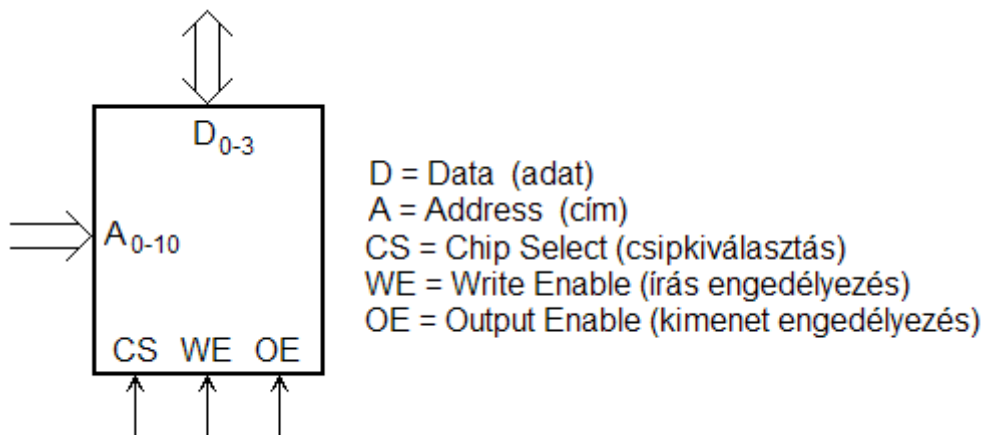


### III. MEMÓRIABLOKK TERVEZÉSE

A memóriablokk tervezésének menetét **egy példán** keresztül fogjuk bemutatni:

Egy 16-bites adatbusszal rendelkező rendszer számára tervezzen meg egy 8 KB-os statikus RAM memóriablokkot, amely az A1000 (hexa) címtől kezdődjön. A felhasználható memória-áramköröknek az alábbi logikai vázlata van:



### MEGOLDÁS

#### 1. Meghatározzuk a szükséges áramkörök számát és elosztjuk tömbökre.

A logikai vázlatról megállapítjuk, hogy

- 11 címbemenete van  $\Rightarrow$  a kapacitása  $2^{11} = 2 \times 2^{10} = 2 \text{ K}$
- 4 adatvonala van  $\Rightarrow$  mindegyik rekesze 4 bites.

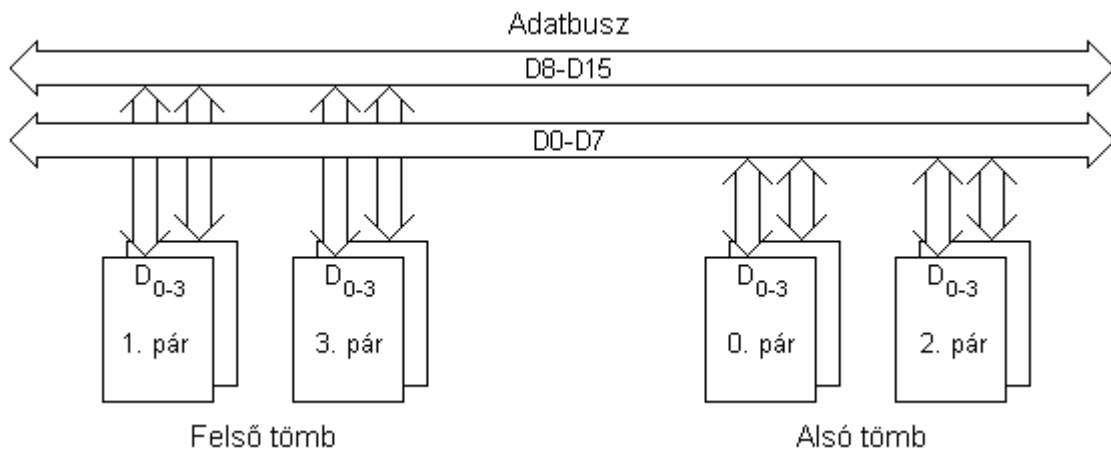
Ebből következik, hogy a memóriacsip szervezése  $2 \text{ K} \times 4 \text{ bit}$ .

A feladat meghatározásában  $8 \text{ KB} = 8 \text{ K} \times 8 \text{ bites}$  kapacitást kértek, tehát:

$$\Rightarrow \text{a szükséges áramkörök száma} = \frac{8 \text{ K} \times 8}{2 \text{ K} \times 4} = 8 \text{ csip.}$$

Mivel alapértelmezetten minden mikroszámítógép memóriája bájtstruktúrájú (8 bites rekeszek), és a mi áramköreink 4 bitesek, ezeket kettesével kell csoportosítsuk.

Mivel a rendszer adatbusza  $2 \times 8 = 16 \text{ bites}$ , az áramkörpárokat két tömbben kell elhelyeznünk (az adatbusz minden 8 vonalas egysége számára egy-egy tömb):



## 2. Elkészítjük a címhatárok táblázatát.

Mivel a megadott kezdőcím 20 bites, a processzor címbusza 20 vonallal rendelkezik. Következésképpen az áramkörök címhatárainak táblázata az A0-tól A19-ig terjedő címeket fogja tartalmazni (a legkisebb helyiértékű címmel a jobboldalon).

A táblázatban függőlegesen elkülönítjük a címcsoportokat, a kiválasztásban betöltött szerepük szerint.

Vízszintesen az áramkörpárok által elfoglalt tartományokat különítjük el. Mindegyik zónában specifikáljuk a kezdő- és befejezőcímet, valamint egy csipkiválasztó jelet (amely – figyeljük meg – a baloldalon megmaradt bitek függvénye):

Csipkiválasztás								Rekeszkiv.			Tömb kiv.	Tartomány	Áramkörök	Szelekció
A19	A18	A17	A16	A15	A14	A13	A12	A11	...	A1	A0			
1	0	1	0	0	0	0	1	0	...	0	0	A1000 A1FFF	0. és 1. pár (4 KB)	SEL <sub>01</sub> =1
1	0	1	0	0	0	0	1	1	...	1	1			
1	0	1	0	0	0	1	0	0	...	0	0	A2000 A2FFF	2. és 3. pár (4 KB)	SEL <sub>23</sub> =1
1	0	1	0	0	0	1	0	1	...	1	1			

## 3. Elkészítjük a dekóder logikai tervét (a kiválasztó jelek generálására).

### A. megoldás

Felírjuk a kiválasztó jelek logikai függvényét a címhatárok táblázata alapján:

$$SEL_{01} = A19 \cdot \overline{A18} \cdot A17 \cdot \overline{A16} \cdot \overline{A15} \cdot \overline{A14} \cdot \overline{A13} \cdot A12$$

$$SEL_{23} = A19 \cdot \overline{A18} \cdot A17 \cdot \overline{A16} \cdot \overline{A15} \cdot \overline{A14} \cdot A13 \cdot \overline{A12}$$

A függvények alapján logikai kapukból implementáljuk a kiválasztó jeleket előállító áramkört.

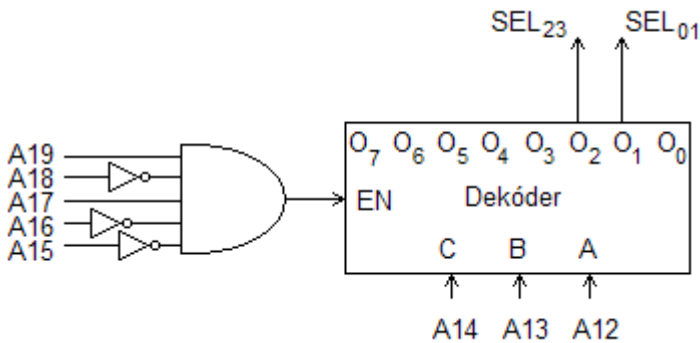
### B. megoldás

Választunk egy dekódoló áramkört, például a *Függelékben* megadott 3 bemenettel,  $2^3 = 8$  kimenettel és 1 engedélyező bemenettel rendelkezőt.

A dekóder bemeneteire helyezzük a csipkiválasztás számára a táblázatban megmaradt címvonalakat, a legkisebb helyiértékűvel kezdődően (A12). A többi bitből, a táblázat alapján, felírjuk a dekóder engedélyező jelének az egyenletét:

$$EN = A_{19} \cdot \overline{A_{18}} \cdot A_{17} \cdot \overline{A_{16}} \cdot \overline{A_{15}}$$

Így elkészíthetjük a kiválasztó jeleket előállító áramkör rajzát, amelyen a két kiválasztó jelet a dekóder O<sub>1</sub>-es és O<sub>2</sub>-es kimenetén kapjuk meg (ezek aktiválódnak az A14 A13 A12 bitek 001-es és 010-ás kombinációjánál):



#### 4. Elkészítjük a memóriablokk teljes logikai tervrajzát.

A memóriacsipek címbemeneteire az áramkörökön belüli rekeszkiválasztásra szánt jeleket alkalmazzuk, a táblázatnak megfelelően (A1–A11).

Az áramkörök csipkiválasztó (CS) jeleinek generálásánál figyelembe kell venni a tömbök kiválasztását, amely a processzor által szolgáltatott BE (Byte Enable) jelekkel történik:

- BE0 – alsó tömb kiválasztása (amelyre jellemző A0 = 0)
- BE1 – felső tömb kiválasztása (amelyre jellemző A0 = 1)

Az adatok átviteléhez alsó tömb a D0–D7, a felső a D8–D15 adatvonalakat használja.

A tömbök kiválasztását is figyelembe véve, a csipkiválasztó jeleket az alábbi egyenletek alapján kapjuk meg:

$$CS_0 = SEL_{01} \cdot BE_0$$

$$CS_1 = SEL_{01} \cdot BE_1$$

$$CS_2 = SEL_{23} \cdot BE_0$$

$$CS_3 = SEL_{23} \cdot BE_1$$

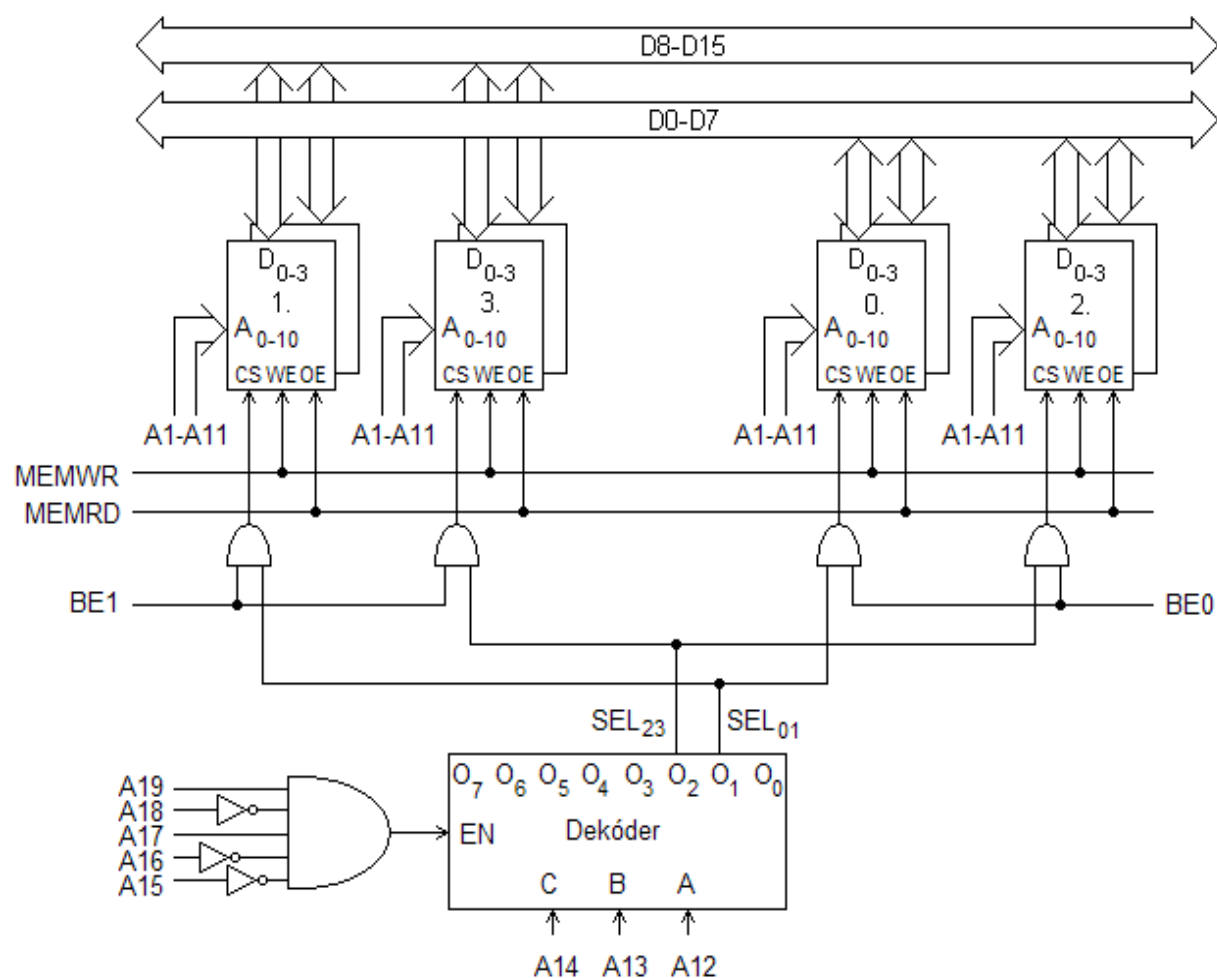
Az áramkörök írást engedélyező bemenetére (WE) a memóriairási vezérlőjelet alkalmazzuk, amely a vezérlőbuszon érkezik a processzortól:

$$WE = \overline{MEMWR}$$

Az áramkörök kimeneteinek engedélyezésére (OE) olvasásnál van szükség, tehát ezeket a memóriaolvasási vezérlőjellel aktiváljuk:

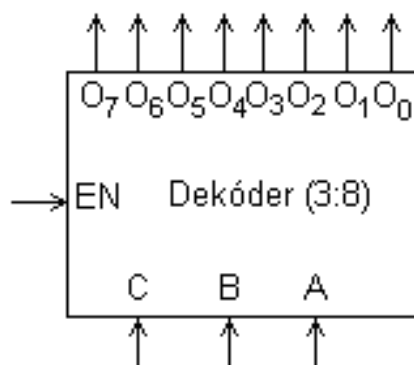
$$OE = \overline{MEMRD}$$

Most már elkészíthetjük a memóriablokk végleges rajzát:



## FÜGGELÉK

A 3 bemenetű és  $2^3 = 8$  kimenetű **dekóder** áramkörnek a logikai vázlatát és a működési táblázatát tartalmazza az alábbi ábra. Az EN (*Enable*) bemenet a dekódolási funkció engedélyezését jelenti.

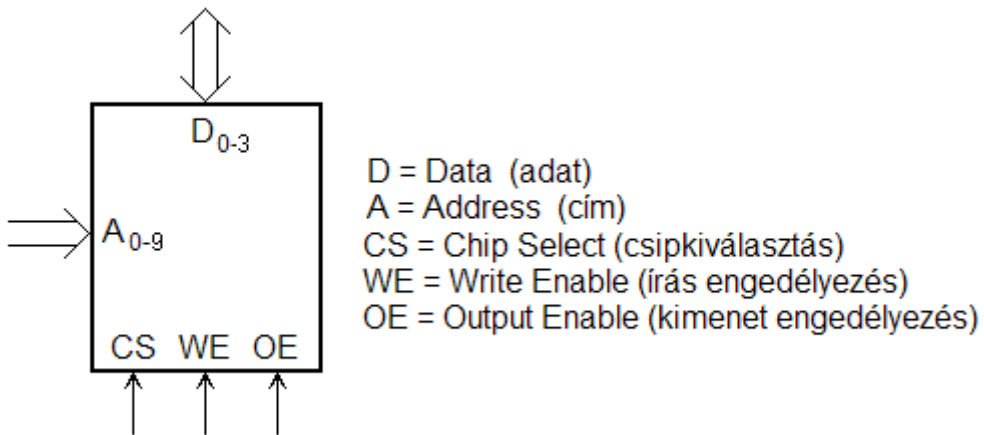


EN	C	B	A	O <sub>7</sub>	O <sub>6</sub>	O <sub>5</sub>	O <sub>4</sub>	O <sub>3</sub>	O <sub>2</sub>	O <sub>1</sub>	O <sub>0</sub>
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

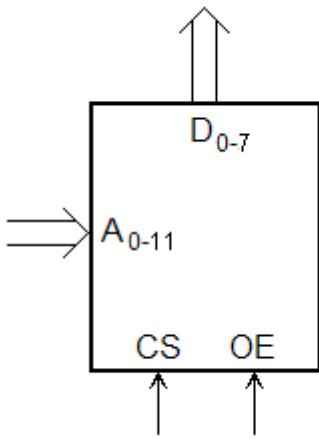
## FELADATOK

III.1. Egy 8-bites adatbusszal rendelkező rendszer számára tervezzen meg egy 8 KB-os ROM memóriablokkot, amely a 4000 (hexa) címtől kezdődjön. A felhasználható memória-áramköröknek  $2\text{ K} \times 8$  bites szervezése van.

III.2. Egy 8-bites adatbusszal rendelkező rendszer számára tervezzen meg egy 6 KB-os RAM memóriablokkot, amely a 6000 (hexa) címtől kezdődjön. A felhasználható memória-áramköröknek az alábbi logikai vázlata van:



III.3. Egy 16-bites adatbusszal rendelkező rendszer számára tervezzen meg egy 16 KB-os ROM memóriablokkot, amely az F0000 (hexa) címtől kezdődjön. A felhasználható memória-áramköröknek az alábbi logikai vázlata van:



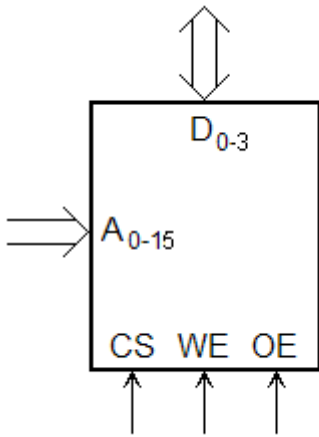
D = Data (adat)  
 A = Address (cím)  
 CS = Chip Select (csipkiválasztás)  
 OE = Output Enable (kimenet engedélyezés)

III.4. Egy 16-bites adatbusszal rendelkező rendszer számára tervezzen meg egy 48 KB-os ROM memóriablokkot, amely a C0000 (hexa) címtől kezdődjön. A felhasználható memória-áramköröknek  $8\text{ K} \times 8$  bites szervezése van.

III.5. Egy 16-bites adatbusszal rendelkező rendszer számára tervezzen meg egy 32 KB-os RAM memóriablokkot, amely a 40000 (hexa) címtől kezdődjön. A felhasználható memória-áramköröknek  $8\text{ K} \times 8$  bites szervezése van.

III.6. Egy 32-bites adatbusszal rendelkező rendszer számára tervezzen meg egy 2 MB-os ROM memóriablokkot, amely a D00000 (hexa) címtől kezdődjön. A felhasználható memória-áramköröknek  $256\text{ K} \times 8$  bites szervezése van.

III.7. Egy 32-bites adatbusszal rendelkező rendszer számára tervezzen meg egy 256 KB-os statikus RAM memóriablokkot, amely a 380000 (hexa) címtől kezdődjön. A felhasználható memória-áramköröknek az alábbi logikai vázlata van:



D = Data (adat)  
 A = Address (cím)  
 CS = Chip Select (csipkiválasztás)  
 WE = Write Enable (írás engedélyezés)  
 OE = Output Enable (kimenet engedélyezés)

III.8. Egy 32-bites adatbusszal rendelkező rendszer számára tervezzen meg egy 1 MB-os statikus RAM memóriablokkot, amely a 800000 (hexa) címtől kezdődjön. A felhasználható memória-áramköröknek  $128\text{ K} \times 8$  bites szervezése van.