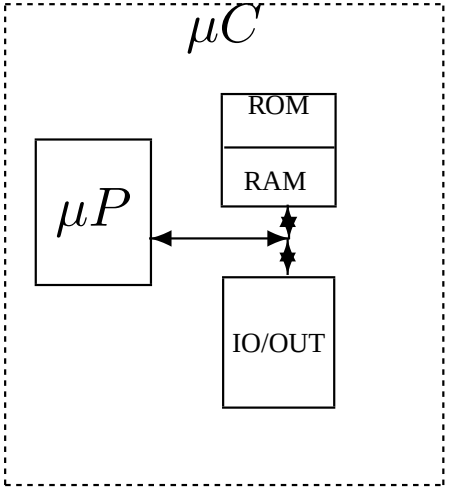
Pamięci półprzewodnikowe - wykład 7

Adam Szmigielski
aszmigie@pjwstk.edu.pl

Pamięć RAM i ROM jako element składowy μC



- RAM - *Random Access Memory* (pamięć z dostępem swobodnym) pamięć, umożliwiająca zarówno odczyt jak i zapis informacji. Pamięć RAM jest pamięcią ulotną (ang. *volatile*), tj. zapisana informacja jest utrzymywana dopóki jest włączone zasilanie.
- ROM - ang. *Read Only Memory* (pamięci tylko do odczytu) pamięć, w których raz zapisana informacja może być w wielokrotnie odczytywana. Zapisana w nich informacja jest utrzymywana niezależnie od tego czy włączone jest napięcie zasilające - pamięć *nieulotna* (ang. *Non-volatile*).

Pamięci ROM jak i pamięci RAM są pamięciami z dostępem swobodnym.

Parametry pamięci półprzewodnikowych

- *Pojemność pamięci*
  - *wielkość komórki pamięci (słowa)* (np. 8-bitowa, 1 bajt),
  - *liczba pamiętanych słów*,

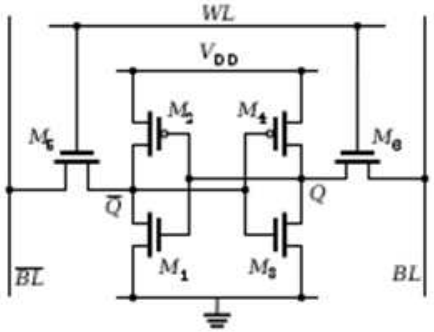
  Pojemność pamięci określa się podając liczbę pamiętanych bitów (np. $16Mb$) albo liczbę pamiętanych słów o określonej długości (na przykład $2M \times 8$, $1M \times 16$).
- *Szybkość pracy pamięci*
  - *Czas dostępu do pamięci* - czas, po do uzyskania zawartości pamięci od momentu jej zażądania,
  - *Czas cyklu maszynowego* - minimalny czas, po którym możemy ponownie zwrócić się do pamięci w celu odczytania kolejnej informacji,

  Czas cyklu maszynowego zazwyczaj jest dłuższy od *czasu dostępu*.

Budowa pamięci RAM

Do budowy pamięci RAM wykorzystuje się różne elementy pamiętające.

- pamięci statyczne (S-RAM) - w których elementem pamiętającym jest przerzutnik (np., teoretycznie, zatrzask),

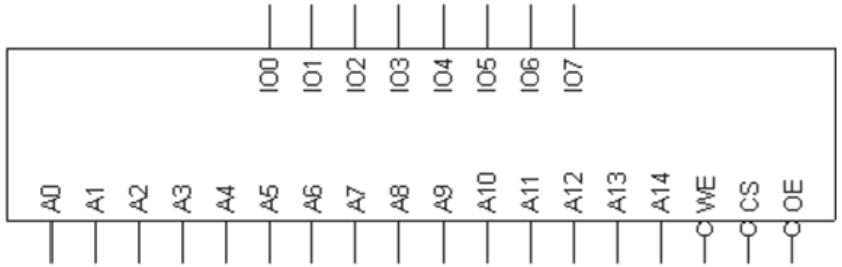


  Dostęp do komórki jest możliwy poprzez linie WL, która sterując tranzystorami M_5 i M_6 umożliwia odczyt stanu przerzutnika (tranzystory M_1, M_2, M_3 i M_4) za pomocą linii $\overline{BL}$ i BL.

- pamięci dynamiczne (DRAM), w których informacja jest pamiętana w postaci ładunku zgromadzonego w kondensatorze.

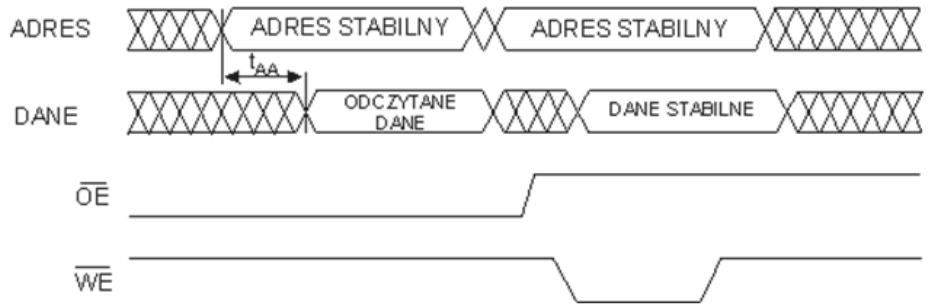
Pamięć RAM - układ scalony

Przykład pamięci S-RAM ($32K \times 8$):



- *Wejścia adresowe* $A_0 \div A_{14}$ - 15 linii adresowych umożliwia zaadresowanie 2^{15} komórek pamięci,
- *Końcówki IO* (wejściowo-wyjściowe) $IO_0 \div IO_7$ - 8 linii umożliwia odczyt albo zapis do wskazanej komórki pamięci (1 bajtu),
- *Sygnały sterujące*
  - $\overline{CS}$ - *sygnał wyboru kości* - przy niskim poziomie sygnału pamięć jest aktywna,
  - $\overline{WE}$ - *sygnał zapisu/odczytu* - przy niskim poziomie następuje zapis, przy wysokim odczyt,
  - $\overline{OE}$ - *sygnał otwierający wyjścia trójstanowe* pamięci w fazie odczytu.

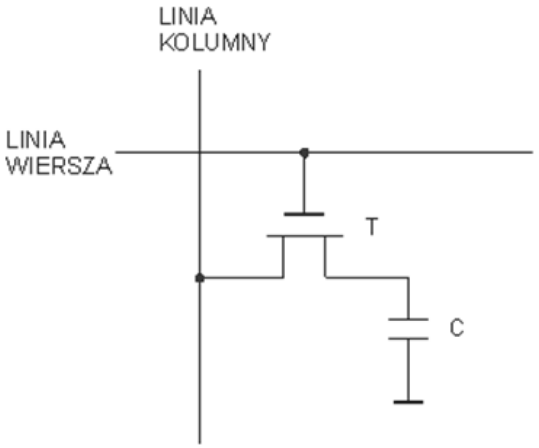
Przebiegi czasowe odczytu, zapisu pamięci S-RAM

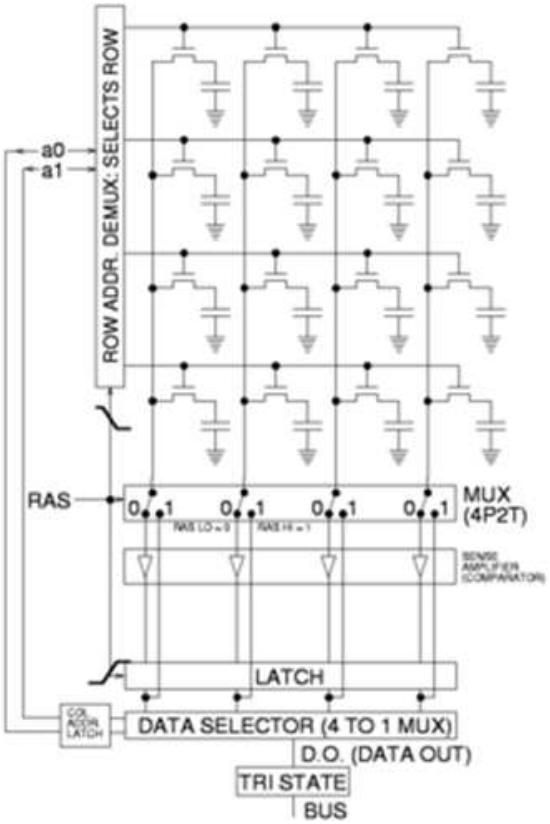


- *Cyklu odczytu* - po czasie dostępu t_{AA}, liczonym od chwili ustalenia się adresu na wejściach pamięci, na wyjściach pojawia się odczytana informacja. Sygnały zapisu i otwierania wyjść są równe $\overline{WE} = 1$ i $\overline{OE} = 0$,
- *Cyklu zapisu* - po ustaleniu się adresu na wejściach pamięci, podawany jest sygnał zapisu $\overline{WE} = 0$ i informacja wejściowa jest zapisywana do pamięci. Sygnał $\overline{OE} = 1$ blokuje układy wyjściowe (końcówki IO pełnią rolę wejść).

Produkowane są również pamięci synchroniczne SSRAM (wykorzystujące sygnał zegara), w których wykorzystywane jest tylko jedno zbocze sygnału zegarowego (SDR SRAM) albo oba zbocza sygnału zegarowego (DDR SRAM).

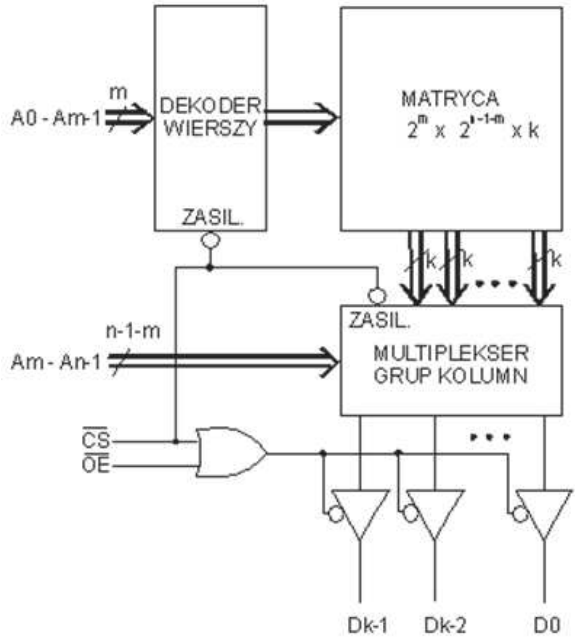
Pamięci D-RAM





- Elementem pamiętającym w tym układzie jest kondensator C o bardzo małej pojemności. Jeżeli kondensator nie jest naładowany, to pamiętane jest zero logiczne.
- *linia wiersza* umożliwia przepływ ładunku elektrycznego do kondensatora,
- *linia kolumny* ładuje kondensator lub odczytuje jego stan.

Pamięci ROM



- Część bitów słowa adresowego (m bitów) jest wykorzystywana do adresowania wierszy matrycy a pozostałe bity słowa adresowego wybierają grupy k kolumn,
- Słowo wyjściowe ma k bitów.

Klasyfikacja pamięci ROM

- ROM - *Read Only Memory*
- PROM - *Programmable* ROM
- EPROM - *Erasable* ROM
- EEPROM - *Electrically Erasable* PROM (w tym również pamięć Flash)

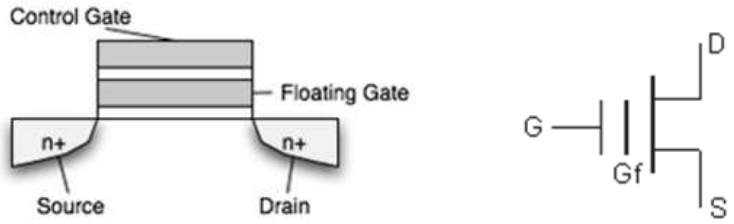
ROM i PROM

- ROM - informacja jest wprowadzana w czasie procesu produkcyjnego. Na podstawie znajomości zawartości pamięci producent projektuje maskę połączeń w matrycy. Informacja jest wprowadzana do pamięci tylko raz i nie można jej zmienić.
- PROM - *Programmable* ROM - Pamięci stałe typu PROM umożliwiają jednokrotne zapisywanie informacji przez użytkownika. W czasie programowania ma miejsce odłączanie odpowiednich tranzystorów od linii matrycy. Odbywa się to na zasadzie przepalenia odpowiednich bezpieczników.

Pamięci EPROM i EEPROM

- EPROM - W pamięciach typu EPROM wykorzystuje się tranzystory z podwójną bramką. Przy braku ładunku w pływającej bramce tranzystor zachowuje się jak zwykły tranzystor, natomiast po wprowadzeniu ładunku do pływającej bramki tranzystor jest na stałe odcięty, co odpowiada sytuacji braku tranzystora.
- EEPROM - W pamięciach typu EEPROM zarówno zapis jak i odczyt pamięci realizowane są w docelowym układzie na zasadzie czysto elektrycznej. Kasowaniu podlegają pojedyncze bity.

Tranzystory MOSFET z podwójną (pływającą) bramką



- Dopóki dodatkowa bramkę Gf, która jest odizolowana od otoczenia, nie zostanie wprowadzony ładunek, to tranzystor zachowuje się jak zwykły tranzystor typu MOS,
- Jeżeli natomiast na dodatkową bramkę zostanie wprowadzony ładunek, to tranzystor jest odcięty i nie przewodzi prądu między drenem D a źródłem S.
- Ładunek wprowadzony na tę bramkę może utrzymywać się przez długi czas (minimum 10 lat).

Pamięci FLESH

- W dużych pamięciach EEPROM możliwe jest równoczesne kasowanie zawartości całych dużych bloków lub nawet całej zawartości pamięci. Stąd pamięci te są określane jako pamięci FLASH (błyskawiczne).
- Pamięci EEPROM mogą być przeprogramowywane jedynie określoną liczbę razy (kilkadziesiąt tysięcy razy).

Pamięci ROM jak i pamięci RAM są pamięciami z dostępem swobodnym.

Funkcje pamięci ROM i pamięci RAM

Zarówno pamięci ROM jak i pamięci RAM są pamięciami z dostępem swobodnym.

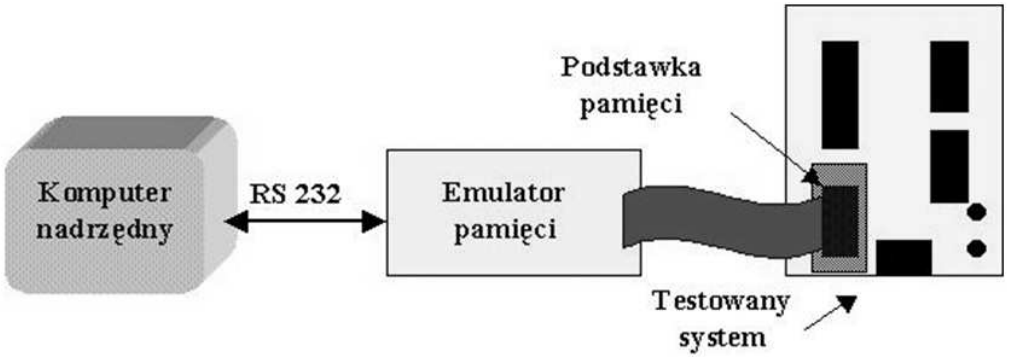
- *Pamięć ROM* - Informacja przechowywana w pamięci ROM nie może być łatwo modyfikowana i pozostaje w pamięci po wyłączeniu zasilania. Pamięć ROM ma zastosowanie jako:
  - *Pamięć programu* - przechowywany jest w niej program,
  - *Nieulotna pamięć danych EEPROM data* albo pamięć *Flesh* - służy do przechowywania danych nawet po wyłączeniu zasilania.
- *Pamięć RAM* - w pamięci RAM przechowuje się dane potrzebne do bieżącego wykonywania programu, *stos* oraz inne bloki.

Sposoby programowania pamięci programu

Pamięci programu ROM można programować na trzy sposoby:

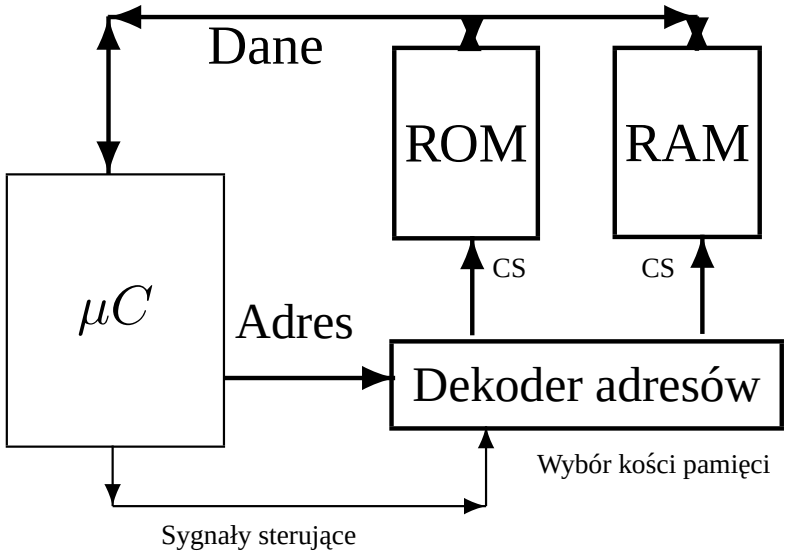
1. *High voltage Programming* czyli sposób programowania wprowadzony ponad 15lat temu do programowania pamięci EPROM za pomocą sygnałów 12V - wymaga programatora.
2. *ISP (In-System Programmable)* które nie wymaga wyjmowania pamięci z systemu w którym pracuje.
3. *Bootloader* - po resecie μC uruchamiany jest program znajdujący się w sekcji Bootloadera, który poprzez łącze (np. port szeregowy) łączy się z komputerem nadrzędnym, pobiera kod programu i umieszcza go w przeznaczonej do tego obszarze pamięci ROM.

Emulator sprzętowy pamięci programu



1. W większości przypadków *emulator programu* jest pamięcią typu RAM, do której wpisuje się program z komputera nadrzędnego np. za pomocą łącza szeregowego lub równoległego,
2. Emulator pamięci jest wyposażony w sondę zakończoną adapterem dopasowanym do podstawki pamięci programu w systemie rzeczywistym,
3. Emulator nadaję się tylko do systemów gdzie możliwa jest praca mikrokontrolera z zewnętrzną pamięcią programu.

Współpraca pamięci ROM i RAM z μC

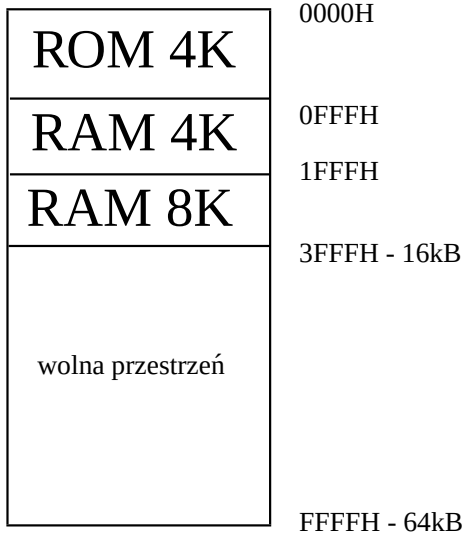


- Pamięć ROM i RAM są umieszczone we wspólnej przestrzeni adresowej jak na rysunku (arch. von Neumana) albo w osobnych przestrzeniach adresowych (arch. Harwardzka),
- Zarówno na pamięć RAM jak i ROM może składać się wiele kości pamięci,
- Zadaniem *dekodera adresu* jest wybór konkretnej kości pamięci, w oparciu o adres i dopowiednie sygnały sterujące procesora (np. żądania dostępu do pamięci $\overline{MREQ}$ albo urządzeń wej/wyj $\overline{IORQ}$)

Wspólna przestrzeń adresowa

- Zadaniem projektanta μC jest zagospodarowanie przestrzeni adresowej,
- Po resecie zawartość *Rejestru Programu* PC jest zerowa - adres pierwszej instrukcji,
- Na początku przestrzeni adresowej μC jest pamięć pamięć ROM,
- Pamięć RAM jest umieszczana za pamięcią ROM (bezpośrednio lub z odstępem).
- W wielu przypadkach szyna adresowa μC 8 bitowego ma 16 linii - *przestrzeń adresowa pamięci* wynosi $2^{16} = 64kB$

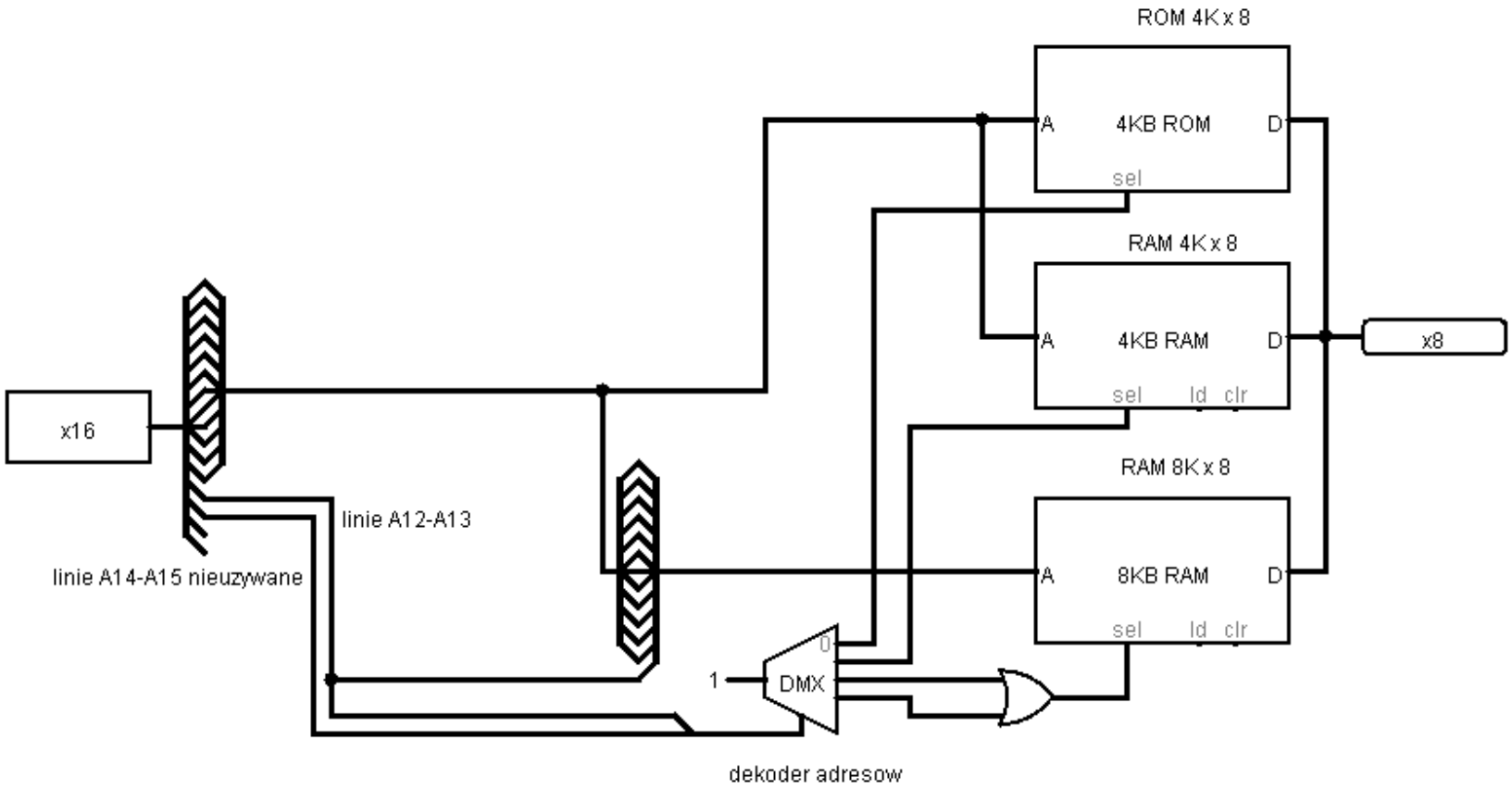
Przykład wspólnej przestrzeni adresowej



Linie A_{13} A_{12}	kość
00	ROM 4K
01	RAM 4K
10	RAM 8K
11	RAM 8K

- Z $64kB$ przestrzeni adresowej wykorzystywanych jest $16kB$,
- Do zaadresowania $16kB = 2^{14}$ potrzeba 14 linii adresowych $A_0 \div A_{13}$. Ostanie dwie linie adresowe A_{15} i A_{14} są nieużywane,
- Pamięci RAM i ROM $4kB = 2^{12}$ posiadają 12 linii adresowych $A_0 \div A_{11}$, pamięć RAM $8kB = 2^{13}$ 13 linii $A_0 \div A_{12}$,
- Linie A_{12} i A_{13} są potrzebne dla *dekodera adresu* do ustalenia wyboru kości.

Realizacja wspólnej przestrzeni adresowej - cd.



Dekodowanie adresów

- *pełne* - w adresowaniu biorą udział wszystkie linie magistrali adresowej, wówczas każda komórka ma jednoznacznie określony adres,
- *niepełne* - w przeciwnym przypadku. Poprzedni przykład ilustruje adresowanie niepełne. Z racji, że $\frac{3}{4}$ przestrzeni adresowej jest niewykorzystana, linie A_{15} i A_{14} nie są potrzebne do adresowania.

Zadania na ćwiczenia

1. Zrealizuj układ, który automatycznie wpisze do wszystkich komórek pamięci RAM 256×8 ich adres (np. do komórki o adresie 12 wartość 12).
2. Zrealizuj układ, który automatycznie przepisze zawartość pamięci ROM do pamięci RAM. Rozmiar obu pamięci wynosi 256×8. W każdej komórce pamięci ROM powinien być umieszczony jej adres (tak jak w poprzednim zadaniu).
3. Trzy kości pamięci:
   - 1 kość pamięci ROM: $16K \times 8$,
   - 1 kość pamięci RAM: $32K \times 8$,
   - 1 kość pamięci RAM: $16K \times 8$.

   umieszczone kolejno po sobie w 16-bitowej przestrzeni adresowej. Zrealizuj dekoder adresów i sprawdź jego działanie. Czy to jest dekodowanie pełne czy niepełne?