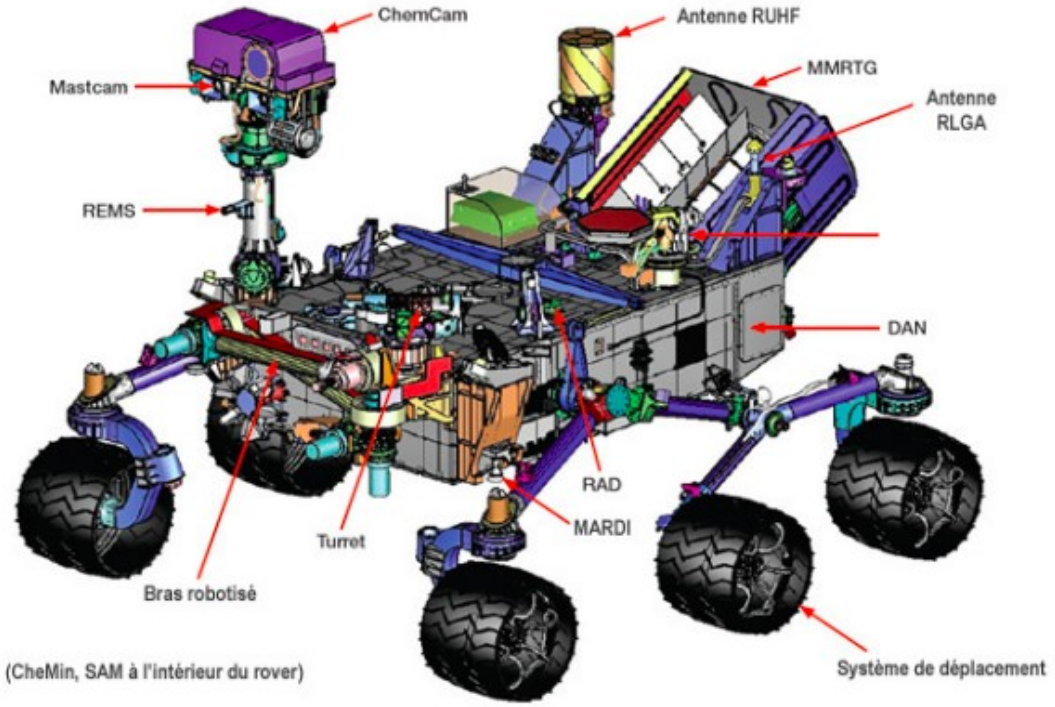
Programmation des composants embarqués et systèmes.

Plan mémoire pour la configuration d’un robot



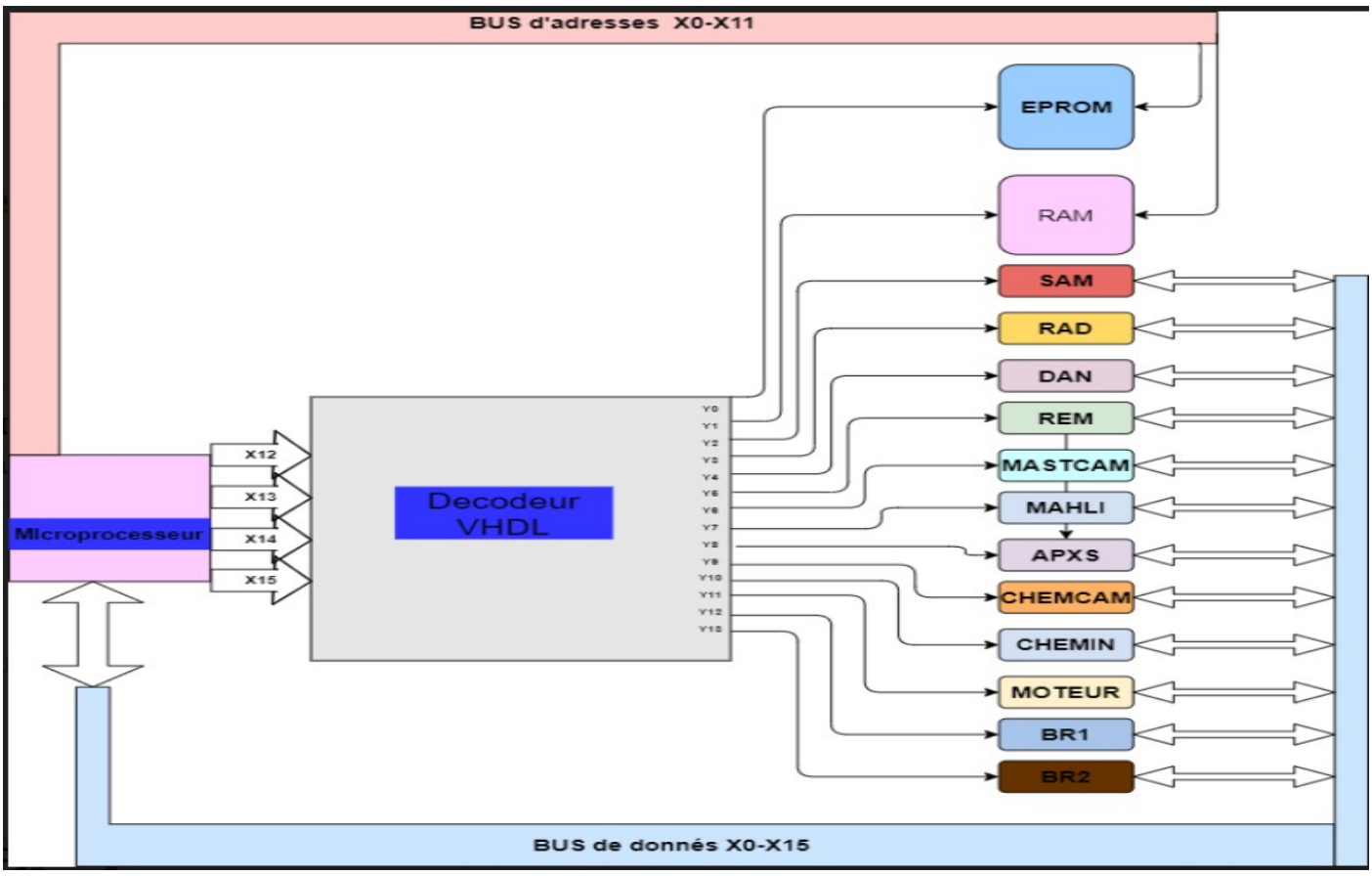
Réalisé par: Micipsa SADJI

16707683

Plan mémoire

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Adresses en hexadecimal
EPROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Ox000
EPROM	//	//	//	//	//	//	//	//	//	//	//	//	//	//	//	//	
EPROM	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0xFFFF
RAM	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0x1000
RAM	//	//	//	//	//	//	//	//	//	//	//	//	//	//	//	//	
RAM	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0X1000
SAM	0	0	1	0	X	X	X	X	X	X	X	X	X	X	X	x	0x2000
RAD	0	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	0x3000
DAN	0	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	0x4000
RED	0	1	0	1	X	X	X	X	X	X	X	X	X	X	X	x	0X5000
MaSTcAM	0	1	1	0	X	X	X	X	X	X	X	X	X	X	X	x	0x6000
Mahli	0	1	1	1	X	X	X	X	X	X	X	X	X	X	X	x	0x7000
APXS	1	0	0	0	X	X	X	X	X	X	X	X	X	X	X	x	0x8000
Chemcam	1	0	0	1	X	X	X	X	X	X	X	X	X	x	X	X	0x9000
CheMin	1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	0xA000
Moteurs	1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	x	0xB000
Bras robotisé n°1	1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	x	0XC000
Bras robotisé n°2	1	1	0	1	X	X	X	X	X	X	X	X	X	X	x	X	0xD000

Le schéma :



Code VHDL(testé et simulé sur modelSIM)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity decode_4to16_top is
 Port ( X : in STD_LOGIC_VECTOR (3 downto 0); -- 4-bit input
 Y : out STD_LOGIC_VECTOR (15 downto 0); -- 16-bit output
 EN : in STD_LOGIC); -- enable input
end decode_4to16_top;
architecture Behavioral of decode_4to16_top is
begin
process (X, EN)
begin
 Y <= "1111111111111111"; -- default output value
 if (EN = '1') then -- active high enable pin
 case X is
   when "0000" => Y <= "0111111111111111";
   when "0001" => Y <= "1011111111111111";
   when "0010" => Y <= "1101111111111111";
   when "0011" => Y <= "1110111111111111";
   when "0100" => Y <= "1111011111111111";
   when "0101" => Y <= "1111101111111111";
   when "0110" => Y <= "1111110111111111";
   when "0111" => Y <= "1111111011111111";
   when "1000" => Y <= "1111111101111111";
   when "1001" => Y <= "1111111110111111";
   when "1010" => Y <="1111111111011111";
   when "1011" => Y <="1111111111101111";
   when "1100" => Y <="1111111111110111";
   when "1101" => Y <="1111111111111011" ;
```

```
  when "1110" => Y <="1111111111111101";
  when "1111" => Y <="1111111111111110";
  when others => Y <= "1111111111111111";
 end case;
 end if;
end process;
end Behavioral;
```

Captures d’écran de la simulation sur modelSim :