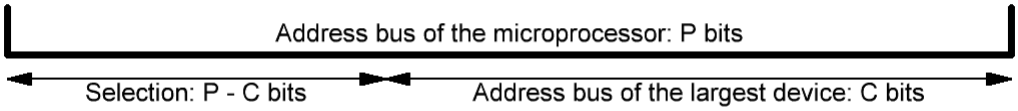
# Key to Tutorial 6
# Address Decoding

## Exercise 1

Let us consider $P$, the number of address lines of a microprocessor, and $C$, the number of address lines of the largest device connected to the microprocessor.

1. Write down the total number of devices that can be connected to the microprocessor using the linear-decoding technique in terms of $C$ and $P$.




   $P - C$ bits are available for device-selection purposes.

   The linear address decoding uses one address line per device. **Therefore, $P - C$ devices can be connected to the microprocessor.**

2. Write down the total number of devices that can be connected to the microprocessor using the block-decoding technique in terms of $C$ and $P$.

   The block address decoding uses one combination of address lines per device. **Therefore, $2^{P-C}$ devices can be connected to the microprocessor.**

## Exercise 2

A microprocessor system includes a ROM device, a RAM device and two peripheral devices (**P1** and **P2**). The capacities of these devices are 8 Mib, 8 Mib, 8 KiB and 4 KiB respectively. The microprocessor has a 24-bit address bus and all the components have an 8-bit data bus. The ROM should be located in the lowest part of the memory space, followed by the RAM, **P1** and **P2**.

1. Calculate the size of the address buses for each device.

   The size of an address bus can be determined by the depth of its device. All the devices are 8 bits wide, therefore **the depth of a device is its number of bytes**. Since the capacities of **P1** and **P2** are given in bytes, their capacities are also their depths.

   - **ROM** : 8 Mib = (8 Mi / 8) bytes = 1 MiB = $2^{20}$ bytes → **20 address lines**
   - **RAM** : 8 Mib = (8 Mi / 8) bytes = 1 MiB = $2^{20}$ bytes → **20 address lines**
   - **P1** : 8 KiB = $2^{13}$ bytes → **13 address lines**
   - **P2** : 4 KiB = $2^{12}$ bytes → **12 address lines**

***To begin with, you should use the linear-decoding technique.***

2. Which address lines are required to select the devices? Write down the devices associated with each of these address lines.

   With the linear address decoding, one address bit is paired with one device. These device-selection bits have to be the most significant. Since we have to connect 4 devices to the microprocessor, these bits have to be the address bits from *A23* to *A20*.

   The ROM should be located in the lowest part of the memory space, followed by the RAM, **P1** and **P2**. Therefore, these devices should be paired with *A20*, *A21*, *A22* and *A23* respectively.

| Address line | Device |
|---|---|
| A20 | ROM |
| A21 | RAM |
| A22 | P1 |
| A23 | P2 |

← The ROM should be located in the lowest part of the memory space.

3. The *AS* (Address Strobe) output signal of the microprocessor is active when its address bus contains a valid address. Take the signal *AS* into account to obtain an expression for each output of the address decoder.

   When the value on the address bus is invalid (*AS* = 0), none of the devices should be activated.

   - $CS_{ROM}$ = AS.A20
   - $CS_{RAM}$ = AS.A21
   - $CS_{P1}$ = AS.A22
   - $CS_{P2}$ = AS.A23

4. Draw the memory map. The lowest and highest addresses of each device should appear on the map.

   For each device, we have to determine the most significant address bits (the device-selection bits) and the least significant address bits (the address of the selected device) of the microprocessor. Any un-used address bits will be set to 0.

   Take the ROM for instance:
   - *A20* is set to 1 to select the ROM device.
   - *A21*, *A22* and *A23* are set to 0 to deactivate all the other devices.
   - To obtain its lowest address, its 20 address bits are set to 0.
   - To obtain its highest address, its 20 address bits are set to 1.

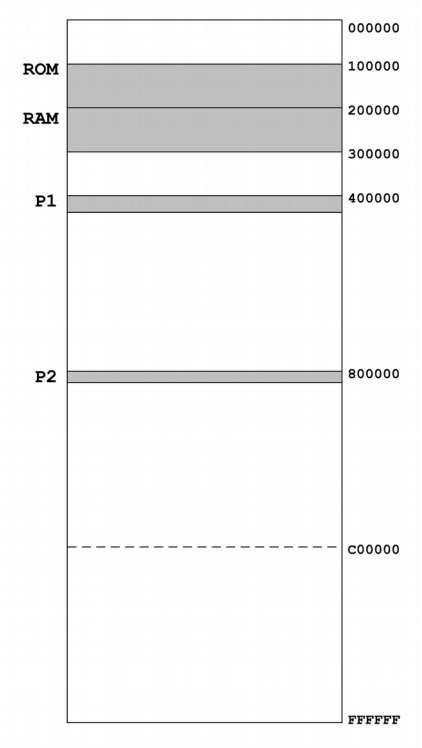
Here is what the memory map should look like:

ROM (lowest): **0001 0000 0000 0000 0000 0000**$_2$ = $100000_{16}$
ROM (highest): **0001 1111 1111 1111 1111 1111**$_2$ = $1FFFFF_{16}$

RAM (lowest): **0010 0000 0000 0000 0000 0000**$_2$ = $200000_{16}$
RAM (highest): **0010 1111 1111 1111 1111 1111**$_2$ = $2FFFFF_{16}$

P1 (lowest): **0100** 0000 0000 **0000 0000 0000**$_2$ = $400000_{16}$
P1 (highest): **0100** 0000 0001 **1111 1111 1111**$_2$ = $401FFF_{16}$

P2 (lowest): **1000** 0000 0000 **0000 0000 0000**$_2$ = $800000_{16}$
P2 (highest): **1000** 0000 0000 **1111 1111 1111**$_2$ = $800FFF_{16}$




5. What is the main drawback of this decoding technique?

   The main drawback of the linear-decoding technique is that several devices can be activated at the same time, which could lead to access conflicts and may cause severe damage.

6. Which addresses should never appear on the address bus of the microprocessor?

   A forbidden address selects several devices at the same time. This means that at least two of its device-selection bits are 1. Therefore, a forbidden address should never appear on the address bus.

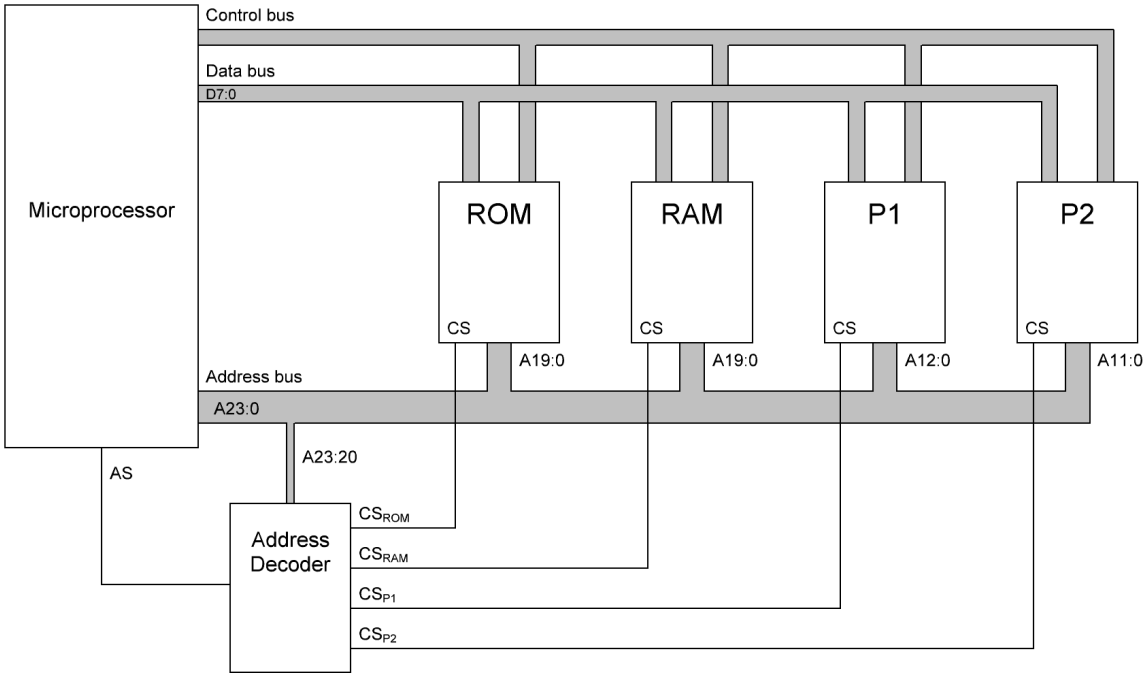
| A23 | A22 | A21 | A20 | Device |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | None of the devices are selected |
| 0 | 0 | 0 | 1 | ROM |
| 0 | 0 | 1 | 0 | RAM |
| **0** | **0** | **1** | **1** | **Forbidden addresses from $300000_{16}$ to $3FFFFF_{16}$** |
| 0 | 1 | 0 | 0 | P1 |
| **0** | **1** | **0** | **1** | **Forbidden addresses from $500000_{16}$ to $7FFFFF_{16}$** |
| **0** | **1** | **1** | **0** | |
| **0** | **1** | **1** | **1** | |
| 1 | 0 | 0 | 0 | P2 |
| **1** | **0** | **0** | **1** | **Forbidden addresses from $900000_{16}$ to $FFFFFF_{16}$** |
| **1** | **0** | **1** | **0** | |
| **1** | **0** | **1** | **1** | |
| **1** | **1** | **0** | **0** | |
| **1** | **1** | **0** | **1** | |
| **1** | **1** | **1** | **0** | |
| **1** | **1** | **1** | **1** | |

7. Suggest a simple solution to solve this problem.

   The address decoder should be changed so that only one device can be selected at a time. When the address decoder activates a device, it should deactivate all the other devices. In other words, the *CS* input of a device should be activated only when its device-selection bit is 1 and those of the other devices are 0.

   - $CS_{ROM} = AS.\overline{A23}.\overline{A22}.\overline{A21}.A20$
   - $CS_{RAM} = AS.\overline{A23}.\overline{A22}.A21.\overline{A20}$
   - $CS_{P1} = AS.\overline{A23}.A22.\overline{A21}.\overline{A20}$
   - $CS_{P2} = AS.A23.\overline{A22}.\overline{A21}.\overline{A20}$

8. Draw the circuit diagram.




***A new peripheral device (P3) should be added. Its capacity is 2 KiB (11 address lines).***

9. Can the linear-decoding technique still be used? Give reasons for your views.

   With the linear address decoding, one address bit of the microprocessor is paired with one device. So, the **5 devices require 5 device-selection bits**. The largest devices are the ROM and RAM devices with 20 address lines. Since the microprocessor has 24 address lines, **4 of them remain available for the selection**. Therefore, **the linear-decoding technique can no longer be used**.

***The block-decoding technique should be used (P3 should still be added). There should be as few blocks as possible.***

10. Which address lines are required to select the devices? Write down the devices associated with each of these address lines.

    To connect 5 devices using as few blocks as possible, we need 3 device-selection bits so that the space memory can be divided into 8 blocks. These bits have to be the most significant: *A23*, *A22* and *A21*.

| A23 | A22 | A21 | Device |
|---|---|---|---|
| 0 | 0 | 0 | ROM |
| 0 | 0 | 1 | RAM |
| 0 | 1 | 0 | P1 |
| 0 | 1 | 1 | P2 |
| 1 | 0 | 0 | P3 |

← The ROM should be located in the lowest part of the memory space.

11. Obtain an expression for each output of the new address decoder.

- $CS_{ROM} = AS.\overline{A23}.\overline{A22}.\overline{A21}$
- $CS_{RAM} = AS.\overline{A23}.\overline{A22}.A21$
- $CS_{P1} = AS.\overline{A23}.A22.\overline{A21}$
- $CS_{P2} = AS.\overline{A23}.A22.A21$
- $CS_{P3} = AS.A23.\overline{A22}.\overline{A21}$

12. Draw the new memory map. The lowest and highest addresses of each device should appear on the map.

    For each device, we have to determine the most significant address bits (the device-selection bits) and the least significant address bits (the address of the selected device) of the microprocessor. Any unused address bits will be set to 0.

    Take the ROM for instance:
    - *A21, A22* and *A23* are set to 0 to select the ROM device.
    - *A20* is set to 0 because it is unused.
    - To obtain its lowest address, its 20 address bits are set to 0.
    - To obtain its highest address, its 20 address bits are set to 1.

Here is what the memory map should look like:

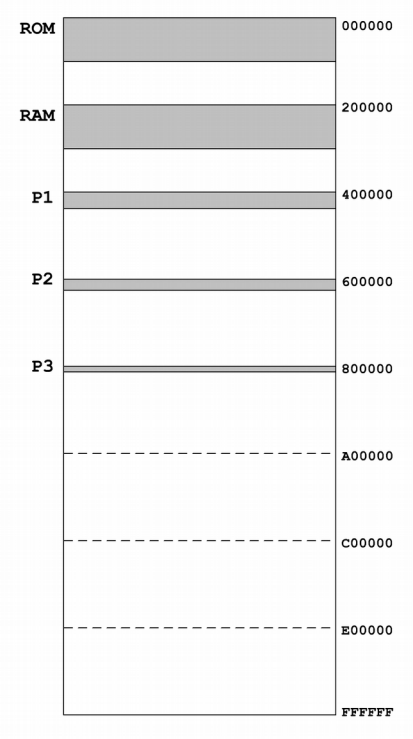
ROM (lowest): **0000 0000 0000 0000 0000 0000**$_2$ = $000000_{16}$
ROM (highest): **0000 1111 1111 1111 1111 1111**$_2$ = $0FFFFF_{16}$

RAM (lowest): **0010 0000 0000 0000 0000 0000**$_2$ = $200000_{16}$
RAM (highest): **0010 1111 1111 1111 1111 1111**$_2$ = $2FFFFF_{16}$

P1 (lowest): **0100** 0000 0000 **0000 0000 0000**$_2$ = $400000_{16}$
P1 (highest): **0100** 0000 0001 **1111 1111 1111**$_2$ = $401FFF_{16}$

P2 (lowest): **0110** 0000 0000 **0000 0000 0000**$_2$ = $600000_{16}$
P2 (highest): **0110** 0000 0000 **1111 1111 1111**$_2$ = $600FFF_{16}$

P3 (lowest): **1000** 0000 0000 **0000 0000 0000**$_2$ = $800000_{16}$
P3 (highest): **1000** 0000 0000 **0111 1111 1111**$_2$ = $8007FF_{16}$




13. Work out the number of redundant images for each device.

The number of redundant images is the number of combinations that can be made with the unused address bits of the microprocessor. These bits are used neither for the selection nor for the address of a device.

*Unused bits = 24 address bits – 3 device-selection bits – Address bits of a device*

- **ROM** : 24 – 3 – 20 = 1 unused bit → $\mathbf{2^1 = 2}$
- **RAM** : 24 – 3 – 20 = 1 unused bit → $\mathbf{2^1 = 2}$
- **P1** : 24 – 3 – 13 = 8 unused bits → $\mathbf{2^8 = 256}$
- **P2** : 24 – 3 – 12 = 9 unused bits → $\mathbf{2^9 = 512}$
- **P3** : 24 – 3 – 11 = 10 unused bits → $\mathbf{2^{10} = 1{,}024}$

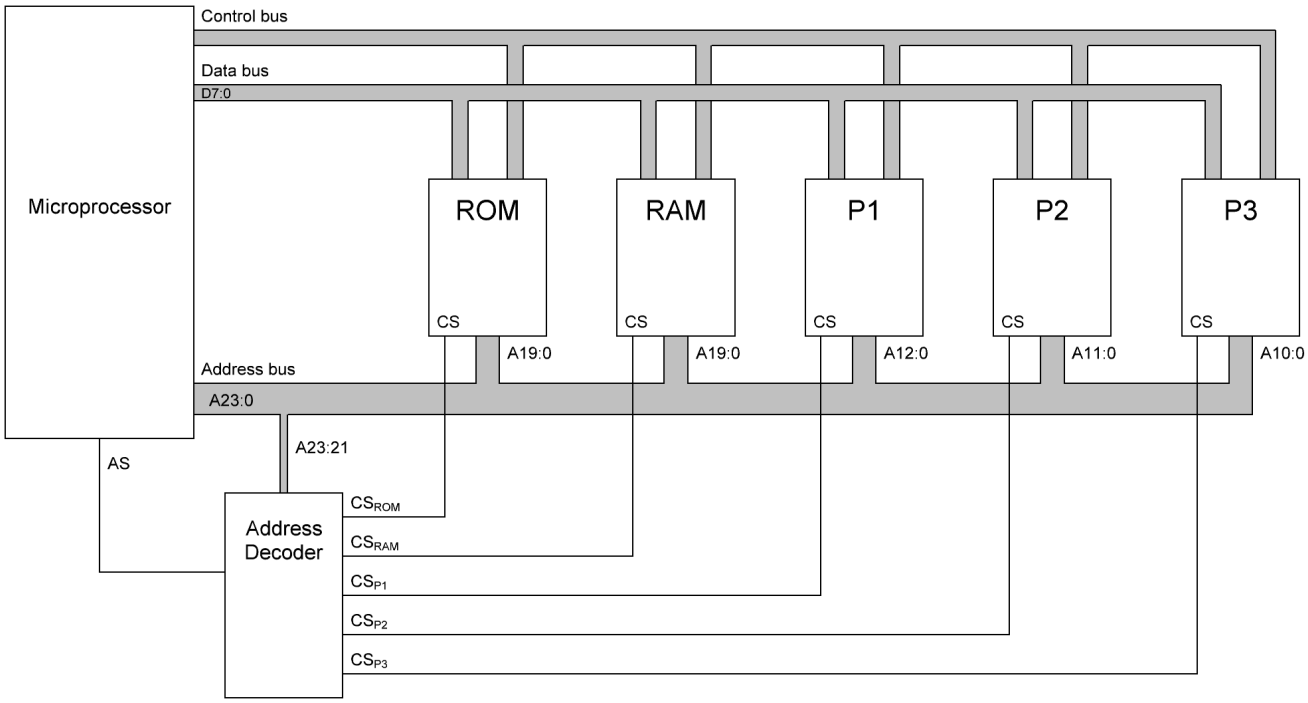
14. Using the 24-bit addresses of the microprocessor, find two different ways to select the address $1F2_{16}$ of each device.

To select the address $1F2_{16}$ of a device, we have to activate the device in question and put the value $1F2_{16}$ on its address bus. If some address bits are unused, they can be either 0 or 1 indifferently. Therefore, there is more than one possibility to select an address of a device.

The following table shows how we obtain two different addresses according to the changing value (0 or 1) of the unused bit *A20*.

| | **A20 = 0** | **A20 = 1** |
|---|---|---|
| **RAM** | $0001F2_{16}$ | $1001F2_{16}$ |
| **ROM** | $2001F2_{16}$ | $3001F2_{16}$ |
| **P1** | $4001F2_{16}$ | $5001F2_{16}$ |
| **P2** | $6001F2_{16}$ | $7001F2_{16}$ |
| **P3** | $8001F2_{16}$ | $9001F2_{16}$ |

15. Modify the circuit diagram.




***The 1-MiB RAM device should be replaced by a 4-MiB RAM device.***

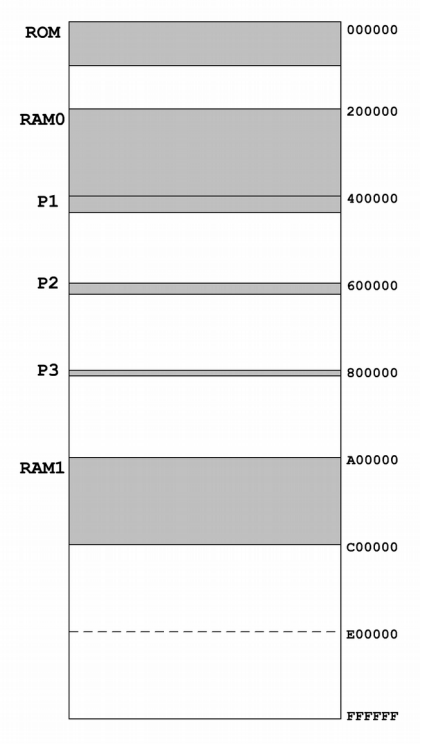
16. Can the block-decoding technique still be used? Give reasons for your views.

    With the block address decoding, one block (a combination of device-selection bits) is paired with one device. So, the **5 devices require 3 device-selection bits (8 blocks)**. The largest device is now the RAM with 22 address lines. Since the microprocessor has 24 address lines, **2 of them remain available for the selection**. Therefore, **the block-decoding technique can no longer be used.**

***In order to connect the 4-MiB RAM device, a derivative of the block-decoding technique should be used. Keeping the previous memory configuration of 8 blocks of 2 MiB each, you should use 2 blocks for the RAM device. The first block should contain the lower 2 MiB of the RAM device (labelled RAM0) and should be the same as the block used for the previous RAM device. The second block should contain the upper 2 MiB of the RAM device (labelled RAM1) and should be the block located after P3.***

17. Draw the new memory map. The lowest and highest addresses of each device should appear on the map.

ROM : $000000_{16}$ - $0FFFFF_{16}$
RAM0 : $200000_{16}$ - $3FFFFF_{16}$
P1 : $400000_{16}$ - $401FFF_{16}$
P2 : $600000_{16}$ - $600FFF_{16}$
P3 : $800000_{16}$ - $8007FF_{16}$
RAM1 : $A00000_{16}$ - $BFFFFF_{16}$



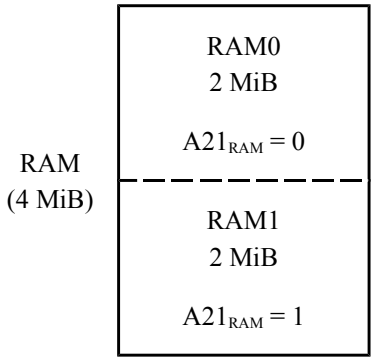

18. The *CS* input of the RAM device should be active for two blocks. Write down its new expression.

$CS_{RAM} = AS.(\overline{A23}.\overline{A22}.A21 + A23.\overline{A22}.A21)$
$CS_{RAM} = AS.\overline{A22}.A21.(\overline{A23} + A23)$
$CS_{RAM} = AS.\overline{A22}.A21$

19. Is this new expression of the *CS* sufficient to decode the RAM device properly?

The 4-MiB RAM is considered as two devices of 2 MiB each (RAM0 and RAM1). Therefore, the value of *A21* will indicate whether an address is in the lower or higher part of the memory. Thus, if $A21_{RAM} = 0$, an address is in the lower part (RAM0) and if $A21_{RAM} = 1$, an address is in the higher part (RAM1).

From now on, we will use these notations:
- *A21* is the address bit number 21 of the microprocessor.
- $A21_{RAM}$ is the address bit number 21 of the RAM.




When the block-decoding technique is used, the least significant bits of the address bus of the microprocessor are connected to the address buses of the devices. Therefore, the bits *A21:0* of the microprocessor are connected to the bits *A21:0* of the RAM and so $A21_{RAM} = A21$.

According to the expression of $CS_{RAM}$, we can see that the RAM device is active only when $A21 = 1$. Since $A21_{RAM} = A21$, we can conclude that the RAM0 is never selected.

**Therefore, this new expression in not sufficient to decode the RAM device properly.**

20. Find a solution to solve this problem.

To solve this problem, $A21_{RAM}$ should be set according to the selected device. When the RAM device is not active, the value on its address bus is ignored. See the table below:

| A23 | A22 | A21 | Selected device | $A21_{RAM}$ | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | ROM | Φ | |
| **0** | 0 | 1 | RAM | **0** | → RAM0 |
| 0 | 1 | 0 | P1 | Φ | |
| 0 | 1 | 1 | P2 | Φ | |
| 1 | 0 | 0 | P3 | Φ | |
| **1** | 0 | 1 | RAM | **1** | → RAM1 |

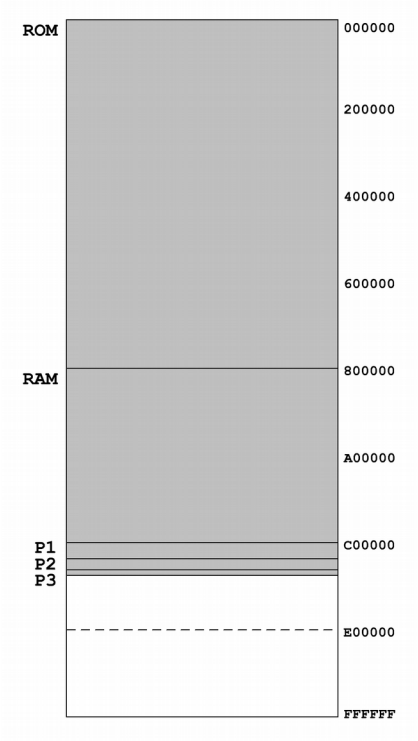
According to this table, we can obviously conclude that $A21_{RAM}$ should be connected to *A23* of the microprocessor (and not *A21* as before).

The expressions of the *CS* inputs do not change.

***The 1-MiB ROM device should be replaced by an 8-MiB ROM device. In order to connect this ROM device, it is proposed to use a new decoding technique: no empty space should be found between the memory spaces occupied by all the devices. The ROM should be located in the lowest part of the memory space, followed by the RAM (4 MiB), P1, P2 and P3.***

21. Draw the new memory map. The lowest and highest addresses of each device should appear on the map.

ROM : $000000_{16}$ - $7FFFFF_{16}$
RAM : $800000_{16}$ - $BFFFFF_{16}$
P1 : $C00000_{16}$ - $C01FFF_{16}$
P2 : $C02000_{16}$ - $C02FFF_{16}$
P3 : $C03000_{16}$ - $C037FF_{16}$




22. Obtain an expression for each output of the new address decoder.

We need to find which address bits are significant enough to select the right device. To begin with, let us convert the addresses into their binary notation as shown below. Each *phi* character represents an address line and can be replaced by either 0 or 1.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ROM | : | **0** | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | (23 address lines) |
| RAM | : | **1** | **0** | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | (22 address lines) |
| P1 | : | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | (13 address lines) |
| P2 | : | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **0** | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | (12 address lines) |
| P3 | : | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | Φ | (11 address lines) |
| | | **A 23** | **A 22** | | | | | | | | | | | **A 13** | **A 12** | | | | | | | | | | | | | | | | | | | |

For each device, the lowest address is obtained when all the *phi* characters are replaced by 0 and the highest address is obtained when all the crosses are replaced by 1.

The first significant bit that can be noticed is *A23*. When *A23* = 0, we should select the ROM and when *A23* = 1, we should select any one of the other devices. The bit *A23* can be used to differentiate the ROM from the other devices. We can begin to write down a first uncompleted expression for the outputs of the address decoder.

- $CS_{ROM} = AS.\overline{A23}$
- $CS_{RAM} = AS.A23$
- $CS_{P1} = AS.A23$
- $CS_{P2} = AS.A23$
- $CS_{P3} = AS.A23$

The second significant bit that can be noticed is *A22*. With the same line of reasoning, this bit can be used to differentiate the RAM from the other remaining devices (**P1**, **P2** and **P3**). The outputs of the address decoder are as follows:

- $CS_{ROM} = AS.\overline{A23}$
- $CS_{RAM} = AS.A23.\overline{A22}$
- $CS_{P1} = AS.A23.A22$
- $CS_{P2} = AS.A23.A22$
- $CS_{P3} = AS.A23.A22$

With the same line of reasoning, the last two significant bits are *A13* and *A12*:

- $CS_{ROM} = AS.\overline{A23}$
- $CS_{RAM} = AS.A23.\overline{A22}$
- $CS_{P1} = AS.A23.A22.\overline{A13}$
- $CS_{P2} = AS.A23.A22.A13.\overline{A12}$
- $CS_{P3} = AS.A23.A22.A13.A12$

It is noteworthy that the line numbers on these expressions match the number of address lines you find in each device.