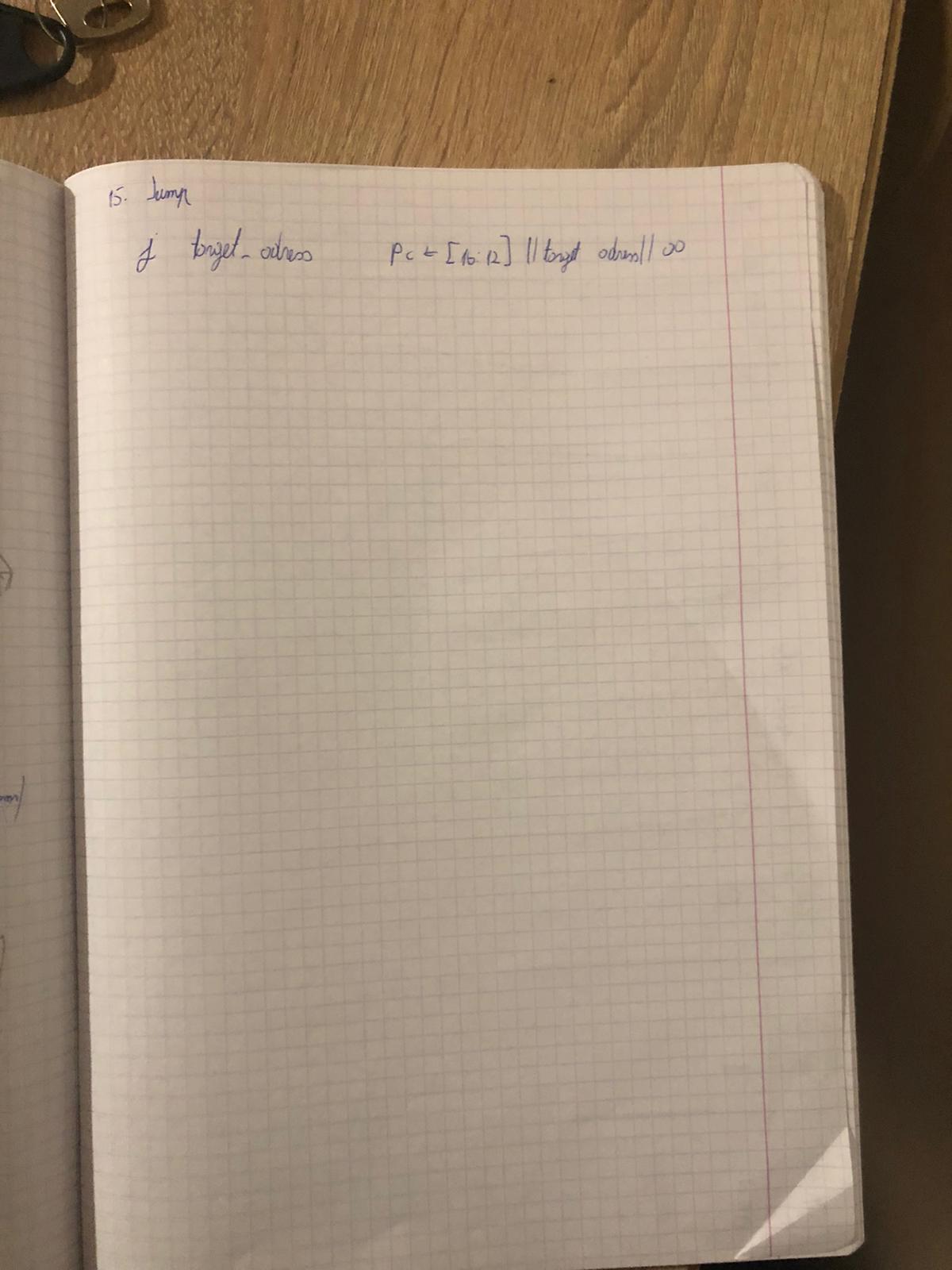
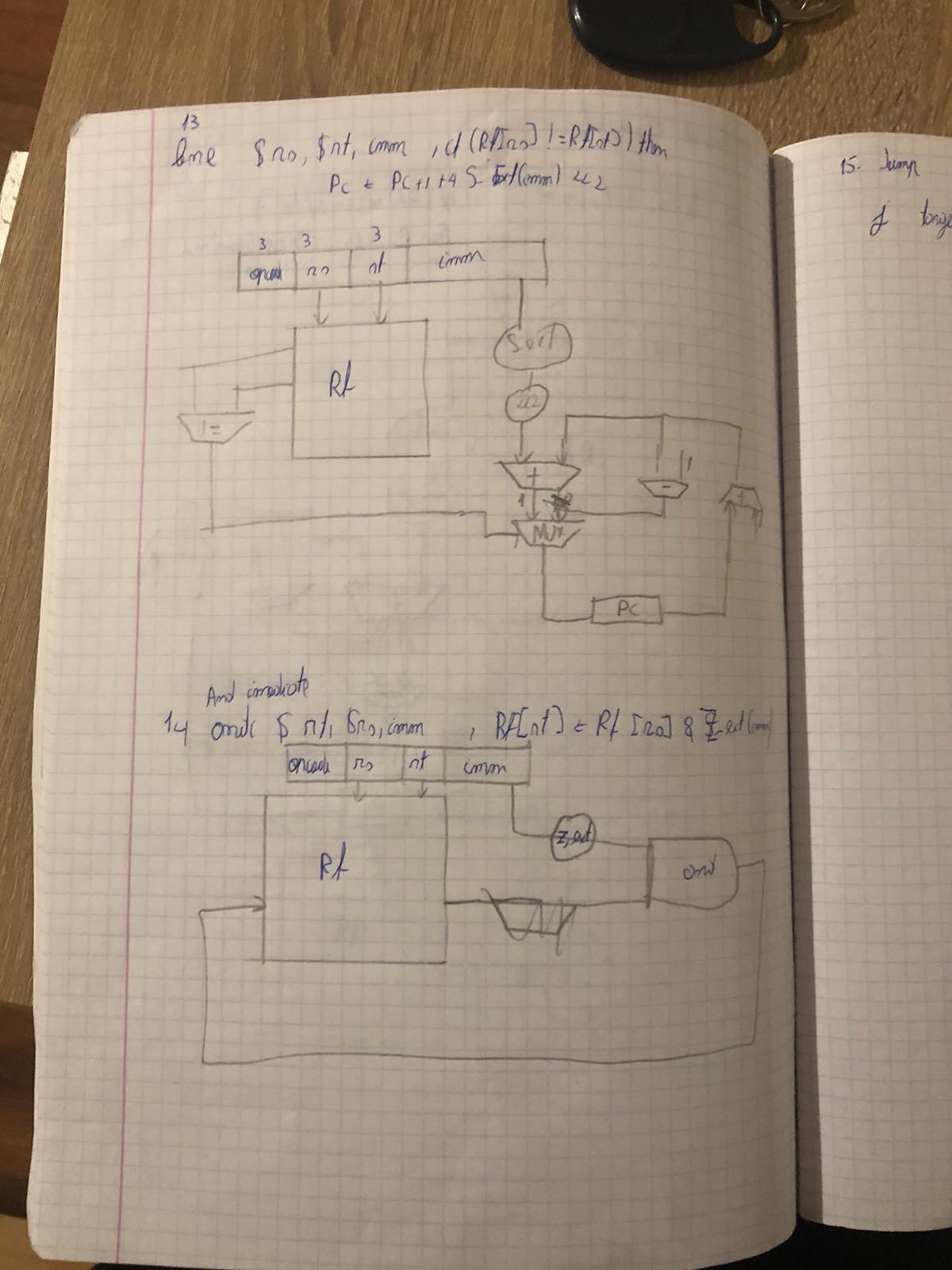
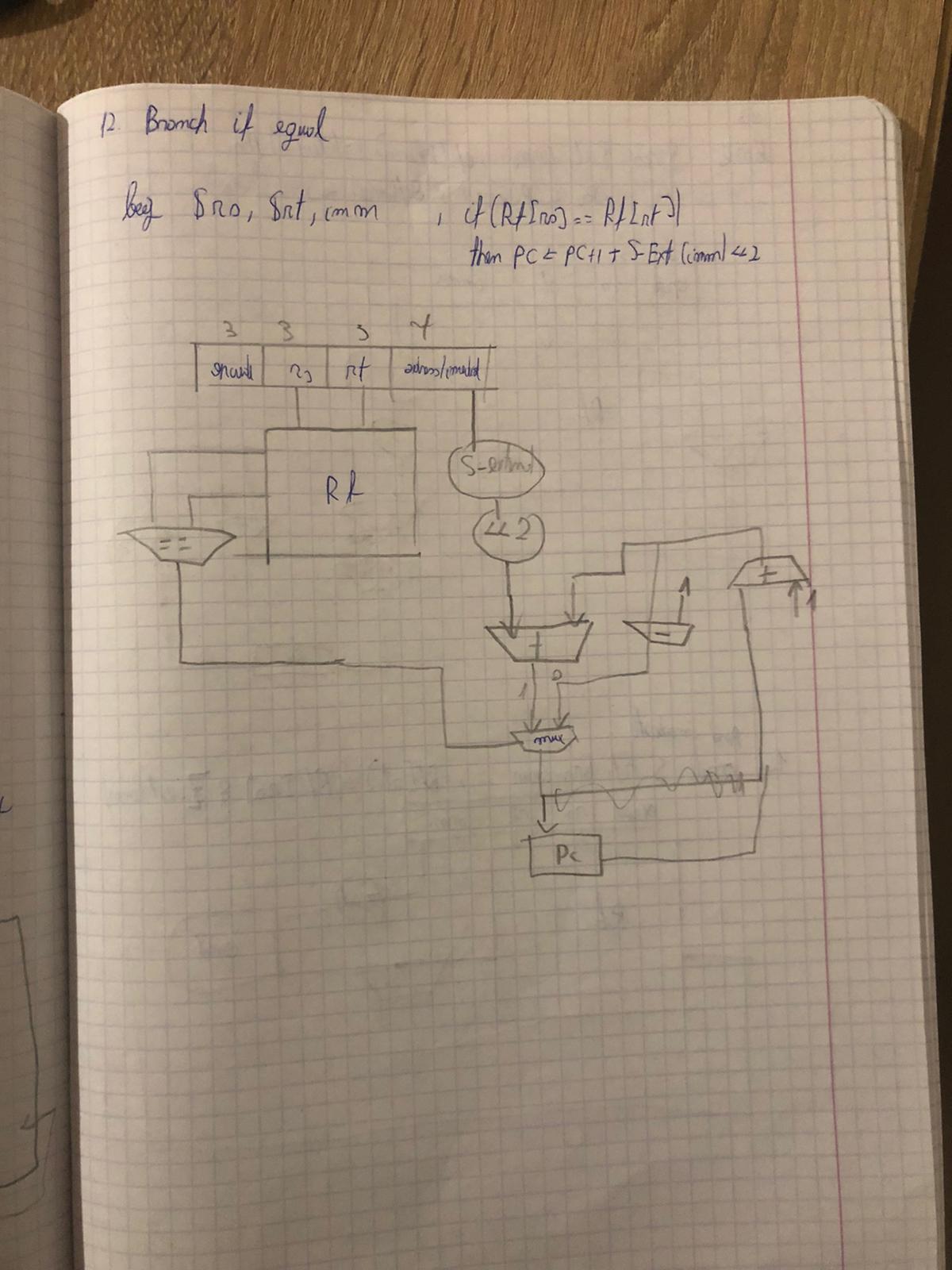
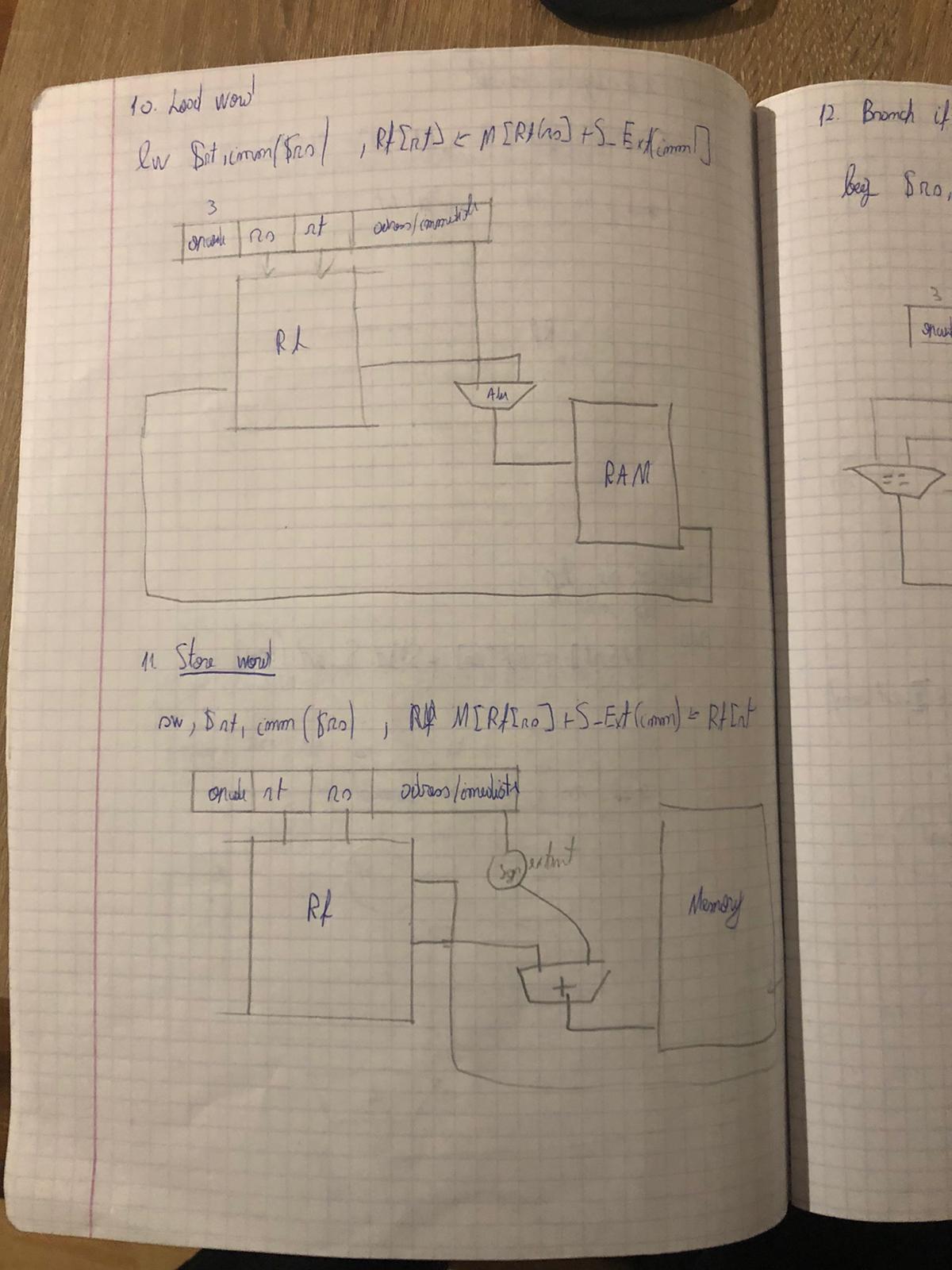
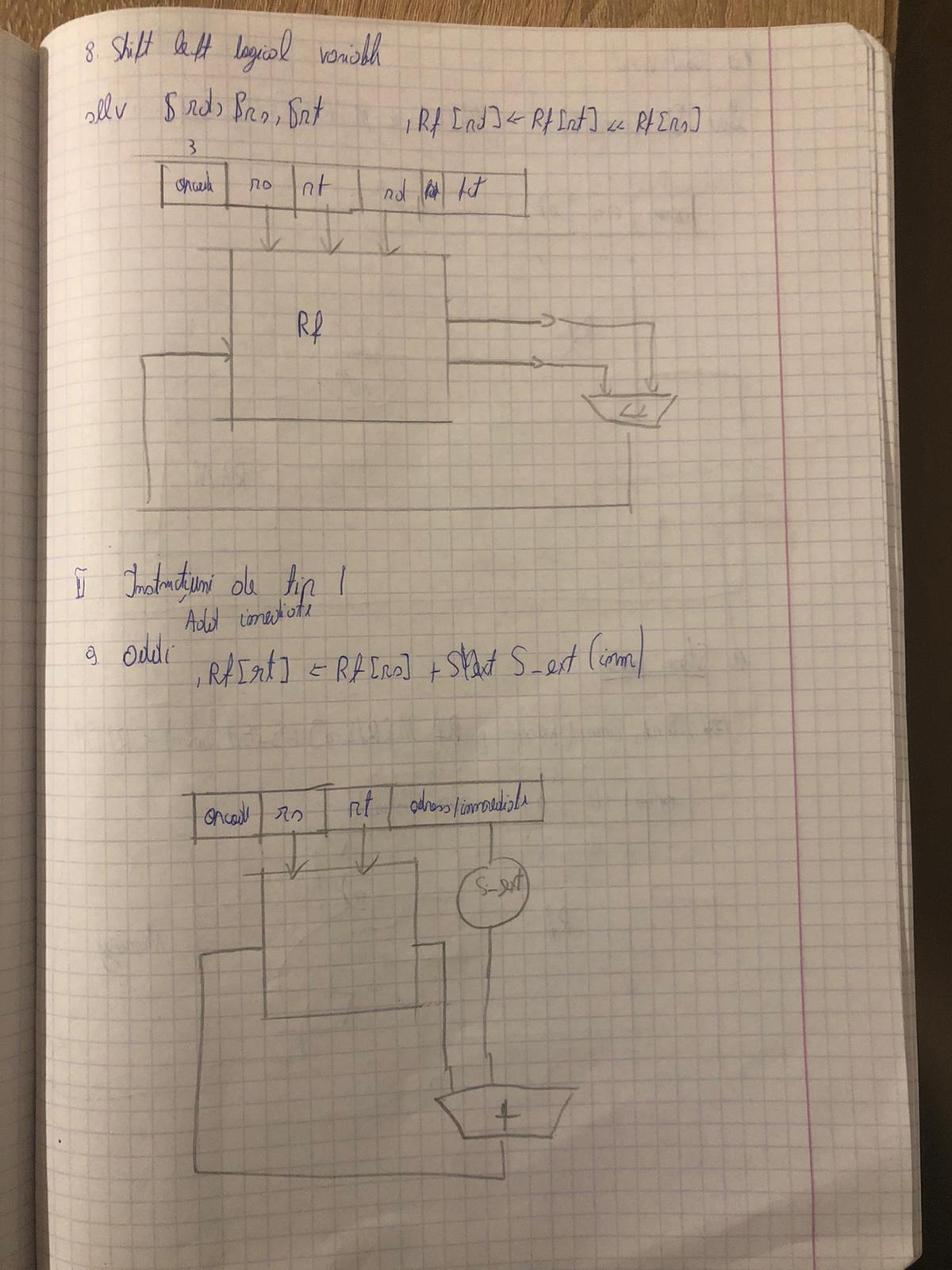
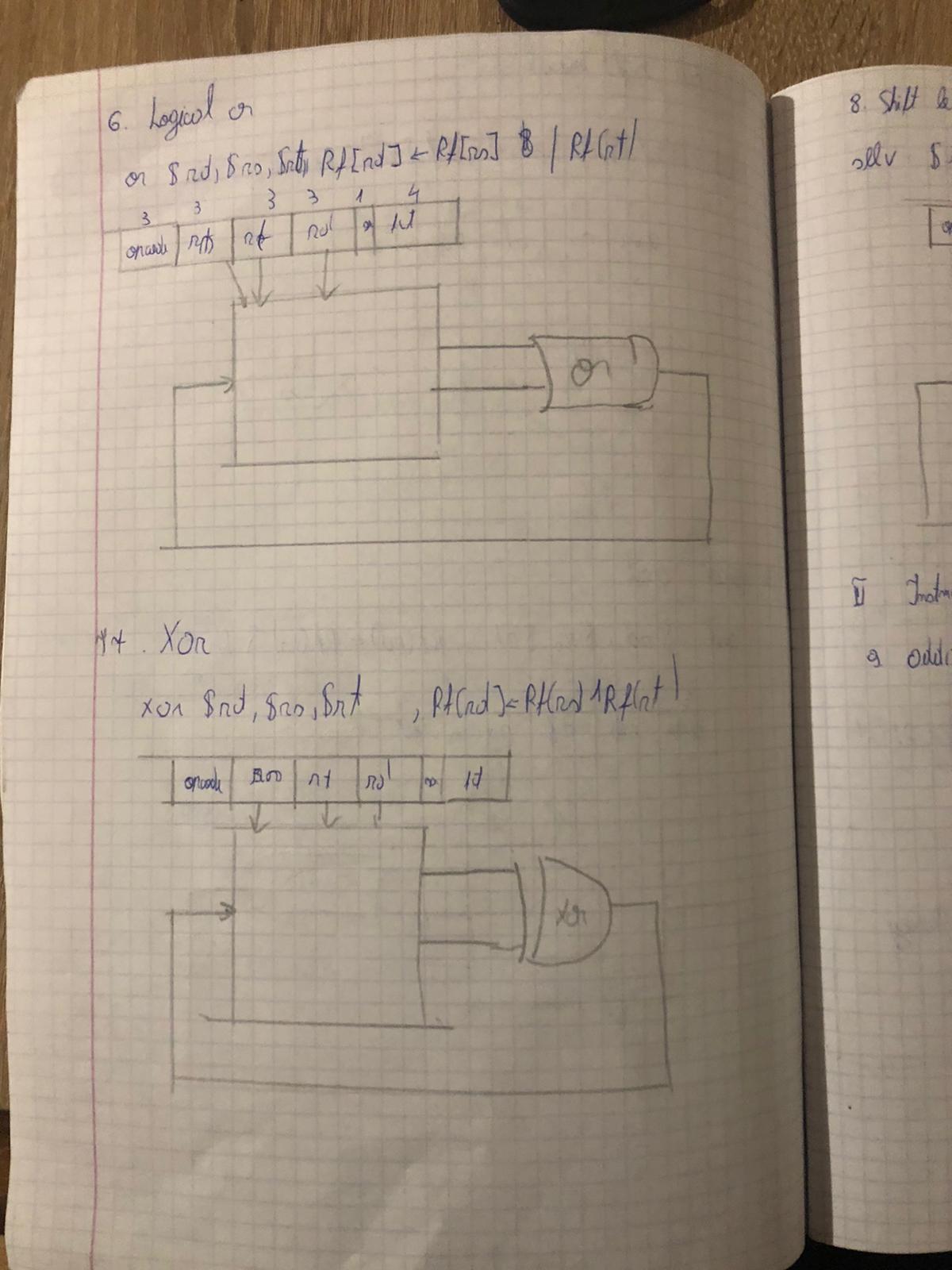
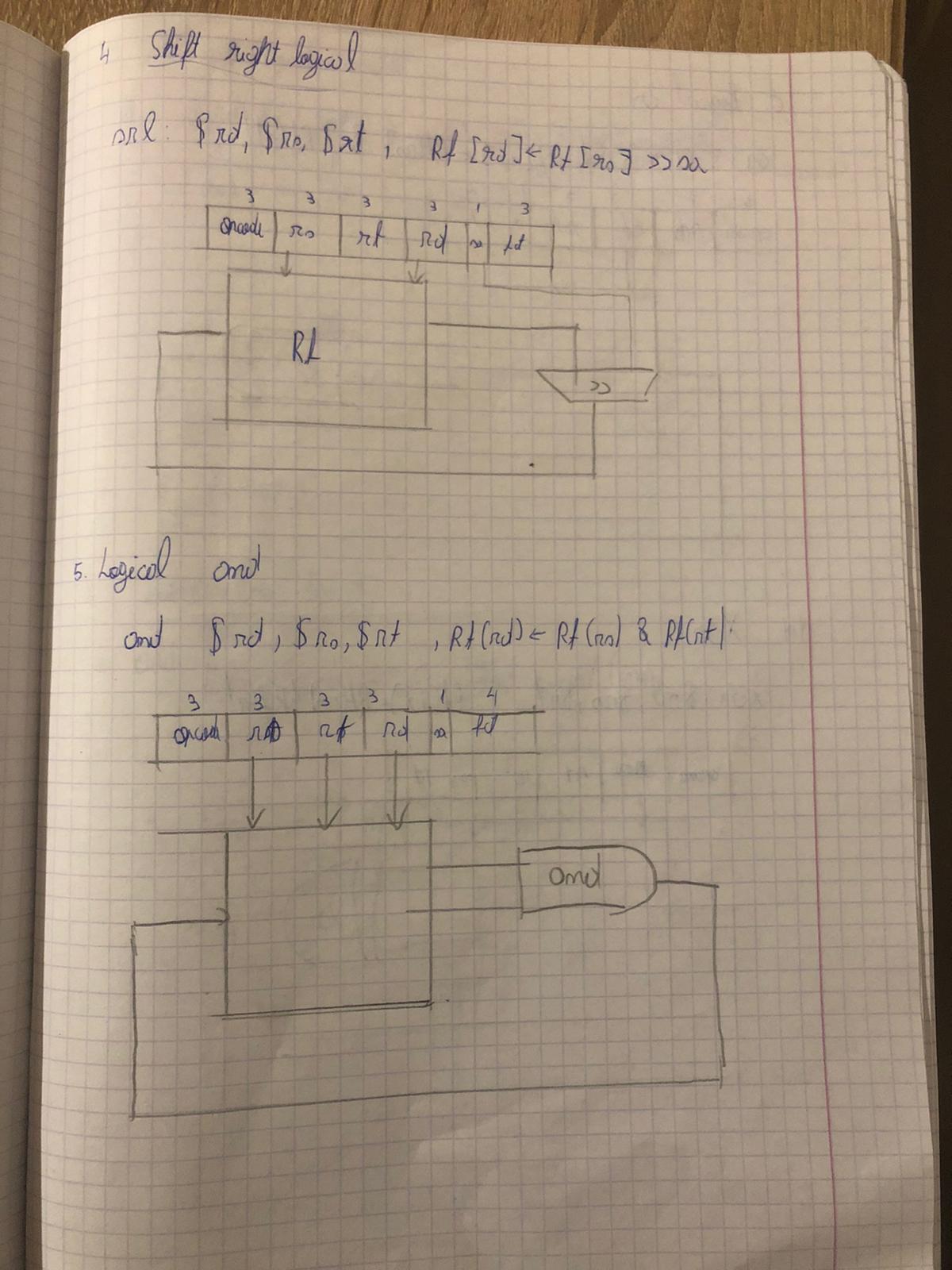
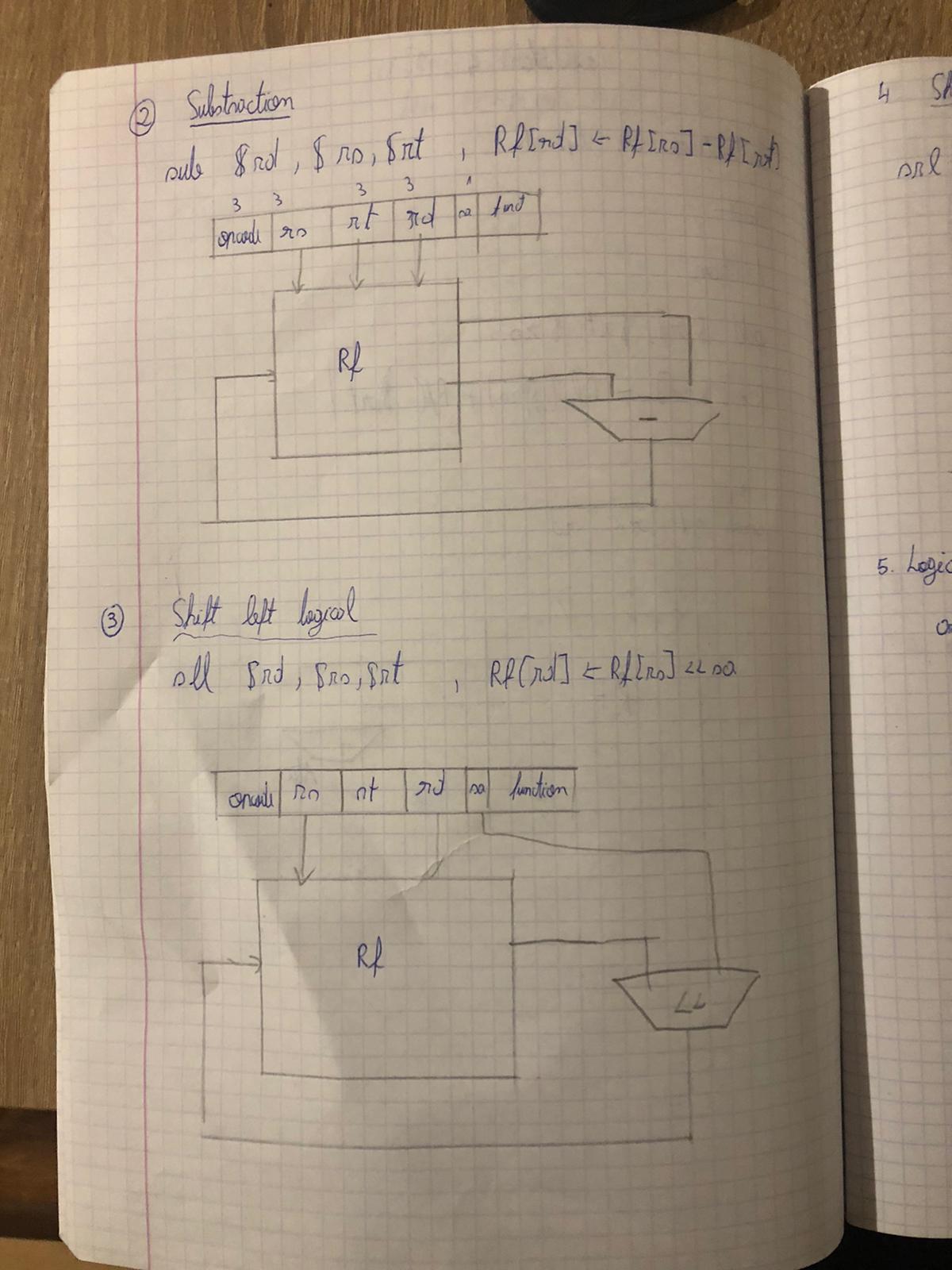
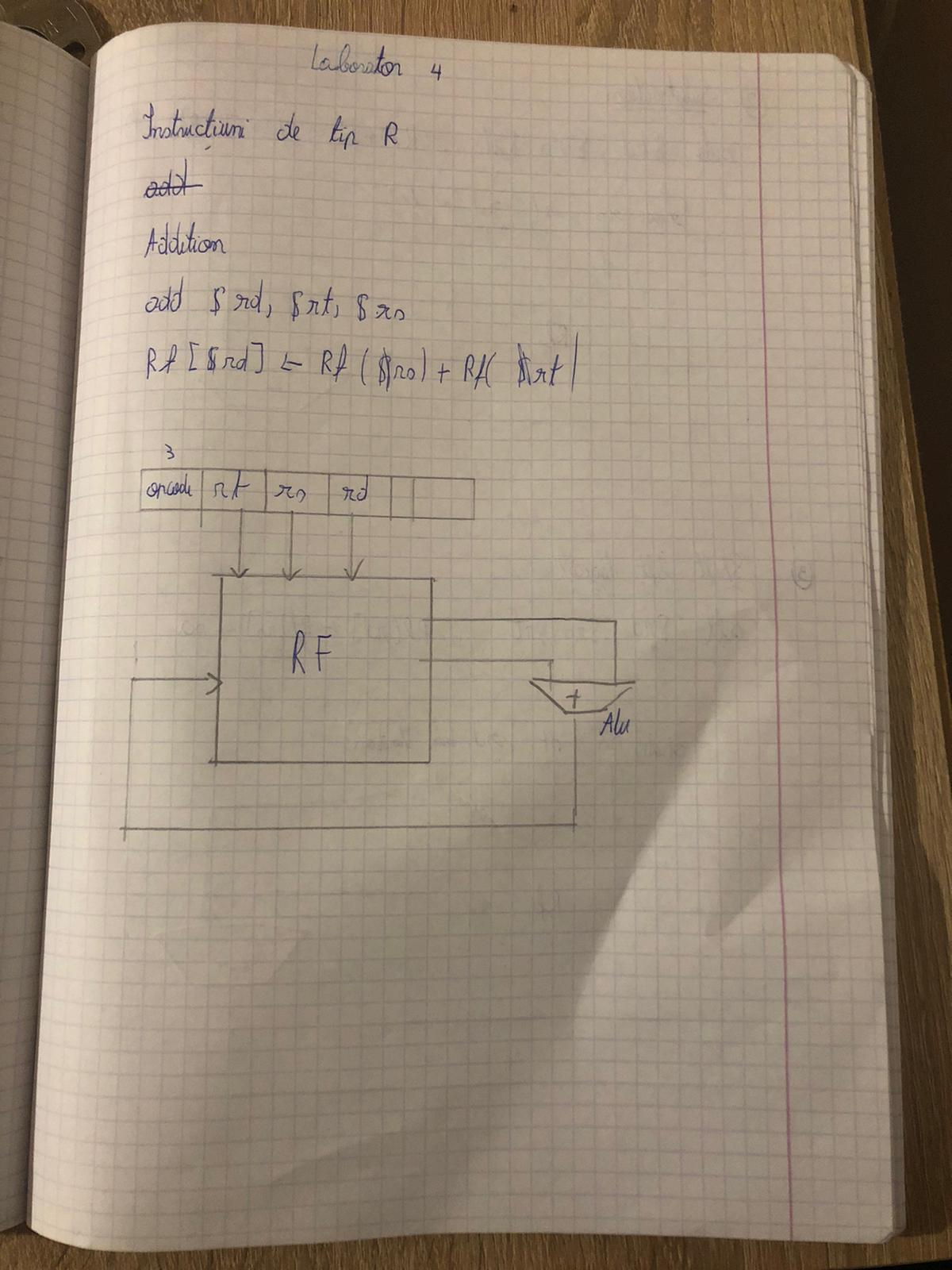
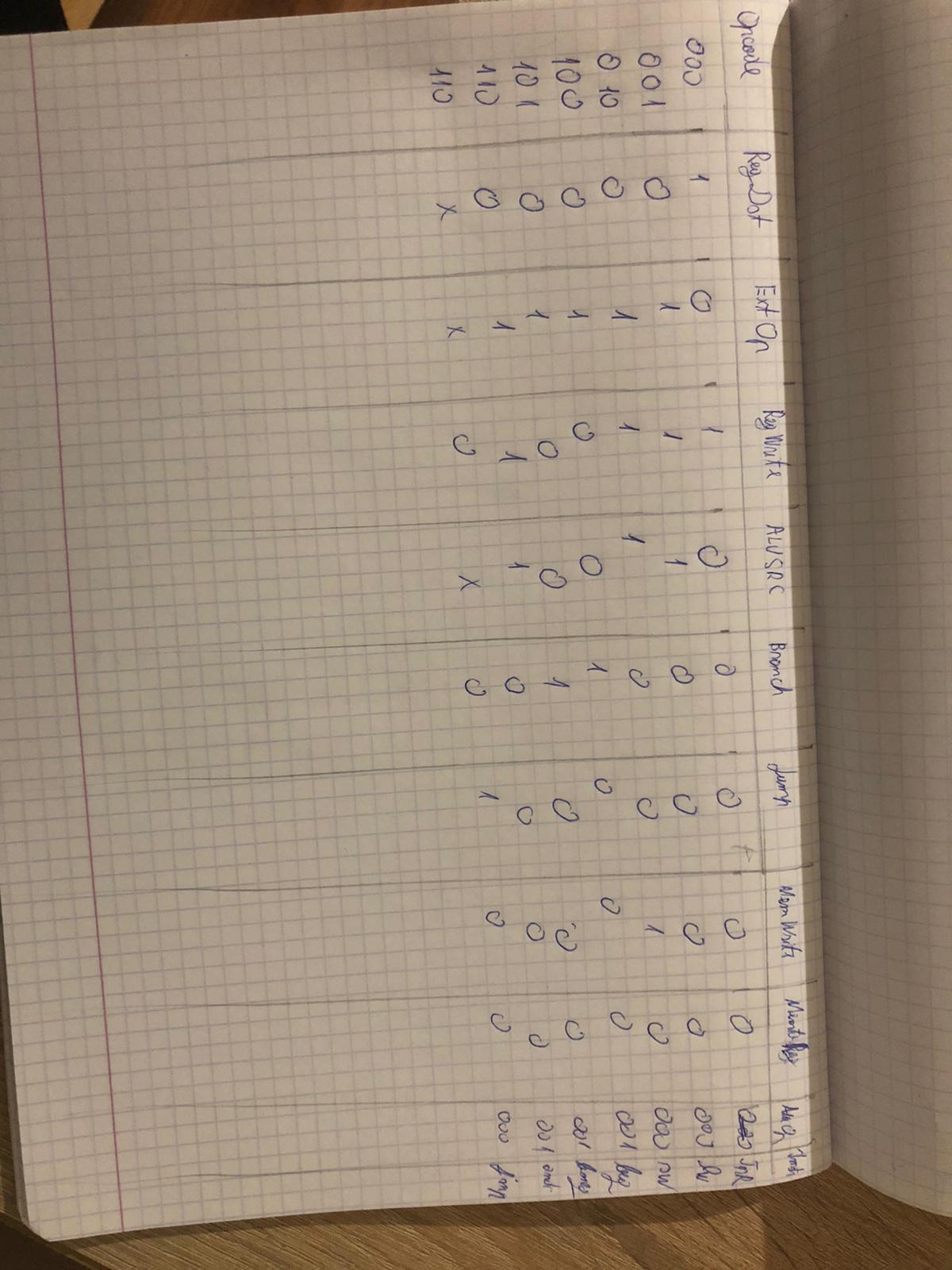
**Raport Mips 16**

Blendea Adrian

Operatii laborator 4 :



Tabel semnale



Trasarea programului

-- (6+2)\*2 - 4/2

b"001\_000\_001\_0000110", --addi $1 $0 6 adaugam 6 la adresa 1,

b"001\_000\_010\_0000010", --addi $2 $0 2 adaugam 2 la adresa 2

b"000\_001\_010\_011\_0\_000", --add $3 $1 $2 punem la adresa 3 , 6+2=8

b"000\_011\_000\_100\_1\_010", --sll $4 $3 1 punem la adresa 4 , 8\*2=16

b"001\_000\_101\_0000100" , --addi $5, $0 4, punem la adresa 5 , 4

b"000\_101\_000\_110\_1\_011", --srl $6 $5 1 punem la adresa , 4/2=2

b"000\_100\_110\_000\_0\_001", --sub $0 $4 $6 rezultat final

Observarea rezultatelor pe afisor:

process(sw(7 downto 5))

begin

case sw(7 downto 5) is

when "000" => afisare <= instruction;

when "001" => afisare <= instruction\_next;

when "010" => afisare <= rd1;

when "011" => afisare <= rd2;

when "100" => afisare <= ExtImm;

when "101" => afisare <= ALURes2;

when "110" => afisare <= MemData;

when "111" => afisare<= wd;

end case;

end process;

1. Instruction = 2086 Rd1 =0 Rd2=0 ExtImm=6 AluRes=6

Prima instructiune adauga la adresa 1 valoarea 6. Rd1 si Rd2 sunt 0 pentru ca nu se citeste nimic .

1. Instruction = 2102 Rd1 =0 Rd2=0 ExtImm=2 AluRes=2

A 2 instructiune adauga la adresa 2 valoarea 2. Rd1 si Rd2 sunt 0 pentru ca nu se citeste nimic .

1. Instruction = 0530 Rd 1=6 Rd2=2 ExtImm nu este relevant (instructiune de tip R) AluRes=8

Instructiune adauga la adresa 3 suma valorile de la adresele 1 si 2. Rd1 =6

Rd2 =2 asadar la adresa 3 se va adauga valoarea 8.

4.Instruction = 0C41 Rd 1 =8 Rd2 nu este folosit ExtImm irelevant (tip R) sa=1 .

Instructiunea pune la adresa 4 a registrului valoarea de la adresa 3 shiftata la stanga cu sa biti , adica shiftata cu 1 bit , rezultand o inmultire cu 2 => AluRes=2\*8=16 (10 pe afisor deoarece afisorul afiseaza in hexa)

5.Instruction = 2284 Rd1 =0 Rd2=0 ExtImm=6 AluRes=4

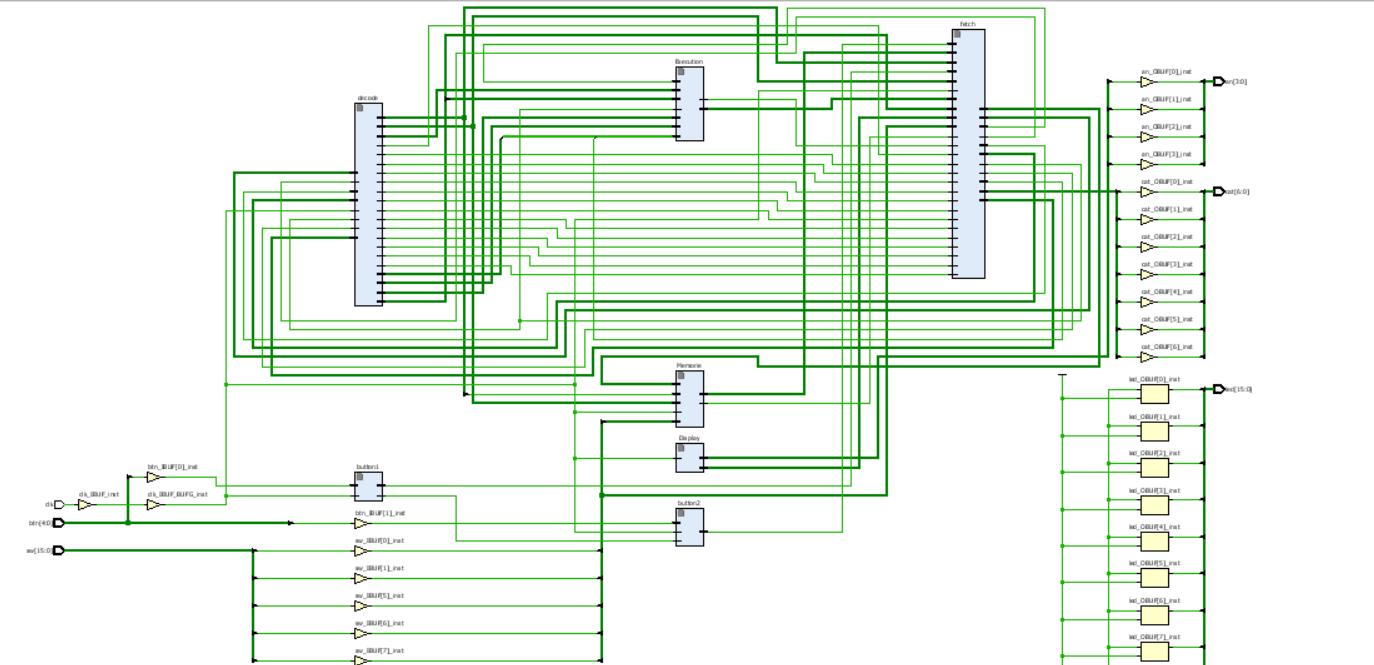
Prima instructiune adauga la adresa 5 valoarea 4.

6.Instruction = 146b Rd 1 =4 Rd2 nu este folosit ExtImm irelevant (tip R) sa=1 .

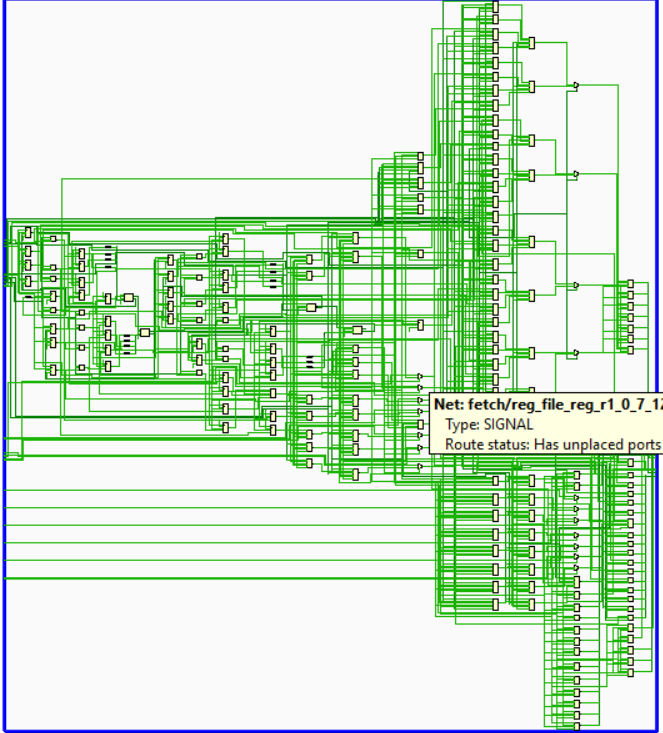
Instructiunea pune la adresa 6 a registrului valoarea de la adresa 5 shiftata la dreapta cu sa biti , adica shiftata cu 1 bit , rezultand o impartire cu 2 => AluRes=4/2=2 (10 pe afisor deoarece afisorul afiseaza in hexa)

7. Instruction=1301 Rd1= 10 Rd2=2 . Instructiunea pune la adresa 0 a registrului valoarea de la adresa 4 – valoarea de la adresa.AluRes = 10-2 (in hexa ) =E (14 in decimal adica rezultatul operatiei (6+2)\*2 - 4/2)

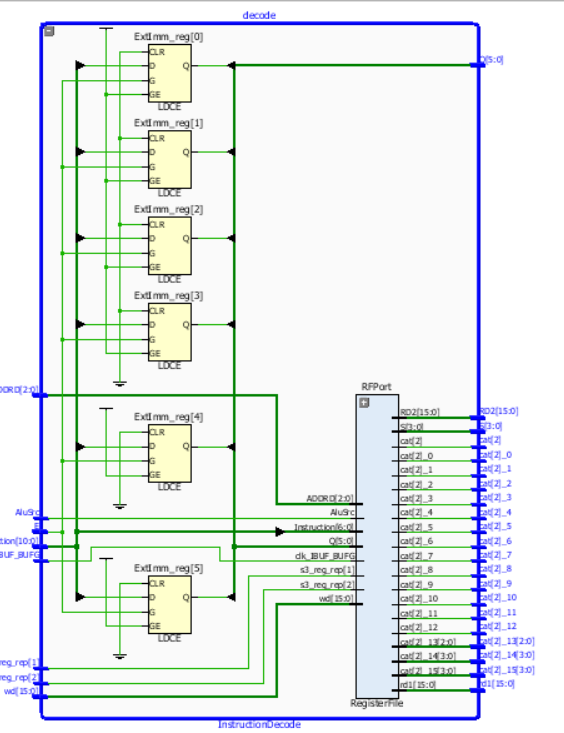
RTL SCHEMATIC:



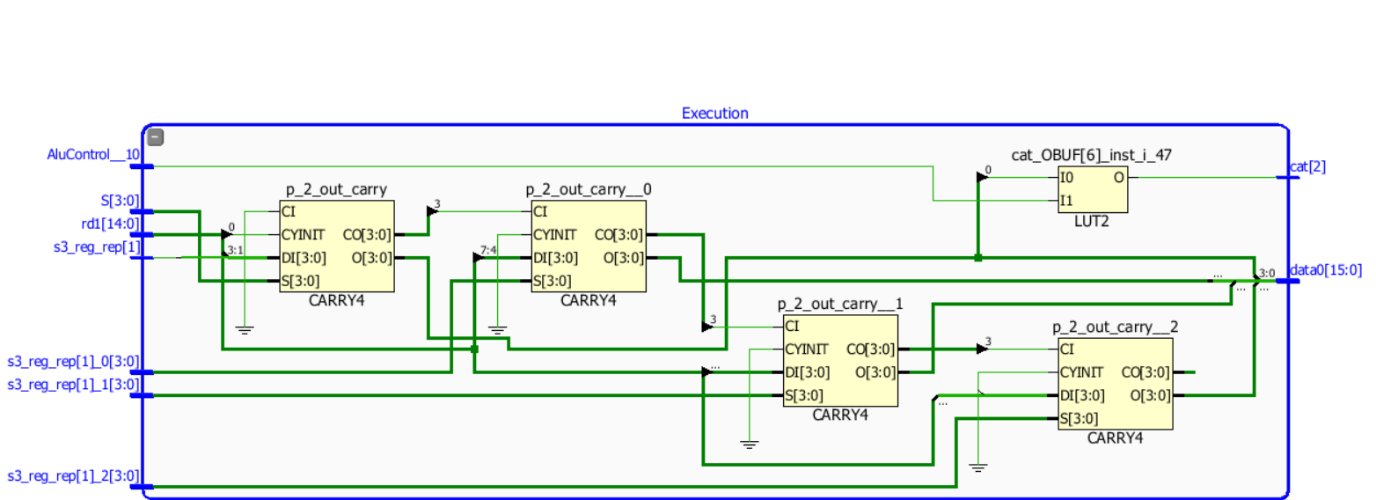
Calea de date:



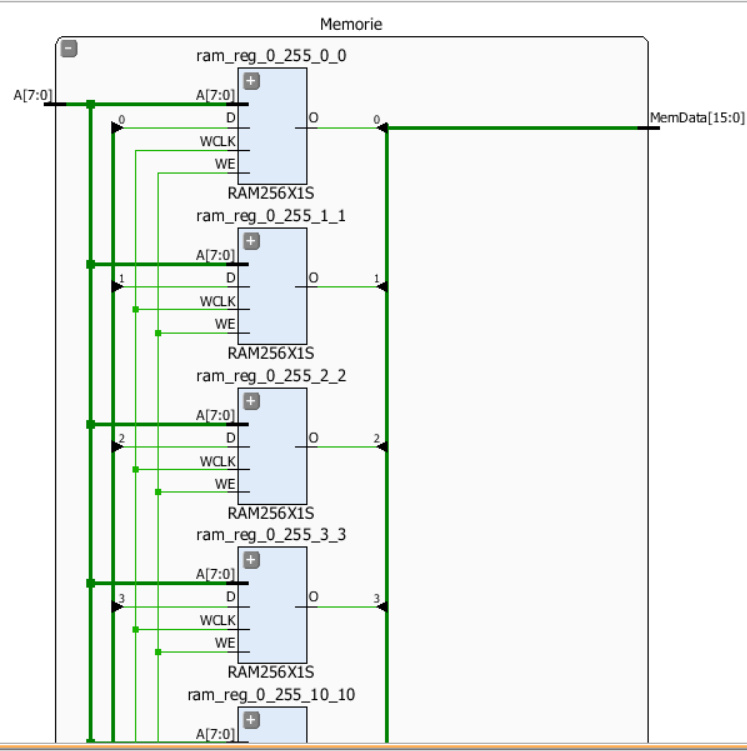
Instruction Decode:



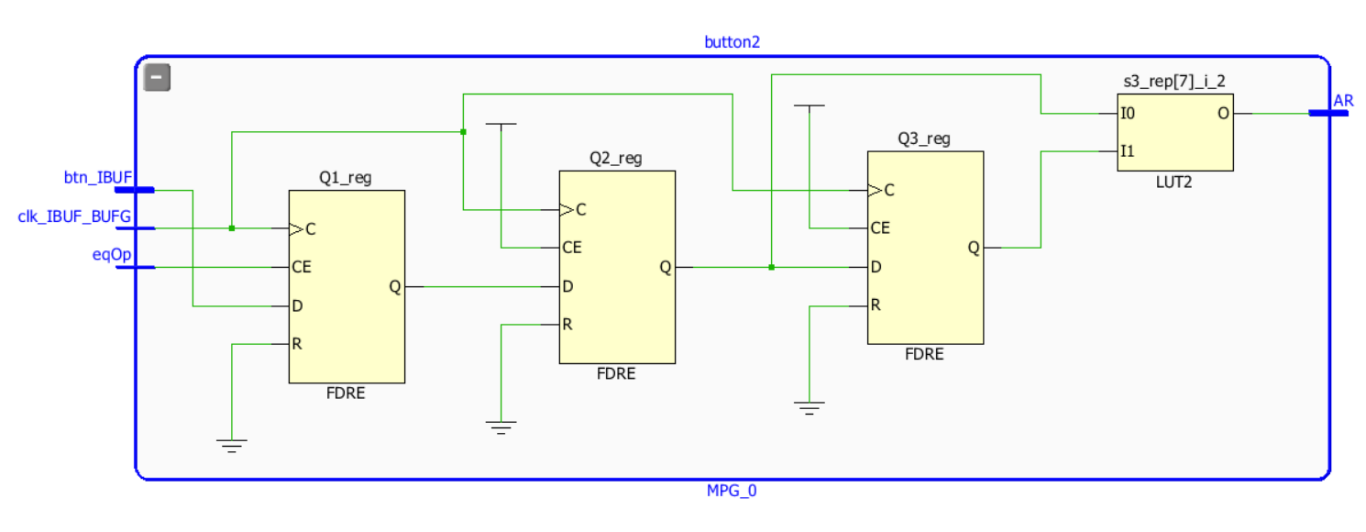
Unitate Executie:



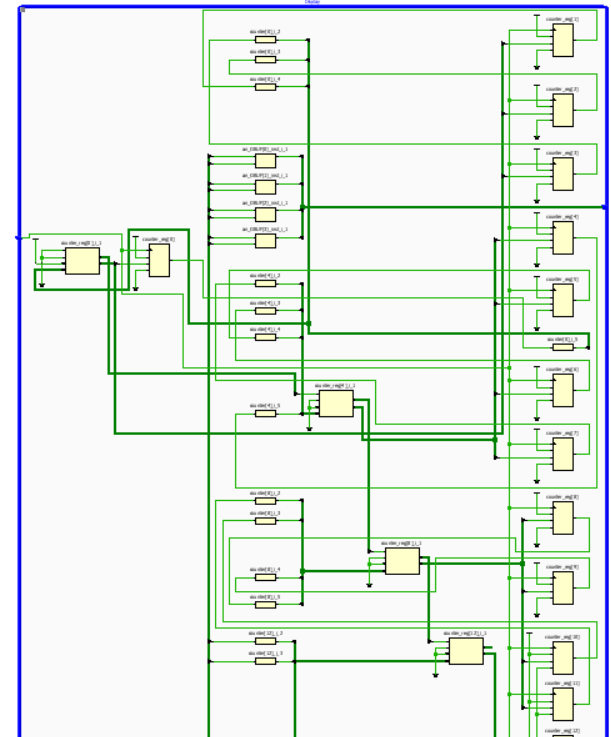
Unitate de memorie:



MPG:



SSD:



Programul a fost testat pe placa si este functional pentru cazul dat , functionarea altor operatii decat cele care sunt in exemplu nu a fost testata.