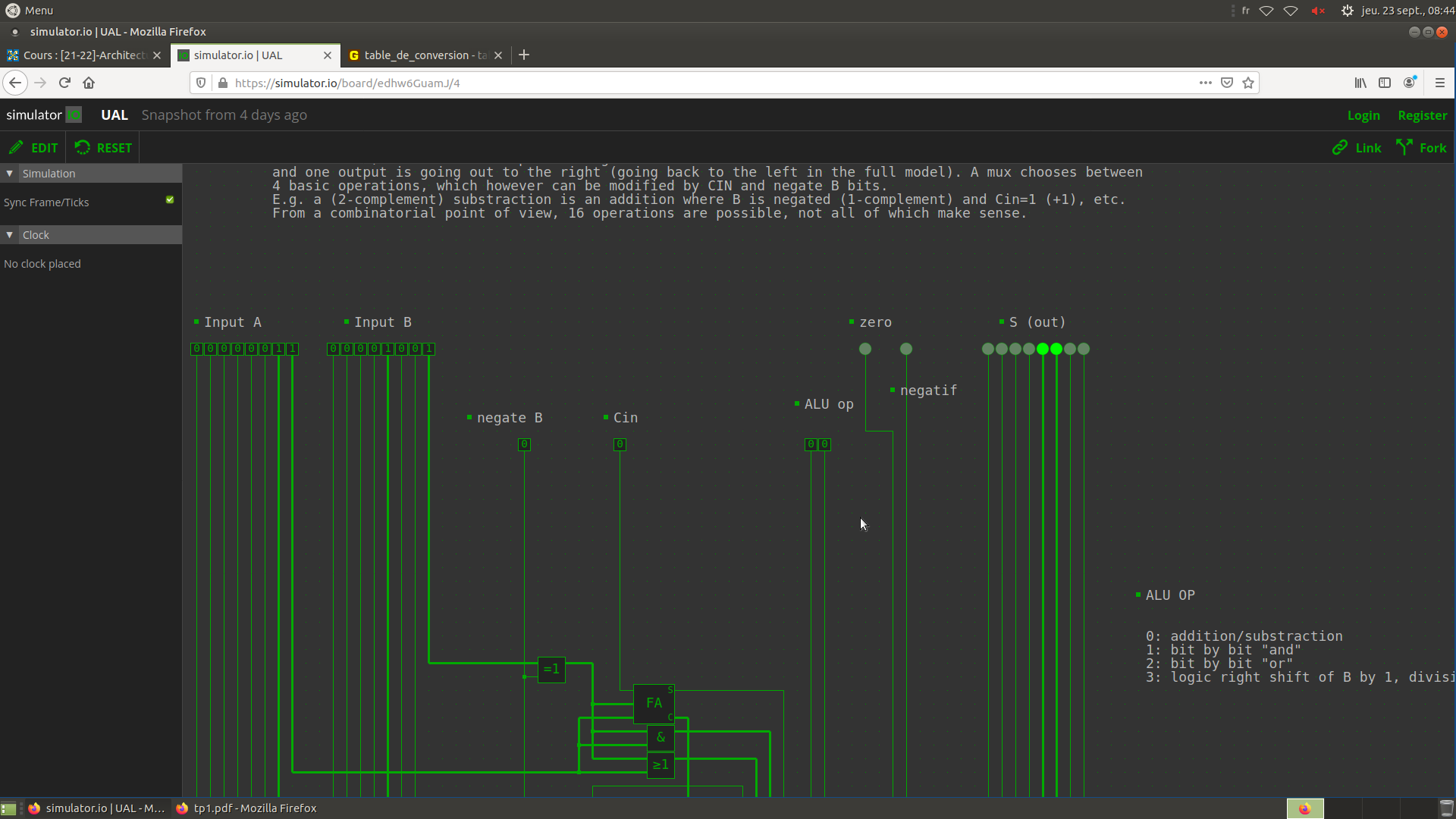
**RAMDANI Meriem**

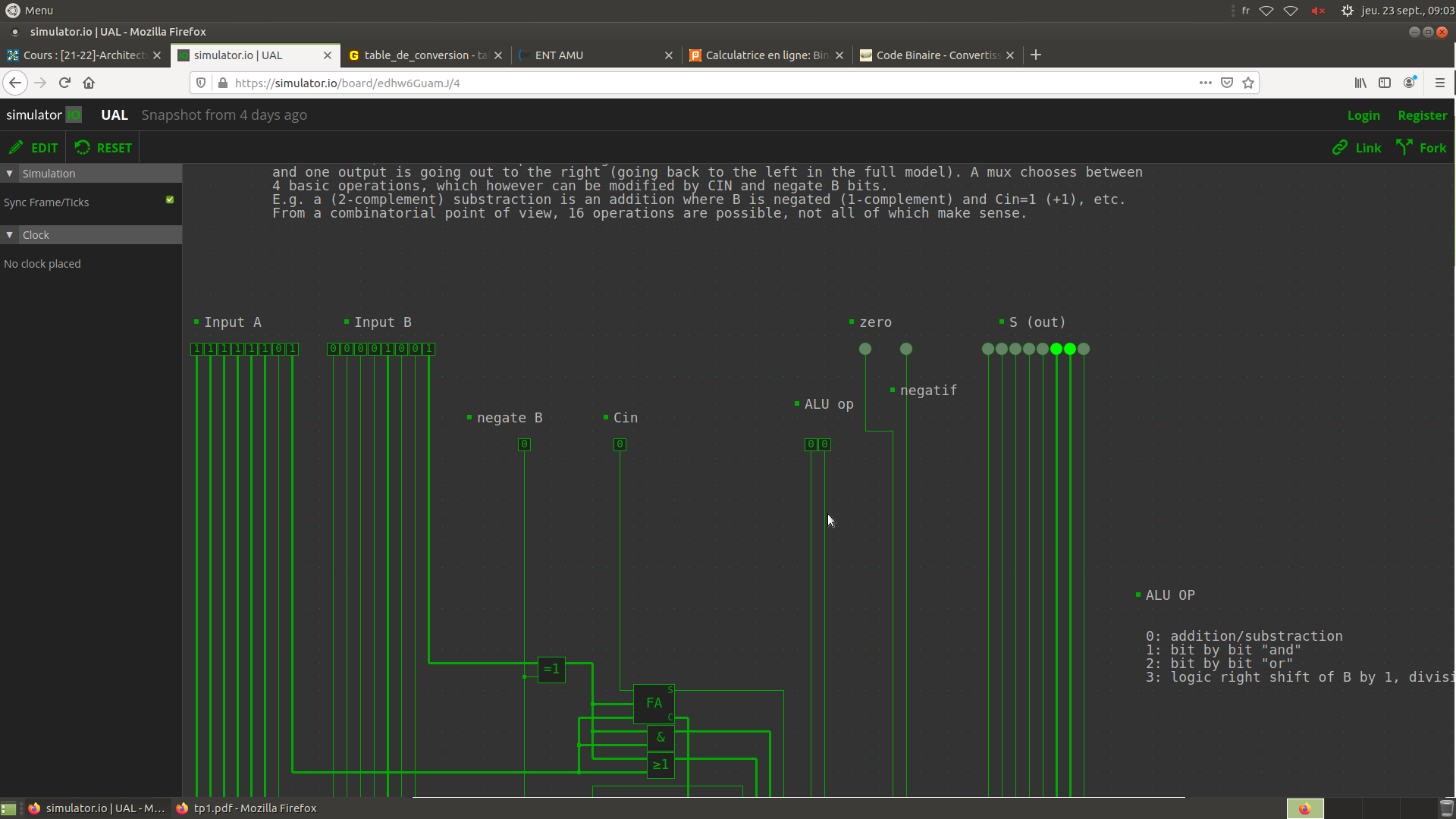
**Groupe 4**

**Compte rendu TP1 : Microprogrammation**

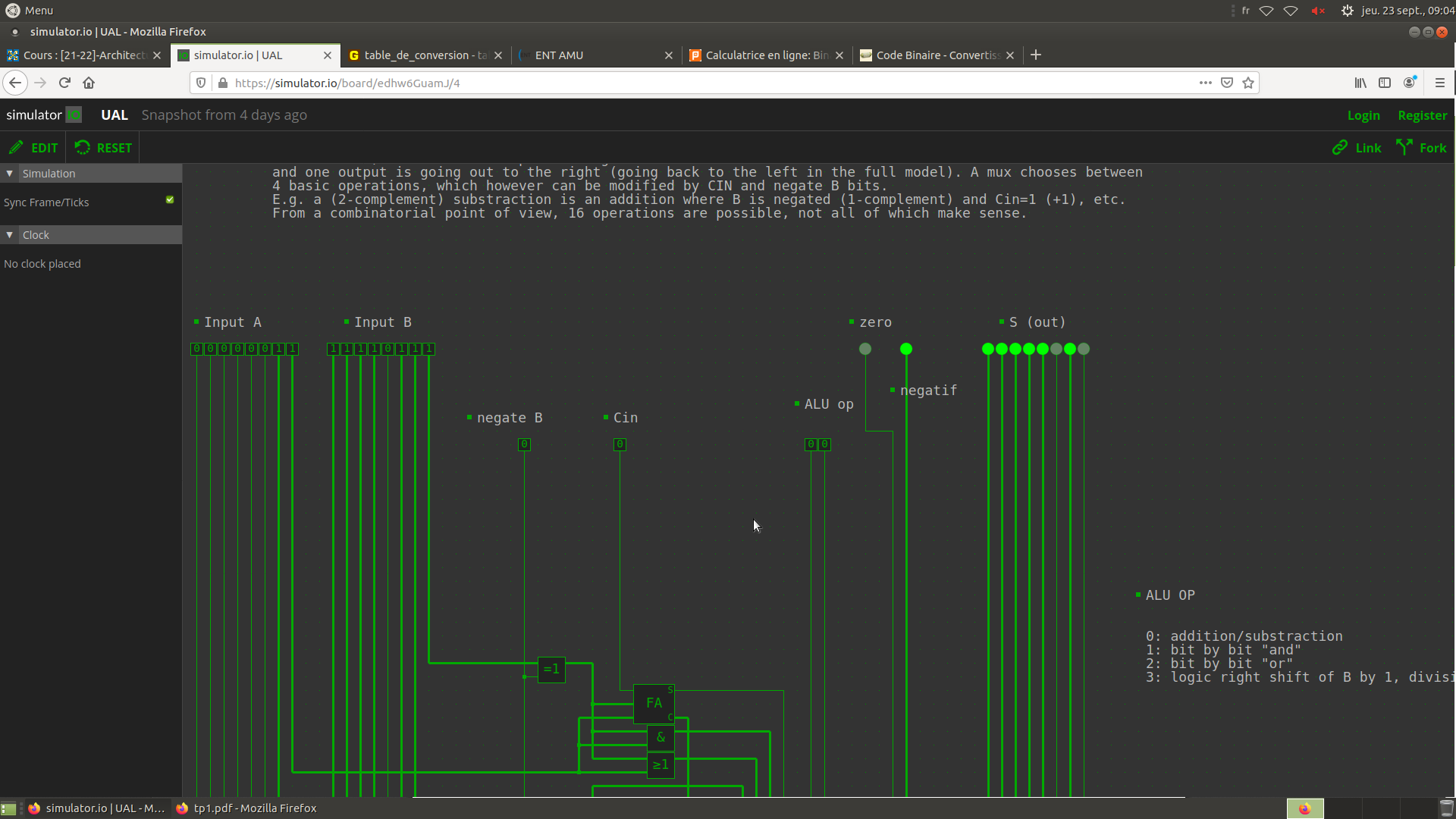
**Exercice 1 :**

1. 3 = 00000011 ; 9 = 00001001
2. -3 en Cà2 : 11111101 ; -9 en Cà2 : 11110111



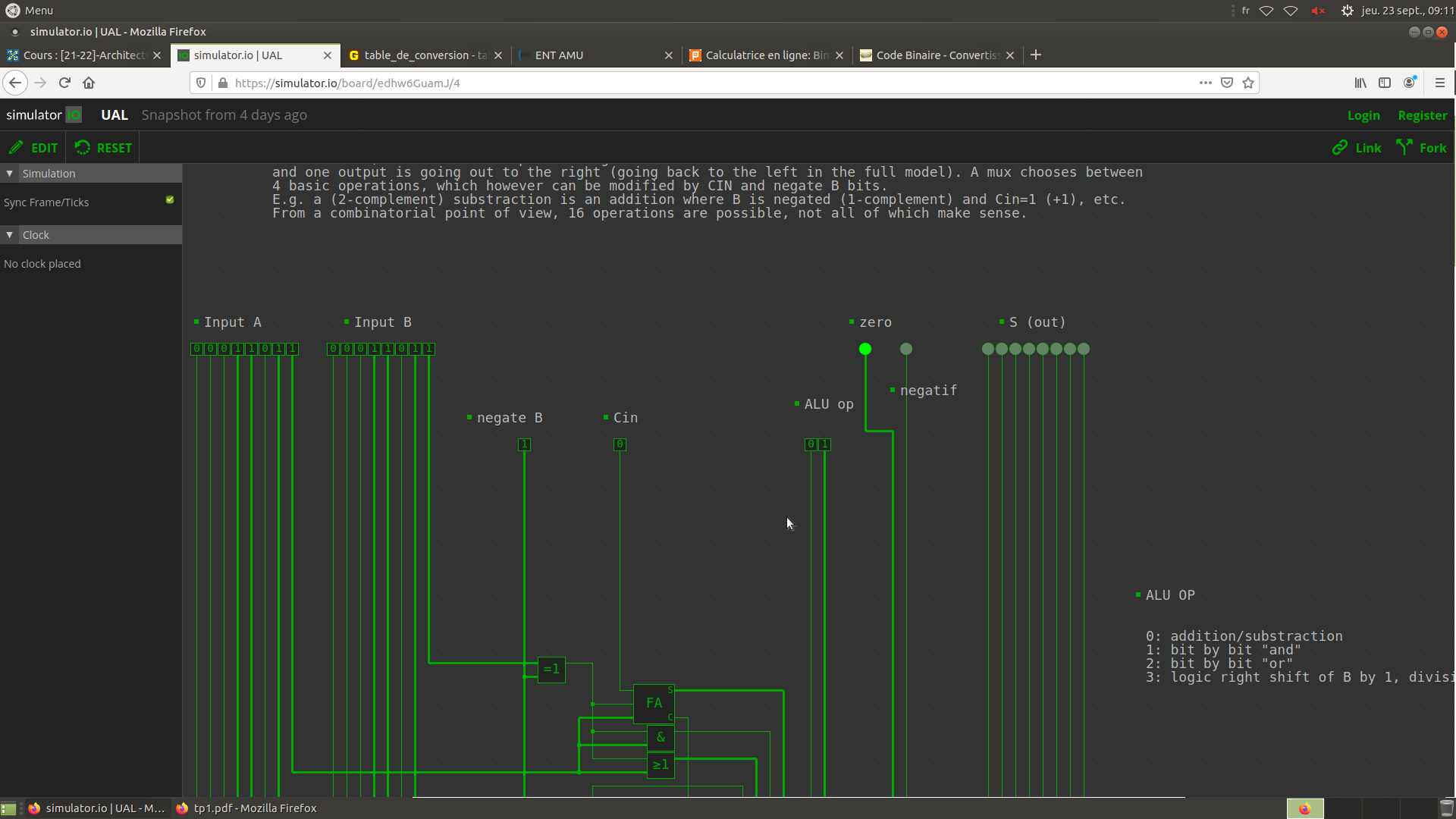
**3) Capture de la soustraction 9-3**

**4) Capture de la soustraction 3-9**



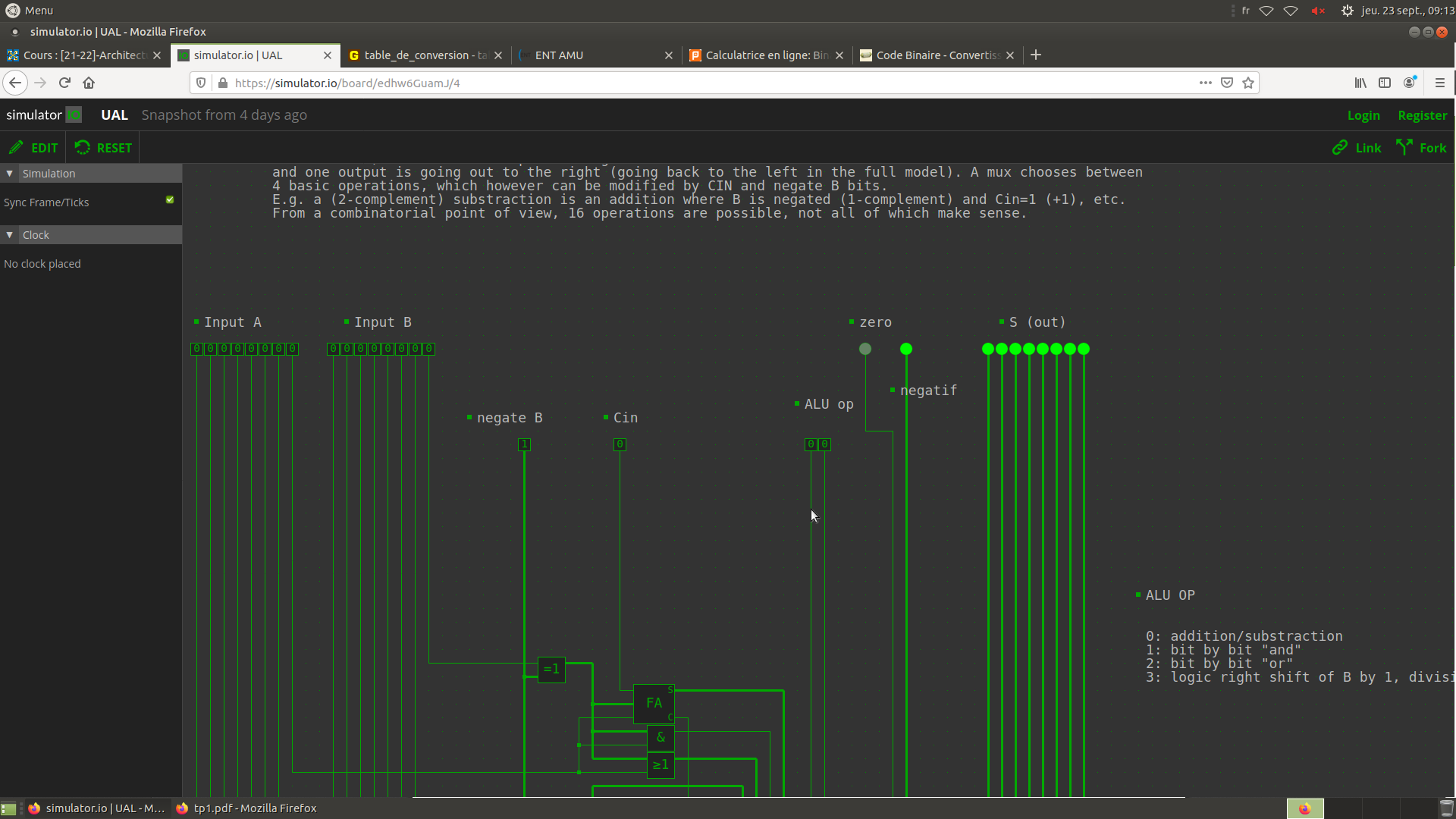
**5)Agencement de l'opération et des variations qui, quel que soit l'entrée, rend un résultat 0 :**

**Pour avoir 0 en sortie avec les mêmes entrées, on devrait faire “negate B” et mettre l’ALU à 1 (AND) et ainsi on aura (A ET NON A bit par bit) ce qui nous donnera 0 en sortie.**



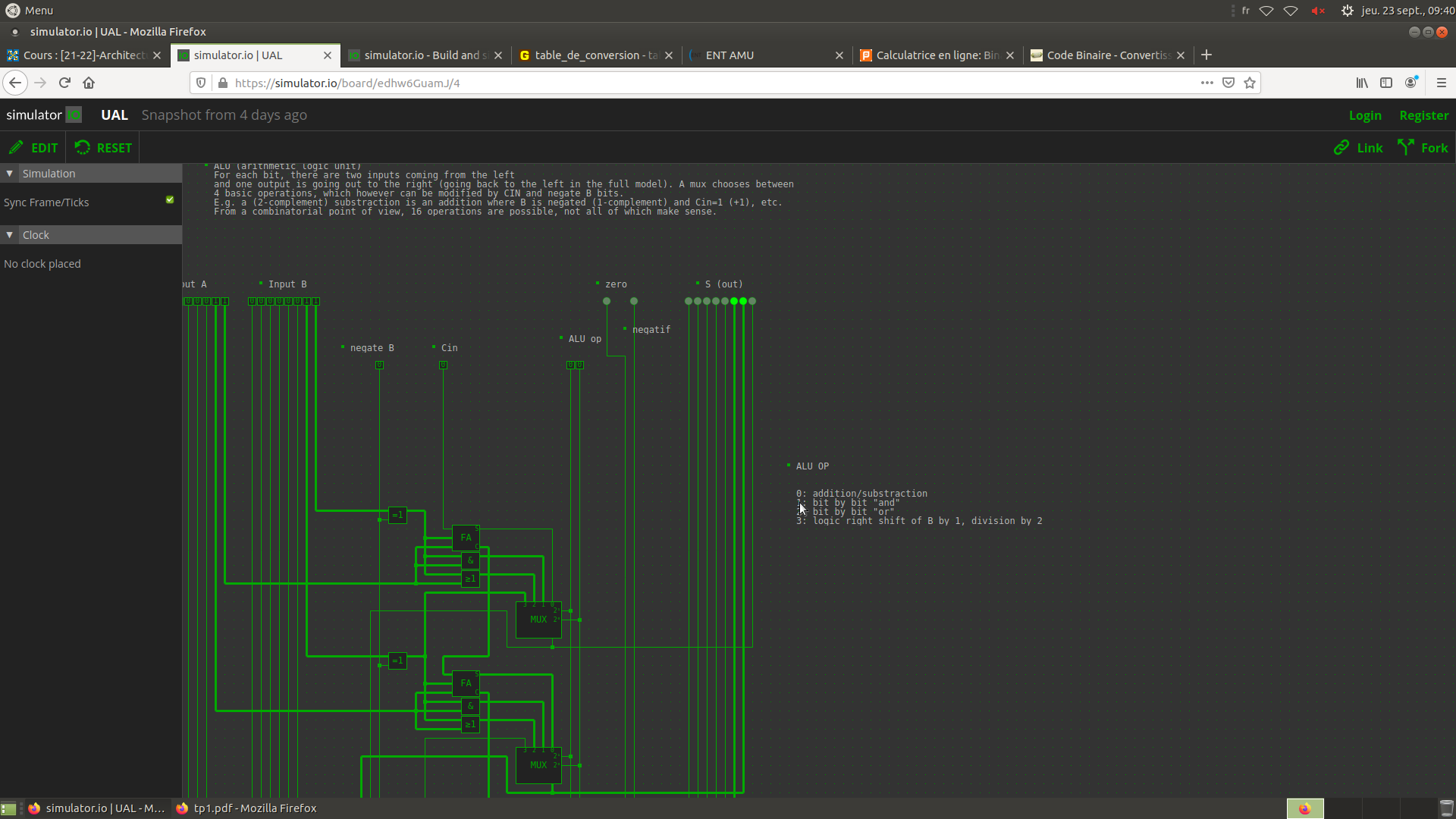
**6) Agencement qui pour deux entrées 0, calcule le résultat 1 :**

**Pour avoir 1 en sortie avec 2 entrées qui sont à 0, on fait negate B et on met l’ALU à 0 pour faire l’addition entre A et NON B ce qui nous donne 1.**



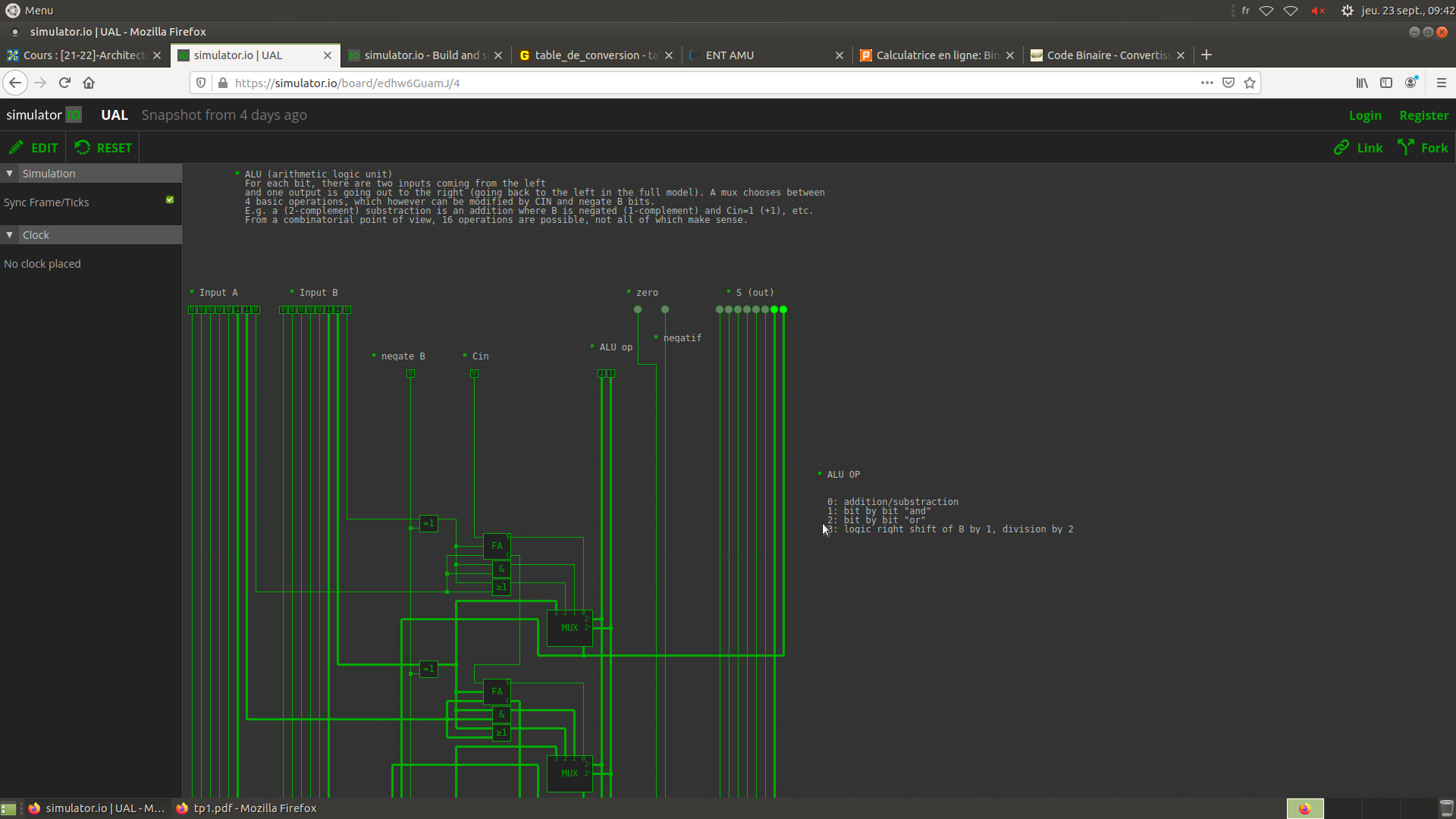
**7) La multiplication par 2 :**

**Pour faire la multiplication par 2, on met le même nombre dans les 2 entrées, puis on met l’ALU à 0 pour faire l’addition entre A et B.**



**8) La division par 2 :**

**Pour faire la division par 2, on met le même nombre dans les 2 entrées et on met l’ALU sur 3.**



**9)Pour tester si 2 entiers sont identiques soit :**

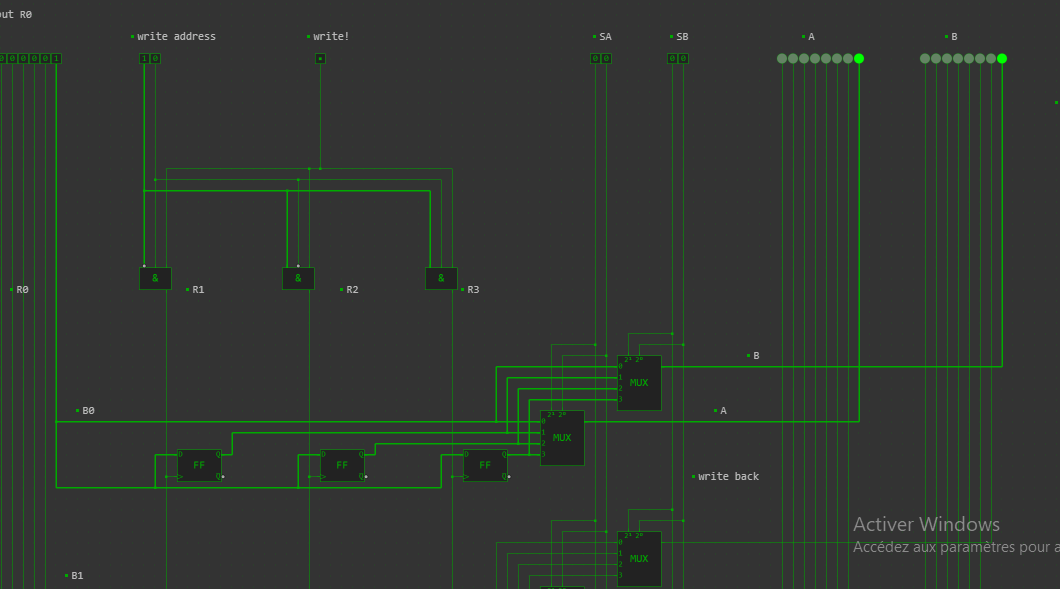
**\_On fait comme la question 5 en mettant l’ALU sur 1, et negate B, et obtient 0 en sortie.**

**\_ On met l’ALU à 0, et on fait negate B, et on met CIN à 0, et ainsi on fait une soustraction.**

**Exercice 2 :**

1) On constate qu’on a le même nombre en sortie A et B.

2)



**Exercice 3 :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **R1** | **R2** | **R3** | **ETAT** | **CONDITION** |
| **1** | 0 | 0 | 0 | 1 | FALSE |
| **2** | 0 | 0 | 1 | 2 | FALSE |
| **3** | 3 | 0 | 1 | 3 | FALSE |
| **4** | 3 | 3 | 1 | 4 | FALSE |
| **5** | 2 | 3 | 1 | 3 | FALSE |
| **6** | 2 | 6 | 1 | 4 | FALSE |
| **7** | 1 | 6 | 1 | 3 | FALSE |
| **8** | 1 | 9 | 1 | 4 | FALSE |
| **9** | 0 | 9 | 1 | 3 | TRUE |
| **10** | 0 | 9 | 1 | 5 | FALSE |

**Exercice 4 :**

\_A l’état 0 on fait AND entre le contenu de rep1 qui est 2 et le négatif du rep2 donc -2, et ça renvoie 0. Puis on met 0 dans le rep2, et on passe à l’état 1.

\_A l’état 1 on fait OR entre le contenu du rep2 et le rep2 qui est à 0 suite à l’état 0, ce qui nous donne 0 et puisque on a fait CIN ça renvoie 1 dans le rep3, et on passe à l’état 2.

\_A l’état 2 on fait AND entre rep0 et rep0 ça renvoie le contenue de rep0 on obtient 3 et on le met dans le rep1 puis on fait le test si = à 0. On passe à l’état 3.

\_A l’état 3, on fait OR entre le contenu de R0 qui est 3 et R2 qui est 0, on obtient 3 et on le renvoie dans R2, puis on fait OR entre le contenu de R0(3) et R0(3) on obtient le 3 et on le met dans R0.

\_ A l’état 4 on fait OR entre le negate du contenu de R3(-1) et le contenu de R1(3) on obtient 2 on le met dans R1, on test la condition, et on passe à l’état 3.

\_ on fait OR entre le contenu de R0 qui est 3 et R2 qui est 3 suite à l’état 3 précédent, on obtient 6 et on le renvoie dans R2, puis on fait OR entre le contenu de R0 et R0 on obtient le 3 et on le met dans R0, on passe à l’état 4.

\_\_ A l’état 4 on fait OR entre le negate du contenu de R3(-1) et le contenu de R1(2) on obtient 1 on le met dans R1, on test la condition, et on passe à l’état 3.

\_ on fait OR entre le contenu de R0 qui est 3 et R2 qui est 3, on obtient 9 et on le renvoie dans R2, puis on fait OR entre le contenu de R0(3) et R0(3) on obtient le 3 et on le met dans R0.

\_ A l’état 4 on fait OR entre le negate du contenu de R3(-1) et le contenu de R1(1) on obtient 0 on le met dans R1, on test la condition, et on passe à l’état 5.

\_A l’état 5, on affiche R2 qui est 9, et on termine.

**Exercice 5 :**

1)\_ Pour avoir 1 dans un registre on fait CIN.

\_ Pour tester que le bit b0 d’un registre vaut 1 : on met dans le premier bit d’un deuxième registre 1, et on fait un AND, et puisque le AND se fait bit par bit, si on obtient 1 dans le premier bit de la sortie c’est que b0 vaut 1, si on obtient 0 c’est qu’il vaut 0.

\_Pour tester que le bit bi d'un registre vaut 1 pour i = 1, 2, on divise le contenu d registre par 2, i fois, et on fait comme la question 2.

2)

R2=0 ;

R3 = 1 ;

R1=R0 ;

While (R1 ! = 0) {

If R1 et R3 != 0{

R2 = R2+R3 ;

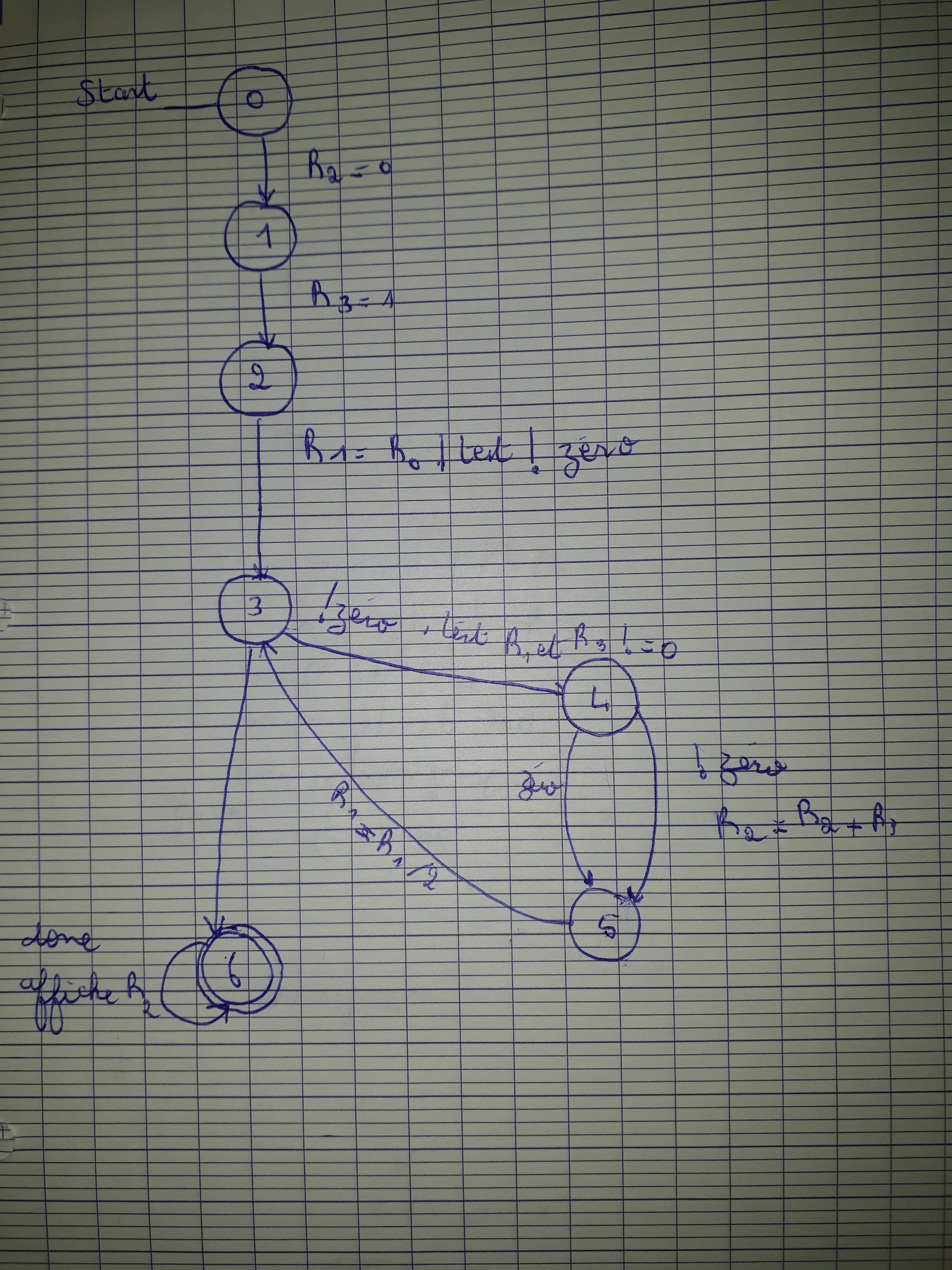
}

R1=R1/2 ;

}

return R2 ;

**3) Le graphe d’état :**



**\_La table d’instructions :**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| State | cond | ALU | test | NegB | Cin | Wen | SB | SA | Wa | succ | done | cmnt |
| 0 |  | AND | \_ | 1 | 0 | 1 | 2 | 2 | 2 | 1 | 0 | R2=0, goto1 |
| 1 |  | + | \_ | 0 | 1 | 1 | 2 | 2 | 3 | 2 | 0 | R3=1, goto2 |
| 2 |  | AND | zero | 0 | 0 | 1 | 0 | 0 | 1 | 3 | 0 | R1=R0, test zero, goto3 |
| 3 | 0  1 | AND\_ | Zero\_ | 0  0 | 0  0 | 0  0 | 1  0 | 3  0 | 0  0 | 4  6 | 0  0 | Test R1 Et R3 = 0 goto4  Goto6 |
| 4 | 0  1 | + | \_  \_ | 0  0 | 0  0 | 1  0 | 2  0 | 3  0 | 2  0 | 5  5 | 0  0 | R2 = R2 +R3, goto5 Goto5 |
| 5 |  | DIV | zero | 0 | 0 | 1 | 1 | 0 | 1 | 3 | 0 | R1=R1/2 goto 3, test zero |
| 6 |  | + | \_ | 0 | 0 | 0 | 2 | 2 | 0 | 6 | 1 | Show R2, done |

**La calculatrice :**

