	0FFF
1000	ROM 4 Ki	SRAM nem aktív	0000
0FFF			
0000			

8.10. ábra

Az elkészített memória-címtérkép ezután már alkalmas arra, hogy a tiltott címtartományokat megjelöljük rajta. Csak ROM- és SRAM-áramkörünk van, azokról pedig tudjuk, hogy azonos címen nem működhetnek. Meg kell tehát keresnünk azokat a területeket, ahol a ROM és a SRAM egyaránt aktív lenne. Ezeket a **8.11. ábrán** sraffozással meg is jelöltük, ezek tehát a tiltott címtartományok.

Ezután még az a feladatunk, hogy meghatározzuk a két memória-áramkör használatára ajánlott címtartományokat. A ROM esetében annak legkisebb című aktivizálódási területe nem esik tiltott címtartományba, az lehet a használt címtartomány, a többi megengedettet minősíthetjük árnyéknak.

Más a helyzet a SRAM-nál, mivel annak legkisebb címtartománya sajnos tiltott területre esik. Végül tehát a SRAM használatát az A000h – AFFF címtartományon oldhatjuk meg, a fennmaradó területek lesznek az árnyékok.

A gyakorlatban a tiltott címtartományokat általában el kell kerülni. Nem célszerű, ha egy mikroszámítógép kezelési útmutatójában az szerepel, hogy bizonyos címtartományokat a számítógépre írt szoftverben nem szabad használni, bár ott a hardver létezik. Egyszerűbb és tisztább megoldás, ha azonos címtartományon nem működtetünk különféle áramköröket.

Az egyetlen, gyakran alkalmazott párosítás az, hogy azonos címre helyezünk bemenőportot és kimenőportot, de ez nem is eredményez tiltott címtartományt, ahogyan már kielemeztük a korábbiakban.

A tiltott címtartományok elkerülésének egyszerű módja az, ha az áramköröket címdekóderrel működtetjük, s külön-külön kimeneteket használunk fel az egyes IC-k kijelölésére. Kapukkal megvalósított cím-kijelöléssel is megoldható a tiltott címtartományok elkerülése.

Memória-címtérkép			
FFFF	ROM   nem  aktív	SRAM árnyék	FFFF
		SRAM árnyék	F000
		SRAM nem aktív	FFFF
			E000
			DFFF
			C000
		SRAM árnyék	BFFF
		SRAM 4 Ki	B000
			AFFF
		SRAM nem aktív	A000
			9FFF
8000			8000
7FFF	ROM árnyék	SRAM árnyék	7FFF
7000			7000
6FFF	ROM árnyék	SRAM árnyék	6FFF
6000			6000
5FFF	ROM árnyék	SRAM nem aktív	5FFF
5000			
4FFF	ROM árnyék		
4000			4000
3FFF	ROM árnyék	SRAM árnyék	3FFF
3000			3000
2FFF	ROM árnyék	SRAM árnyék	2FFF
2000			2000
1FFF	ROM árnyék	SRAM nem aktív	1FFF
1000			
0FFF	ROM 4 Ki		
0000			0000

8.11. ábra

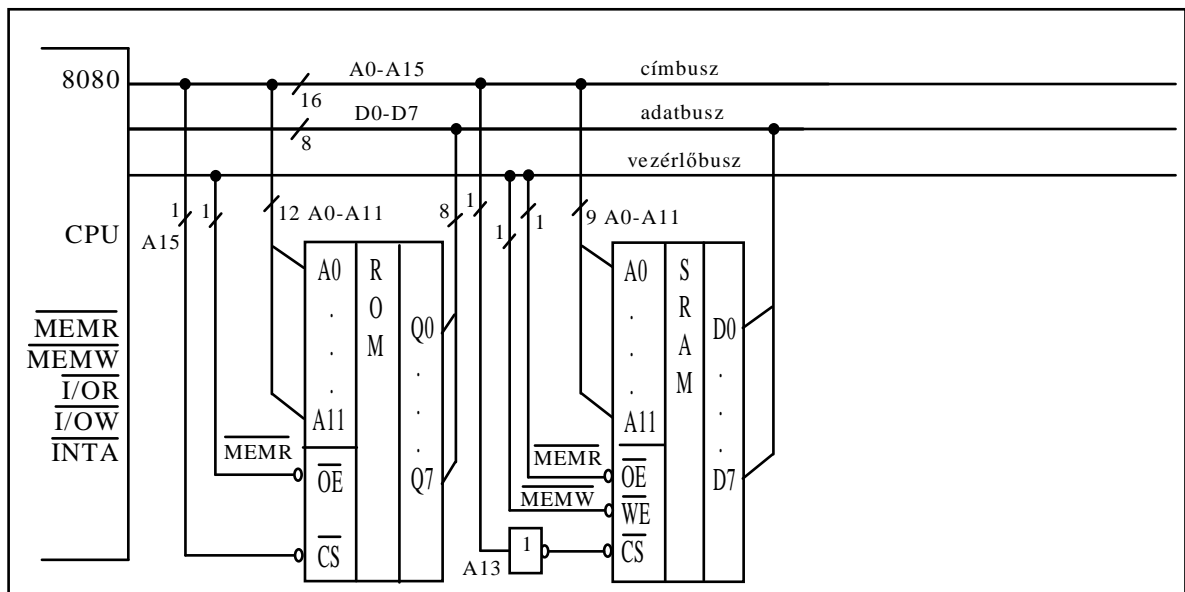
Általában a jól megtervezett mikroszámítógépekben nem fordulnak elő tiltott címtartományok. Azért kell mégis ismernünk ezt a fogalmat, hogy szükség esetén ellenőrizni tudjuk, nem lépnek-e fel mégis ilyenek, s azt is tudjuk, mi a teendő velük kapcsolatban.

### 8.5. Feladat

A fejezetünk végére értünk. Próbálja ki magát tiltott címtartomány vadászaton is! A 8.8. ábrát rajzolja át, építsen be még egy SRAM-áramkört, amit az A14 címbit aktivizáljon, az A14=1 értékével! Ez a SRAM is többször fog aktivizálódni a 64 Ki memória-címtartományon, így tovább fogja színesíteni a kezelhető és tiltott címtartományok megjelenését.

Ezután rajzolja meg a memóriák aktivitási térképét, majd keresse meg és vonalkázással jelölje a tiltott tartományokat! Természetesen minden megjelenő tartománynál írja fel a kezdő- és a záró címeket! Utolsó lépésként adja meg a ROM és a két SRAM használható címtartományát is!

A kiegészített logikai kapcsolást rajzolja meg a következő oldalon! Segítségképpen megismételtük a 8.8. ábrát, de úgy, hogy a rajta szereplő elemeket balra eltoltuk, így elegendő helye van az új memória-áramkör behelyezésére (8.12. ábra)!



8.12. ábra

Rajzolja fel a memória-címtérképet (három hasábbal), jelölje be az aktivizálódási területeket!

Rajzolja be a memóriák pontos szószámának megfelelő területeket egy új címtérképen!

Végül vonalkázza be a tiltott címtartományokat, majd adja meg az egyes memória-áramkörök ajánlott címtérületét!

Az eredményeit ellenőrizheti a 10. fejezetben!

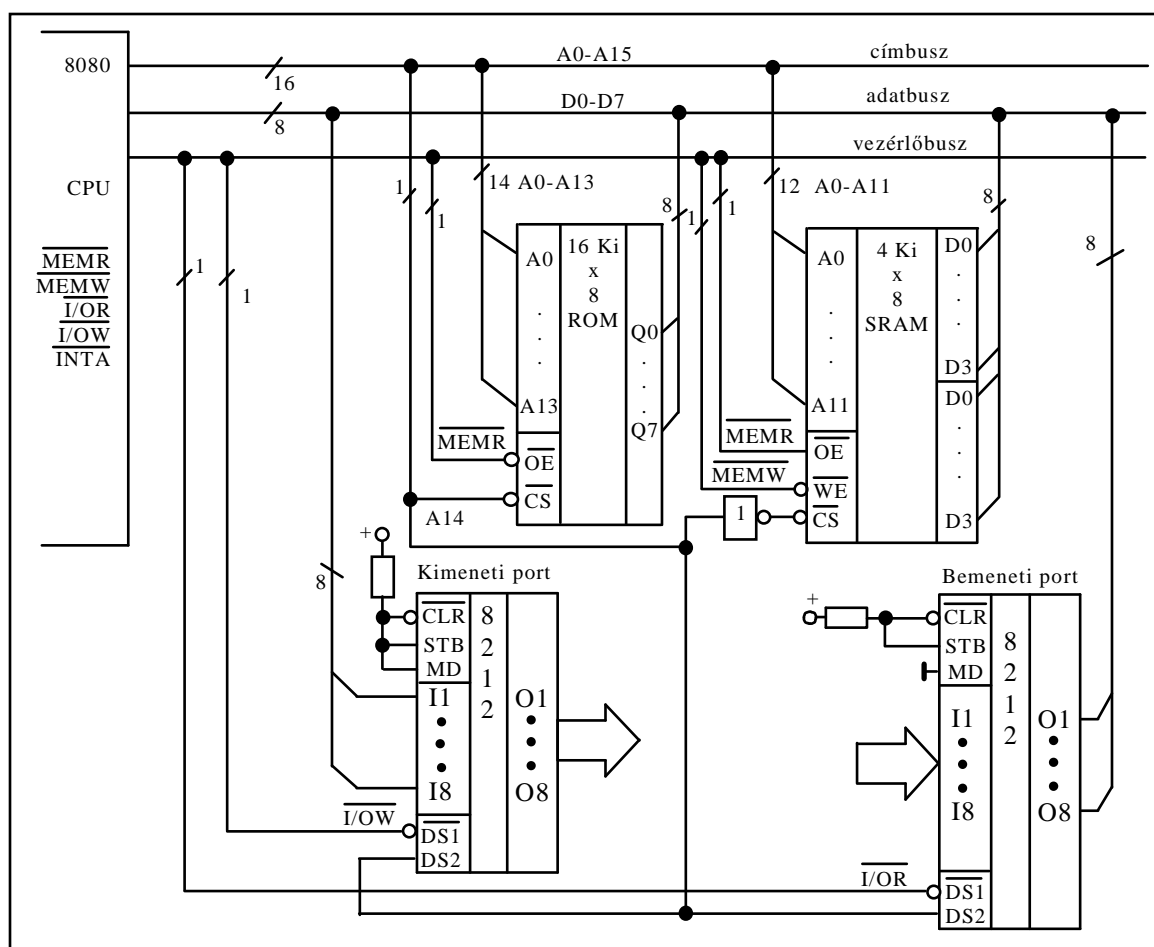


## 9. Kész mikroszámítógép memória- és I/O-rendszerének bővítése

Bár egyszerűnek tűnik a fejezet címében megjelölt feladat, ez az összes eddigi ismereteink alkalmazását kívánja meg! Ha egy mikroszámítógép memória- vagy I/O-rendszerét bővíteni kívánjuk, azaz újabb memória-áramkört vagy portot kell beépíteni a kapcsolásba, előbb elemezni kell a meglévő áramkört megoldásokat, fel kell rajzolni a címtérképeket, esetenként meg kell szüntetni árnyékokat, tiltott címtartományokat, ezután lehet az új elemet beépíteni és így a mikrogépet bővíteni.

### 9.1. Feladat

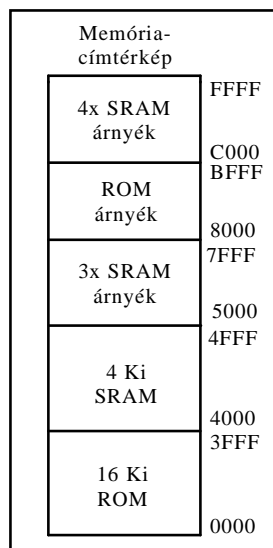
A 9.1. ábrán egy mikroszámítógép látható. Ezt kell bővítenünk egy 4 Ki x 8 kapacitású, memóriába ágyazott ROM-mal és egy 128 x 8 kapacitású, I/O-ba ágyazott ROM-áramkörrel.



9.1. ábra

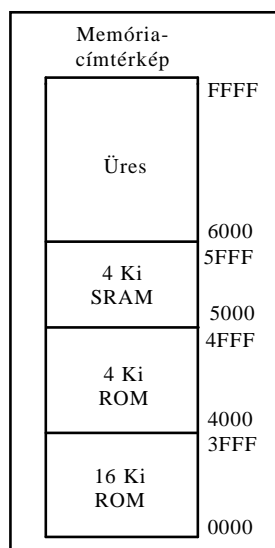
A CPU Intel 8080 mikroprocesszorra épül, ennek megfelelőek a vezérlőjelek. Először vegyük szemügyre a memóriába ágyazott áramköröket. Két memória-IC fog szerepelni a memória-címtérképén, egy ROM, 16 Ki x 8 kapacitással és egy SRAM, ennek a kapacitása 4 Ki x 8. A címkijelölés nagyon egyszerű megoldású, az A14 van rávezetve mindkét memória-áramkör  $\overline{\text{CS}}$  bemenetére. Tudjuk, hogy az A14 a memória-címtérkép 64 Ki teljes területén négy darab 16 Ki széles területet jelöl ki, kettőt az A14 = 0 érték, kettőt az A14 = 1 érték. Ez a méret éppen megfelel a ROM-áramkörnek, a SRAM számára túl széles, az négyszer is elfér rajta.

A SRAM tehát minden előfordulásakor árnyékokkal fog megjelenni a térképen. Mivel a ROM-ra az A14 közvetlenül kapcsolódik rá, azt az A14 = 0 érték választja ki, az inverter miatt a SRAM-ot az A14 = 1 érték aktivizálja. Mindezek alapján meg lehet rajzolni a memória-címtérképet (9.2. ábra). Az ábrán természetesen feltüntettük a címhatárokat is.



9.2. ábra

tett memória-áramkör így lefedi a teljes 64 Ki címterületet, nincs hely a bővítésre. Ez azt jelenti, hogy a bővítés előtt árnyékmentesíteni kell a memória-rendszert.



9.4. ábra

Az árnyékmentesített és az új ROM-áramkörrel bővített rendszer memória-címtérképét a 9.4. ábrán mutatjuk be. Az új és az eredeti ROM egymás mellé került, mert tudjuk, hogy a mikroszámítógépekben a ROM területnek is és a SRAM területnek is egybefüggőnek kell lennie. Leolvasható az ábráról, hogy címdekóderrel lehet megoldani a feladatot, úgy, hogy 8 x 4 Ki területeket fogunk vele kijelölni. Ehhez az A12, A13, A14 címbiteket kell dekódolni, az ismétlődés ellen pedig az egyik alacsony szinten aktív engedélyezőjelet az A15-tel állítjuk elő.

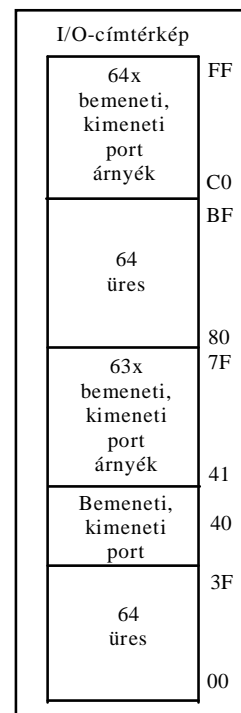
A bővített I/O-rendszer címtérképe a 9.5. ábrán látható. Itt is szükséges az árnyékmentesítés!

Mindkét memória-IC esetében a legelső előfordulást jelöltük meg valódi működési területként, a továbbiakat árnyékként.

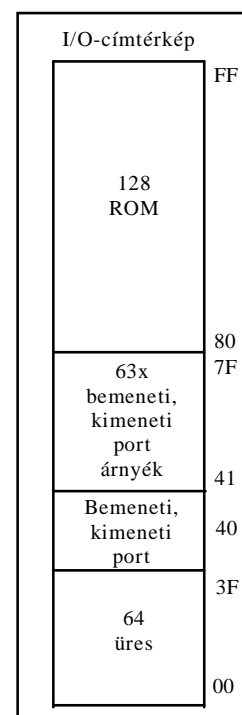
A végeredmény a memóriarendszer bővítése szempontjából kellemetlen, ugyanis a teljes címterületet lefoglalja a két eredeti áramkör. A 0000 ... 3FFF sávban (ami 16 Ki szélességű, az A14 = 0 jelöli ki) a ROM memória aktivizálódik, ez a valódi területe.

A következő 16 Ki méretű területen (3000 ... 7FFF, az A14 = 1 választja ki) a SRAM működik. Mivel csak 4 Ki a szószáma, az alsó 4 Ki méretű sávban tekintjük valódinak, felette háromszor árnyék jellegűnek a megjelenését. Ezután ismét a ROM működik (mivel az A14 = 0 megint, ez a 8000 ... BFFF címterület), a legfelső 16 Ki-ban pedig megint a SRAM aktivizálódik (az A14 = 1 miatt, a C000 ... FFFF területen). Az összesen 20 Ki beépített

Az I/O-rendszert egy bemeneti és egy kimeneti port alkotja, ezek vannak a vezérlőjelek alapján I/O-ba ágyazva. A címkiválasztásukat továbbra is az A14 címvezetékekkel oldottuk meg, ez az I/O címezésére szolgáló címbitek közül megfelel az A6-nak, azaz azzal azonosan viselkedik. Az A6 pedig az I/O-címtérképet négy darab 64 címértéket tartalmazó sávra bontja, kettőben az A6 = 0, kettőben pedig 1 értékű. A portok DS2 (magas aktív szintű) kijelölőjelét állítja elő az A14, ezért azokban a sávokban jelennek majd meg a címtérképen, ahol az A6 = 1 (40 ... 7F és C0 ... FF). A legkisebb előfordulási cím a 4000, ezt tekinthetjük valódi címnak, a többi árnyéknak. Mindezek alapján az I/O-címtérképet a 9.3. ábrának megfelelően lehet megrajzolni. Az I/O-címterületen van szabad hely, de két különálló 64 címértéket tartalmazó sáv, ami kevés a bővítéshez. Itt tehát vagy árnyékmentesíteni kell, vagy jelentősen át kell alakítani a kapcsolást.

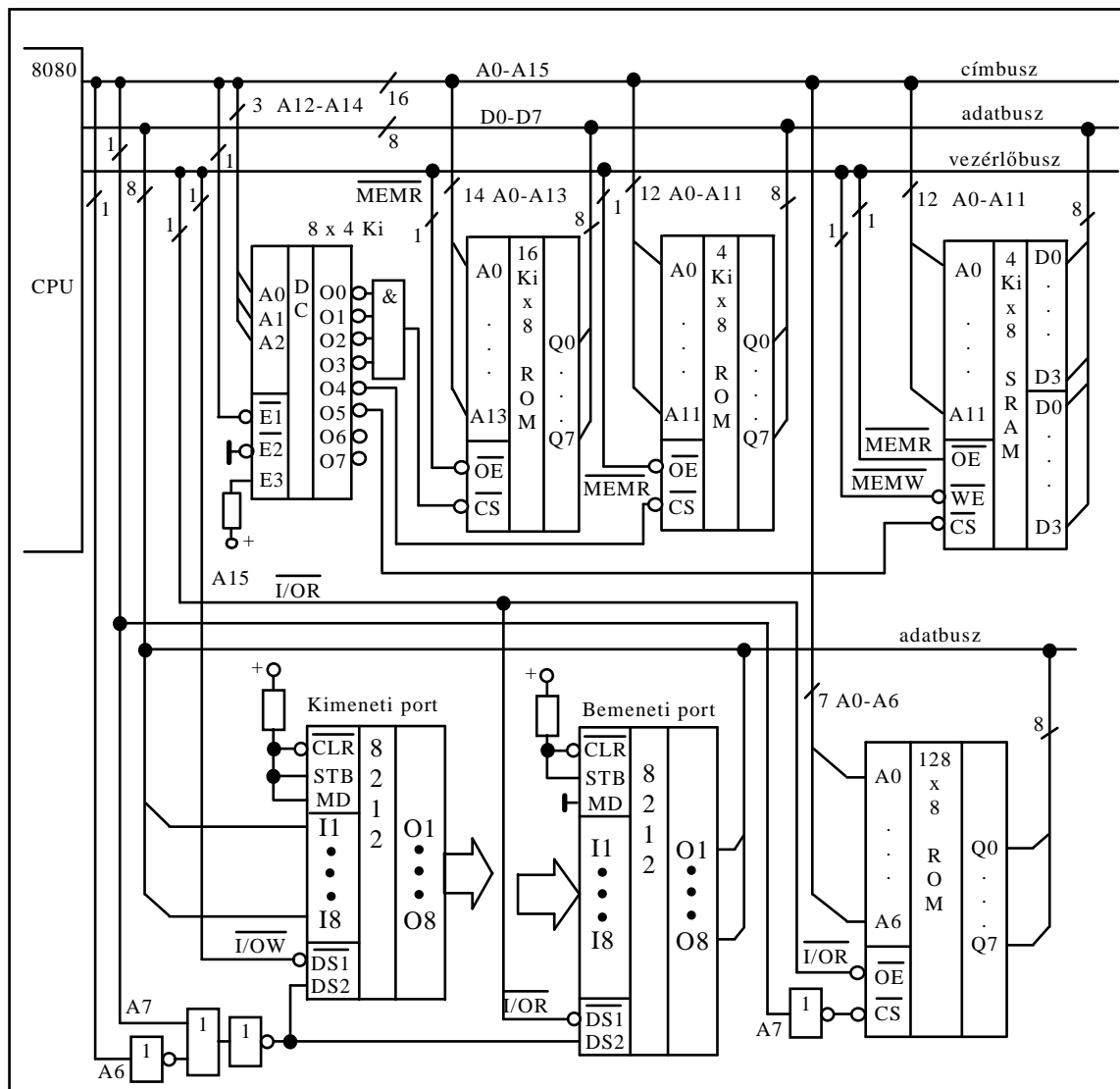


9.3. ábra



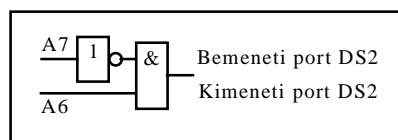
9.5. ábra

Ha lehetséges, a bővítéskor az eredeti elemek címzeit célszerű megtartani. Ezért az eredetileg beépített bemeneti és kimeneti port valódi címe továbbra is a 4000 maradt. Az I/O-címtérkép felső részéről tüntetjük el az ismétlődést, így kialakult az egybefüggő, 128 címhelyes tartomány, ahova a bővítő ROM elfér. A bővített áramkör kapcsolási rajzát a 9.6. ábra mutatja be.



9.6. ábra

A címdekóder bekötését már az előzőekben részleteztük. A portok címkijelölésénél egy VAGY-kaput használtunk fel, ami az A7 = 0-ra korlátozza a portok működését, így szabadul fel az I/O-címtérkép felső része.



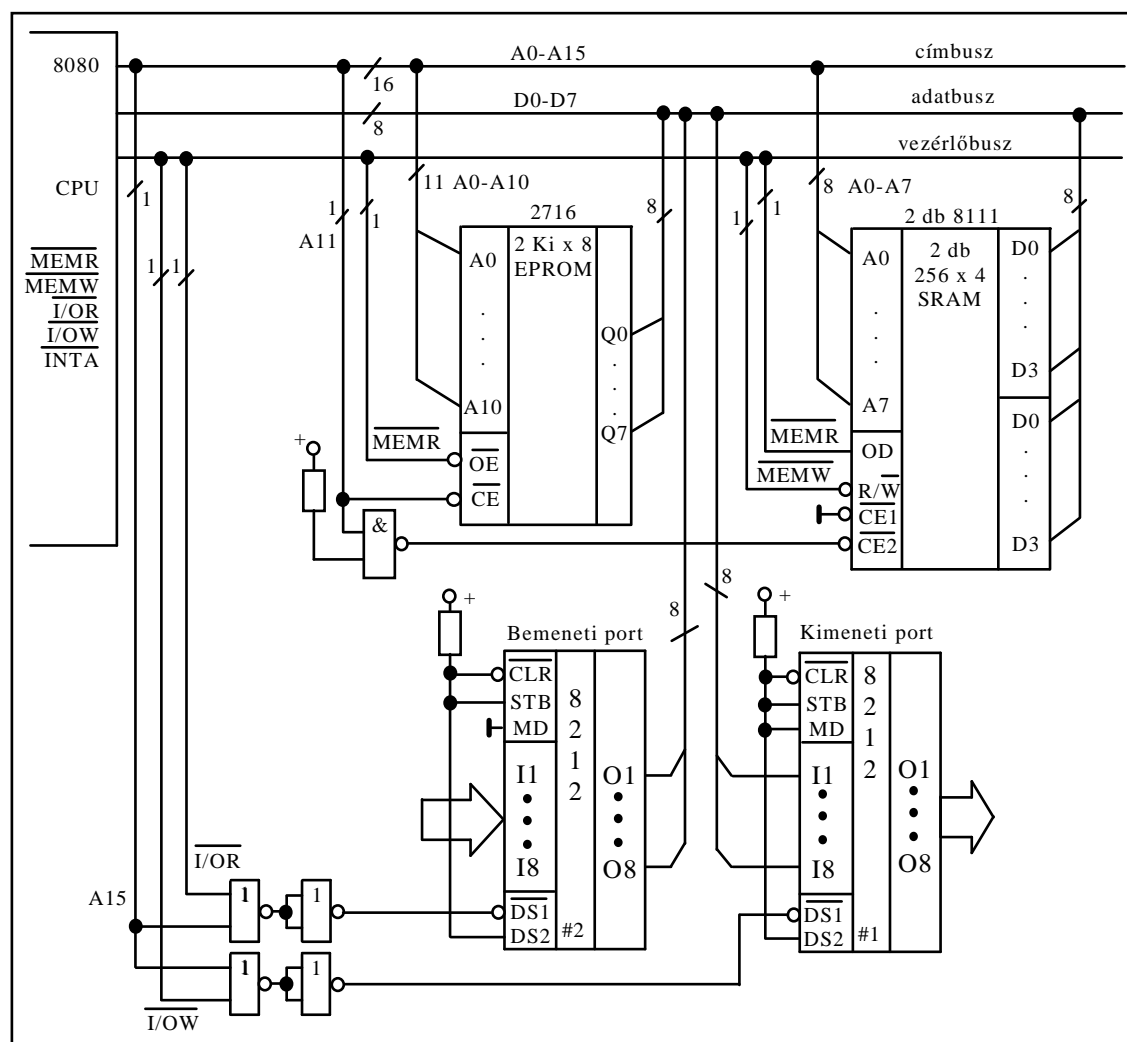
9.7. ábra

A portok címzőegysége egyébként más kapuhálózattal is kialakítható (9.7. ábra). Mivel a teljes címtérkép felső felén dolgozik a 128 x 8 kapacitású ROM-áramkör, az A7 címbit egy inverteren át lép be a CS pontjára. Ügyelni kellett még arra, hogy a bővítő elemek annak megfelelő vezérlőjeleket kapjanak, hogy melyik tartományba ágyazódnak be.

A példa megoldása igazolja, hogy minden eddig tanult ismeretünkre szükség van a bővítések szakszerű, működőképes végrehajtásához!

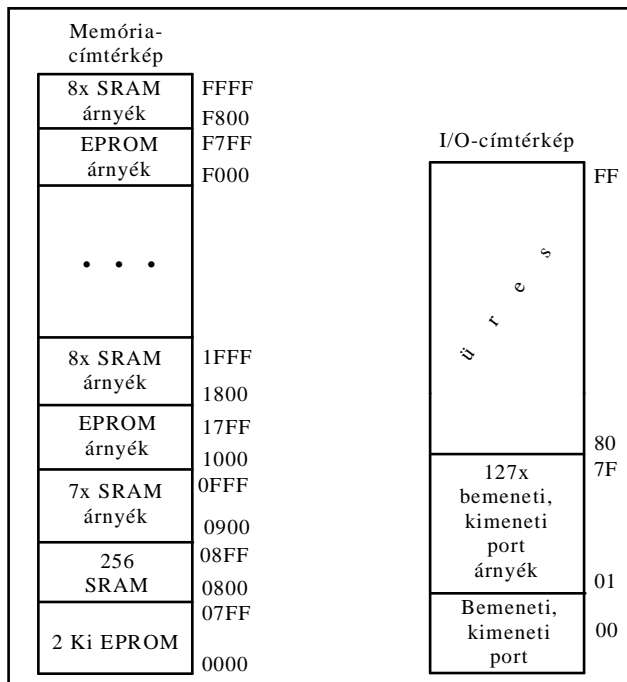
## A Demo- $\mu$ C

A Mikroelektronikai gyakorlatok V. című jegyzetünkben ismertük meg a Demo- $\mu$ C (demonstrációs mikroszámítógép) elnevezésű egyszerű, 8080-ra épülő mikroszámítógépet. Abban a jegyzetben alaposan elemezzük is a felépítését, megrajzoljuk a címtérképeit. Azért, hogy példákat tudjunk megoldani a Demo- $\mu$ C bővítésére, itt is megadjuk a logikai kapcsolási rajzát (**9.8. ábra**) és a címtérképeit (**9.9. ábra**).



9.8. ábra

A mikroszámítógépben memóriába ágyazva egy 2 Ki x 8 kapacitású, 2716 típusjelű EPROM és egy 1/4 Ki x 8 méretű SRAM-egység található. Utóbbit két 1/4 Ki x 4 kapacitású, 8111 típusjelű SRAM-IC alkotja. Az EPROM a szokásos vezérlőjelekkel működik, de a SRAM vezérlőjelei sajátosak. Különválasztott írásvezérlő és kimenetengedélyező jele van, és ezeken kívül két chip-engedélyező jele. Az írásengedélyező jel a szokásos  $\overline{WR}$  elnevezés helyett az  $R/\overline{W}$  nevet viseli, de ugyanúgy működik, mint egy  $\overline{WR}$  jel. Tulajdonképpen ugyanez igaz a kimenetet kezelő jelre is, amit itt OD névvel látunk (Output Disable). Mivel ez H szinten tiltja a kimenetet és L szinten engedélyezi, végül is ugyanúgy működik, mint egy  $\overline{OE}$  jel, nyugodtan azt is írhatták volna a katalógusban erre a csatlakozóra!



9.9. ábra

a többi 127-et pedig árnyéknak.

Az elmondottak alapján lehet megrajzolni a memória- és az I/O-címtérképet (9.9. ábra). A bővítések szempontjából nem túl kedvező a helyzet. Az összesen  $2\frac{1}{4}$  Ki beépített memória-áramkör a teljes 64 Ki memória-címterületet letakarja az árnyékaival, egyáltalán nincs hely bővítésre. Az I/O-címtérképen kicsit kedvezőbb a helyzet, a felső 128 címhelyre be lehet helyezni kiegészítő elemeket. Ha nagyobb címterületre van szükség, itt is árnyékmentesíteni kell előbb.

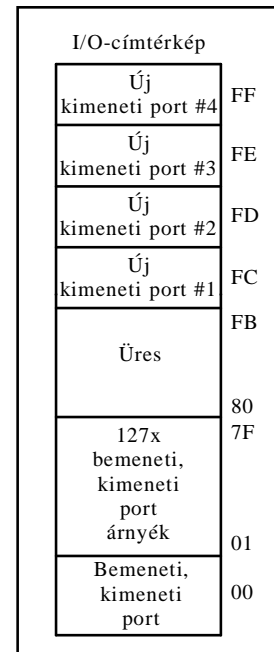
## 9.2. Feladat

Első feladatként bővítsük a Demo- $\mu$ C-t négy darab kimeneti porttal, amik I/O-ba legyenek ágyazva, legyenek árnyékmentesek, és a lehető legmagasabb címeken helyezkedjenek el!

Mivel az I/O-címtérképen a felső terület (128 címhellyel) üres, az eredeti kapcsolást nem kell változtatni, csak kiegészíteni. Ezért csak a kiegészítő részt fogjuk megrajzolni, valamint annak a buszokhoz csatlakozó részeit.

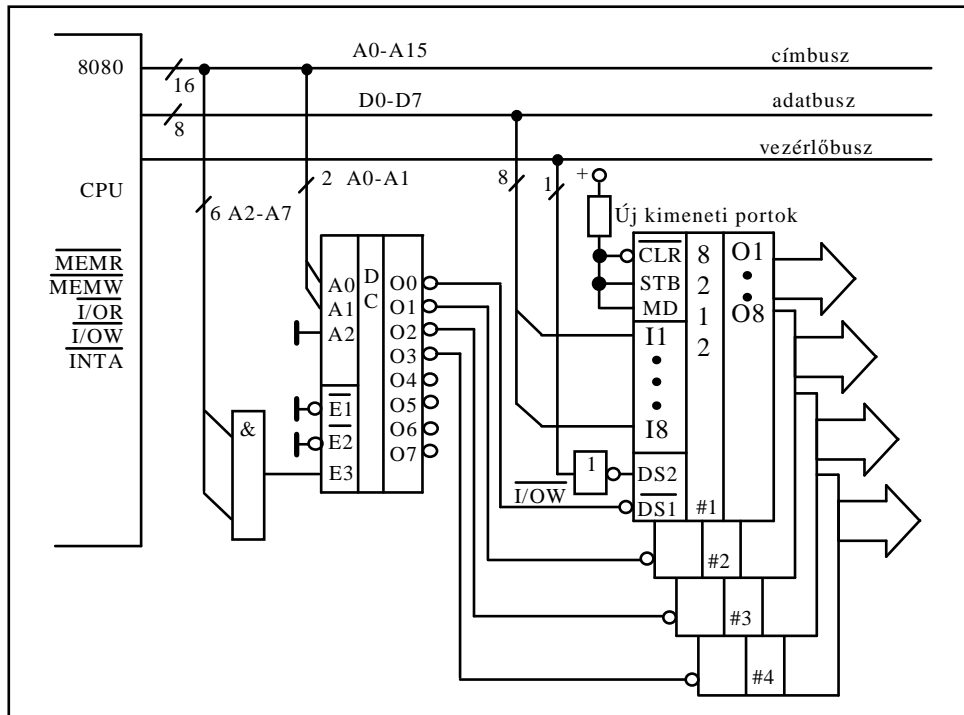
A **9.10. ábrán** megrajzoltuk az új I/O-címtérképet, rajta a négy új kimenőporttal, olyan elrendezésben, ami megfelel a feladat követelményeinek (I/O-ba vannak ágyazva, nincsenek árnyékait, és a legmagasabb címeken jelennek meg, az FC ... FF négy címértéken).

A címkezelést több módszerrel is megoldhatjuk, most a címdekódert fogjuk választani. A négy egyedi cím kiválasztásához a címdekóder az A0 és az A1 címbiteket fogja szétkódolni (4 x 1 jellegű a dekódolás). Azért, hogy a legfelső címterületen jelenjen meg ez a négy cím, a többi címbitnek (A2 ... A7) az 1 értékét kell összefogni, amit egyszerűen  $\bar{E}S$  kapuval lehet megoldani. Az  $\bar{E}S$  bemeneteire vezetjük ezeket a magasabb címbiteket, s a kapu kimenete csak akkor lesz 1 értékű, ha minden címbit 1 állapotú. Mivel most a kapu kimenetén a magas szint jelzi a megfelelő címtartományt, a címdekóder magas szinten aktív engedélyezőjelekként ( $\bar{E}3$ ) fogjuk hasznosítani. A kimeneti portokat az  $\overline{I/O}W$  jel működteti, így valósul meg az I/O-ba ágyazás.



9.10. ábra

A bővítő részleteket a **9.11. ábra** szemlélteti. Az eredeti kapcsolat minden részlete továbbra is szükséges, itt csak az új, bővítő elemek szerepelnek. Még egyszer elmondjuk, hogy ezt azért tehetjük meg, mert a bővítés miatt az eredeti kapcsoláson semmit nem kell változtatni, csak kiegészítésről van szó.



9.11. ábra

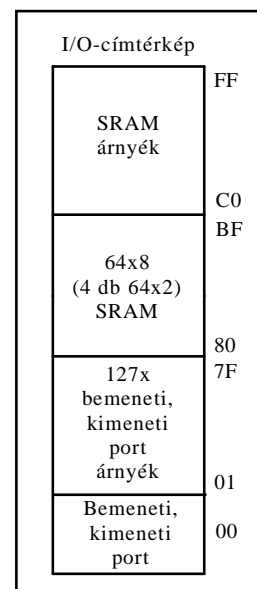
### 9.3. Feladat

Bővítsük a Demo- $\mu$ C-t négy darab I/O-ba ágyazott, 64 x 2 kapacitású, három vezérlőjeles SRAM-áramkörrel!

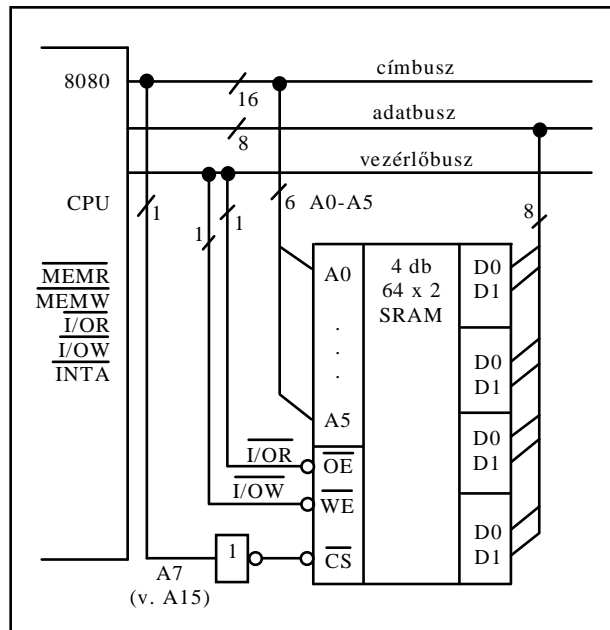
A négy darab memória-IC egyetlen memóriaegységet alkot, aminek a kapacitása 64 x 8. Ezt kell tehát elhelyezni az I/O címtérképen, tetszőleges helyre (mivel az elhelyezésére nézve a feladatban nincs kikötés, még az árnyékmentességet sem követeli meg a kiírás). Mivel az I/O-címtérkép felső (128 címhelyes) fele üres, továbbra sem kell az eredeti kapcsolást megváltoztatni, csak bővítő részleteket kell rajzolnunk!.

Helyezzük a 64 x 8 méretű SRAM-egységet a 80 ... BF címtartományra, s engedjük meg, hogy a felette lévő C0 ... FF címeken az árnyéka is megjelenjen! Így ugyanis rendkívül egyszerű lesz a címkijelölés, az A7 (vagy A15) és egy inverter lesz csak szükséges hozzá!

Az elmondottnak megfelelő I/O-címtérképet a **9.12. ábra** mutatja be. A kapcsolási rajz ismét csak a bővítő elemeket fogja tartalmazni. A memóriaegységet egyszerűsített technikával rajzoljuk meg, majd az ökölszabályok szerint bekötjük az adatpontokat és a saját címpontokat. Az üzemmódok vezérlésére az I/O-kezelő vezérlőjeleket használjuk fel. Végül a címkijelölés az A7 (vagy A15) címbittel történik meg, egy inverteren keresztül (így az A7 vagy az A15 magas szintje fogja működtetni a SRAM-áramkört). A bővítés kapcsolási rajzát a **9.13. ábrán** láthatjuk.



9.12. ábra



9.13. ábra

#### 9.4. Feladat

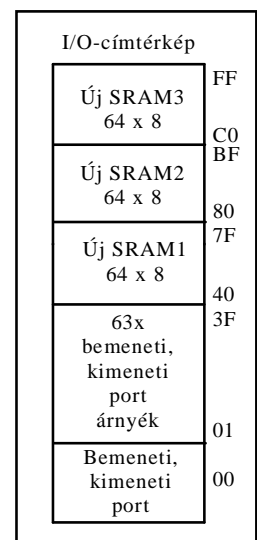
Bővítsük a Demo- $\mu$ C-t I/O-ba ágyazott három darab 64 x 8 kapacitású SRAM-áramkörrel. Ezek a lehető legmagasabb címeken helyezkedjenek el, árnyékmentesen, és legyenek három vezérlőjelesek.

Mivel a Demo- $\mu$ C eredeti I/O-címtérképén csak a felső 128 címhely üres, a három SRAM számára nincs elegendő hely. Át kell alakítanunk a mikroszámítógép eredeti I/O-rendszerét! Járjunk el úgy, hogy az eredeti portokat ne 128, csak 64 címhelyen engedjük megjelenni, s az ezután rendelkezésre álló 3 x 64 címértéket használjuk fel a SRAM-áramkörökhöz.

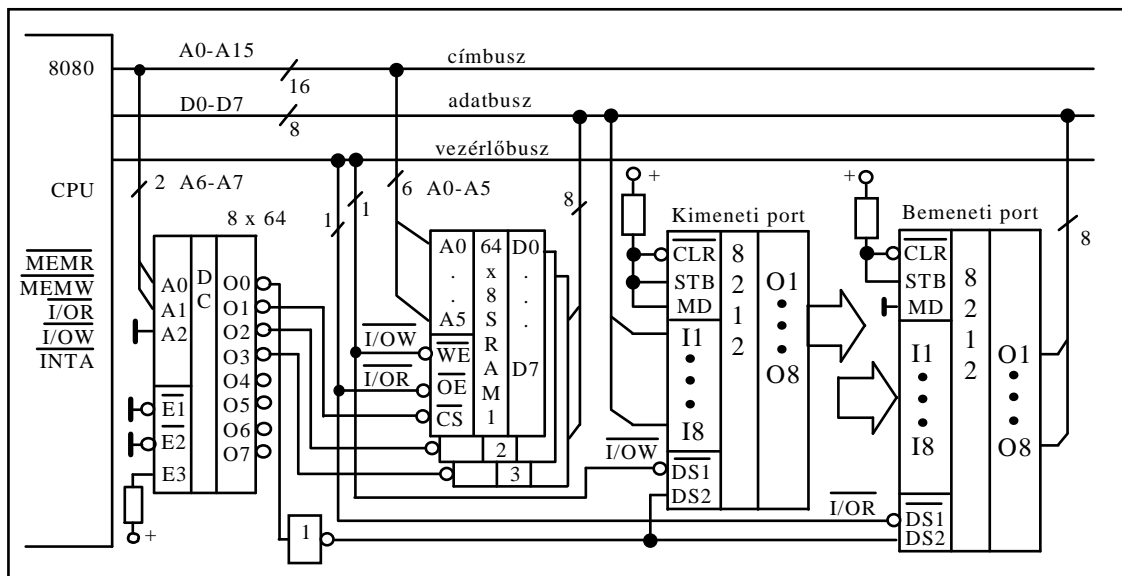
Az elképzelésünknek megfelelő I/O-címtérképet a **9.14. ábra** mutatja be. Jól látható, hogy a hardver kialakításához négy darab 64 címhelyes területet kell majd kijelölni, ezt a legegyszerűbben címdekóderrel tehetjük meg. A dekóder az A6, A7 (vagy A14, A15) címbiteket fogadja majd és azokat kódolja szét négyfelé. Magasabb címbitek nincsenek, így további árnyékmentesítéssel nem kell foglalkozni. A legkisebb című tartományt kapja meg a két port, így az eredeti valódi címük (00) továbbra is használható marad. A másik három kimenet engedélyezi az egyes SRAM-áramkörök működését. Az üzemmódokat az I/O-ba ágyazásnak megfelelően az I/O-kezelő vezérlőjelekkel kell megoldani.

Mivel az eredeti Demo- $\mu$ C nem teszi lehetővé ezt a bővítést, nem lehet egyszerűen a bővítő elemeket lerajzolnunk. Meg kell mutatnunk azt is, hogy az eredeti kapcsoláson milyen módosítást kell végrehajtani. A memória-rendszert nem érinti a változtatás, azt továbbra sem kell lerajzolnunk, tehát most az I/O-rendszer teljes rajzát kell megadni.

Felrajzoljuk a CPU-t és a buszrendszert, a címdekódert és az áramköri elemeket, a három SRAM-IC-t és a portokat. Az ökölszabályoknak megfelelően bekötjük az adatpontokat és a memóriák saját címbemeneteit, majd az I/O-ba ágyazáshoz szükséges vezérlőjeleket. A címdekóder az A6, A7 címbitek dekódolja, egy kimenete a portokat, három a memóriákat aktivizálja. A portokat működtető dekóder-kimenet inverteren át kell a DS2 pontokra vezetni, mivel ez az engedélyező bemenet magas aktív szintű. A bővített I/O-endszer kapcsolási rajzát a **9.15. ábrán** láthatjuk.



9.14. ábra



9.15. ábra

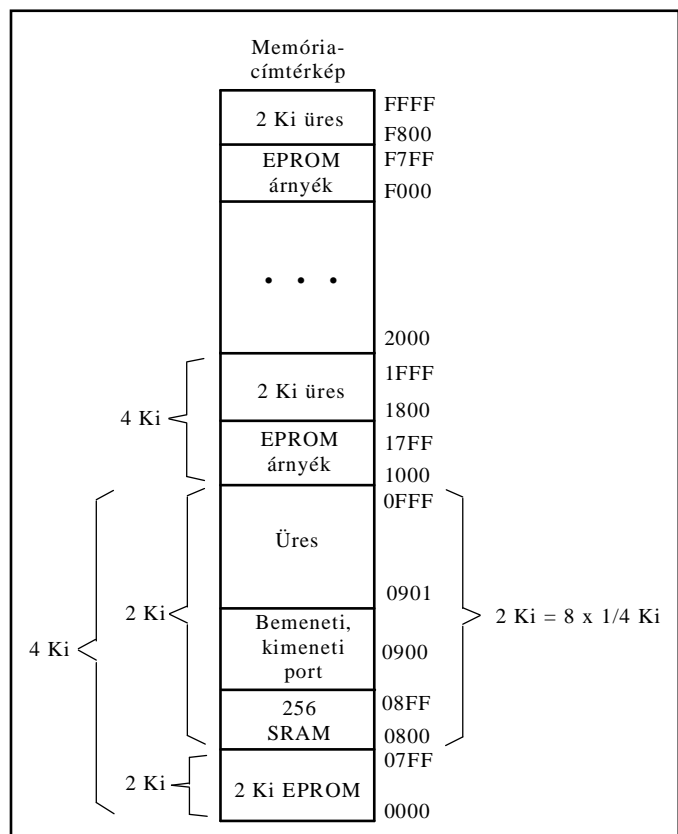
### 9.5. Feladat

Bővítjük a Demo- $\mu$ C-t egy-egy memóriába ágyazott bemenőporttal és kimenőporttal. A portok a lehető legkisebb címen jelenjenek meg, árnyékmentesen!

Mint a Demo- $\mu$ C elemzésekor kimutattuk, eredeti állapotában a memória-címterkép teljesen lefedett, semmilyen bővítésre sem ad lehetőséget. Az egy bemenőport és egy kimenőport összesen egyetlen címet igényelne a memória-címterképen, de ennyi sincs szabadon. Mindenképpen módosítani kell az eredeti kapcsolást, de ezt úgy kell megtenni, hogy az eredeti elemek valódi megjelenése ne változzon meg, az eredeti címtartományokon legyenek kezelhetők!

A portokat a SRAM áramköröket követő első címre lehet helyezni, így megfelelnek a feladat követelményeinek. Az új memória-címterképet a 9.16. ábra mutatja be. Az EPROM árnyékait meghagytuk, csak a SRAM címezésén módosítottunk, annak el is tűnnek az árnyékai. Az új portokat pedig árnyékmentesen kell beépítenünk.

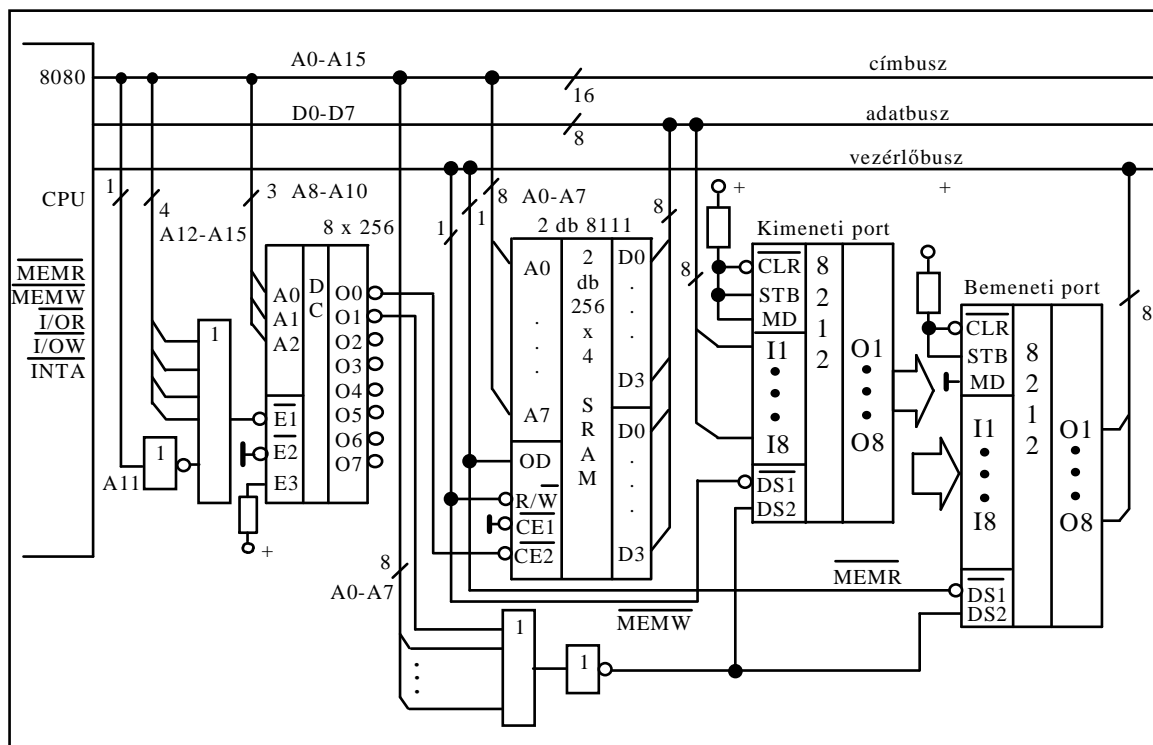
A címterkép még nem árul el mindent az áramkörtől, hiszen három eltérő címmennyiséget igénylő áramkört kell elhelyeznünk. Az EPROM szószáma 2 Ki, a SRAM 256 rekeszes, a portok egyetlen címet igényelnek. Egy címdekóderrel oldjuk meg a feladatot, amivel  $8 \times \frac{1}{4}$  Ki területet aktivizálunk, a 0800 ... 0FFF (2 Ki méretű) tartományban. Az első kimenete vezérli a SRAM-ot.



9.16. ábra



A második kimenet engedélyezi a portokat egy VAGY kapun át, amire még rávezetjük a nem kódolt alsóbb címbiteket (A0 ... A7). Így a portok nem a teljes ¼ Ki tartományban, csak annak legkisebb címén fognak működni (ahol az A0 ... A7 bitek mind 0 értékűek). A címdekóder az A8 ... A10 biteket kódolja szét, a 0800 ... 0FFF tartományra egy VAGY kapu segítségével helyezzük el, ami az A11-et inverteren át, az A12 ... A15 címbiteket közvetlenül fogadja. A kapcsolás a **9.17. ábrán** látható.



9.17. ábra

## 9.6. Feladat

Bővítsük a Demo- $\mu$ C-t két új 2716 EPROM-mal! Ezek legyenek memóriába ágyazva, az eredeti EPROM címtérületének folytatásában. A SRAM a három EPROM után jelenhet meg. Az EPROM-ok nem lehetnek árnyékosak!

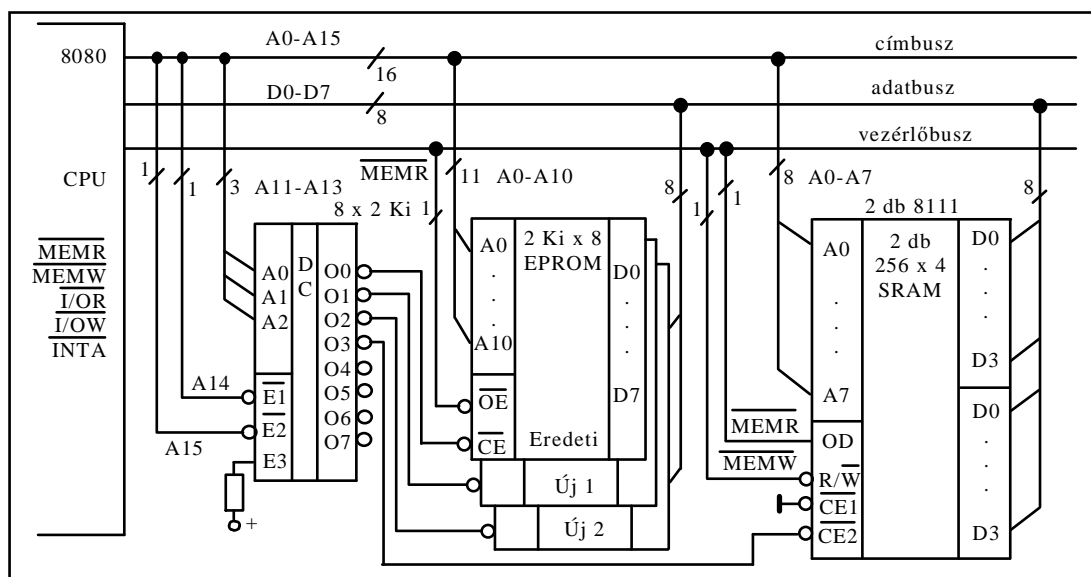
Az egyik lehetséges elrendezés, ami megfelel a feladat igényeinek, a 9.18. ábrán látható. A memória-címtérkép alsó három darab, 2 Ki széles tartományán helyezkedik el az eredeti 2716, majd felette a két új áramkör. Ezek után közvetlenül a SRAM aktivizálódik, de továbbra is egy teljes 2 Ki széles tartományban, ezért hét árnyéka követi. A továbbiakban a memóriaterület üres, magasabb címeken nem aktivizálódnak az áramkörök.

Jól láthatóan címdekóderrel oldható meg a feladat, úgy, hogy 8 x 2 Ki területet jelölünk ki a kimeneteivel. A szétkódolásra kerülő címbitek az A11, A12, A13 lesznek. Mivel ez a legalsó 8 x 2 Ki = 16 Ki, a magasabb címbiteket (A14, A15) a címdekóder alacsony aktív szintű két engedélyezőjeleként lehet felhasználni, így nem szükséges VAGY kapu alkalmazása.

A három EPROM azonos típus, ezért az egyszerűsítő ábrázolási módszer használható, egymás mögé lehet ezeket rajzolni. Mivel az I/O-rendszert ez a bővítés nem érintette, azt nem rajzoltuk meg, csak a CPU-t, a buszrendszert és a memóriarendszert. Az áramkört a **9.19. ábra** mutatja be. Ha a SRAM árnyékait is meg kívánjuk szüntetni, a kijelölőjelét VAGY kapun vezetjük át és rácsatlakoztatjuk az A8 ... A10 címbiteket.

Memória-címtérkép	
Üres	FFFF
	2000
7x SRAM árnyék	1FFF
	1900
256x8 SRAM	18FF
	1800
2 Ki EPROM új 2	17FF
	1000
2 Ki EPROM új 1	0FFF
	0800
2 Ki EPROM eredeti	07FF
	0000

9.18. ábra



9.19. ábra

### 9.7. Feladat

Bővítsük a Demo- $\mu$ C-t az eredetit folytató, memóriába ágyazott 6 darab 256 x 4 kapacitású SRAM áramkörrel, és két bemenőporttal, két kimenőporttal, melyeket I/O-ba ágyazunk, az eredeti portok utáni címeken. A portok legyenek árnyékmentesek!

Miután végiggondolta a feladatot, írja fel, milyen egységekkel kell bővítenie a memóriarendszert!

és az I/O-rendszert:

Az I/O-rendszernél kell-e változtatni az eredeti kapcsoláson, vagy csak bővíteni kell? Indoklást is írjon!

A memóriarendszernél kell-e változtatni az eredeti kapcsoláson, vagy csak bővíteni kell? Indoklást is írjon!

Rajzolja meg az új I/O-címtérképet!

Rajzolja meg az új memória-címtérképet!

Készítse el az új kapcsolási rajzot!

## 10. Megoldások

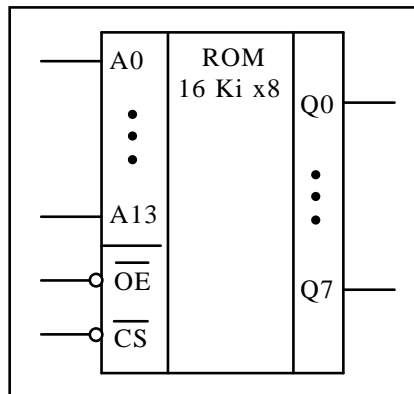
### 1.1. Feladat

13 darab címbemenet van ( $A_0 \dots A_{12}$ ), tehát a szószám 8 Ki.

8 darab adatbemenet van, így a szóhosszúság 8 bit.

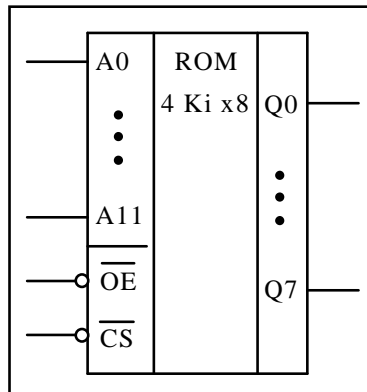
A memória kapacitása tehát  $8 \text{ Ki} \times 8$ .

### 1.2. Feladat



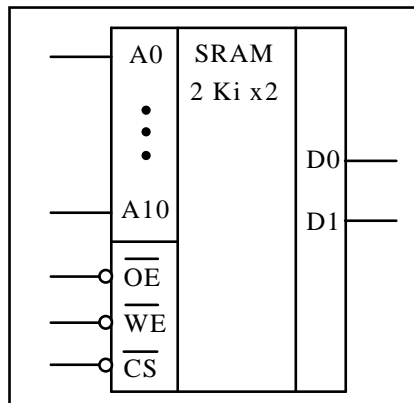
10.1. ábra

### 1.3. Feladat



10.2. ábra

### 1.6. Feladat



10.3. ábra

### 1.7. Feladat

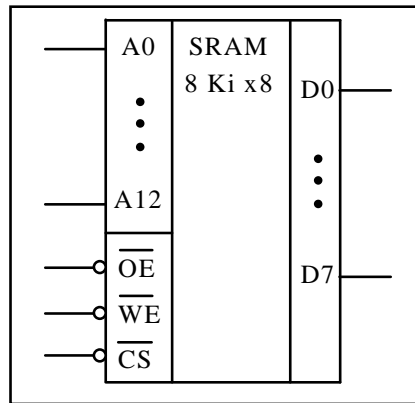
8 darab címbemenet (A0 ... A7) van, tehát a szószám 256 (1/4 Ki).

4 darab adatpont van, tehát a szóhosszúság 4 bit.

A kapacitás 256 x 4, azaz 1/4 Ki x 4.

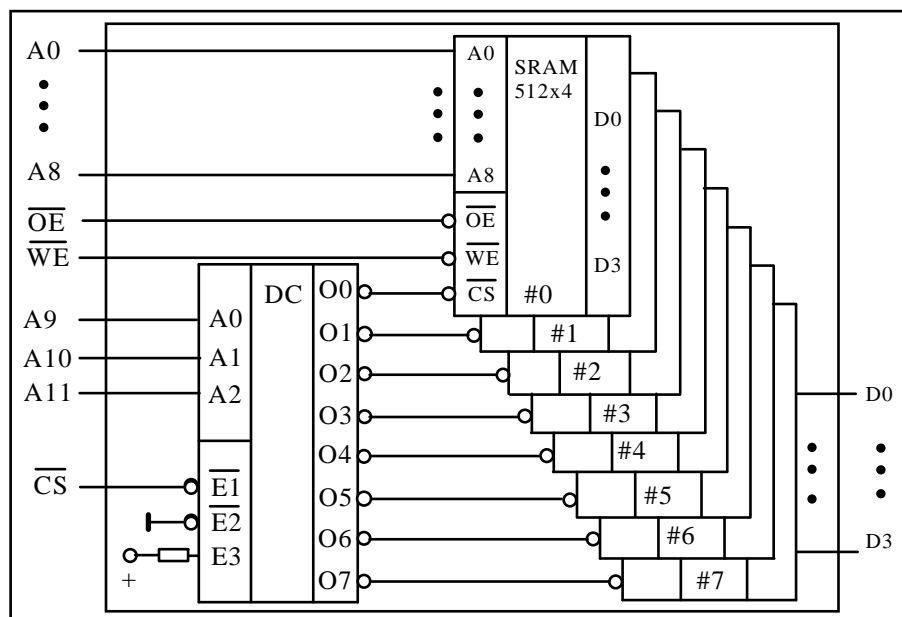
A címhatárok: 00 ... FF.

### 1.8. Feladat



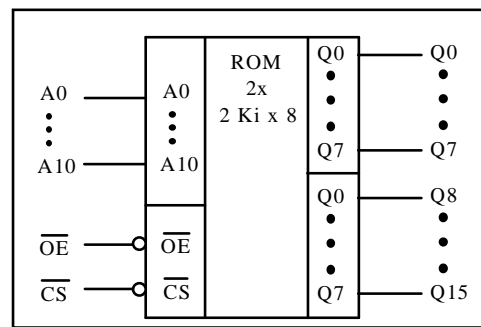
10.4. ábra

### 2.2. Feladat



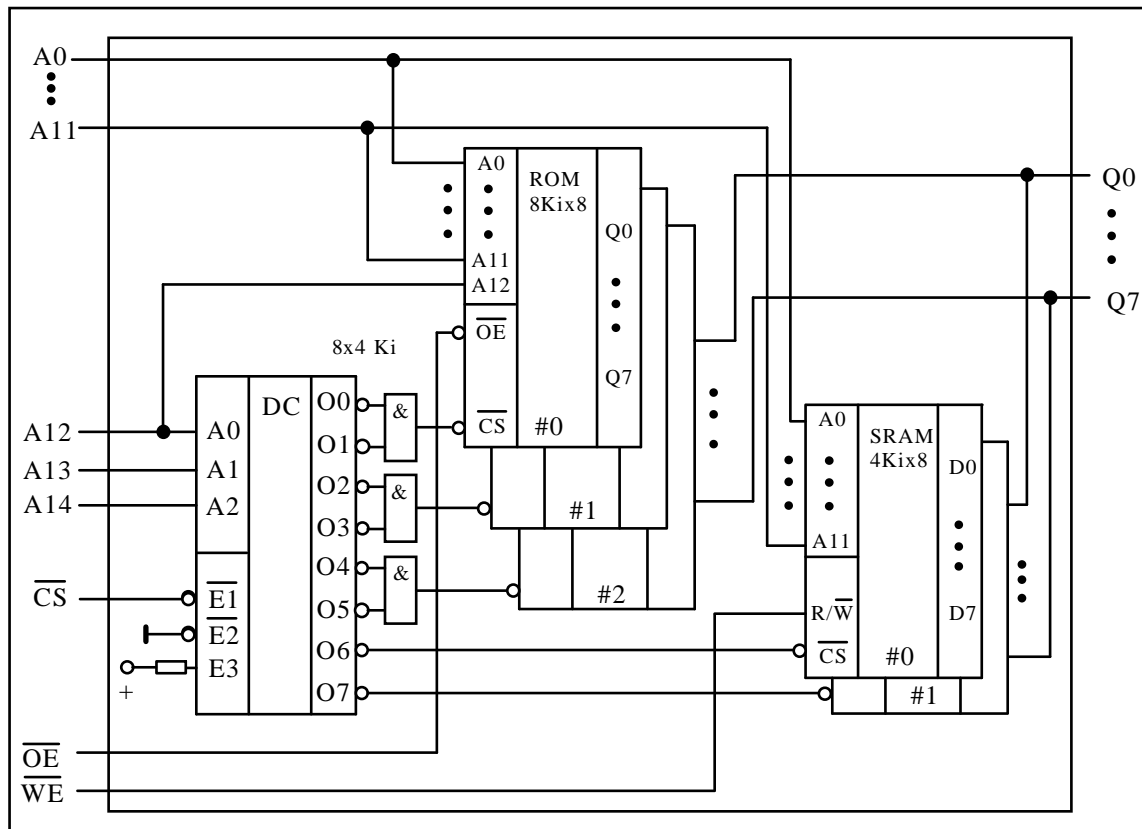
10.5. ábra

## 2.4. Feladat



10.6. ábra

## 3.3. Feladat



10.7. ábra

### 3.4. Feladat

A következő kérdéseket tegye fel magának!

Mekkora a legkisebb címterület, amit a címdekóderrel ki kell majd jelölni?

Hány ekkora méretű terület fedi le a teljes kezelt címterületet?

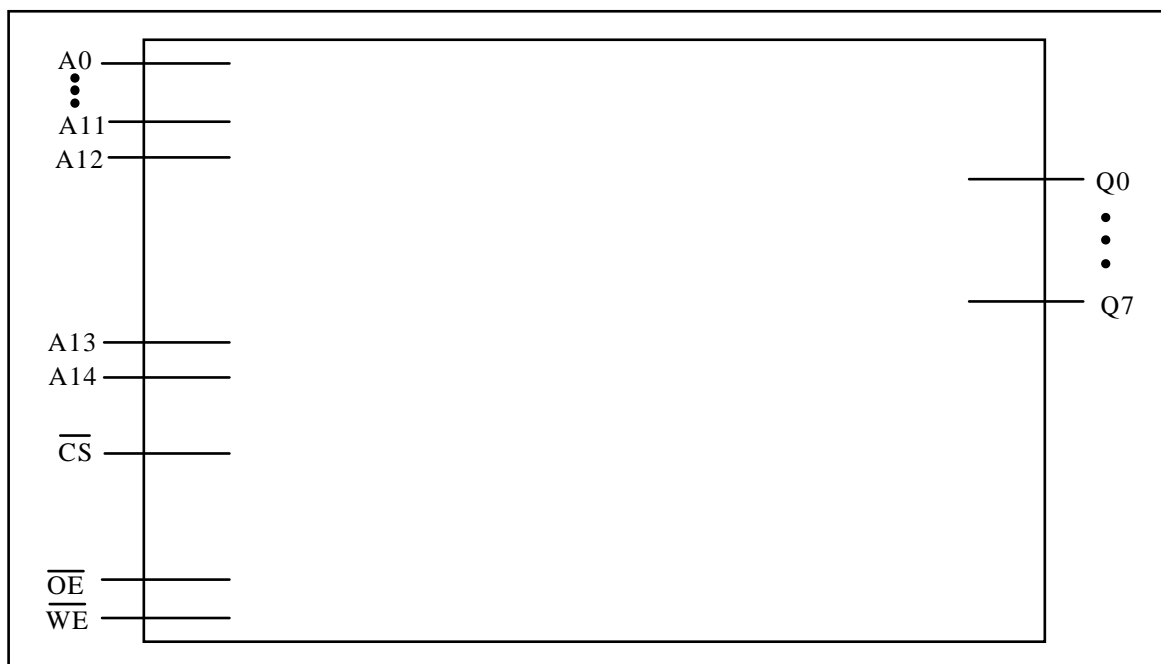
Hány kimenetű címdekóderre van szükség?

Melyik a legkisebb, dekódolásra kerülő címbit?

Hány címbitet kell dekódolni?

Melyek lesznek a dekódolásra kerülő címbitek?

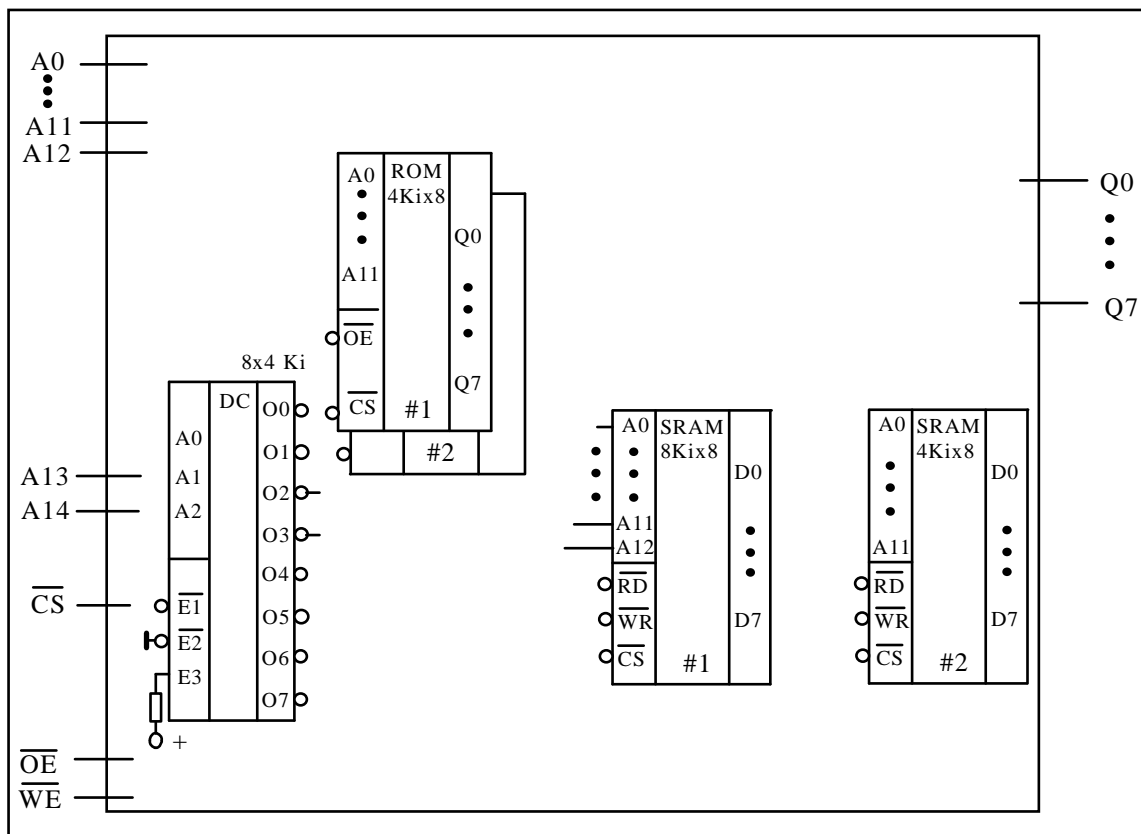
Ellenőrizze a memóriakártya csatlakozópontjait (ne ide rajzolja be az áramköröket)!



10.8. ábra

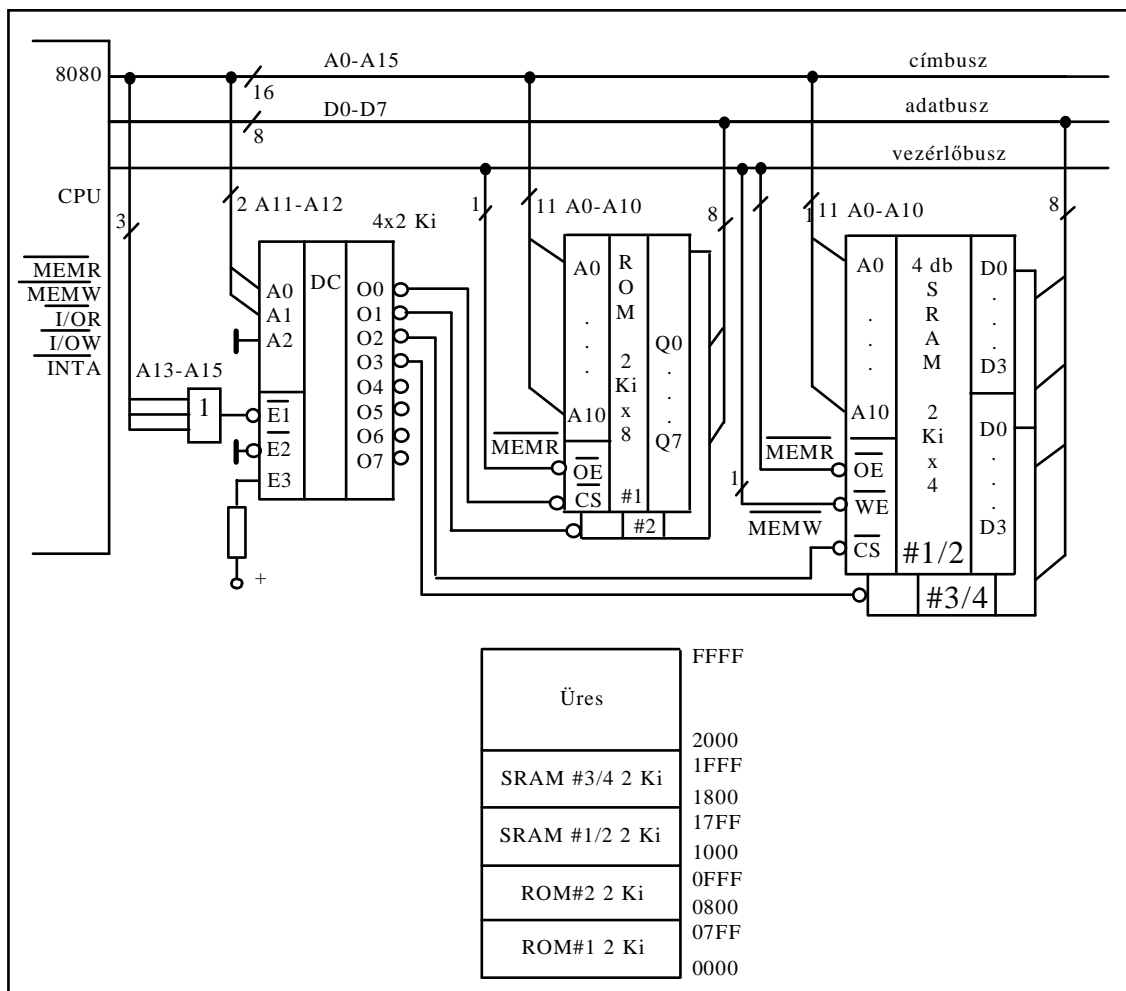


Az ábrába berajzoltuk a felhasználásra kerülő IC-eket, az előzőek alapján végezze el a csatlakozópon-  
tok és az áramkörök összekötését!



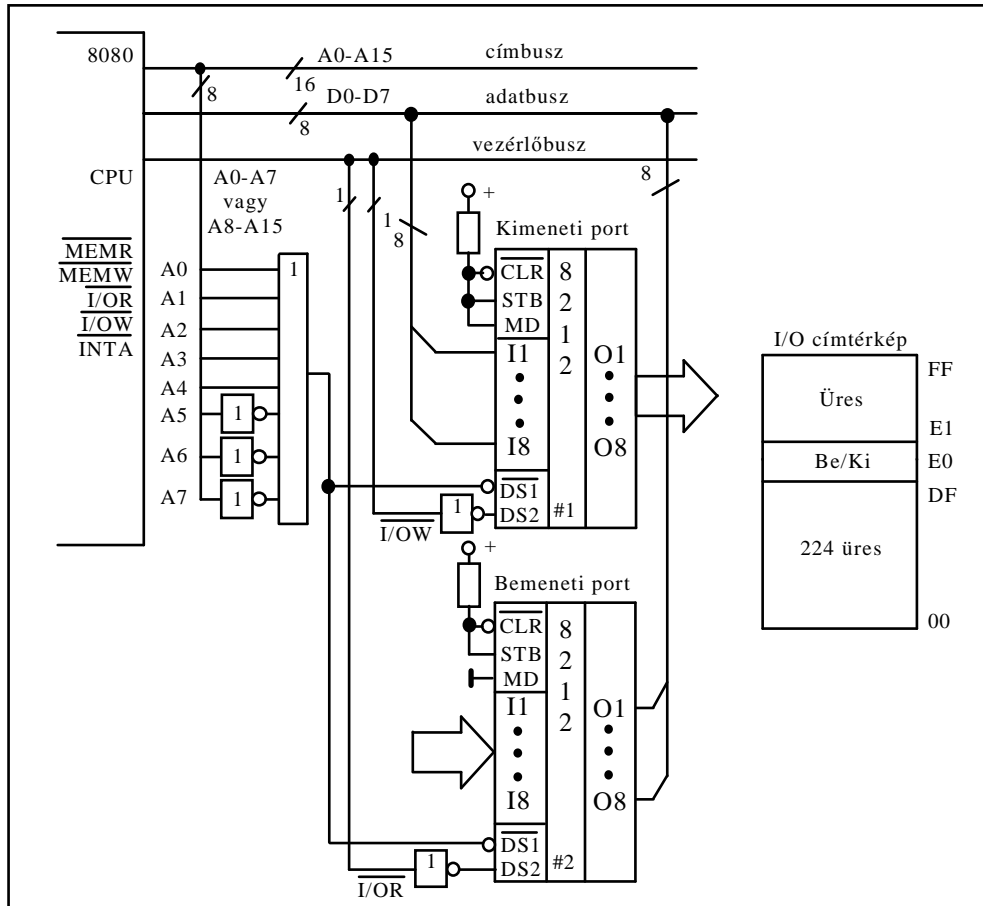
10.9. ábra

#### 4.4. Feladat



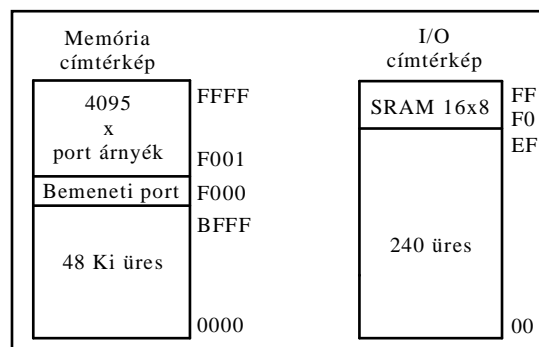
10.10. ábra

## 5.2. Feladat



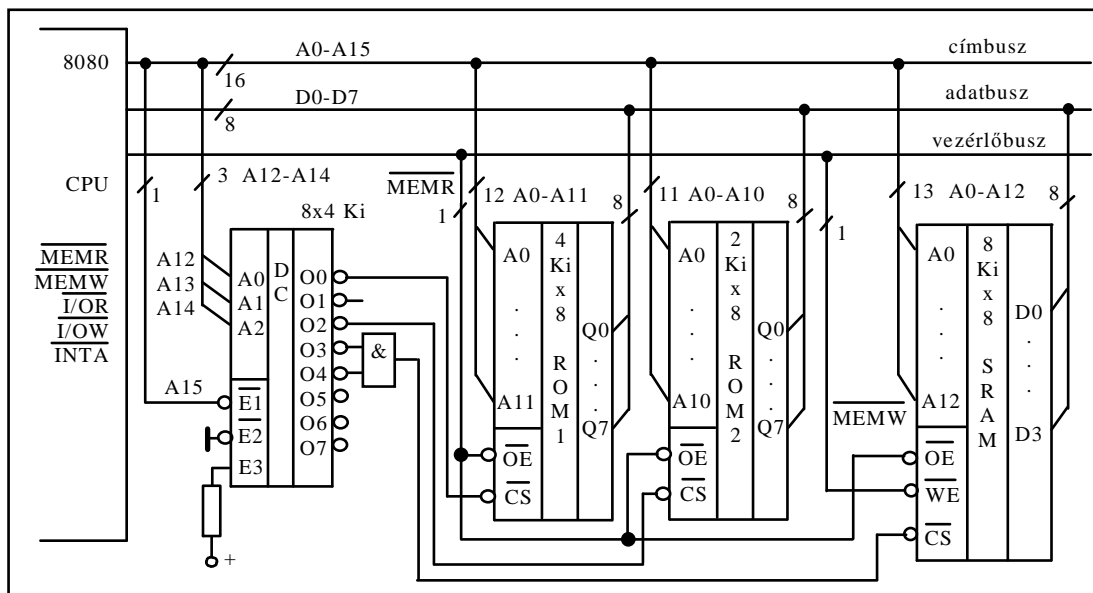
10.11. ábra

## 6.2. Feladat



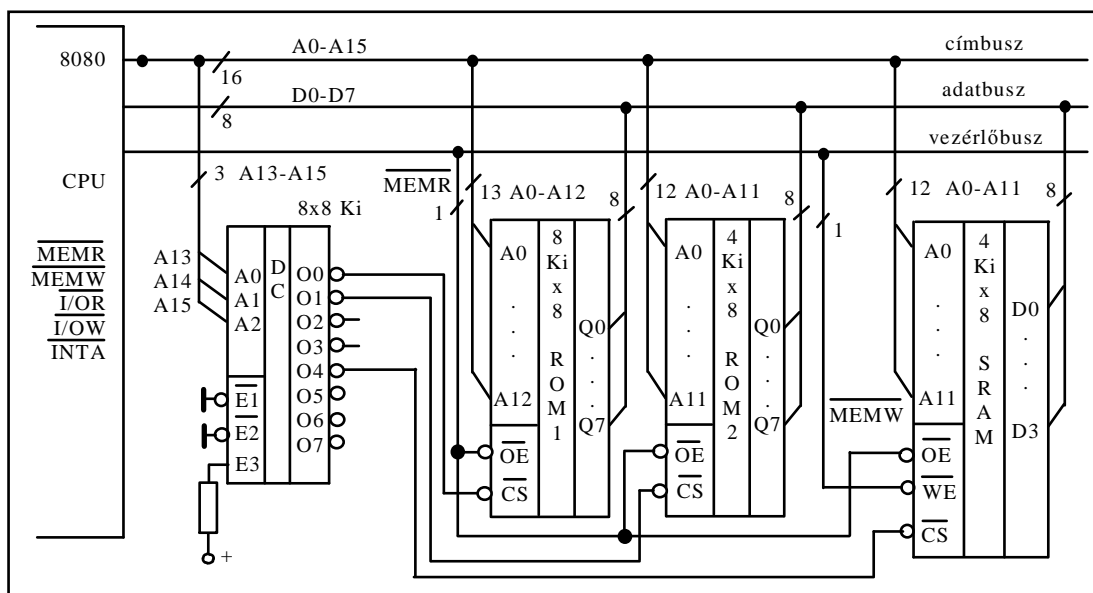
10.12. ábra

### 7.3. Feladat



10.13. ábra

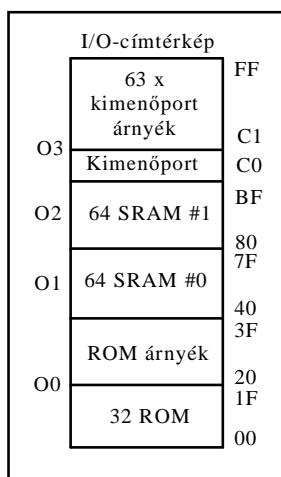
### 7.4. Feladat



10.14. ábra

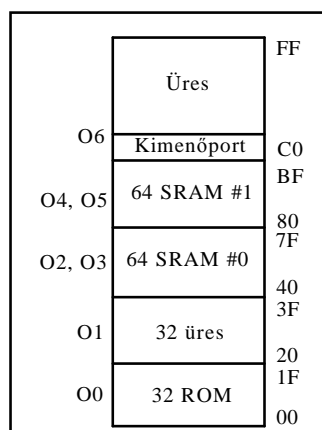
## 8.2. Feladat

Az eredeti kapcsolás I/O-címtérképe:



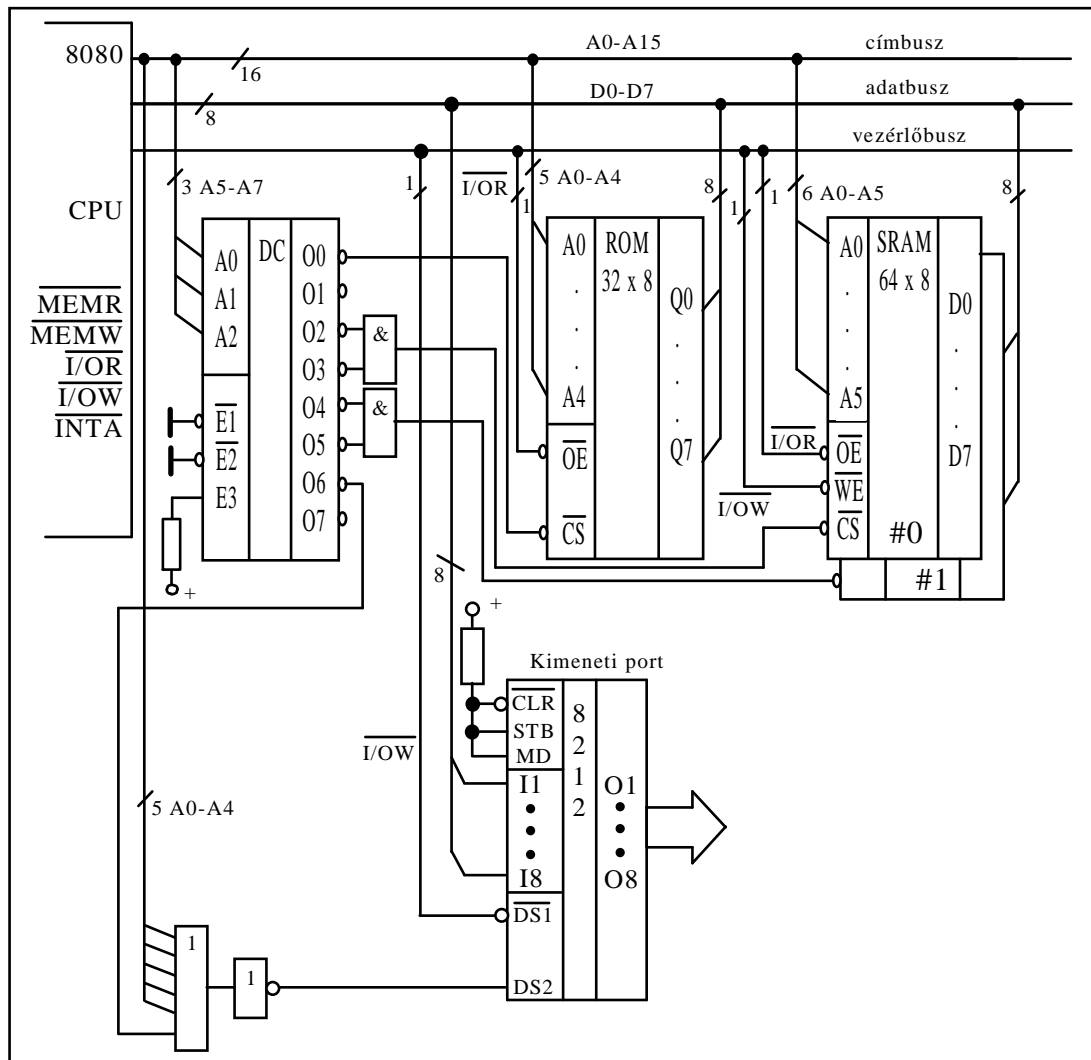
10.15. ábra

Az árnyékmentesített kapcsolás I/O-címtérképe:



10.16. ábra

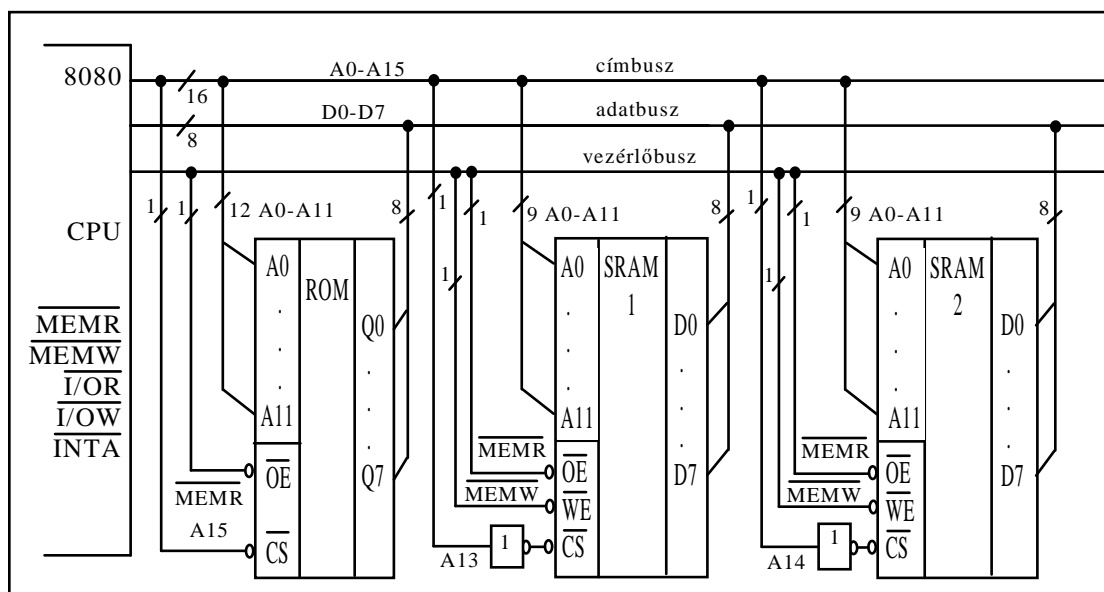
Az árnyékmentesített kapcsolás:



10.17. ábra

## 8.5. Feladat

A SRAM2 memória-áramkörrel kiegészített mikroszámítógép kapcsolási rajza:



10.18. ábra

Az egyes memória-áramkörök aktivitását jelző memória-címtérkép:

Memória-címtérkép				
FFFF      8000 7FFF     0000	32 Ki,  EPROM  nem aktív	8 Ki, SRAM1 aktív	16 Ki, SRAM2 aktív	FFFF
		8 Ki, SRAM1 nem aktív		E000 DFFF
		8 Ki, SRAM1 aktív	16 Ki, SRAM2 nem aktív	C000 BFFF
		8 Ki, SRAM1 nem aktív		A000 9FFF
	32 Ki,  EPROM  aktív	8 Ki, SRAM1 aktív	16 Ki, SRAM2 aktív	8000 7FFF
		8 Ki, SRAM1 nem aktív		6000 5FFF
		8 Ki, SRAM1 aktív	16 Ki, SRAM2 nem aktív	4000 3FFF
		8 Ki, SRAM1 nem aktív		2000 1FFF  0000

10.19. ábra



A valódi működések és az árnyék-területek a memória-címtérképen:

Memória-címtérkép								
FFFF	EPROM		SRAM1 árnyék	SRAM2 árnyék	FFFF			
			SRAM1 árnyék	SRAM2 árnyék		E000 DFFF		
		nem	SRAM1 nem aktív	SRAM2 árnyék				
				SRAM2 árnyék				
		aktív	SRAM1 árnyék	SRAM2 nem aktív		C000 BFFF		
			SRAM1 árnyék					
			SRAM1 nem aktív					
		8000						8000
		7FFF	EPROM árnyék	SRAM1 árnyék		SRAM2 árnyék	7FFF	
		7000						
6FFF	EPROM árnyék	SRAM1 árnyék	SRAM2 árnyék					
6000				6000				
5FFF	EPROM árnyék	SRAM1 nem aktív	SRAM2 árnyék	5FFF				
5000								
4FFF	EPROM árnyék		SRAM2					
4000				4000				
3FFF	EPROM árnyék	SRAM1 árnyék	SRAM2 nem aktív	3FFF				
3000								
2FFF	EPROM árnyék	SRAM1						
2000								
1FFF	EPROM árnyék	SRAM1 nem aktív						
1000								
0FFF								
0000	EPROM 4 Ki			0000				

10.20. ábra

A tiltott címtartományok kijelölése, majd ezek alapján az egyes memória-elemek valódi címtartományának megválasztása:

Memória-címtérkép				
FFFF		SRAM1 árnyék	SRAM2 árnyék	FFFF
		SRAM1 árnyék	SRAM2 árnyék	
	EPROM	SRAM1 nem aktív	SRAM2 árnyék	E000
			SRAM2 árnyék	DFFF
	nem	SRAM1 árnyék		C000
		SRAM1 árnyék		BFFF
	aktív		SRAM2 nem aktív	A000
		SRAM1 nem aktív		9FFF
8000				8000
7FFF	EPROM árnyék	SRAM1 árnyék	SRAM2 árnyék	7FFF
7000				
6FFF	EPROM árnyék	SRAM1 árnyék	SRAM2 árnyék	6000
6000				5FFF
5FFF	EPROM árnyék	SRAM1	SRAM2 árnyék	
5000		nem aktív		
4FFF	EPROM árnyék		SRAM2	4000
4000				3FFF
3FFF	EPROM árnyék	SRAM1 árnyék		
3000				
2FFF	EPROM árnyék	SRAM1		2000
2000			SRAM2 nem aktív	1FFF
1FFF	EPROM árnyék	SRAM1 nem aktív		
1000				
0FFF	EPROM 4 Ki			
0000				0000

10.21. ábra

## Tartalomjegyzék

<b>Bevezetés.....</b>	<b>3</b>
<b>1. A memória-áramkörök és az I/O-egységek vezérlése.....</b>	<b>5</b>
<b>2. Memória bővítése szószámra, szóhosszra.....</b>	<b>11</b>
<b>3. Önálló memóriaegységek, memóriakártyák kialakítása.....</b>	<b>19</b>
<b>4. Memóriák csatlakoztatása buszrendszerhez, címtérképek.....</b>	<b>25</b>
<b>5. I/O-elemek csatlakoztatása buszrendszerhez.....</b>	<b>41</b>
<b>6. A négyféle tervezési lehetőség: memóriába ágyazott memória és I/O, I/O-ba     ágyazott memória és I/O .....</b>	<b>51</b>
<b>7. Logikai kapcsolások tervezése címtérképek alapján.....</b>	<b>63</b>
<b>8. Kész mikroszámítógép memória- és I/O-rendszerének elemzése .....</b>	<b>73</b>
<b>9. Kész mikroszámítógép memória- és I/O-rendszerének bővítése .....</b>	<b>89</b>
<b>10. Megoldások .....</b>	<b>101</b>