

Adresový dekodér

Ve všech příkladech předpokládejte 16-ti bitovou adresovou sběrnici, kde hexadecimální vyjádření adresy představuje stav adresových vodičů na sběrnici A15,A14,A13,A12 A11,A10,A9,A8 A7,A6,A5,A4 A3,A2,A1,A0

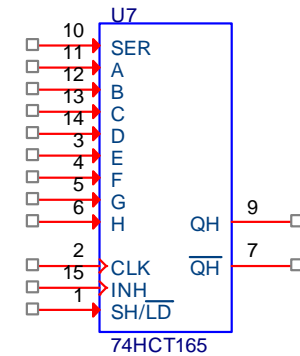
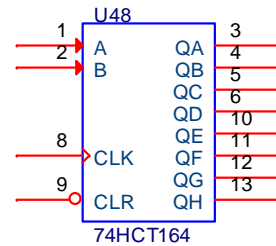
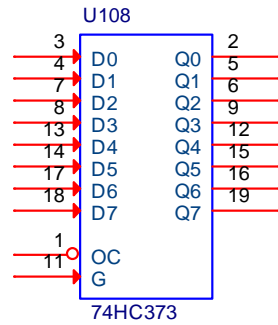
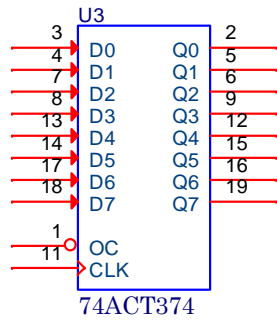
1. Napište logickou funkci indikující log.0 adresový rozsah 0000÷3FFFh
2. Napište logickou funkci indikující log.0 adresový rozsah 8000÷9FFFh
3. Napište logickou funkci indikující log.0 adresový rozsah 7000÷8FFFh
4. Napište logickou funkci indikující log.0 adresový rozsah 0000÷4FFFh
5. Napište logickou funkci indikující log.0 adresový rozsah 0000÷EFFFh

6. Logickou funkci z příkladu 2 realizujte obvodem 74138
7. Logickou funkci z příkladu 4 realizujte obvody 74138
8. Logickou funkci z příkladu 3 realizujte obvodem 74138

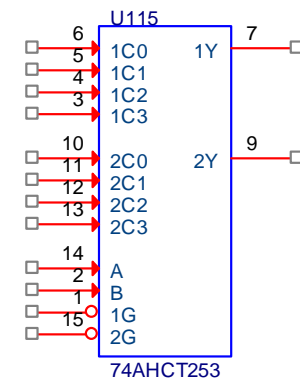
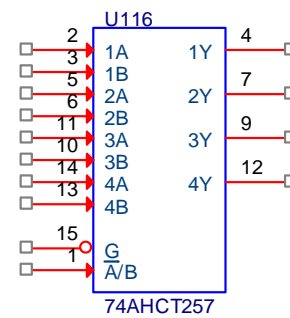
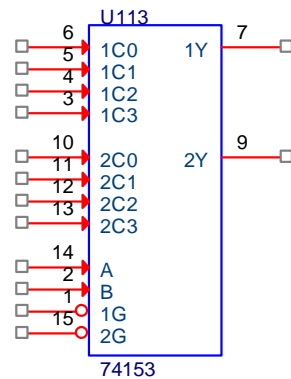
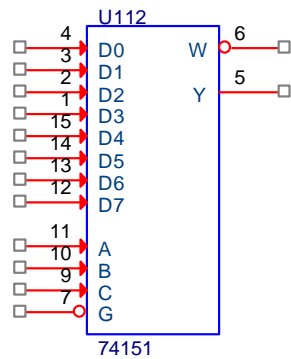
9. Kolik obvodů 74138 potřebují k aktivaci paměti RAM s kapacitou 2kB

*Registry, latch,
posuvné registry*

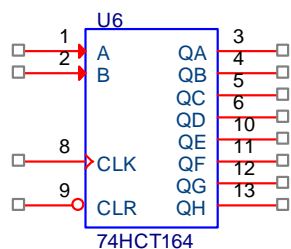
Nakreslete časový průběh signálů na vstupech obvodů potřebný k zápisu hodnoty 56h do následujících obvodů a stanovte časové odstupy pro zajištění správné činnosti.



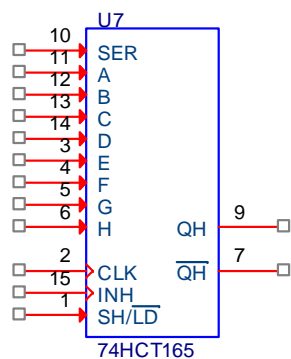
Popište, jak byste přenesli hodnotu ze vstupů multiplexerů s dvoustavovým a třístavovým výstupem do procesoru.



1. Napište v jazyce C program, který zapíše hodnotu 56h do následujícího obvodu

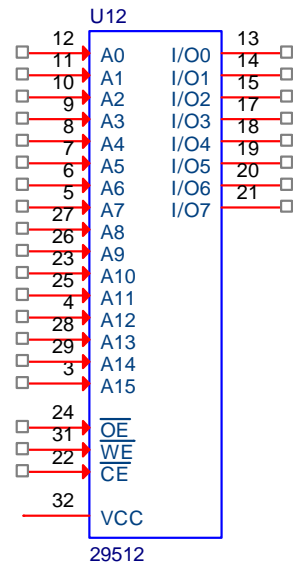


2. Napište v jazyce C program, který přečte hodnotu ze vstupů SER, A,B, až H a uloží ji do proměnné vstup.

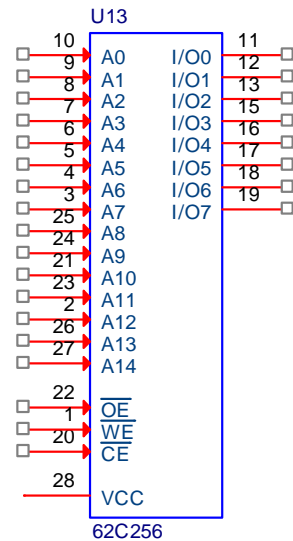


Paměti

1. Nastavte vývody paměti flash tak, aby jste přečetli hodnotu z adresy 5A7h.



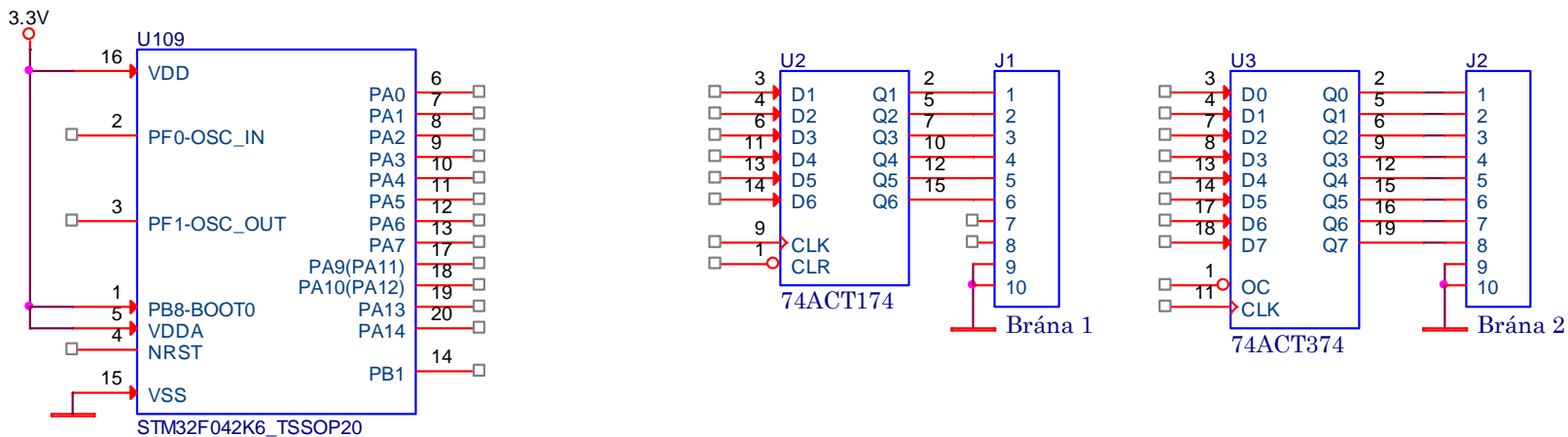
2. Nastavte vývody paměti RAM tak, aby jste uložili hodnotu BEh na adresu 13DCh. Jaký průběh bude muset mít signál WE (WR). Nakreslete časovou posloupnost signálů (adresy, OE, WE a CE) pro zápis hodnoty.



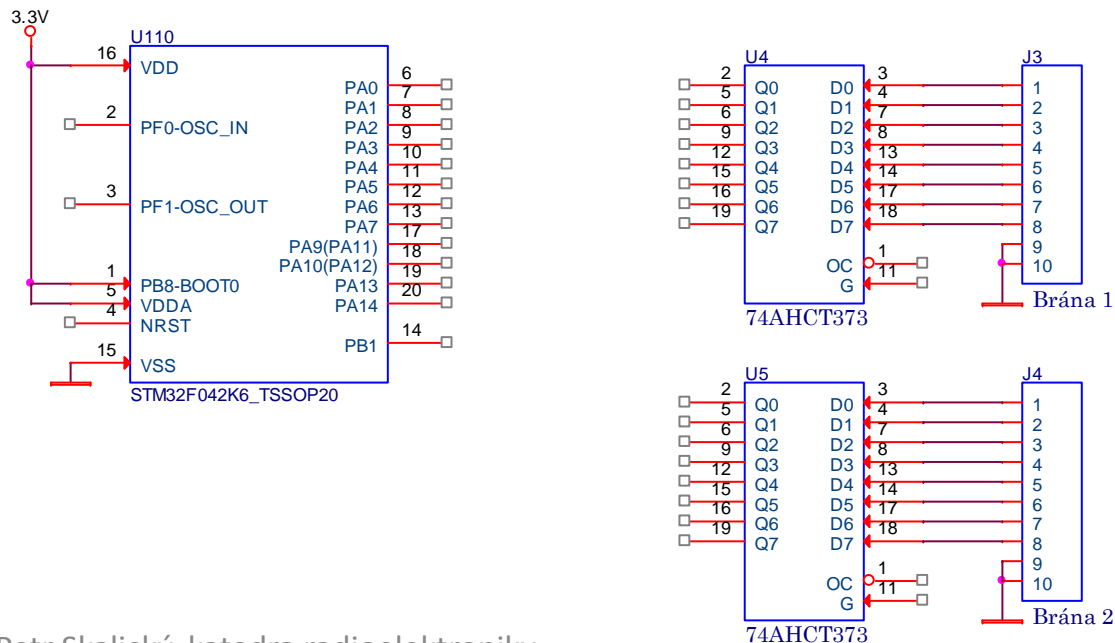
3. Jaké paměti byste použili při realizaci přístupového systému pro 40000 uživatelů s identifikací 20 byty. Systém by měl umožňovat vkládání a vyřazení uživatelů. Identifikace uživatele nesmí trvat déle než 0,3s.

Připojení vstupů a výstupů k I/O vývodům procesoru

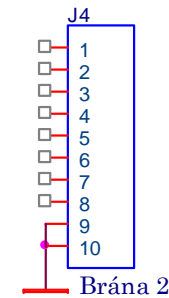
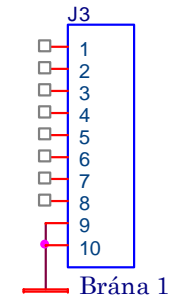
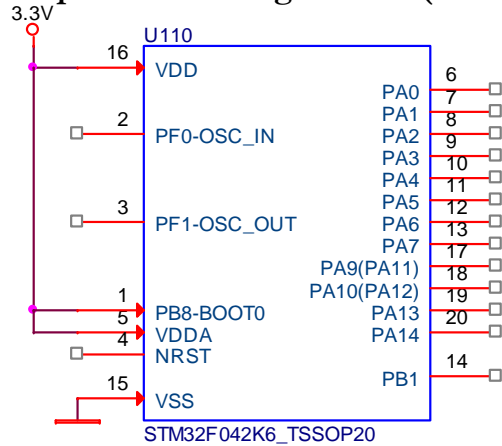
1. Připojte dvě výstupní brány k procesoru a napište obslužný podprogram pro zápis hodnoty z registru R7 (assembler) a hodnoty char (jazyk C) do brány 1.



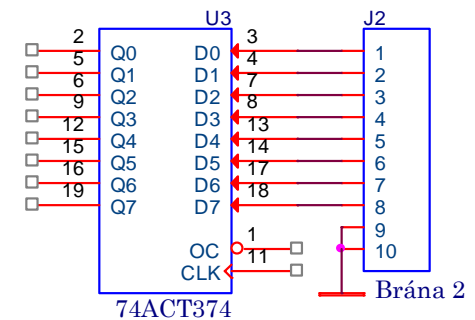
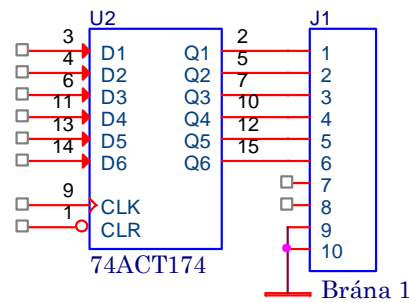
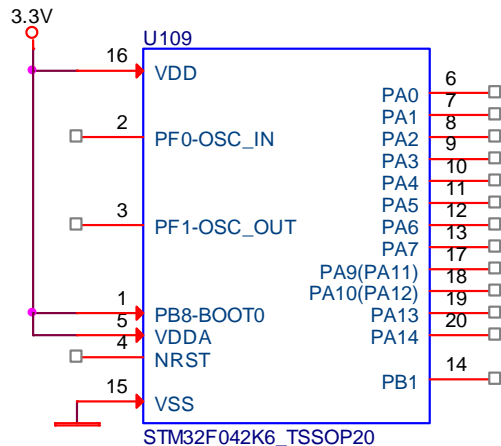
2. Připojte dvě vstupní brány k procesoru a napište obslužný podprogram pro čtení hodnoty ze vstupní brány 2 do registru R7 (assembler) a hodnoty char (jazyk C).



Připojte dvě vstupní brány k bráně PA procesoru STM32F042K6 bez použití obvodů s třístavovým výstupem a napište obslužný program pro čtení do registru R7 (assembler) a hodnoty char (jazyk C) z brány 1.



Připojte jednu vstupní a jednu výstupní bránu k procesoru a napište obslužný podprogram pro zápis hodnoty z registru R7 (assembler) a hodnoty char (jazyk C) do výstupní brány a obslužný podprogram pro čtení hodnoty ze vstupní brány 2 do registru R7 (assembler) a hodnoty char (jazyk C).



Práce s čísly

Napište program v assembleru a jazyce C, který

1. Sečte dvě 16-ti bitová čísla
2. Sečte dvě 16-ti bitová čísla v BCD kódu
3. Vynásobí dvě 16-ti bitová čísla bez znaménka
4. Vynásobte 16-ti bitové číslo hodnotou 2,125 s využitím násobení
5. Vynásobte 16-ti bitové číslo hodnotou 2,125 s využitím posunů
6. Kolik bitů bude potřeba rezervovat na číslo v pohyblivé čárce mantisu a exponent s přesností 4 desetinných míst a rozsahem $10^{\pm 3}$
7. Máme čítač, jehož načítaná hodnota bude násobena 6 místnou konstantou s libovolně umístěnou desetinnou tečkou tj. od 0,00001 ÷ 999999. Součin bude zobrazován na 6 místném displeji ze sedmisegmentových zobrazovačů a uživatel si zadá, kde bude na displeji umístěna desetinná tečka.

Úkol:

1. Určete, kolika bitový musí být čítač ?
2. Určete, kolika bitová musí být aritmetika s pevnou nebo pohyblivou čárkou (je tím míněno Vámi vytvořená nikoliv nějaká standardizovaná) ?
3. Jak získáte cifry, které budou zobrazovány na displeji ?

Výsledky

Ve všech příkladech předpokládejte 16-ti bitovou adresovou sběrnici, kde hexadecimální vyjádření adresy představuje stav adresových vodičů na sběrnici A15,A14,A13,A12 A11,A10,A9,A8 A7,A6,A5,A4 A3,A2,A1,A0

1. Napište logickou funkci indikující log.0 adresový rozsah 0000÷3FFFh

$$CS=A15+A14$$

2. Napište logickou funkci indikující log.0 adresový rozsah 8000÷9FFFh

$$CS=/A15+A14+A13$$

3. Napište logickou funkci indikující log.0 adresový rozsah 7000÷8FFFh

$$CS=(A15+/A14+/A13+/A12).(/A15+A14+A13+A12)$$

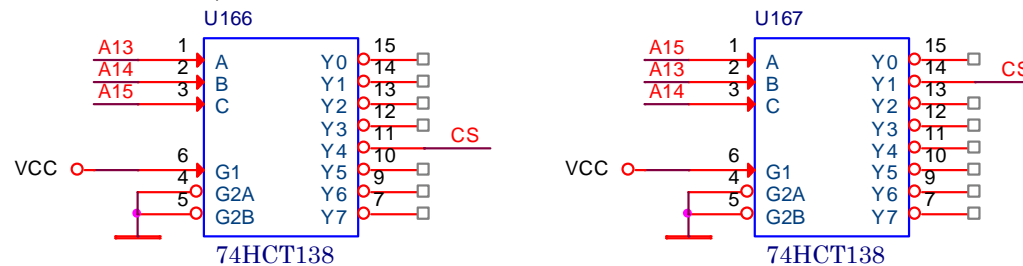
4. Napište logickou funkci indikující log.0 adresový rozsah 0000÷4FFFh

$$CS=(A15+A14).(A15+/A14+A13+A12)$$

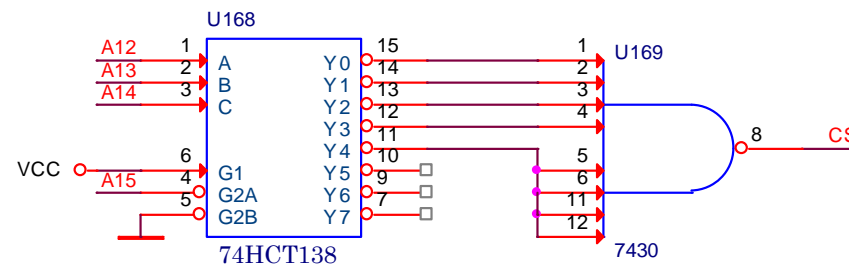
5. Napište logickou funkci indikující log.0 adresový rozsah 0000÷EFFFh

$$CS=A15.A14.A13.A12$$

6. Logickou funkci z příkladu 2 realizujte obvodem 74138



7. Logickou funkci z příkladu 4 realizujte obvodem 74138

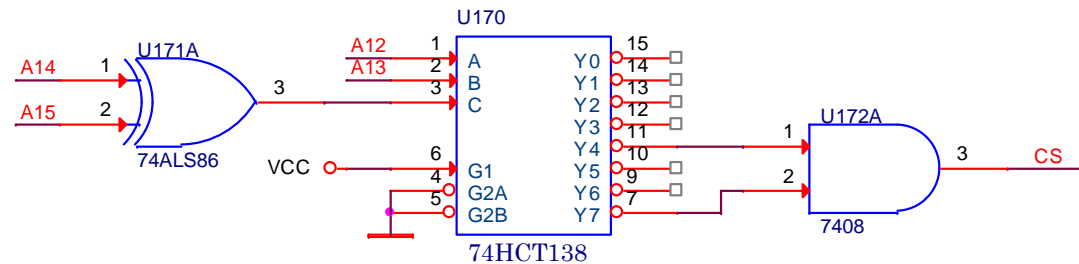


Ve všech příkladech předpokládejte 16-ti bitovou adresovou sběrnici, kde hexadecimální vyjádření adresy představuje stav adresových vodičů na sběrnici A15,A14,A13,A12 A11,A10,A9,A8 A7,A6,A5,A4 A3,A2,A1,A0

3. Napište logickou funkci indikující log.0 adresový rozsah 7000÷8FFFh

$$CS = (A15 + /A14 + /A13 + /A12) \cdot (/A15 + A14 + A13 + A12)$$

8. Logickou funkci z příkladu 3 realizujte obvodem 74138



9. Kolik obvodů 74138 potřebuji k aktivaci paměti RAM s kapacitou 2kB

Dvou kB bloků se do prostoru 62kB vejde 32 ⇒ dekódování 5 adresových vodičů

Pro 31 prostorů od 0000÷07FF až po F700÷F7FF bude jeden adresový vodič nulový ⇒ stačí 1x74138

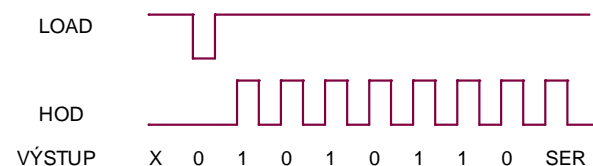
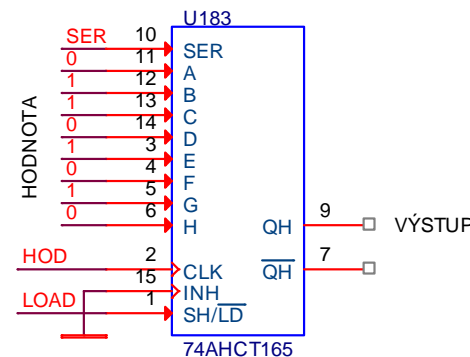
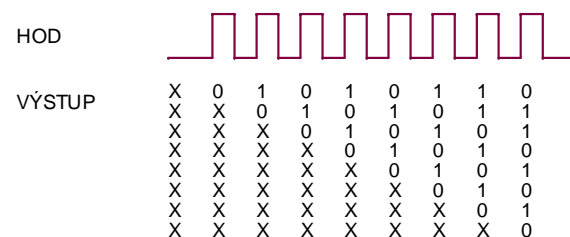
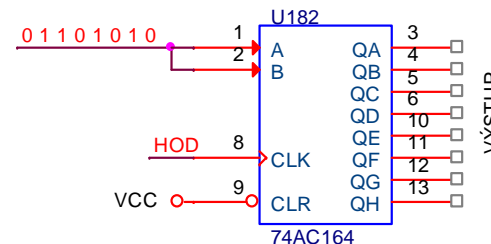
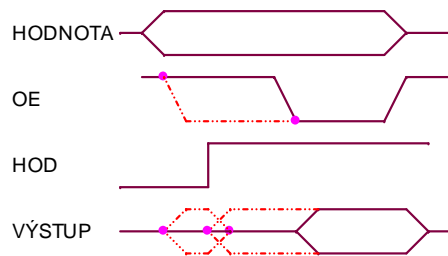
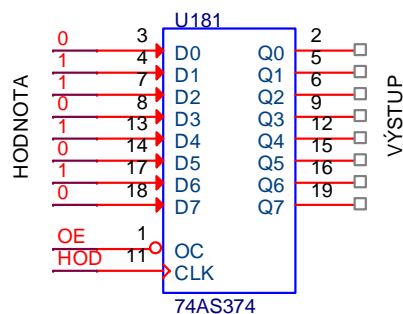
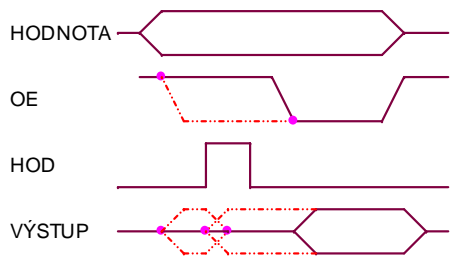
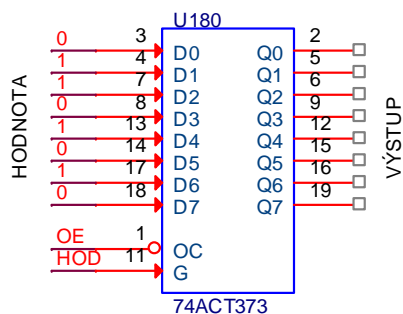
Pro prostor F800÷FFFF jsou všechny dekódované adresové vodiče rovny log.1

⇒ 2x 74138

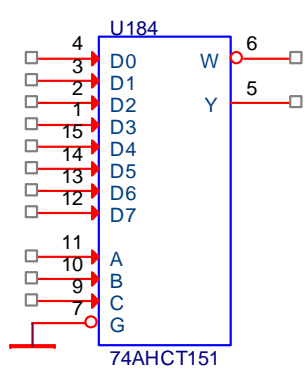
nebo

⇒ 1x74138 plus 1xnegace

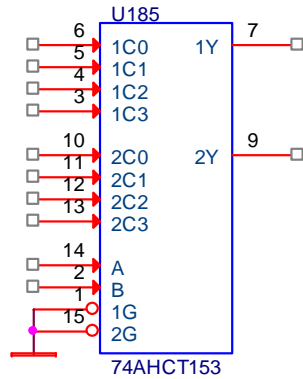
Nakreslete časový průběh signálů na vstupech obvodů potřebný k zápisu hodnoty 56h do následujících obvodů a stanovte časové odstupy pro zajištění správné činnosti.



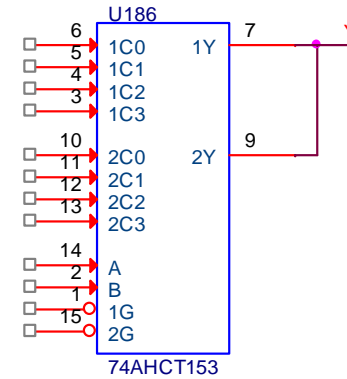
Popište, jak byste přenesli hodnotu ze vstupů multiplexerů s dvoustavovým a třístavovým výstupem do procesoru.



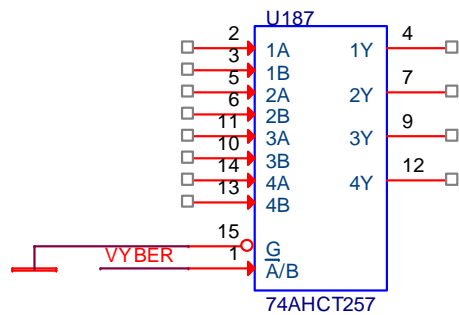
C	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
A	0	1	0	1	0	1	0	1
Y	D0	D1	D2	D3	D4	D5	D6	D7



B	0	0	1	1
A	0	1	0	1
1Y	1C0	1C1	1C2	1C3
2Y	2C0	2C1	2C2	2C3

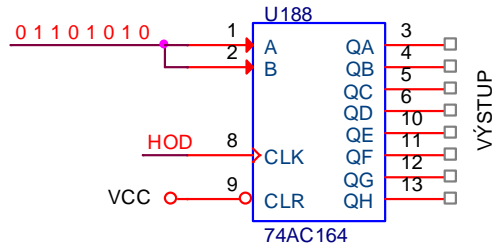


B	0	0	1	1	0	0	0	1	1
A	0	1	0	1	0	0	1	0	1
1G	0	0	0	0	1	1	1	1	1
2G	1	1	1	1	1	0	1	0	0
Y	1C0	1C1	1C2	1C3	Z	2C0	2C1	2C2	2C3



VYBER	0	1
1Y	1A	1B
2Y	2A	2B
3Y	3A	3B
4Y	4A	4B

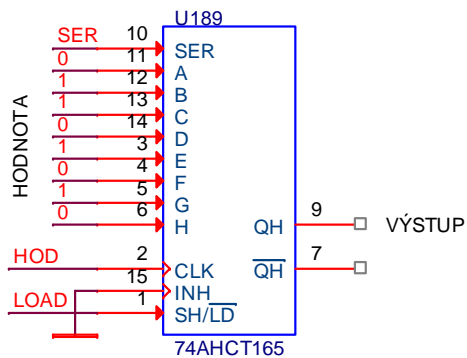
Proč nemohu spojit výstupy u obvodu 74257 ?



1. Napište v jazyce C program, který zapíše hodnotu 56h do následujícího obvodu

```

hodnota=0x56; hod=0;
for (i=0; i<8; i++)
{ if ((hodnota&0x80) !=0) AB=1; else AB=0;
  hod=1; hod=0;
}
    
```

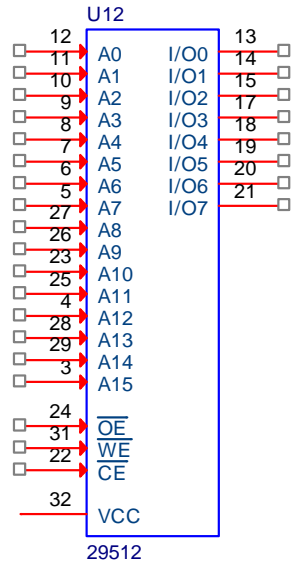


2. Napište v jazyce C program, který přečte hodnotu ze vstupů SER, A,B, až H a uloží ji do proměnné vstup.

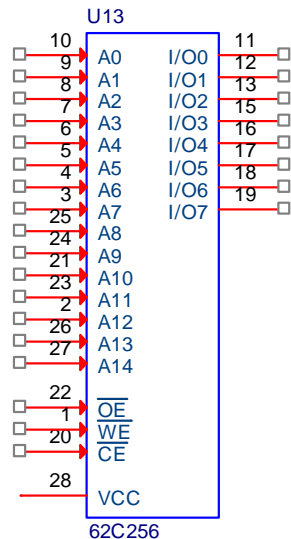
```

hodnota=0; clk=0; load=1;
load=0; load=1; // Načtení hodnoty
for (i=0; i<9; i++)
{ hodnota=hodnota+hodnota;
  if (vystup !=0) hodnota|=1;
  clk=1; clk=0;
}
    
```

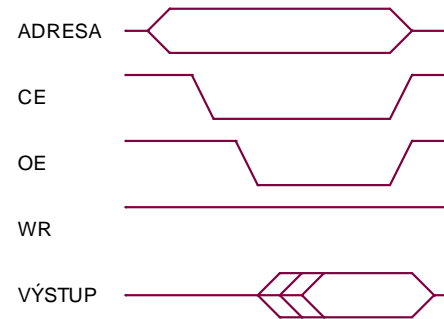
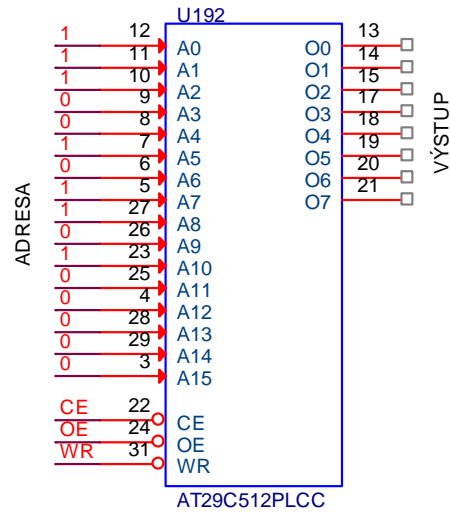
4. Nastavte vývody paměti flash tak, aby jste přečetli hodnotu z adresy 5A7h.



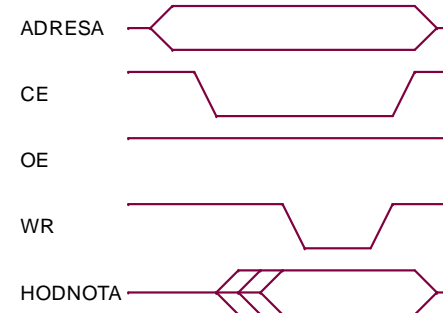
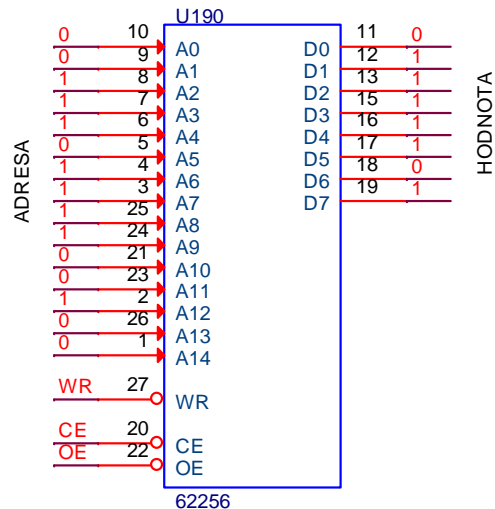
5. Nastavte vývody paměti RAM tak, aby jste uložili hodnotu BEh na adresu 13DCh. Jaký průběh bude muset mít signál WE (WR). Nakreslete časovou posloupnost signálů (adresy, OE, WE a CE) pro zápis hodnoty.



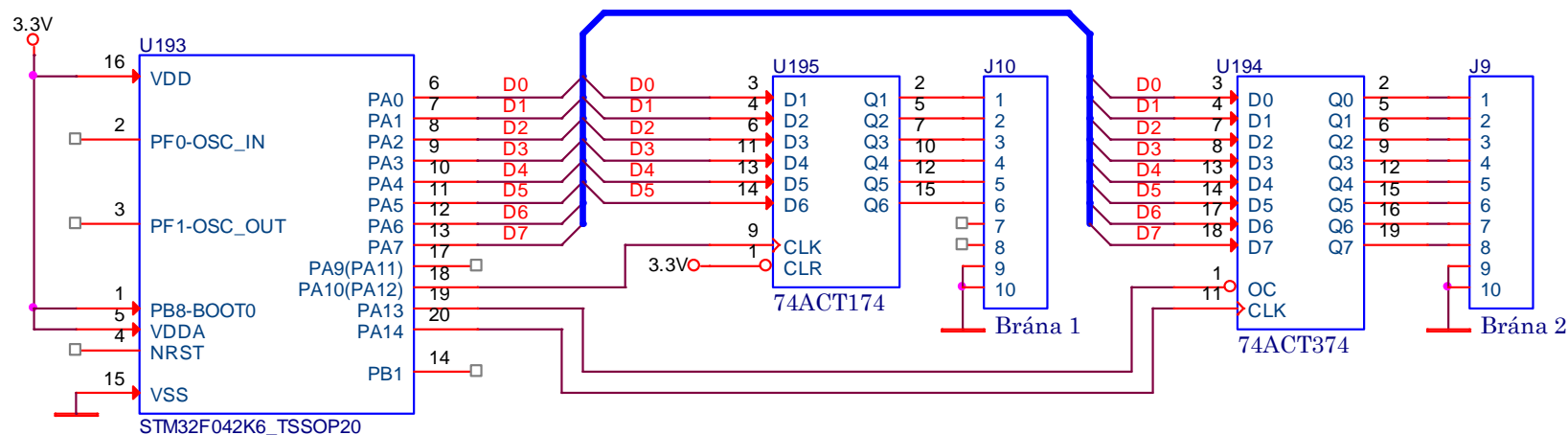
- Nastavte vývody paměti flash tak, aby jste přečetli hodnotu z adresy 5A7h.



- Nastavte vývody paměti RAM tak, aby jste uložili hodnotu BEh na adresu 13DCh. Jaký průběh bude muset mít signál WE (WR). Nakreslete časovou posloupnost signálů (adresy, OE, WE a CE) pro zápis hodnoty.



1. Připojte dvě výstupní brány k procesoru a napište obslužný podprogram pro zápis hodnoty z registru R7 (assembler) a hodnoty char (jazyk C) do brány 1.



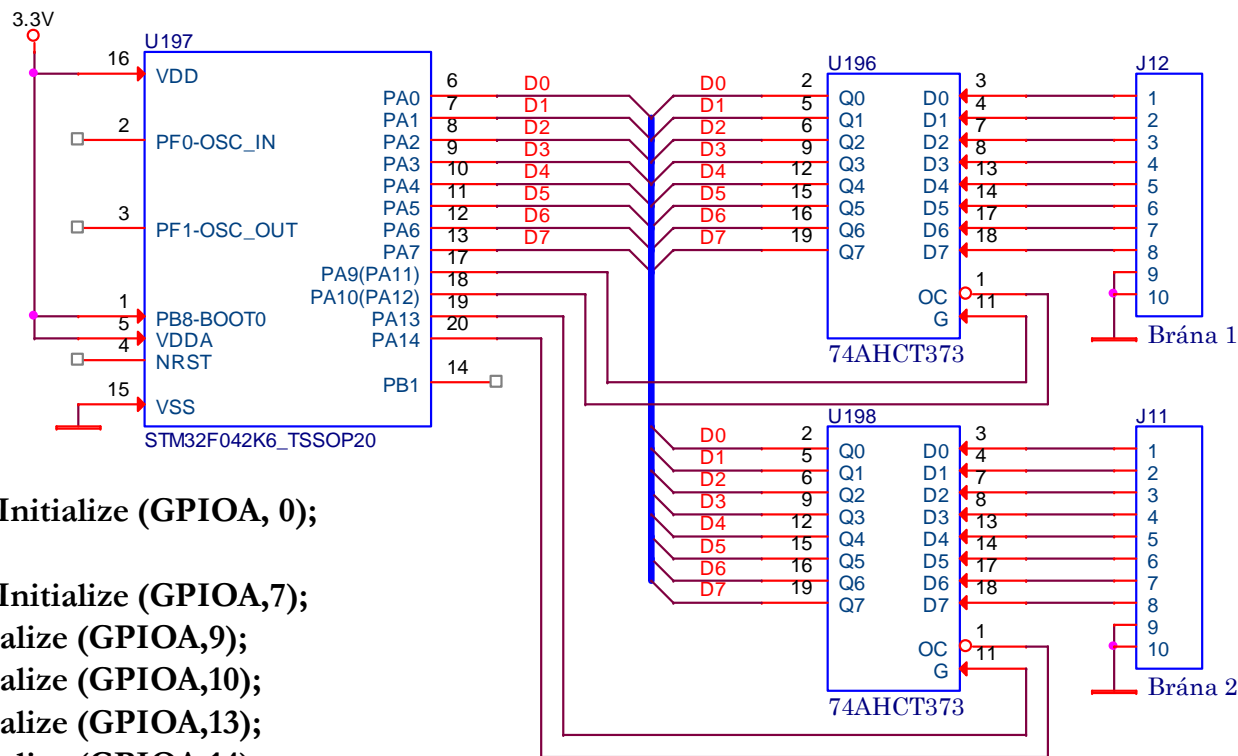
```

PIN_OUTPP_Initialize (GPIOA, 0);
    až
PIN_OUTPP_Initialize (GPIOA,7);
PIN_OUTPP_Initialize (GPIOA,10);
PIN_OUTPP_Initialize (GPIOA,13);
PIN_OUTPP_Initialize (GPIOA,14);
    
```

```

void zapis_do_brany(unsigned char ktera, unsigned char hodnota)
{
    GPIOA->BSRR |= (0xFF << 16); // Nulování předchozího stavu na vývodech PA7÷PA0
    GPIOA->BSRR |= (hodnota&0xFF); // Nastavení zapisované hodnoty na vývody PA7÷PA0
    if (ktera == 1)
    {
        clearbit(GPIOA->ODR, 10); setbit(GPIOA->ODR, 10); } // Zápis do 74174
    if (ktera == 2)
    {
        clearbit(GPIOA->ODR, 14); setbit(GPIOA->ODR, 14); // Zápis do 74374
        clearbit(GPIOA->ODR, 13); } // Aktivace třístavového budiče
}
    
```

1. Připojte dvě vstupní brány k procesoru a napište obslužný podprogram pro čtení hodnoty ze vstupní brány 2 do hodnoty char (jazyk C).



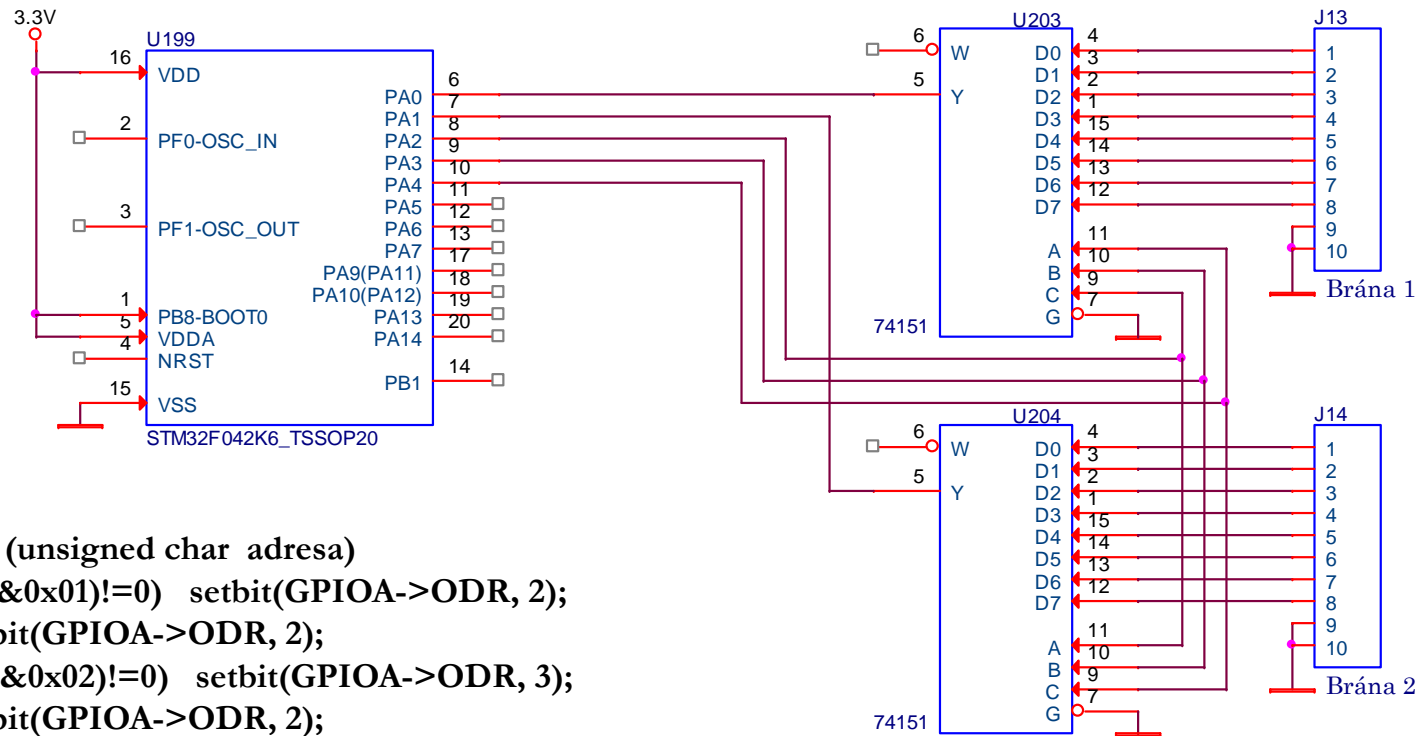
```

PIN_PIN_IN_nn_Initialize (GPIOA, 0);
    až
PIN_PIN_IN_nn_Initialize (GPIOA,7);
PIN_OUTPP_Initialize (GPIOA,9);
PIN_OUTPP_Initialize (GPIOA,10);
PIN_OUTPP_Initialize (GPIOA,13);
PIN_OUTPP_Initialize (GPIOA,14);
    
```

```

unsigned char cti_z_brany(unsigned char ktera)
{
    unsigned char hodnota;
    setbit(GPIOA->ODR, 10); setbit(GPIOA->ODR, 14);
    if (ktera == 1)
    {
        setbit(GPIOA->ODR, 9); clearbit(GPIOA->ODR, 9); clearbit(GPIOA->ODR, 10);
        hodnota = (GPIOA->IDR)&0xFF; setbit(GPIOA->ODR, 10);
    }
    if (ktera == 2)
    {
        setbit(GPIOA->ODR, 13); clearbit(GPIOA->ODR, 13); clearbit(GPIOA->ODR, 14);
        hodnota = (GPIOA->IDR)&0xFF; setbit(GPIOA->ODR, 14);
    }
    return (hodnota);
}
    
```

Připojte dvě vstupní brány k bráně PA procesoru STM32F042K6 bez použití obvodů s třístavovým výstupem a napište obslužný program pro čtení do proměnné char.

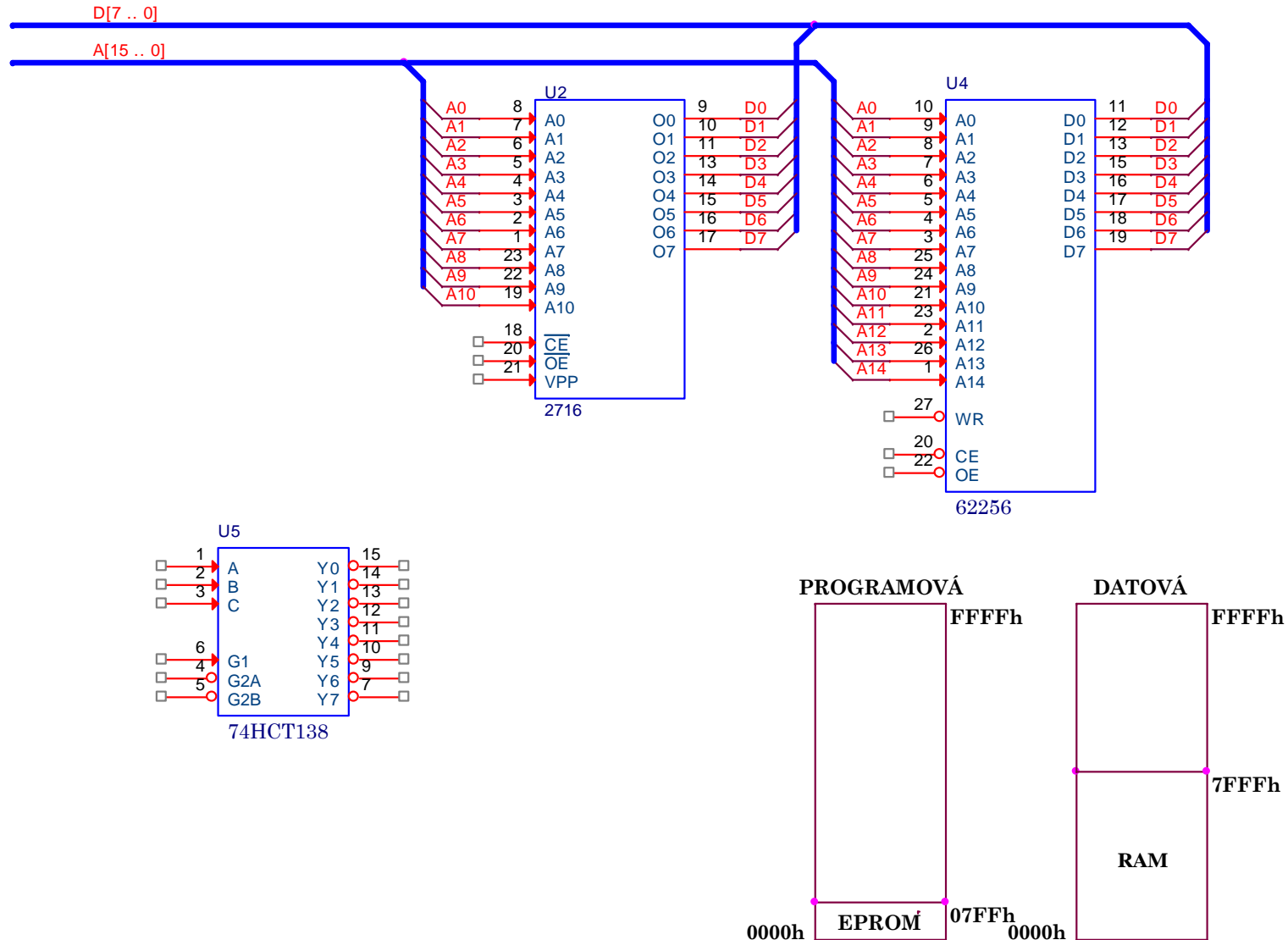


```
void nastav_adr (unsigned char adresa)
{
    if ((adresa&0x01)!=0) setbit(GPIOA->ODR, 2);
    else clearbit(GPIOA->ODR, 2);
    if ((adresa&0x02)!=0) setbit(GPIOA->ODR, 3);
    else clearbit(GPIOA->ODR, 2);
    if ((adresa&0x04)!=0) setbit(GPIOA->ODR, 4);
    else clearbit(GPIOA->ODR, 2);
}
```

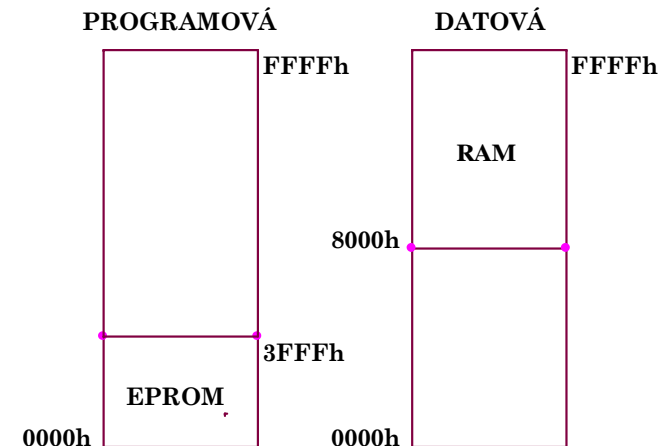
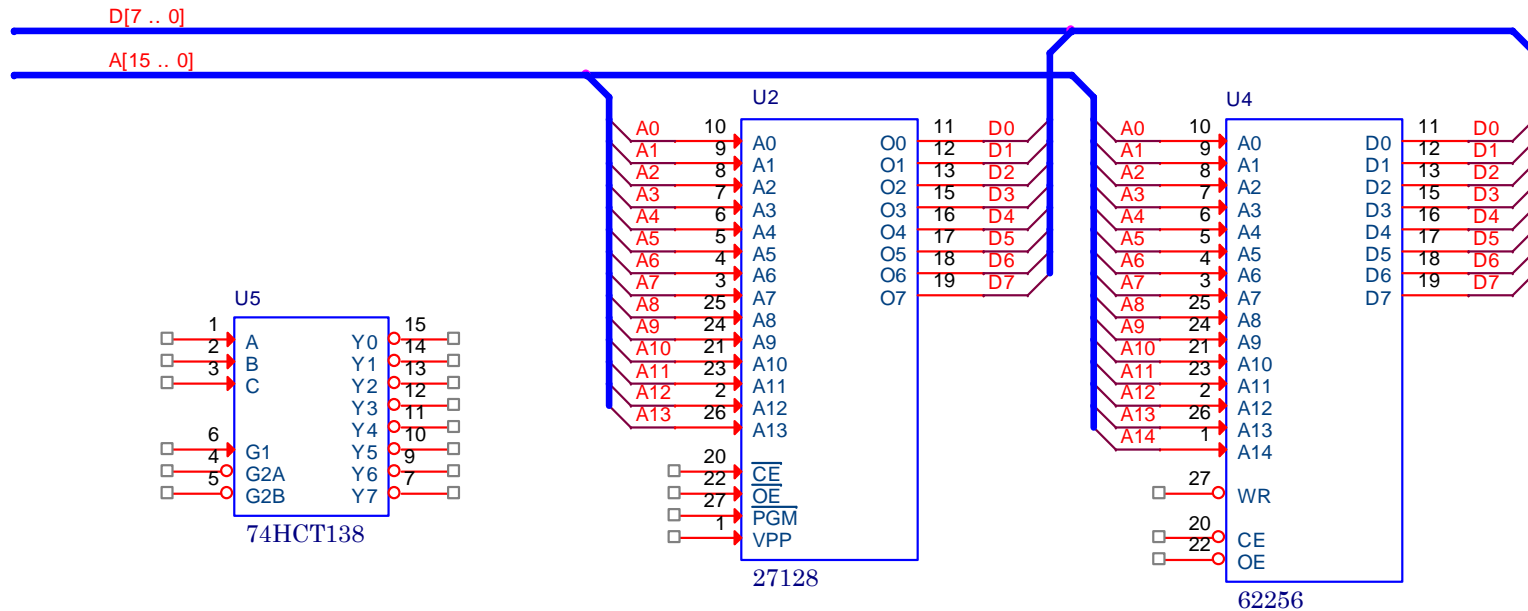
```
unsigned char brána (unsigned char ktera )
{
    unsigned char i, brana1, brana2;
    brana1=brana2=0;
    for (i=0; i<8; i++) { nastav_adr (7-i); brana1+=brana1; brana2+=brana2;
        if (getbit(GPIOA->IDR, 0)!=0) brana1|=1;
        if (getbit(GPIOA->IDR, 1)!=0) brana2|=1;
    }
    if (která==1) return (brana1); if (která==2) return (brana2);
}
```


*Připojení pamětí a
paměťových V/V obvodů
ke společné sběrnici
(standardní)*

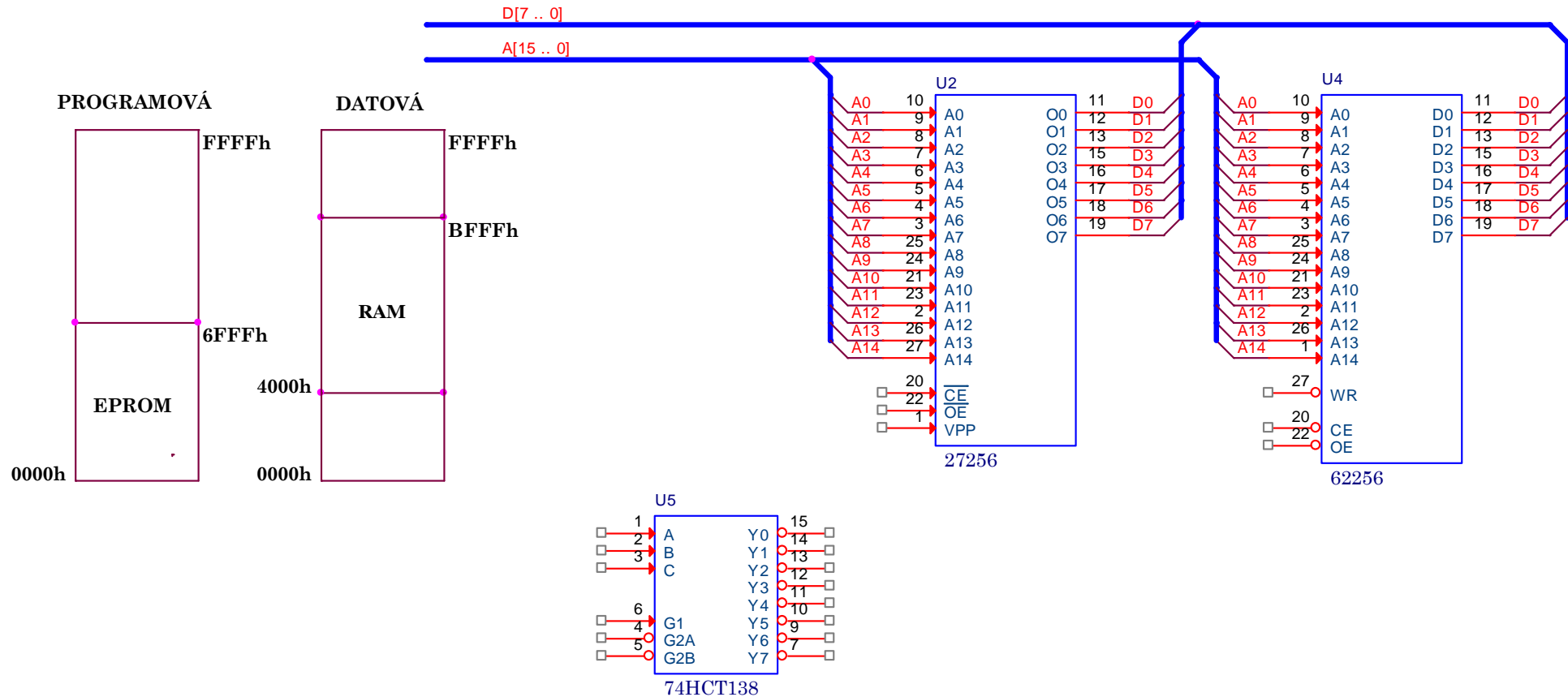
1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 2kB paměť EPROM 2716 a 32kB paměť RAM 62256 do paměťových prostorů uvedených v levé části obrázku. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).
2. Zrealizujte zapojení bez obvodu 138.



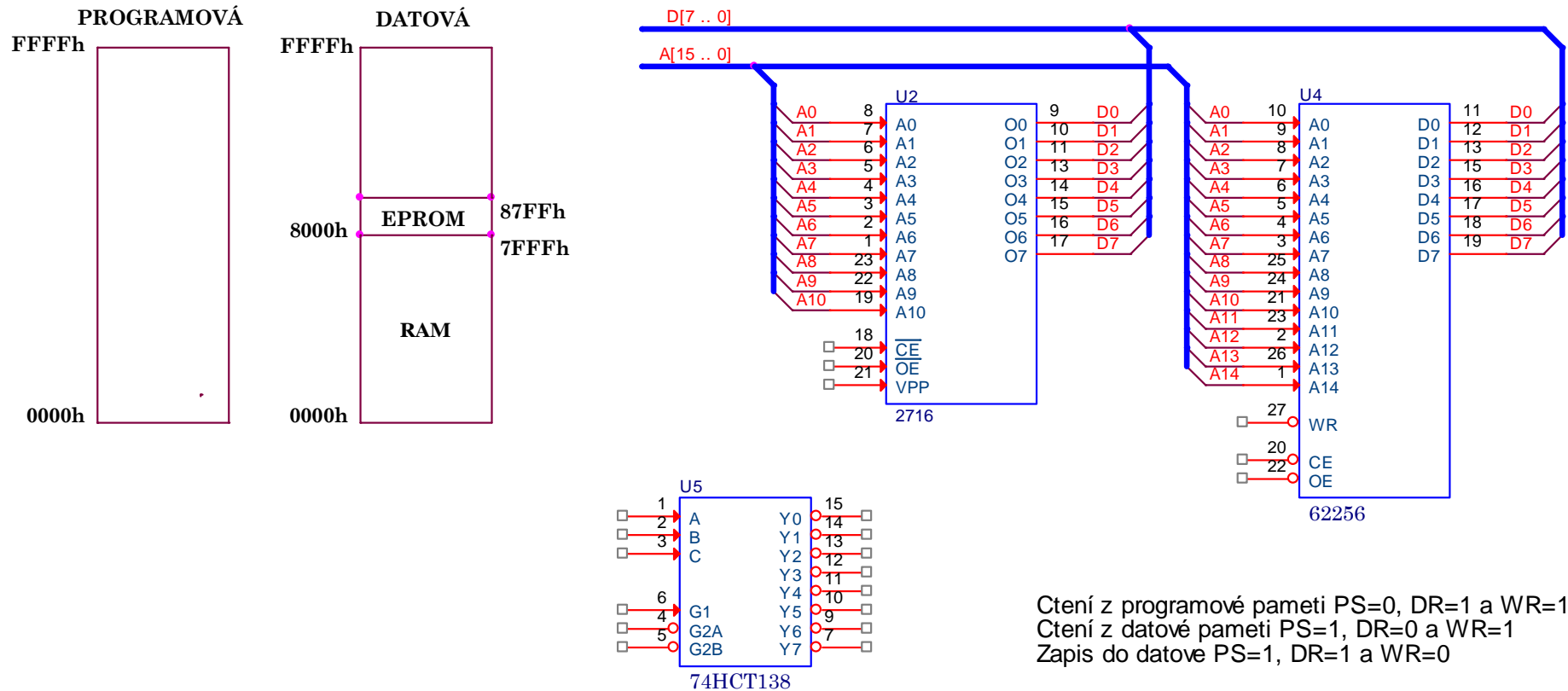
1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 16kB paměti EPROM 27128 a 32kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).
2. Zrealizujte zapojení bez obvodu 138.



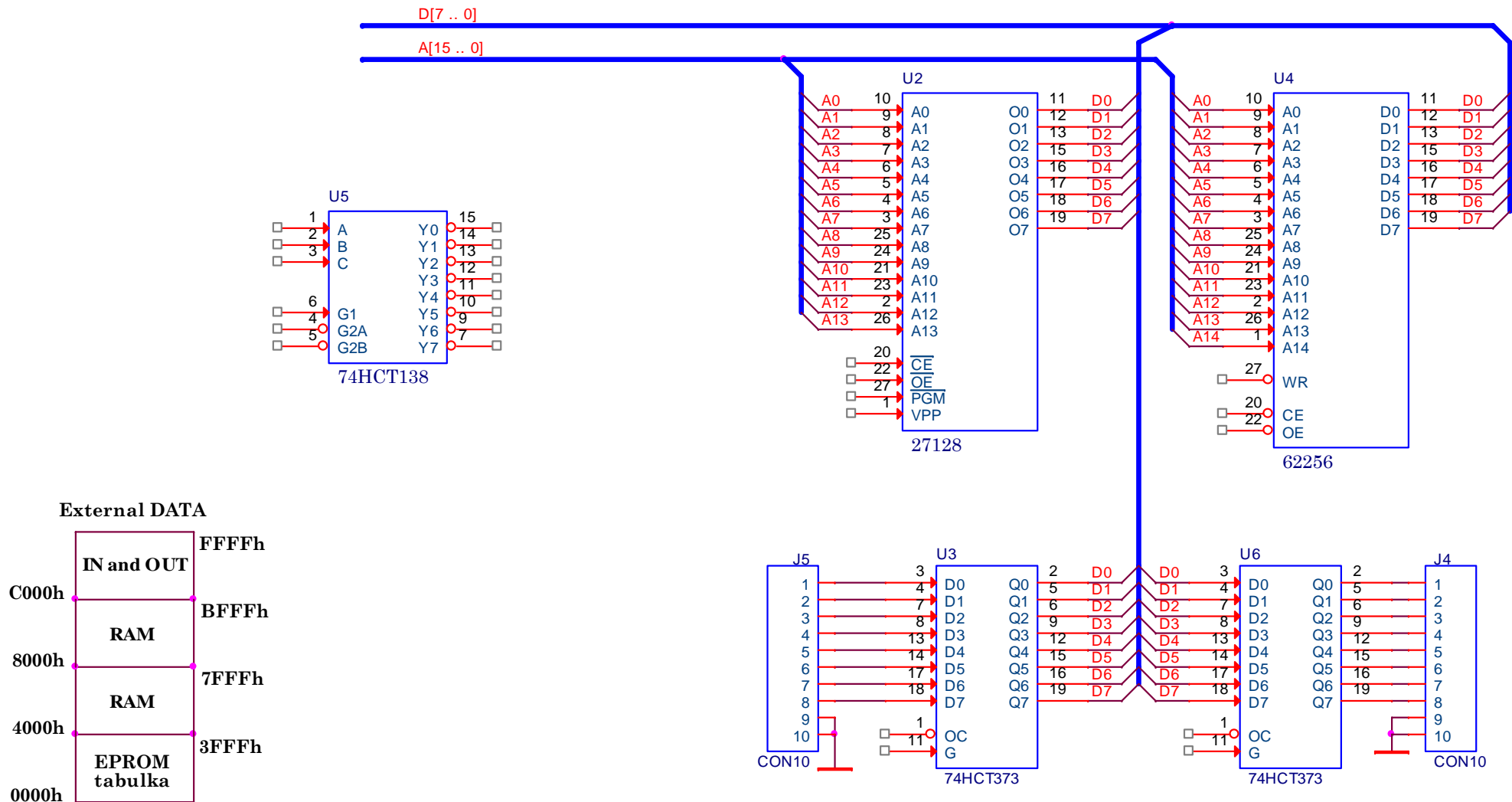
1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 28kB paměti EPROM 27128 a 32kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené. Přístup do programového prostoru ($PS=0$), čtení z datového ($RD=0$) a zápis do datového ($WR=0$).
2. Zrealizujte zapojení bez obvodu 138.



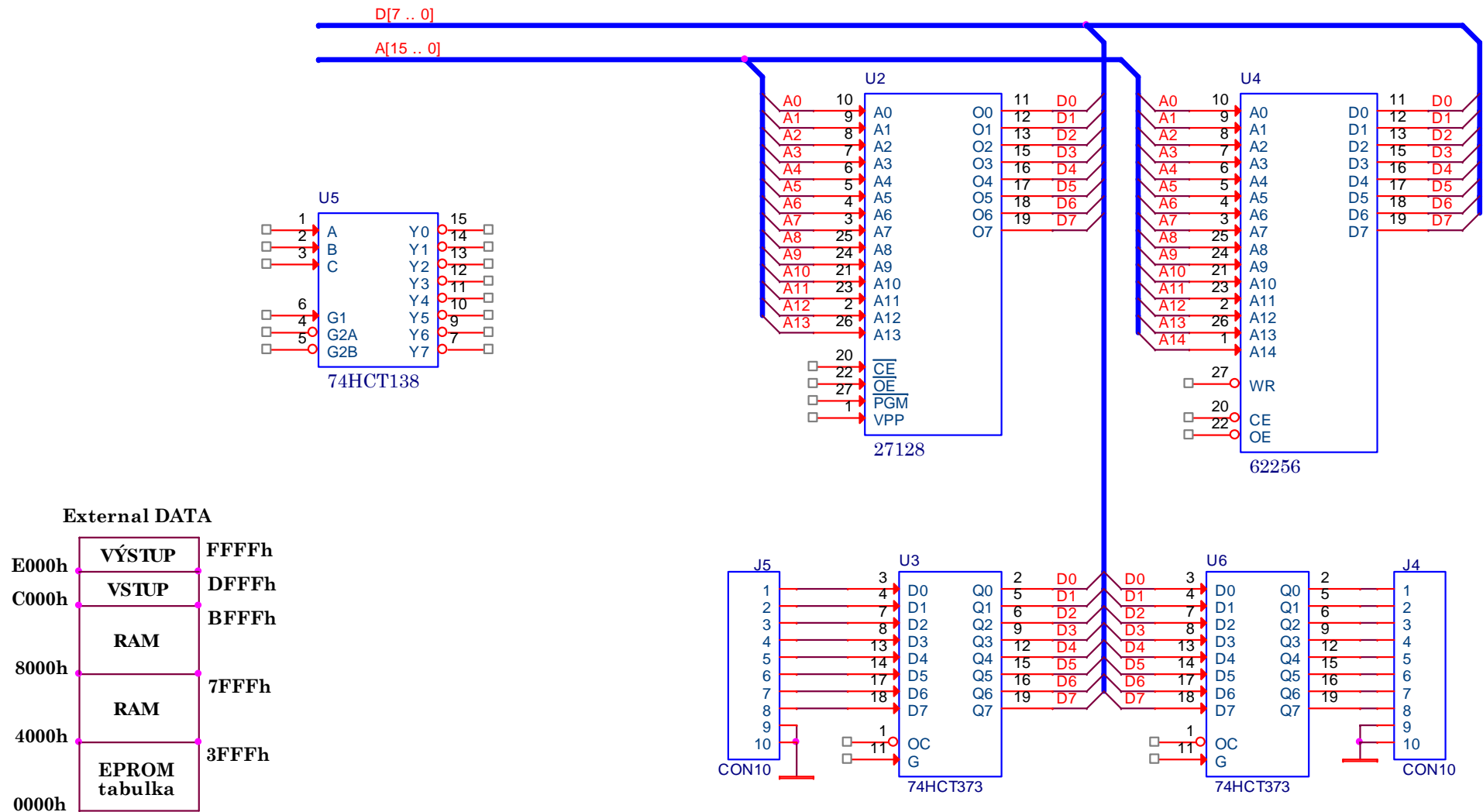
1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 2kB paměti EPROM 2716 a 32kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).



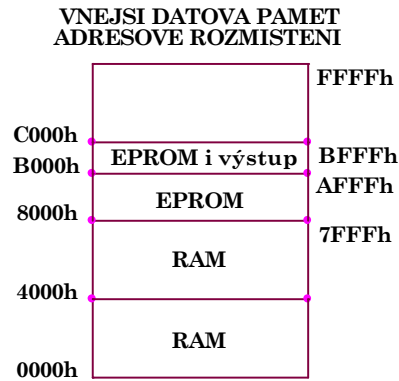
1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 16kB paměti EPROM 27128 a 32kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů U2, U3, U4, U5 a U6 musí zapojené. Přístup do datového prostoru je charakterizován těmito řídicími signály čtení (RD=0) a zápis (WR=0).
2. Zrealizujte zapojení bez obvodu 138.



1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 16kB paměti EPROM 27128 a 32kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů U2, U3, U4, U5 a U6 musí zapojené. Přístup do datového prostoru je charakterizován těmito řídicími signály čtení (RD=0) a zápis (WR=0).
2. Zrealizujte zapojení bez obvodu 138.



1. Odvod'te logické rovnice pro vstupy CSR, OER, WRR, CSE, OE, OC a ZAP tak, aby jednotlivé obvody měly přiřazené adresové prostory z obrázku. Aktivačním signálem (aktivním v log.0) pro datový prostor jsou signály RD a WR.



PŘIPOJENI PAMETI RAM

CSR =

OER =

WRR =

PŘIPOJENI PAMETI EPROM

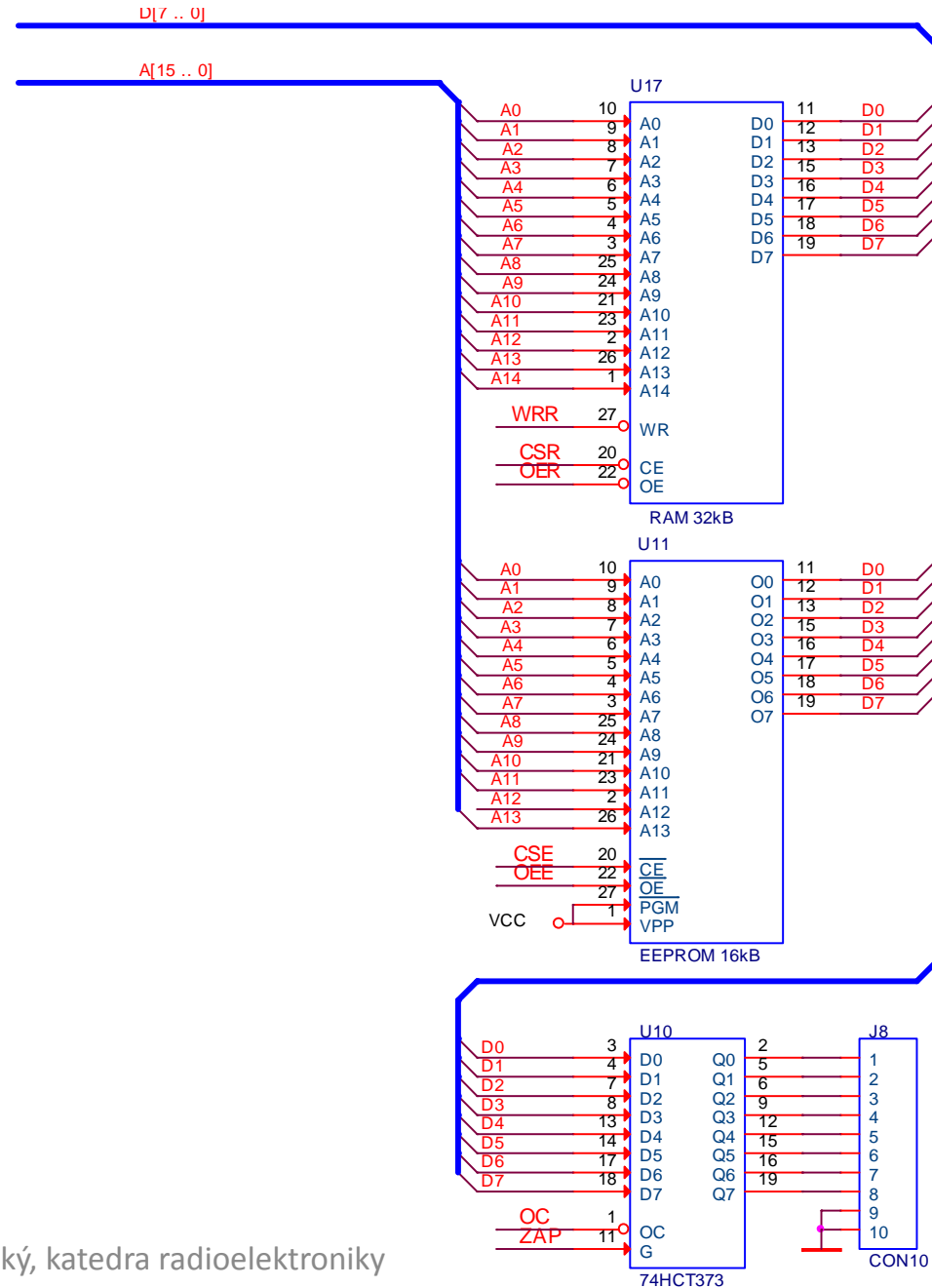
CSE =

OEE =

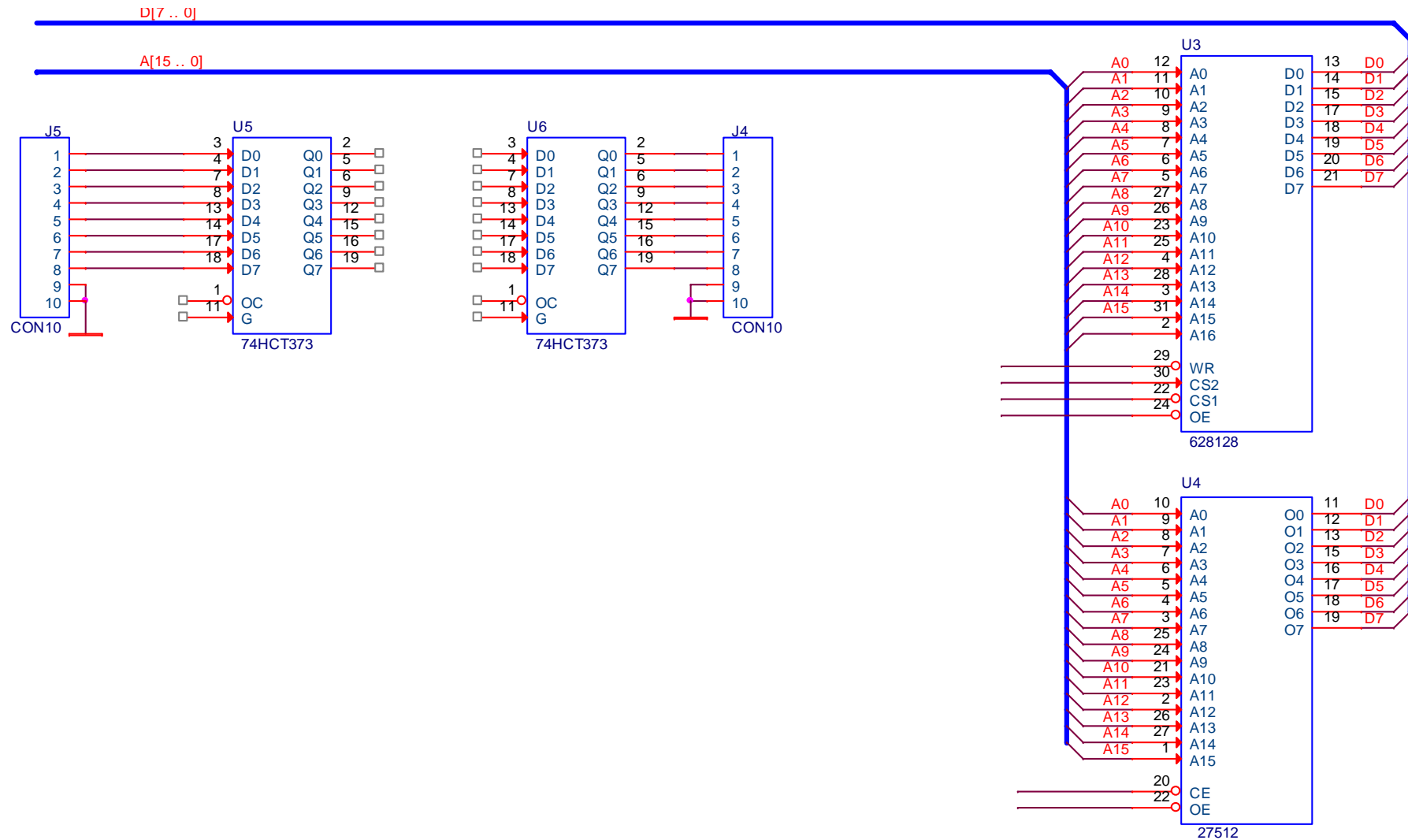
PŘIPOJENI VYSTUPNI BRANY

OC =

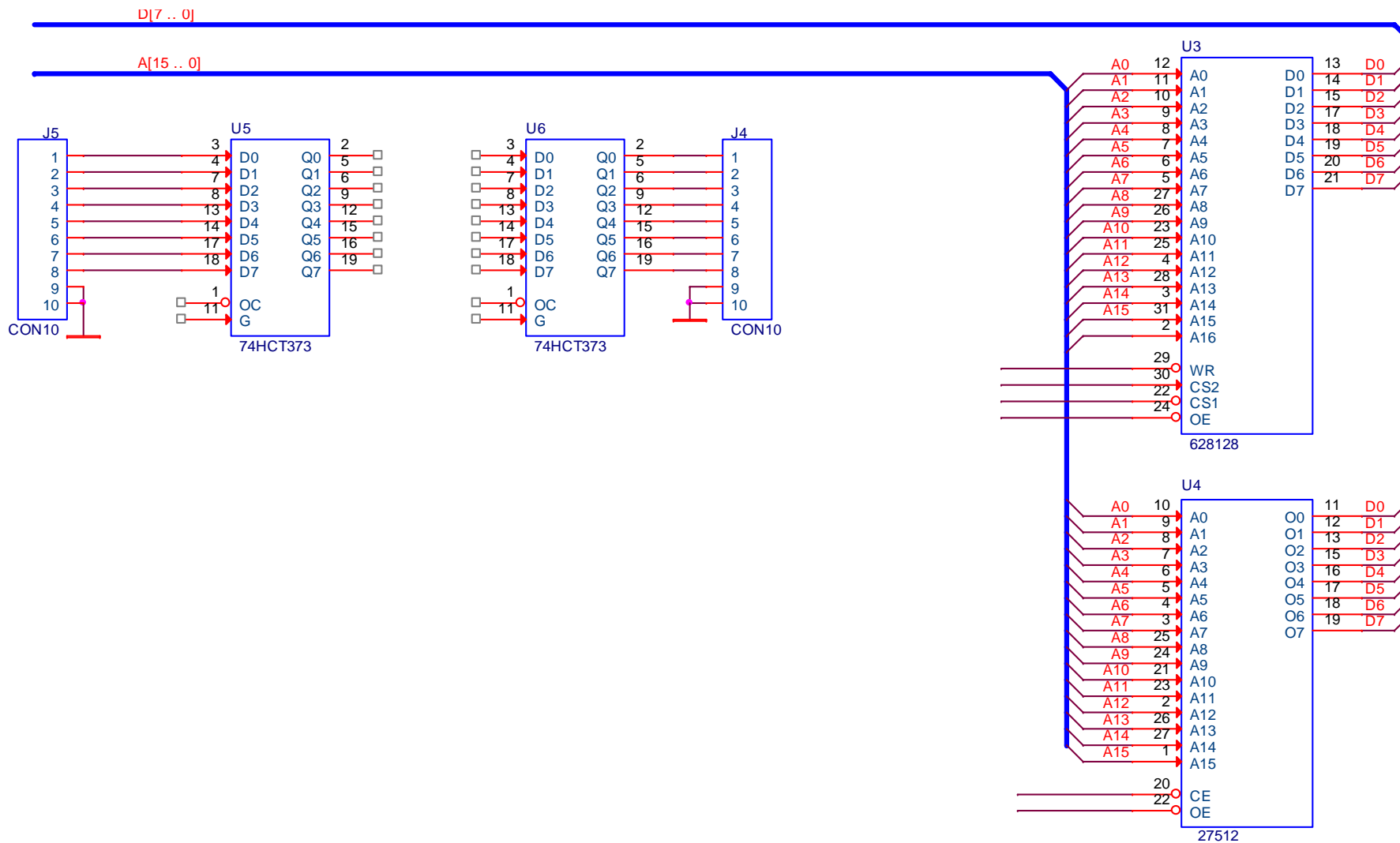
ZAP =



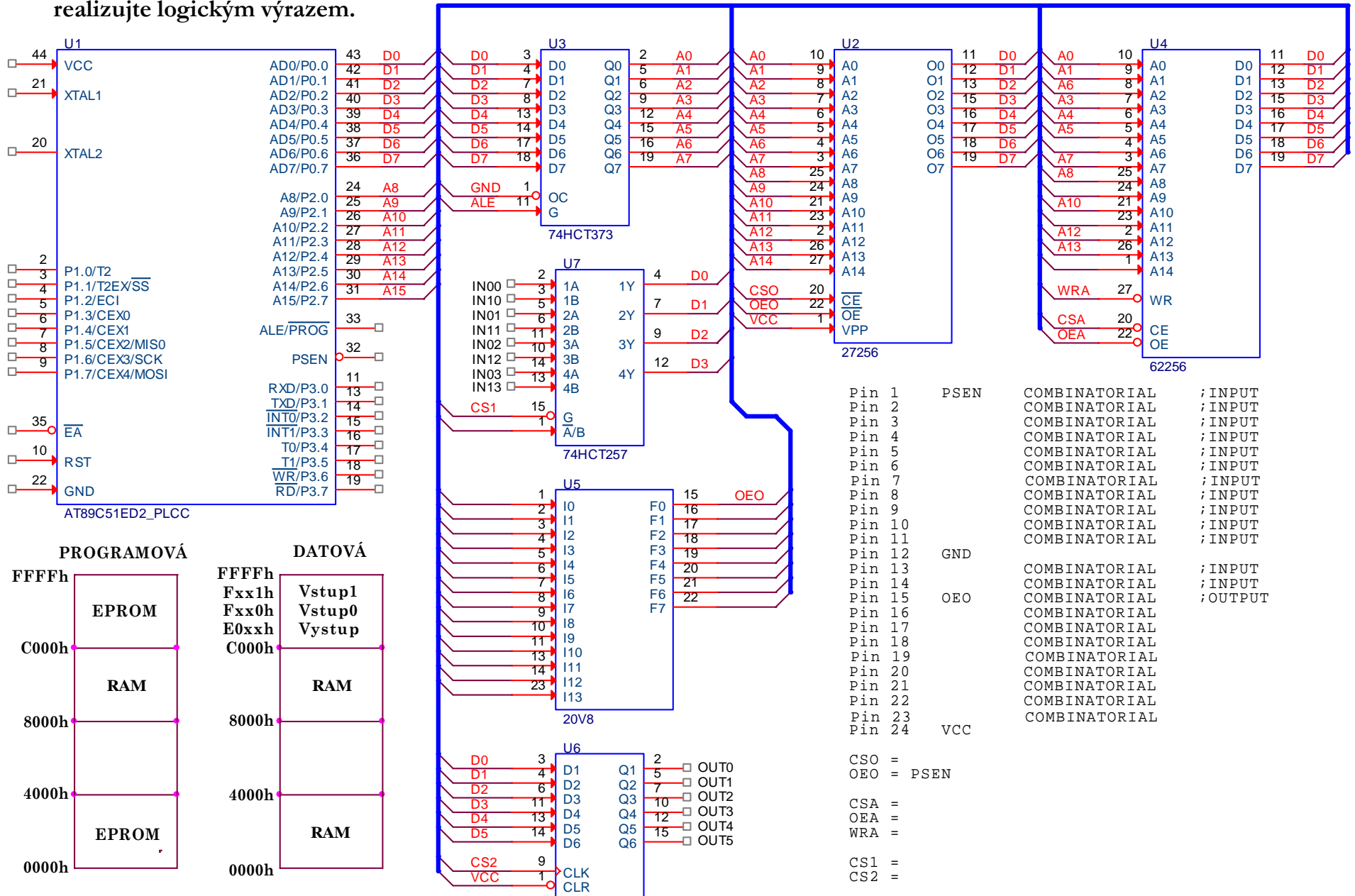
1. K procesoru s přímo dostupným adresovým prostorem 64kB připojte do vnějšího datového prostoru (RD=0 nebo WR=0) paměť EPROM (64kB), paměť RAM (128kB) a současně s jednou výstupní bránu tvořenou obvodem U6 a jednou vstupní bránu U5. Řešení realizujte logickými obvody, obvodem 138 a zápisem logických funkcí v součtové nebo součinnové formě.



1. V adresovém prostoru 8000h až FFFFh stránkujte paměť RAM o kapacitě 128KB, paměť EPROM o kapacitě 64kB a jednu vstupní a výstupní bránu. Řešení realizujte logickými obvody, obvodem 138 a zápisem logických funkcí v součtové nebo součinnové formě.

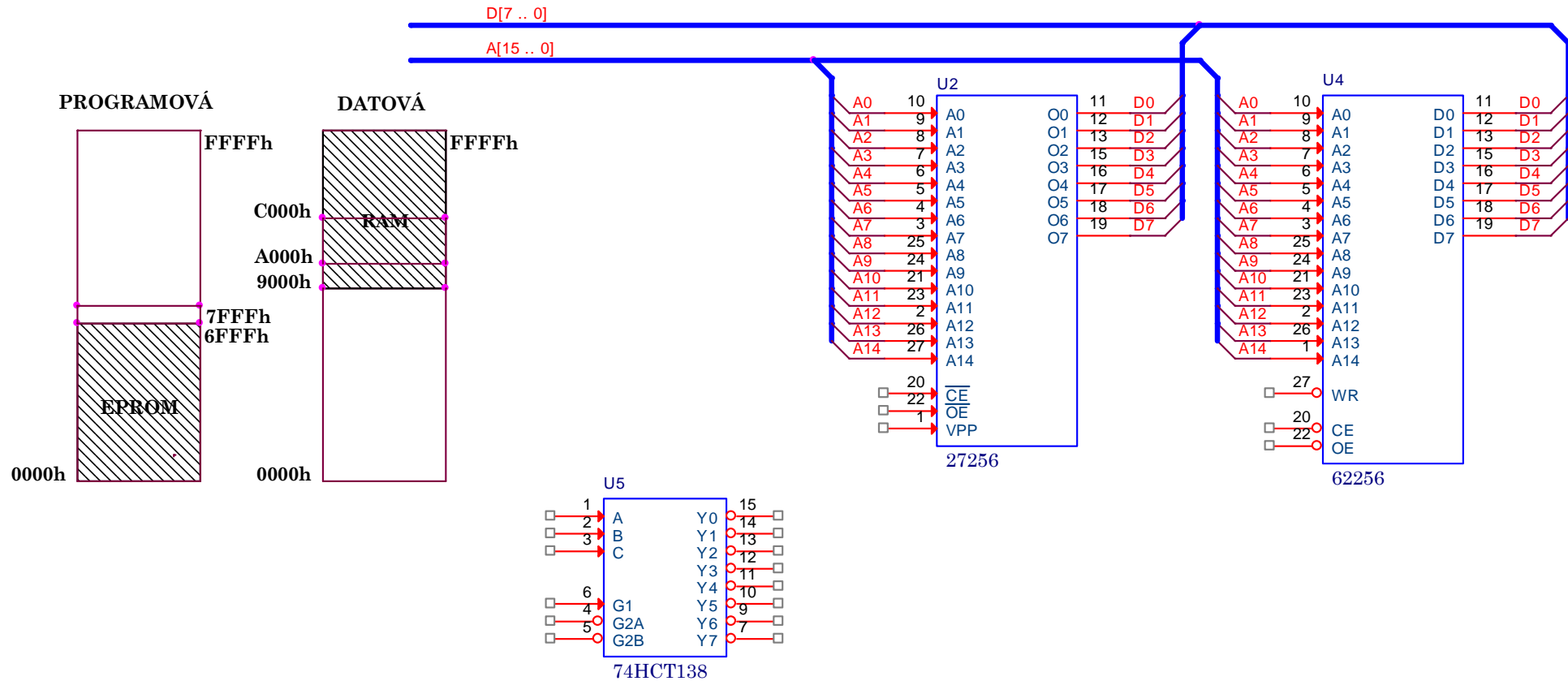


1. Pomocí programovatelného pole GAL20V8 připojte paměti 27256, 61256, vstupní brány 2x4bity a výstup 1x6bitů do paměťových prostorů uvedených v levé části obrázku. Všechny vývody obvodu U2 až U7 musí zapojené. Aktivační signály realizujte logickým výrazem.

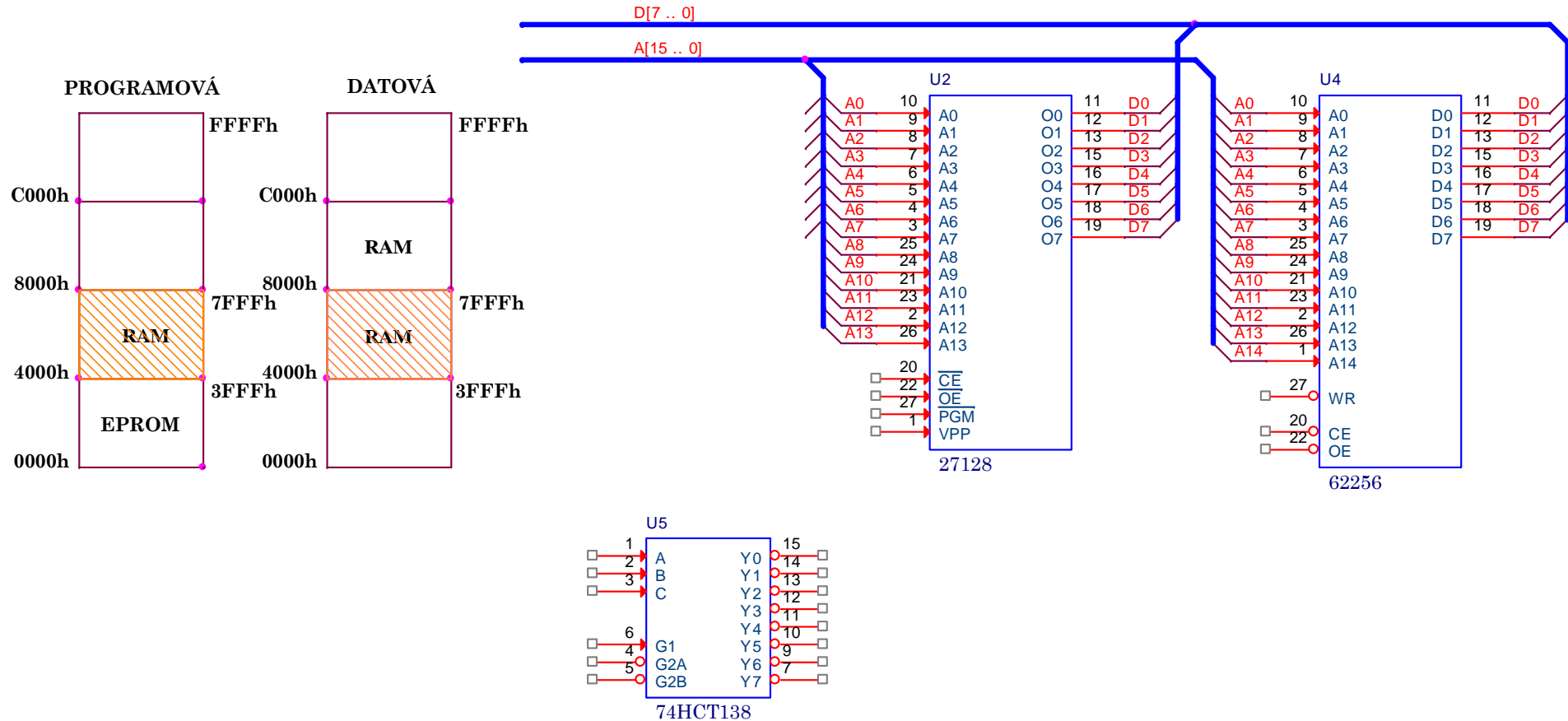


*Připojení pamětí
ke společné sběrnici
(atypická)*

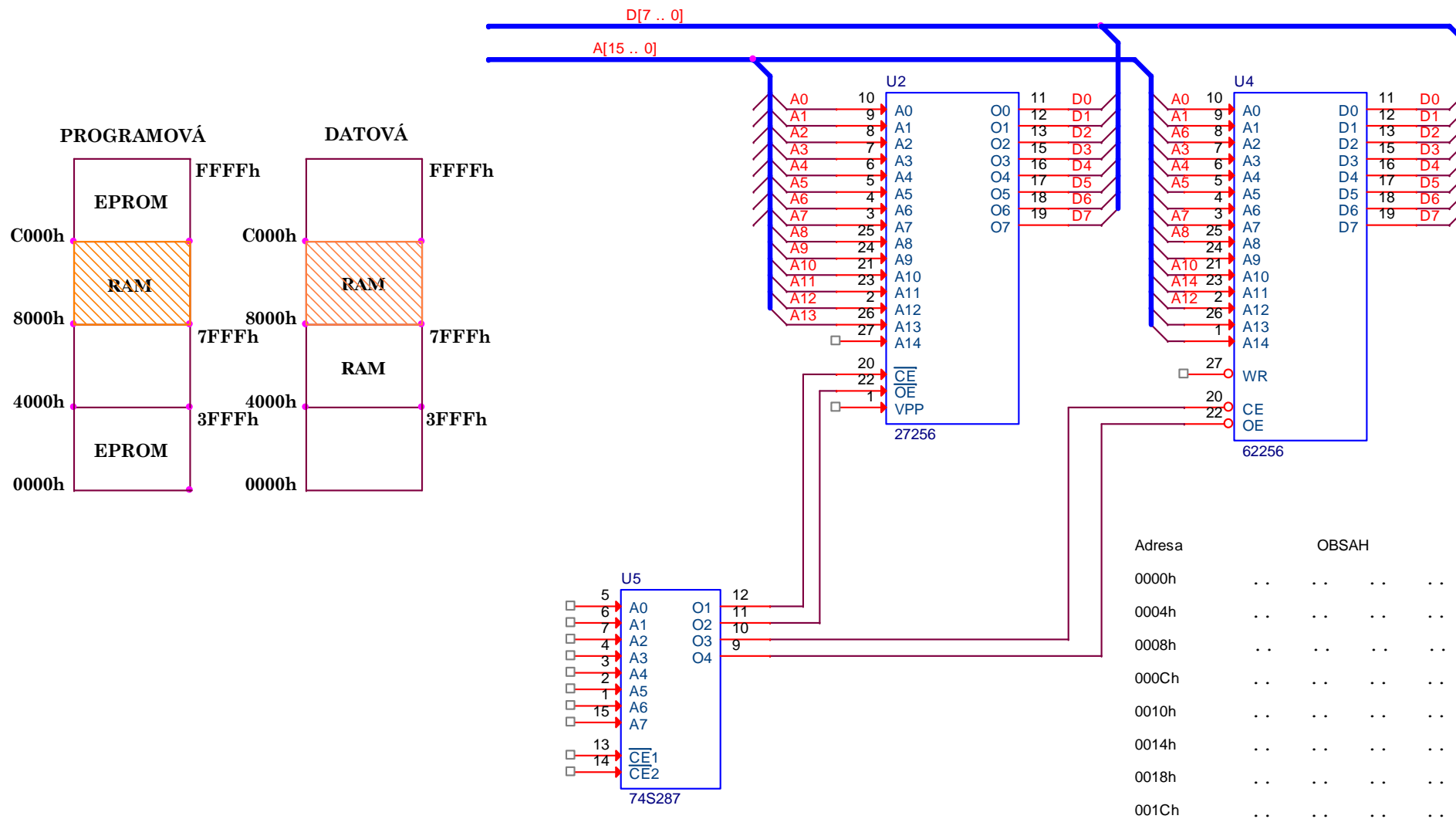
1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 28kB paměti EPROM 27128 a 28kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).
2. Zrealizujte zapojení bez obvodu 138.



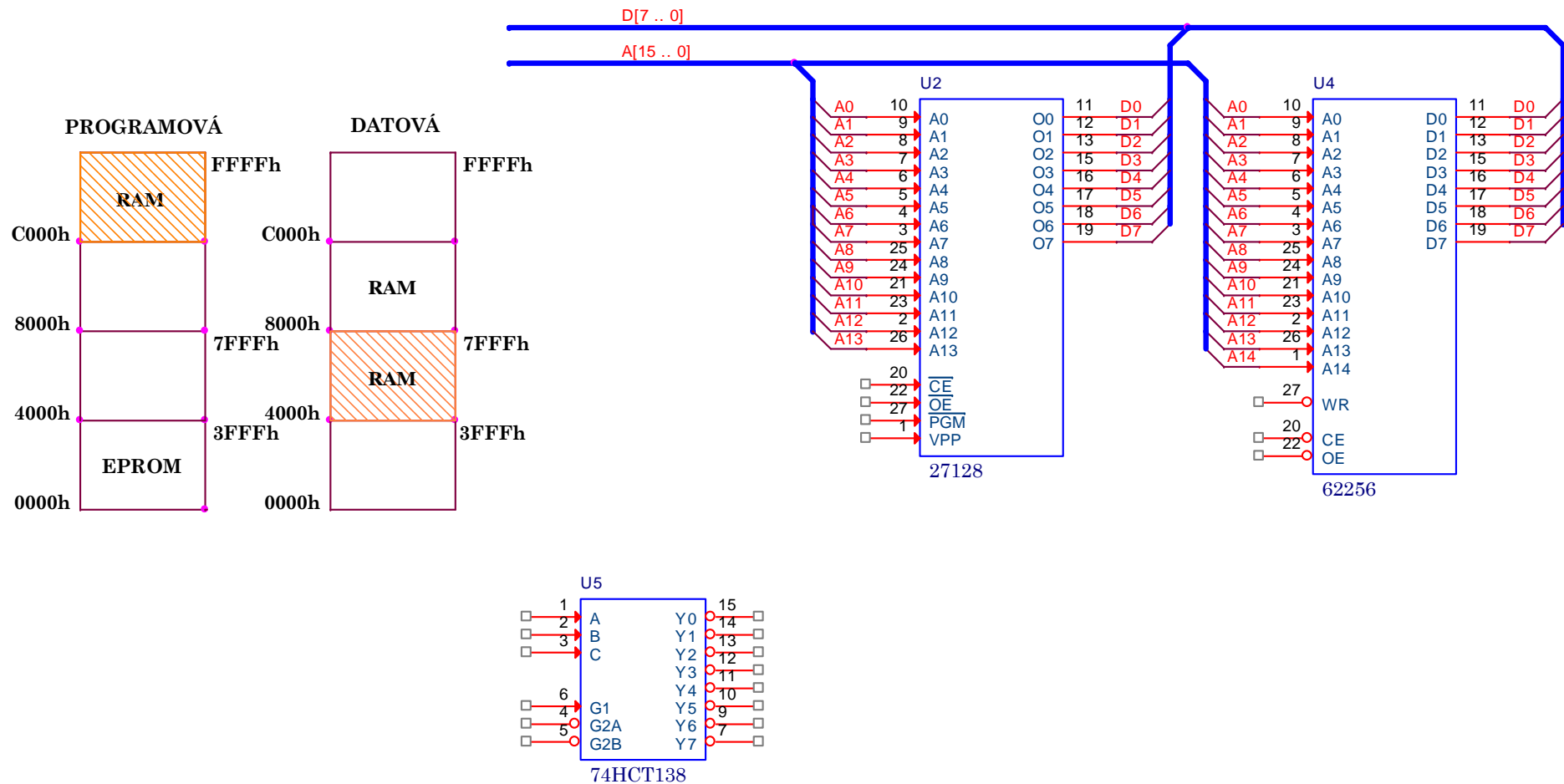
1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte paměť 27128 a 61256 do paměťových prostorů uvedených v levé části obrázku. Polovina paměti RAM leží současně v datovém i programovém prostoru. Zapojení vhodné pro vývojové systémy. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené.
2. Napište logické rovnice pro aktivací vstupy paměti.



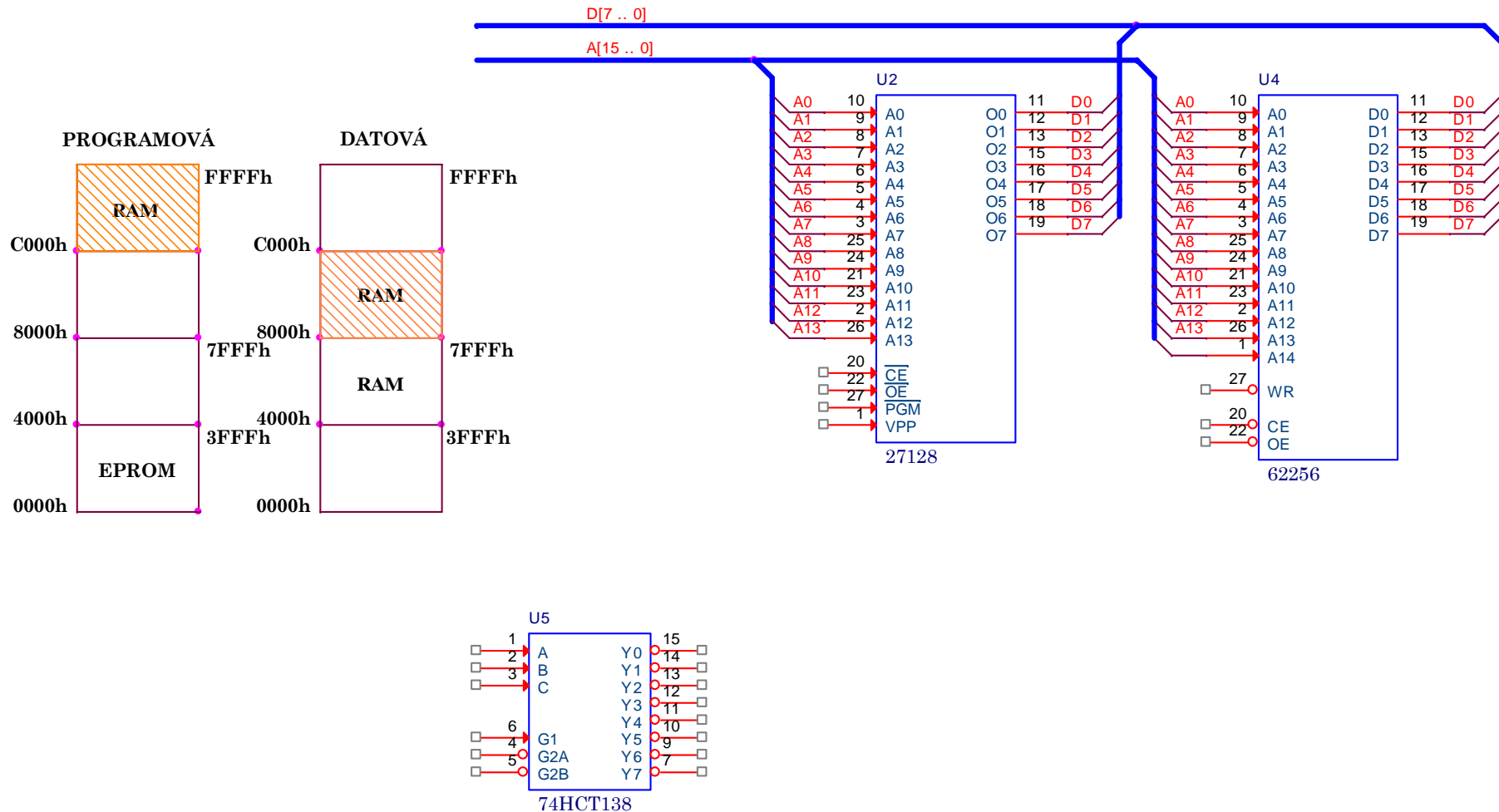
1. Pomocí bipolární paměti 74S287 připojte paměť 27256 a 61256 do paměťových prostorů uvedených v levé části obrázku a vyplňte v hex formě její obsah. Všechny nezbytné vývody obvodů U2 až U5 musí zapojené.



1. Připojte za pomoci obvodu 138 a log. člemů AND a OR 16kB paměť EPROM a 32kB paměť RAM do 16kB prostorů programové a datové paměti dle obrázku. Paměť RAM ve šrafovaném prostoru datové paměti v prostoru 4000÷7FFFh je současně pamětí programu v prostoru C000÷FFFFh.



1. Bylo by možné využívat adresový prostor datové paměti od 8000÷BFFFh jako paměť programu od adresy C000÷BFFFh, jak je zobrazeno na obrázku? Jak by muselo být zapojení upraveno?



ŘEŠENÍ

1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 2kB paměti EPROM 2716 a 32kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).

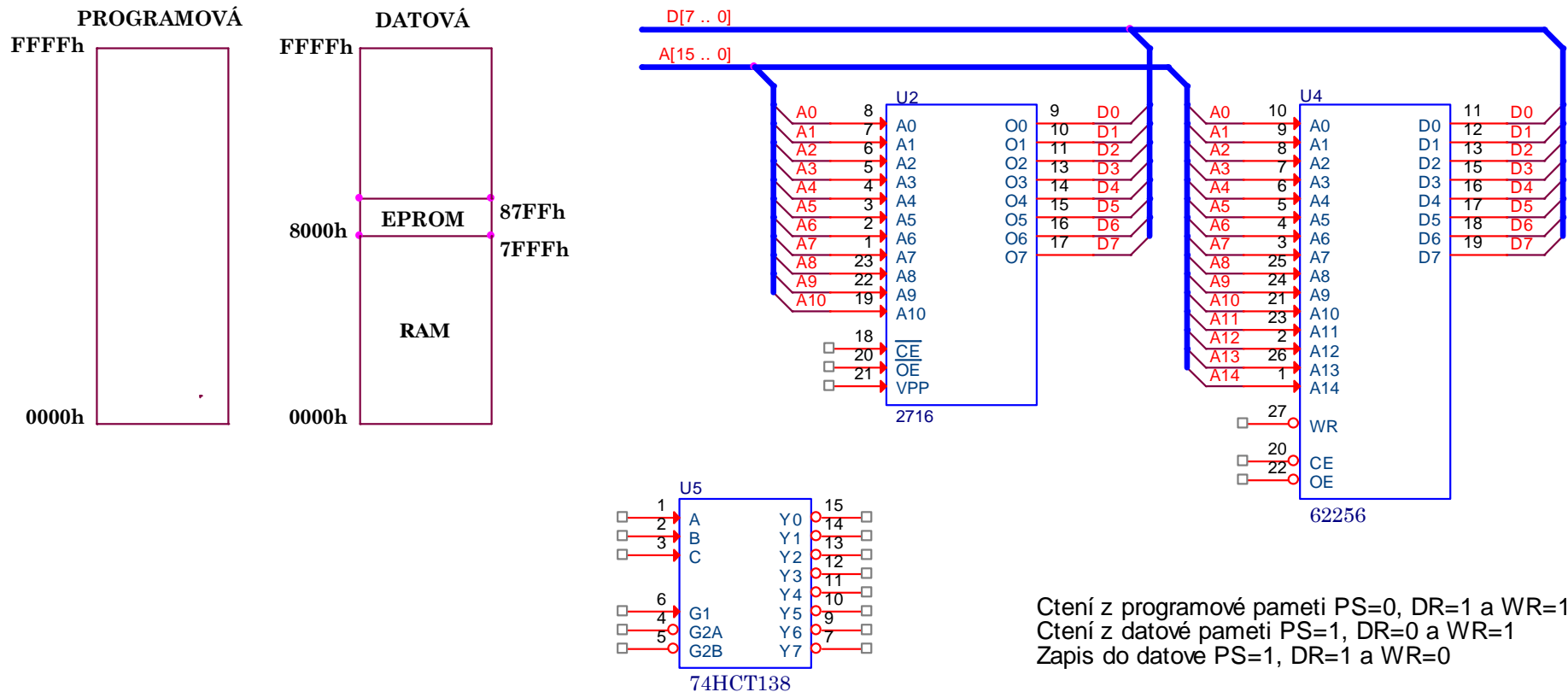
CS_EEPROM = $\neg A_{15} + A_{14} + A_{13} + A_{12}$ tabulka v datovém prostoru

OE_EEPROM = RD

CS_RAM = A15

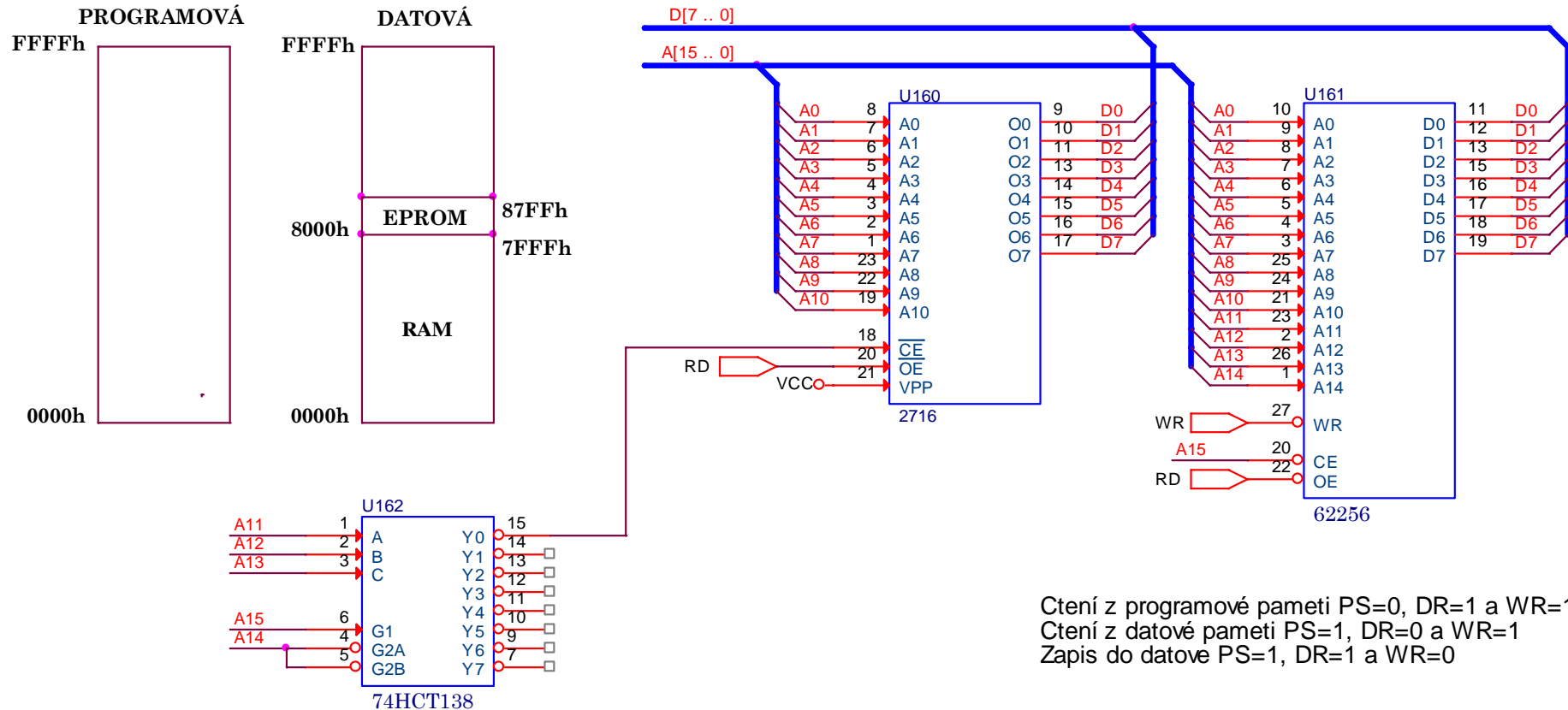
OE_RAM = RD

WR_RAM. = WR



Čtení z programové paměti PS=0, DR=1 a WR=1
 Čtení z datové paměti PS=1, DR=0 a WR=1
 Zápis do datové PS=1, DR=1 a WR=0

1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 2kB paměti EPROM 2716 a 32kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).



Ctení z programové paměti PS=0, DR=1 a WR=1
 Ctení z datové paměti PS=1, DR=0 a WR=1
 Zápis do datové PS=1, DR=1 a WR=0

V tomto případě na připojení A11, A12 a A13 nezáleží.

1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 28kB paměti EPROM 27128 a 32kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).
2. Zrealizujte zapojení bez obvodu 138.

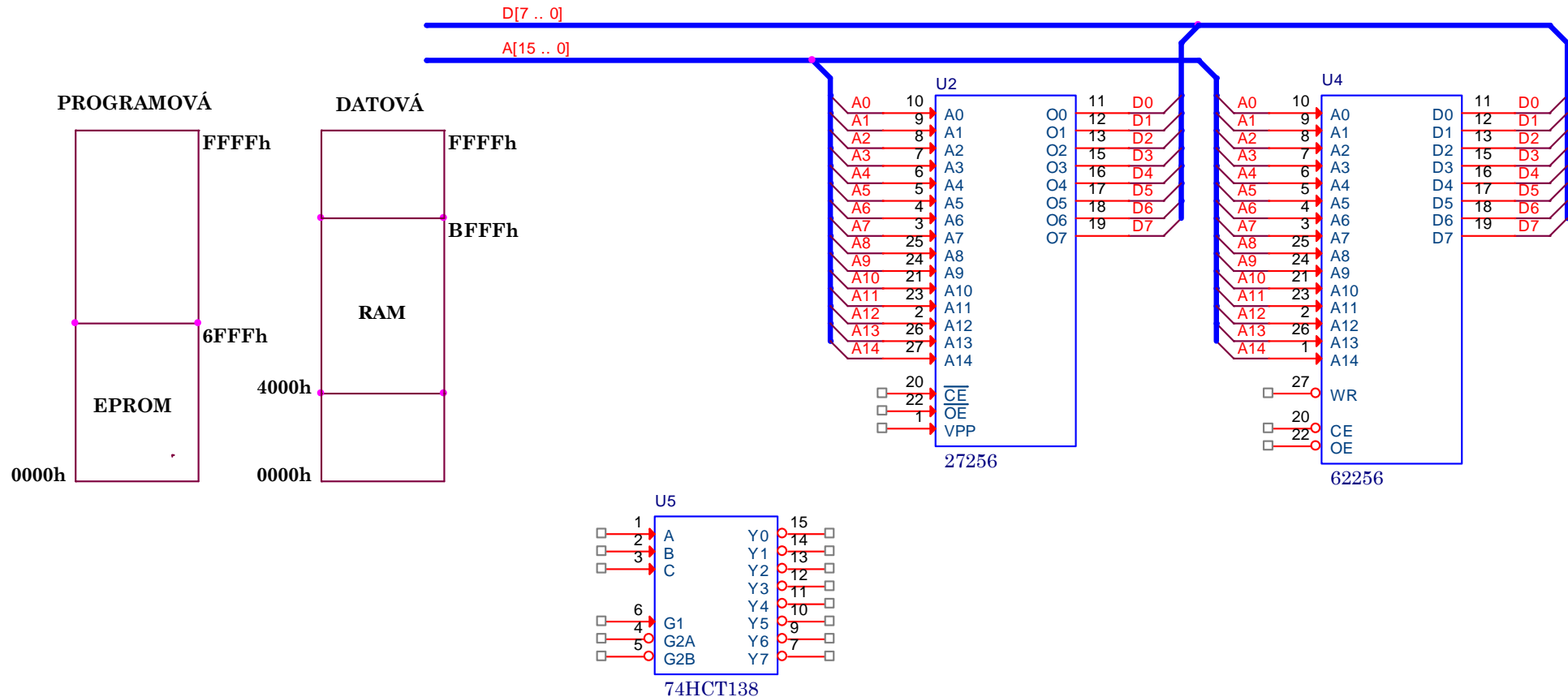
$$CS_EEPROM = A15 + A14.A13.A12 \quad \text{nebo} \quad CS_EEPROM = (A15 + A14).(A15 + /A14 + A13).(A15 + /A14 + /A13 + A12)$$

$$OE_EEPROM = PS$$

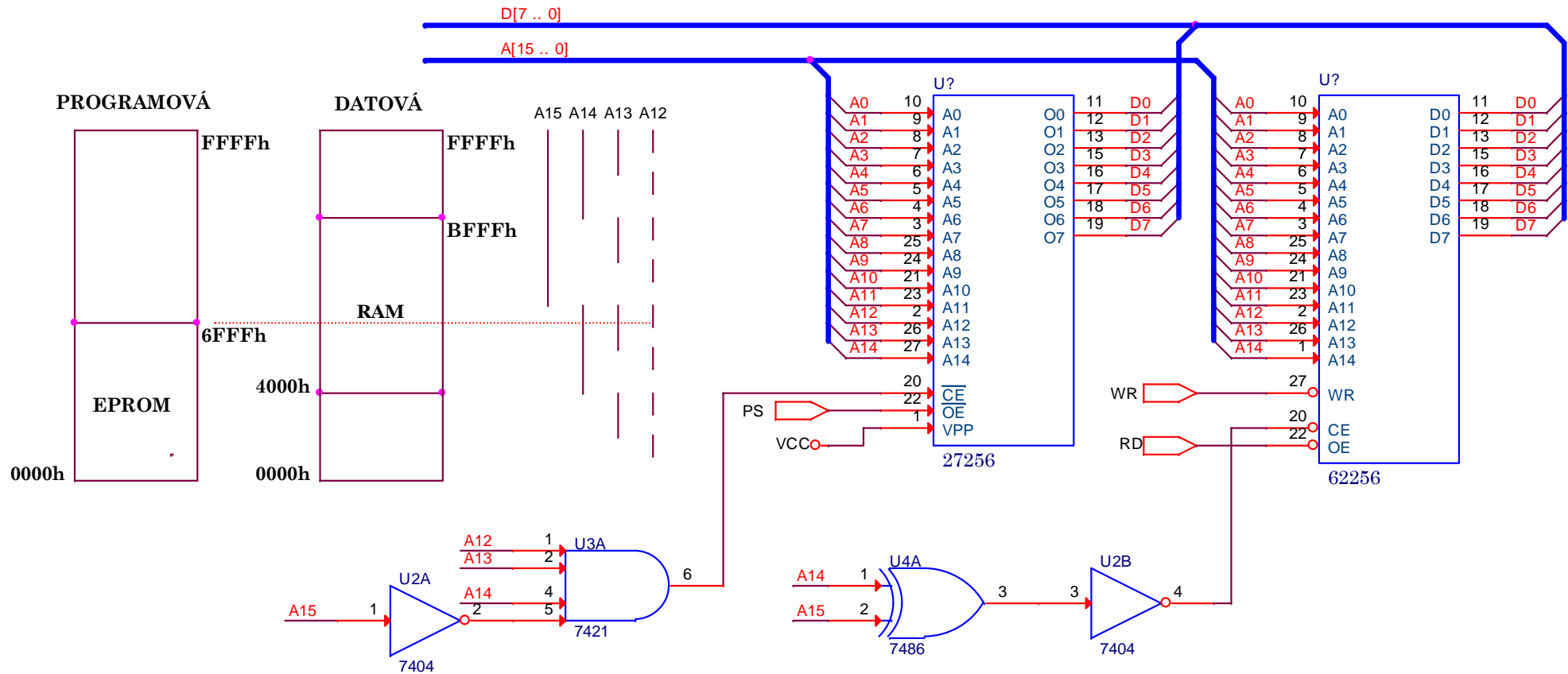
$$CS_RAM = A15 \otimes A14$$

$$OE_RAM = RD$$

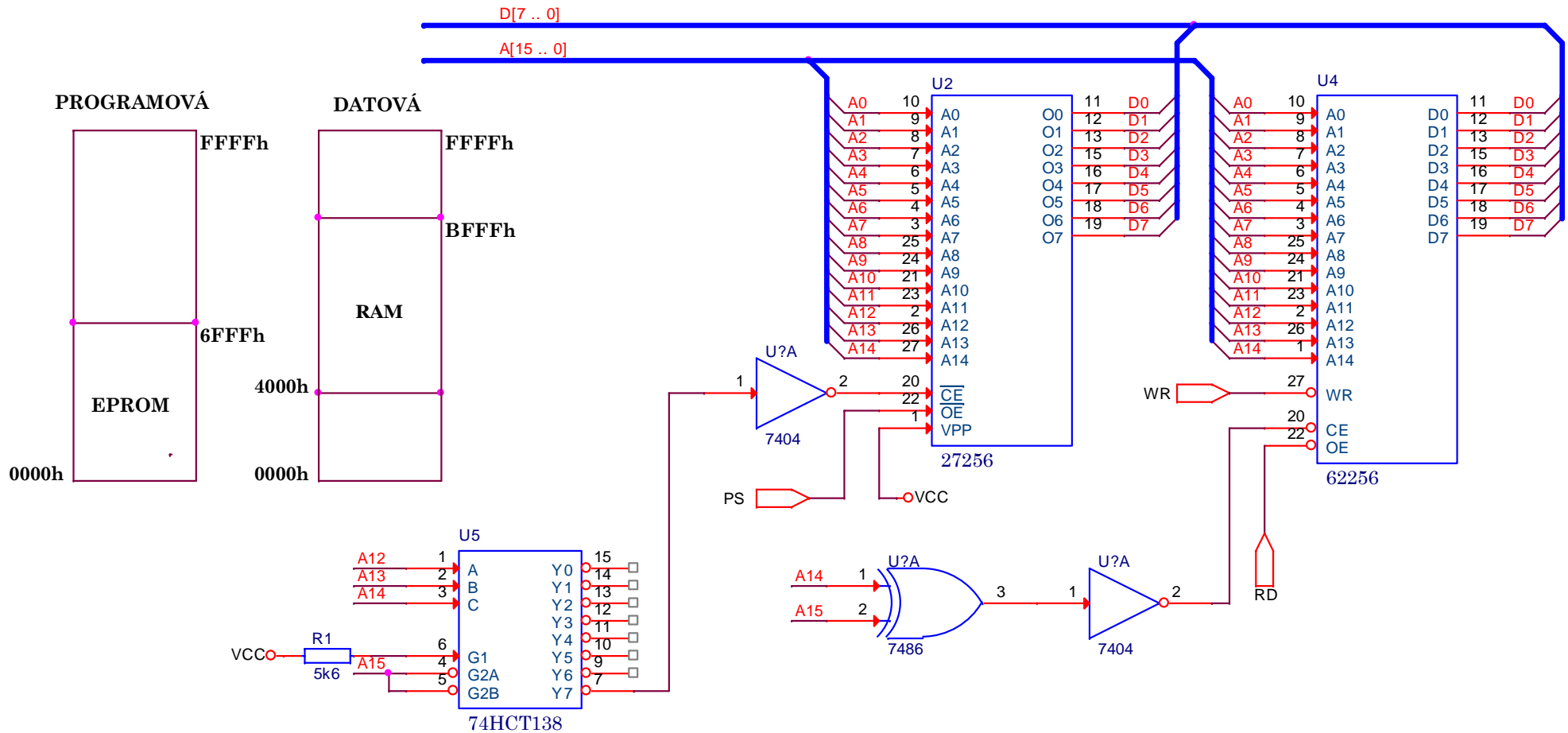
$$WR_RAM. = WR$$



1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 28kB paměti EPROM 27128 a 32kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).
2. Zrealizujte zapojení bez obvodu 138.



1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 28kB paměti EPROM 27128 a 32kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).
2. Zrealizujte zapojení bez obvodu 138.



1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 28kB paměti EPROM 27128 a 28kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).
2. Zrealizujte zapojení bez obvodu 138.

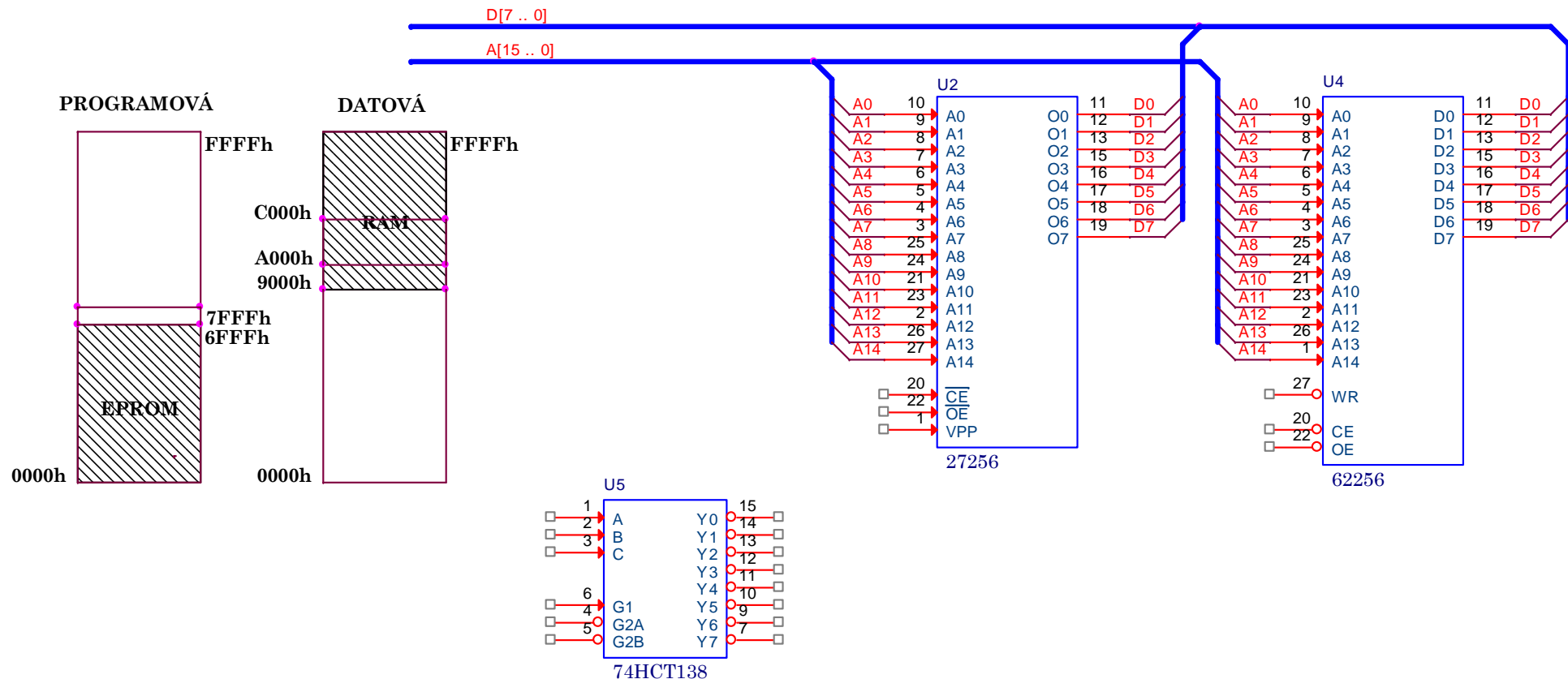
$$CS_EEPROM = \overline{A15} \cdot A14 \cdot A13 \cdot A12 \quad \text{nebo} \quad CS_EEPROM = (A15 + A14) \cdot (A15 + \overline{A14} + A13) \cdot (A15 + \overline{A14} + \overline{A13} + A12)$$

$$OE_EEPROM = PS$$

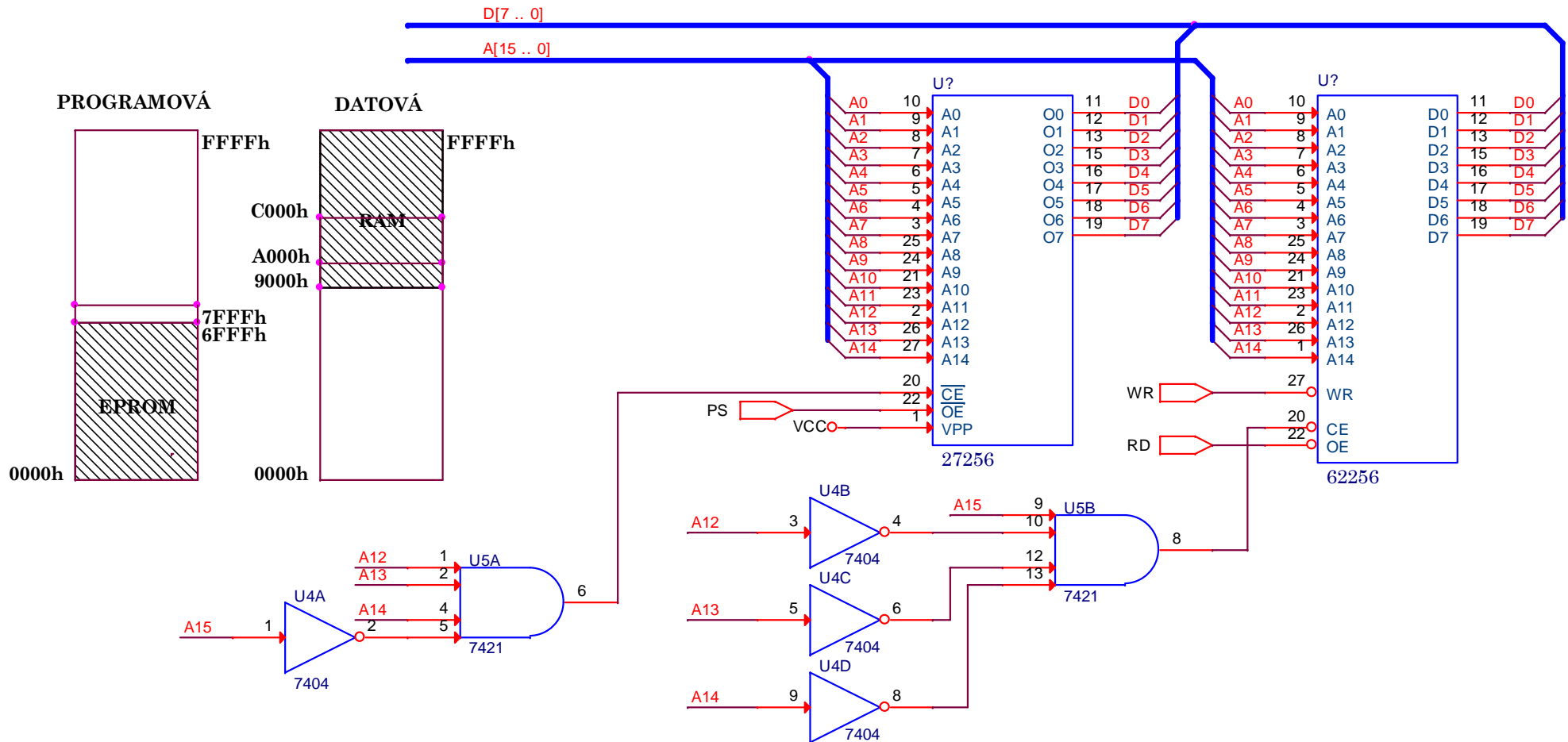
$$CS_RAM = A15 \cdot \overline{A14} \cdot \overline{A13} \cdot \overline{A12} \quad \text{nebo} \quad CS_EEPROM = (\overline{A15} + \overline{A14}) \cdot (\overline{A15} + A14 + \overline{A13}) \cdot (\overline{A15} + A14 + A13 + \overline{A12})$$

$$OE_RAM = RD$$

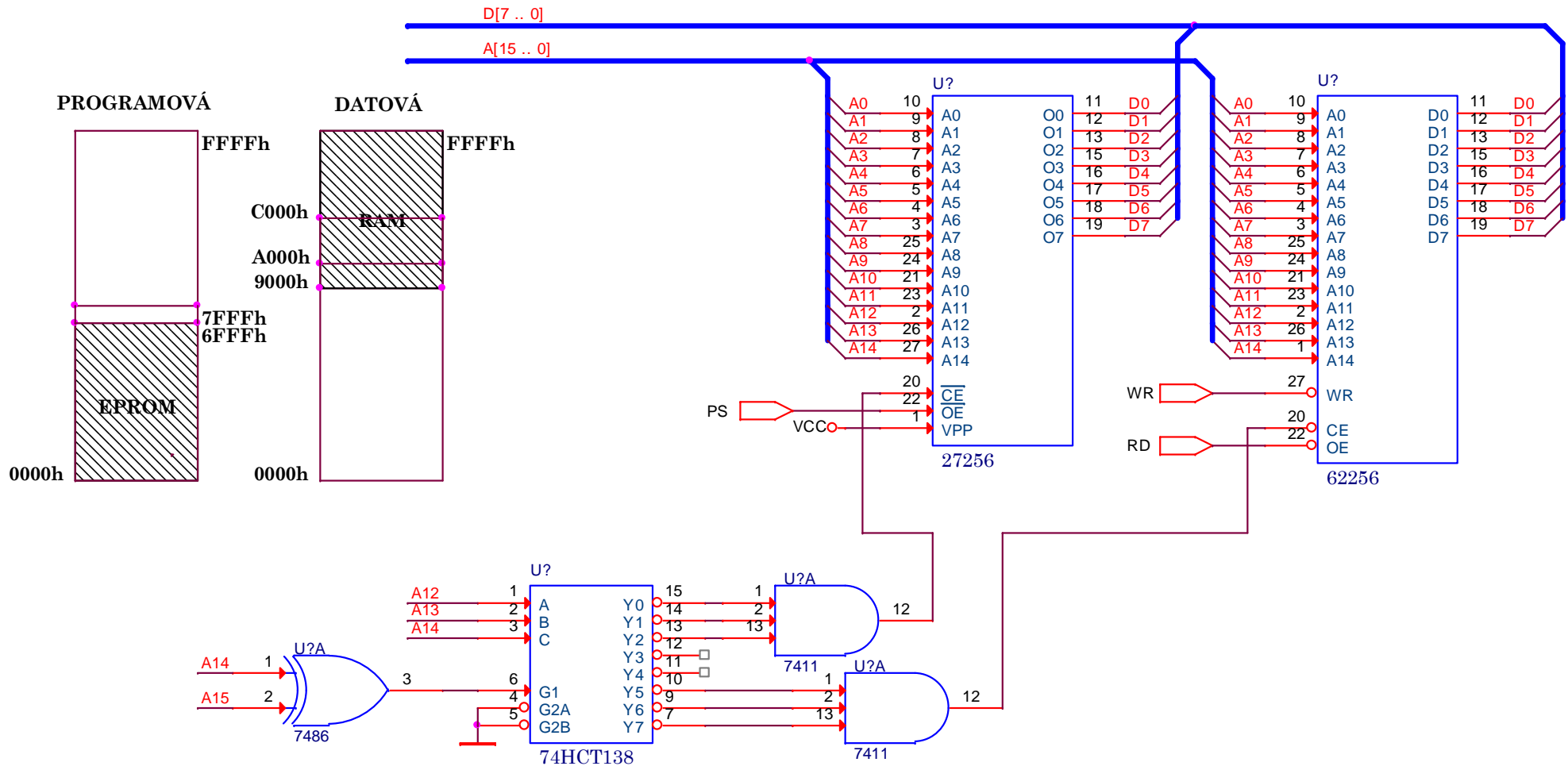
$$WR_RAM = WR$$



1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 28kB paměti EPROM 27128 a 28kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).
2. Zrealizujte zapojení bez obvodu 138.



1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte 28kB paměti EPROM 27128 a 28kB paměť RAM 62256 do paměťových prostorů uvedených na obrázku. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené. Přístup do programového prostoru (PS=0), čtení z datového (RD=0) a zápis do datového (WR=0).
2. Zrealizujte zapojení bez obvodu 138.



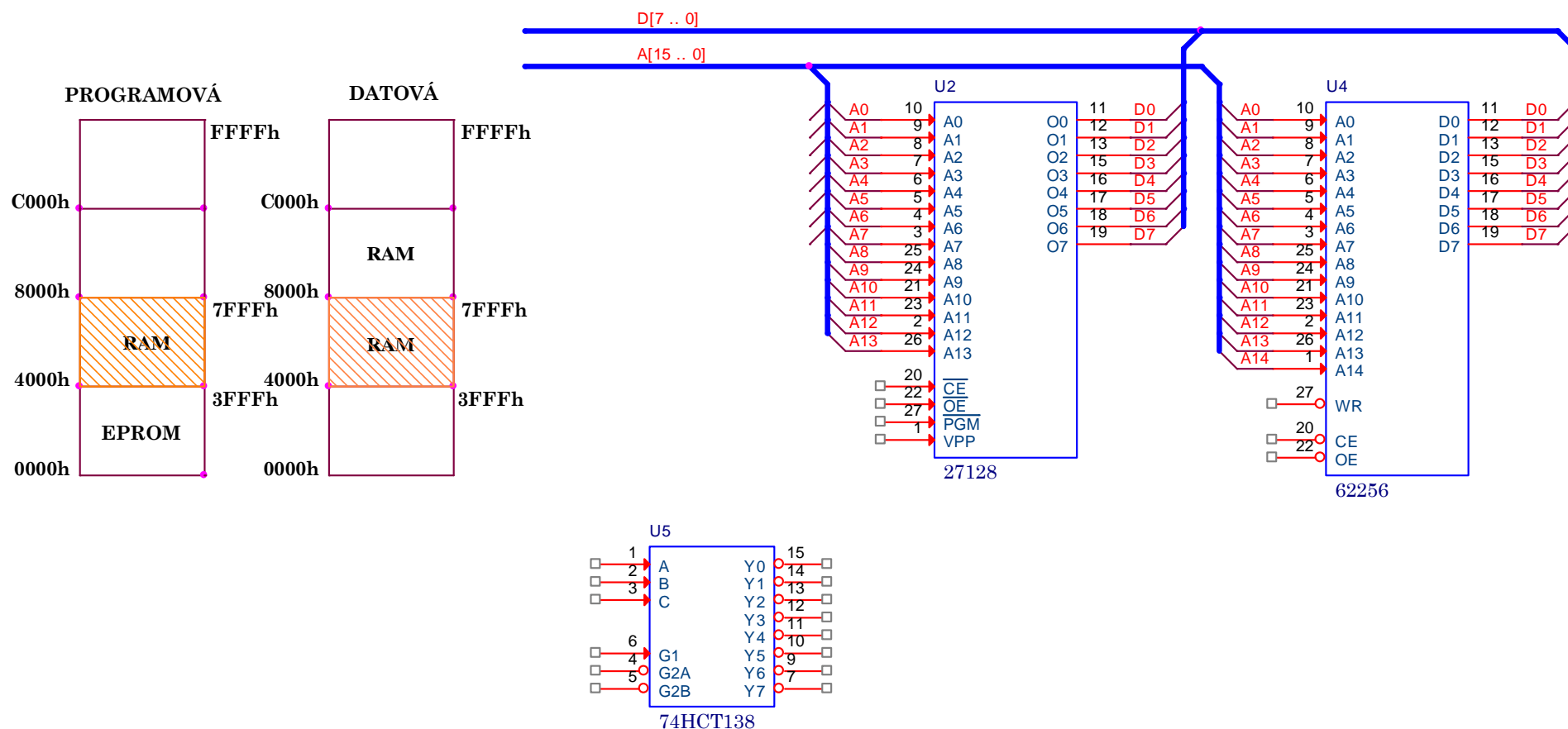
1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte paměť 27128 a 61256 do paměťových prostorů uvedených v levé části obrázku. Polovina paměti RAM leží současně v datovém i programovém prostoru. Zapojení vhodné pro vývojové systémy. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené.
2. Napište logické rovnice pro aktivací vstupy pamětí.

$$CS_EEPROM = A_{15} + A_{14} \quad \text{nebo} \quad CS_EEPROM = OE_EEPROM = A_{15} + A_{14} + PS$$

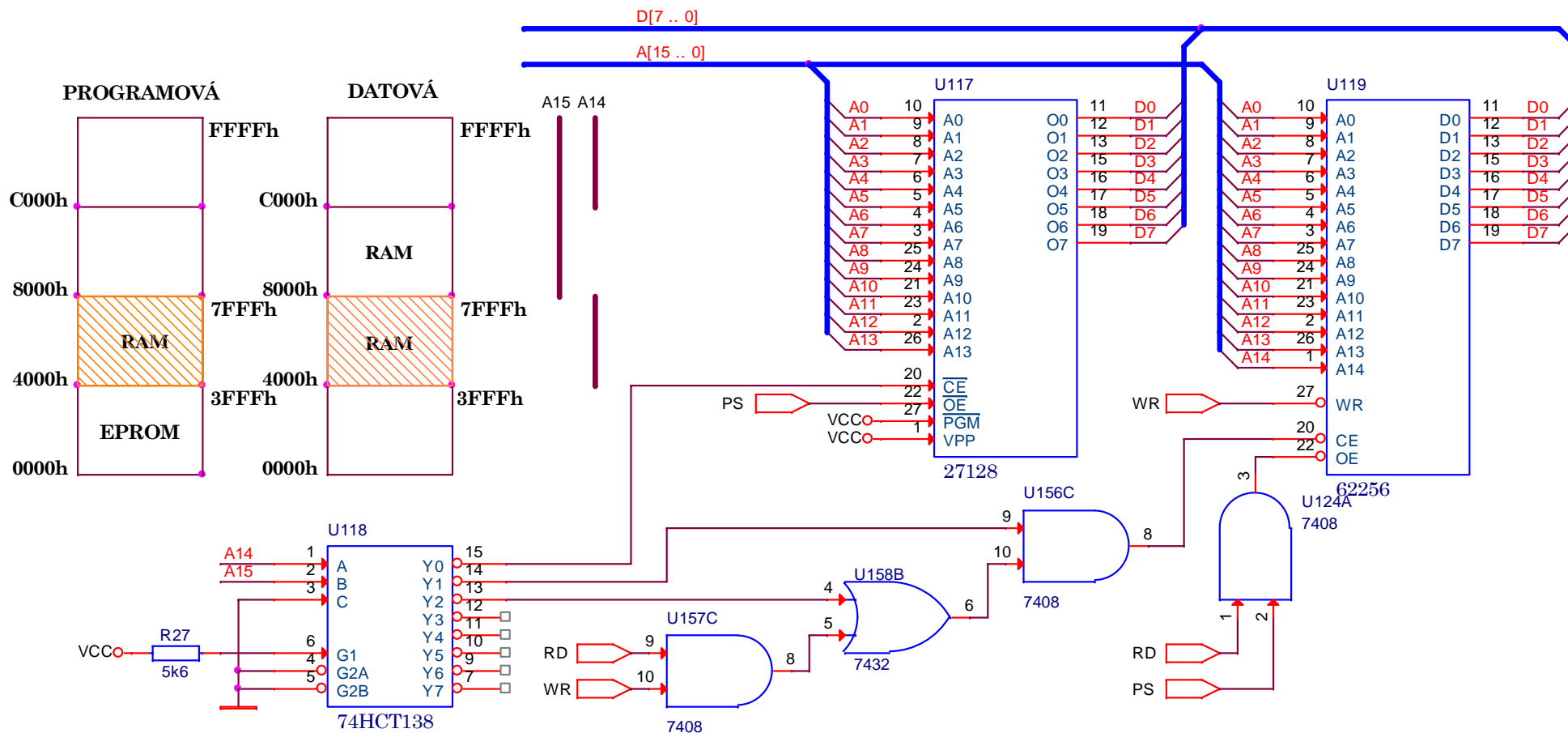
$$OE_EEPROM = PS$$

$$CS_RAM = A_{15} \otimes A_{14}$$

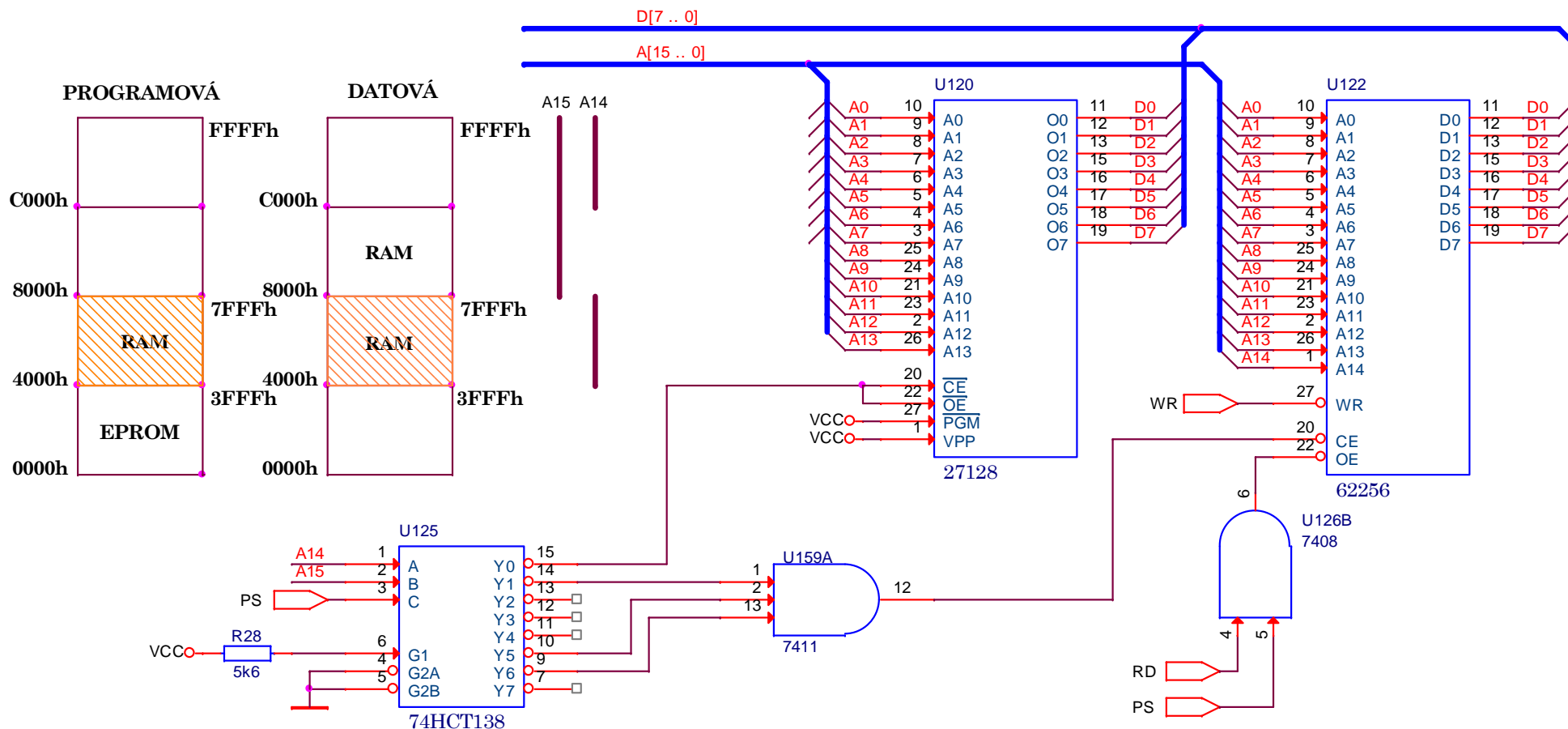
$$OE_EEPROM = (PS + A_{15} + / A_{14}).((A_{15} \otimes A_{14}) + RD.WR)$$



1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte paměť 27128 a 61256 do paměťových prostorů uvedených v levé části obrázku. Polovina paměti RAM leží současně v datovém i programovém prostoru. Zapojení vhodné pro vývojové systémy. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené.
2. Napište logické rovnice pro aktivací vstupy pamětí.



1. Pomocí hradel AND, OR, NAND nebo NOR a obvodu 138 zapojte paměť 27128 a 61256 do paměťových prostorů uvedených v levé části obrázku. Polovina paměti RAM leží současně v datovém i programovém prostoru. Zapojení vhodné pro vývojové systémy. Všechny nezbytné vývody obvodů U2, U4 a U5 musí zapojené.
2. Napište logické rovnice pro aktivační vstupy pamětí.



1. Pomocí bipolární paměti 74S287 připojte paměť 27256 a 61256 do paměťových prostorů uvedených v levé části obrázku a vyplňte v hex formě její obsah. Všechny nezbytné vývody obvodů U2 až U5 musí zapojené.

