

Exécution du programme : *soft_8b_int_mult[V2]*, pas-à-pas

Description :

- Entrées : **A**, **B** : 2 entiers non-signés de 8 bits maximum
- Sortie : Leur produit sur 16 bits : **C** = **A** x **B**

1. **(Optionnel) : génération du code machine.** La 1^{ère} étape est de disposer d'un fichier contenant les instructions *texte* ainsi que le segment *données* du programme. Ce fichier flashMe.hex est déjà fourni mais vous pouvez le re-générer comme suit (par exemple, si vous souhaitez changer les valeurs des 2 nombres multipliés).

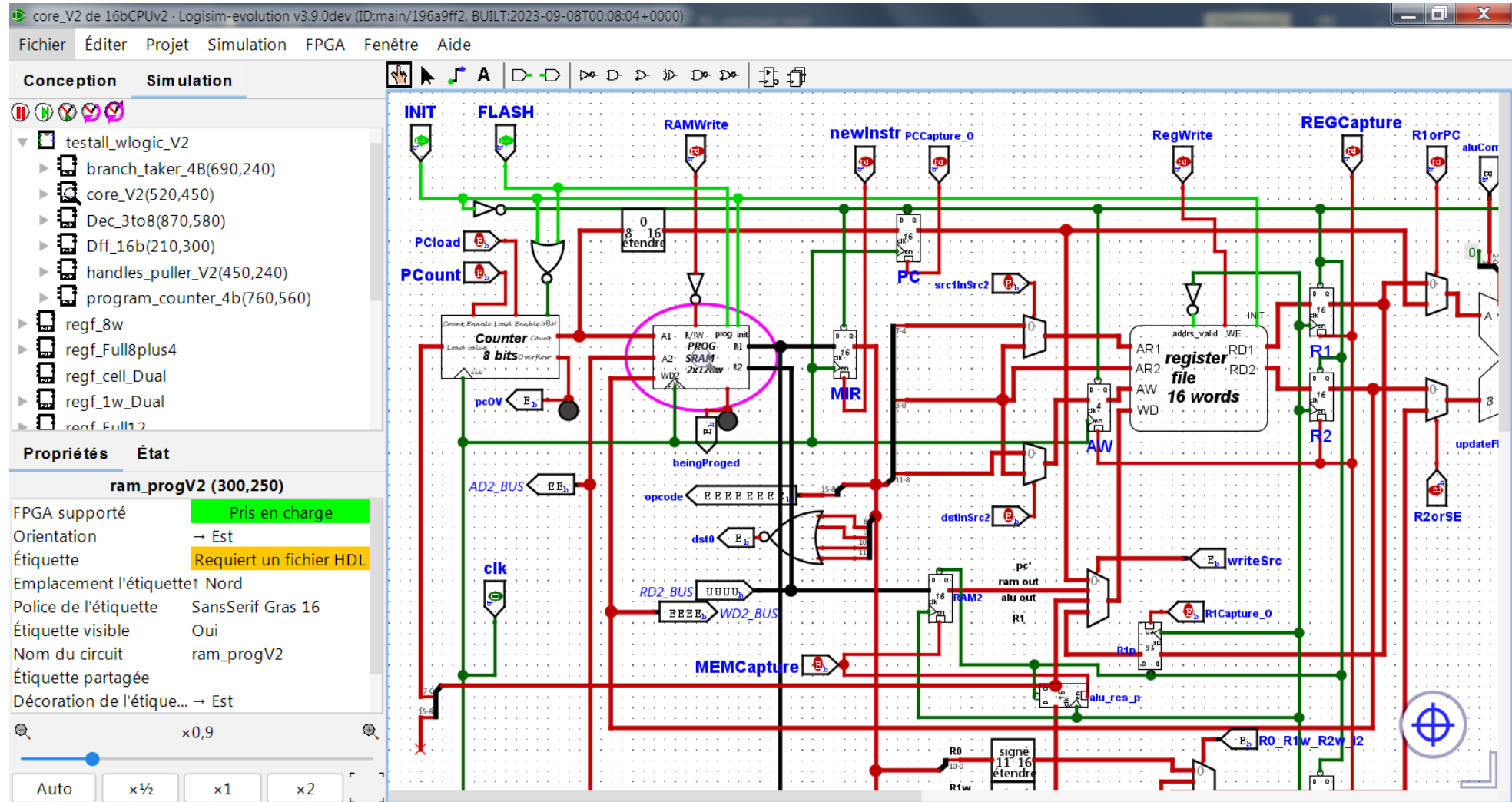
- a. Ouvrir le script assemblerV2.py dans un IDE
- b. Changer la ligne 29 par votre répertoire courant (i.e., le répertoire dans lequel se trouve ce script)

```
28 SCRIPTDIR = "your-local-repo-here"
29 assemblyFile = "./test_programs/soft_8b_int_mult[V2].s"
30 HEXPRGFILE = "./test_programs/flashMe.hex"
```

- c. Sauvegarder et exécuter le script.

2. **Flashage : Copie du code machine sur la RAM.** Ouvrir le projet Logisim (double-cliquer sur 16bCPUv2.circ). Vous arrivez sur le circuit principal appelé testall_wlogic_V2.

3. Faites de même avec le composant : **PROG SRAM 2x128w**



4. Faites un clic-droit sur la mémoire ROM affichée et sélectionnez *Charger une image* :

ram_progV2 de 16bCPUv2 · Logisim-evolution v3.9.0dev (ID:main/196a9ff2, BUILT:2023-09-08T00:08:04+0000)

Fichier Éditer Projet Simulation FPGA Fenêtre Aide

Conception **Simulation**

MIR
PC
RAM2
alu_res_p
R1p
R1
R2
SE
Dff_4b(1120,130)
AW
program_counter_8b(160,230)
ram_progV2(300,250)

Propriétés **État**

ROM (450,250)

FPGA supporté	Pris en charge
Largeur d'adresse	8
Largeur de donnée	16
Nombre de lignes	Simple
Autoriser accès non-ali...	Non
Contenu	(cliquer pour modifier)
Étiquette	Requiert un fichier HDL
Police de l'étiquette	SansSerif Gras 16
Étiquette visible	Non

x0,9

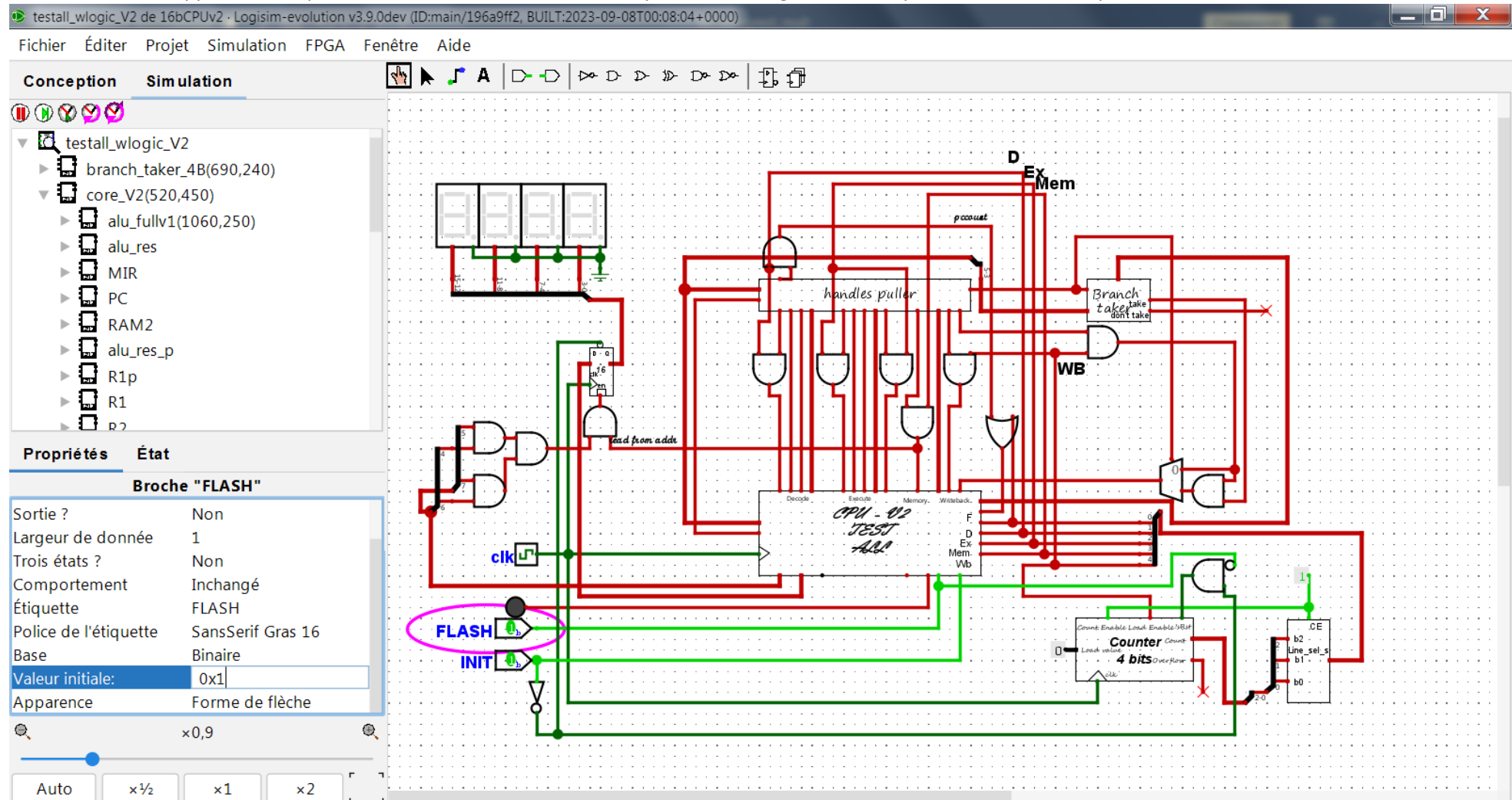
Auto x½ x1 x2

Diagram showing a circuit with components: Counter 8 bits, SRAM V2 (128 x 16), ROM (256 x 16), and various registers (R1, R2, R_Wb, WD2_16b). The ROM is highlighted with a pink oval, and a context menu is open over it, showing options: Effacer, Afficher les attributs, Modifier le contenu..., Effacer le contenu, **Charger l'image...**, and Sauvegarder l'image... The ROM content is displayed as follows:

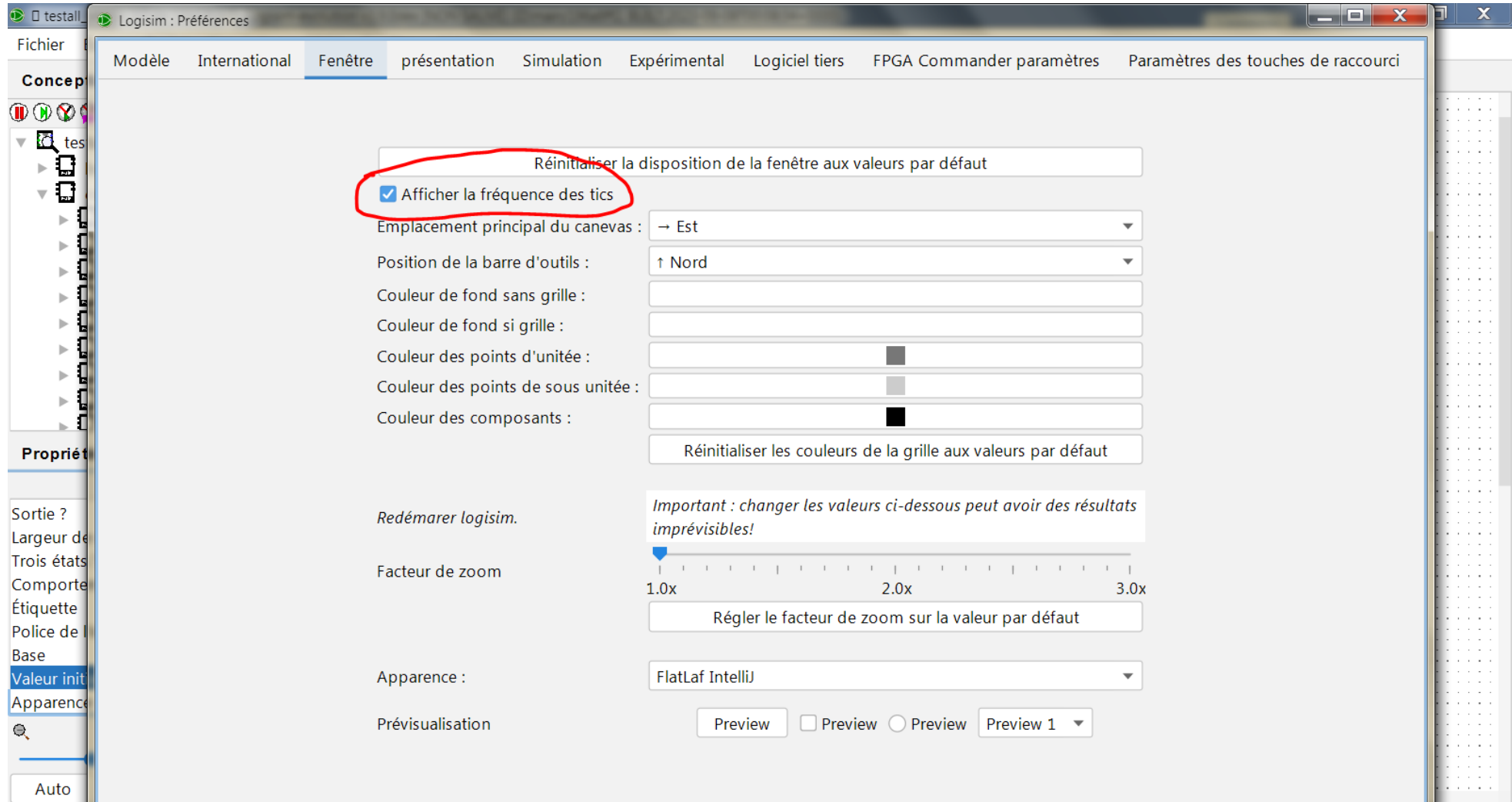
ROM 256 x 16	
00	0000 2d9e e201 0080
04	c01d c11e 200b e207
08	0008 e205 0001 e20f
0c	00f0 e000
10	8002 20c0
14	60ee 50c0

Include last 16w block from writable addresses (i.e., [0xf0..0xff] is A2_8b no-avail range)

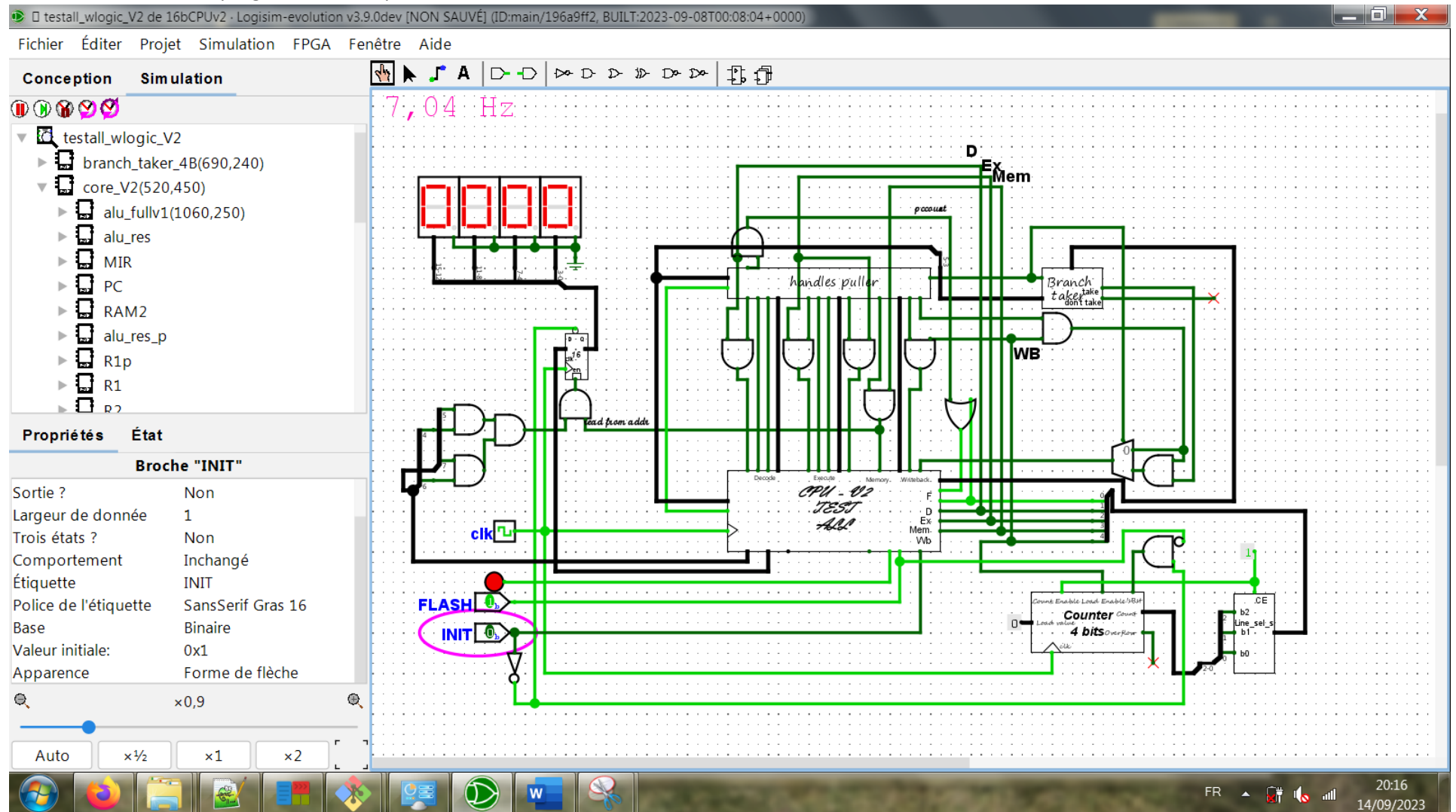
5. Naviguez dans le répertoire *test_programs* pour charger le fichier *flashMe.hex*
6. Revenir de 2 niveaux hiérarchiques en arrière (**ctrl+left arrow**) afin de retrouver la page du projet principal (*testall_wlogic_V2*)
7. Lancer la simulation (ctrl+K). Par défaut, les 2 entrées INIT et FLASH valent '1'. Si ce n'est pas le cas, une erreur « **Oscillations apparentes** » va être affichée et la simulation stoppée. Notez que les valeurs initiales d'une broche peut être changée dans son panneau latéral « *Propriétés* ».



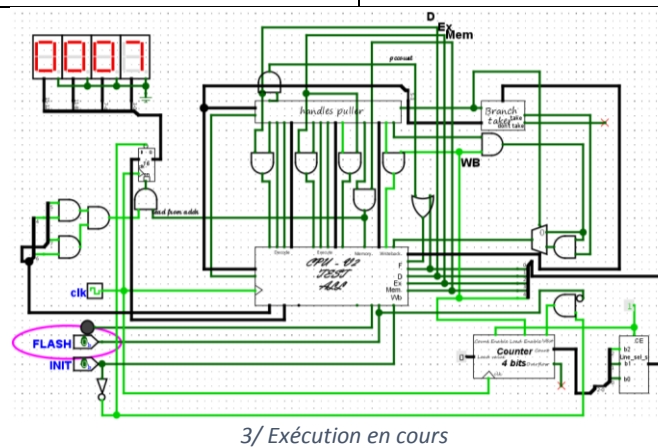
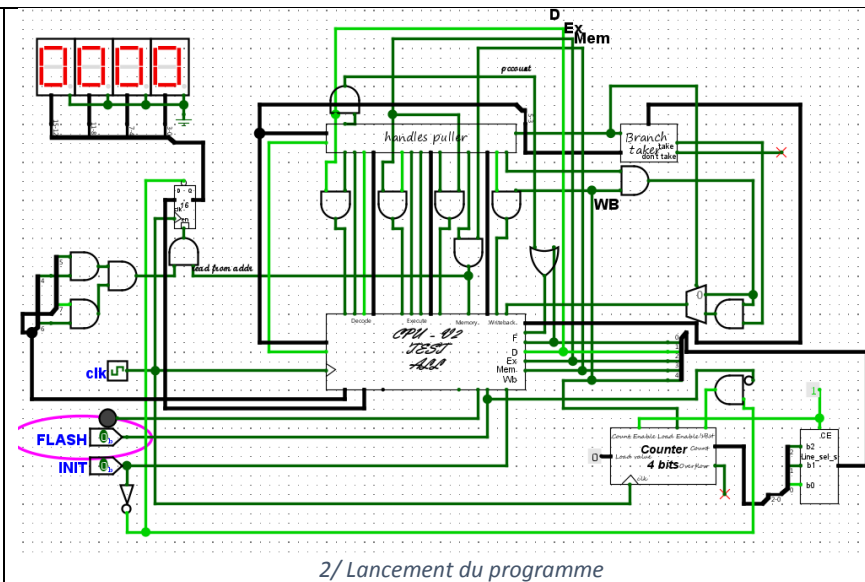
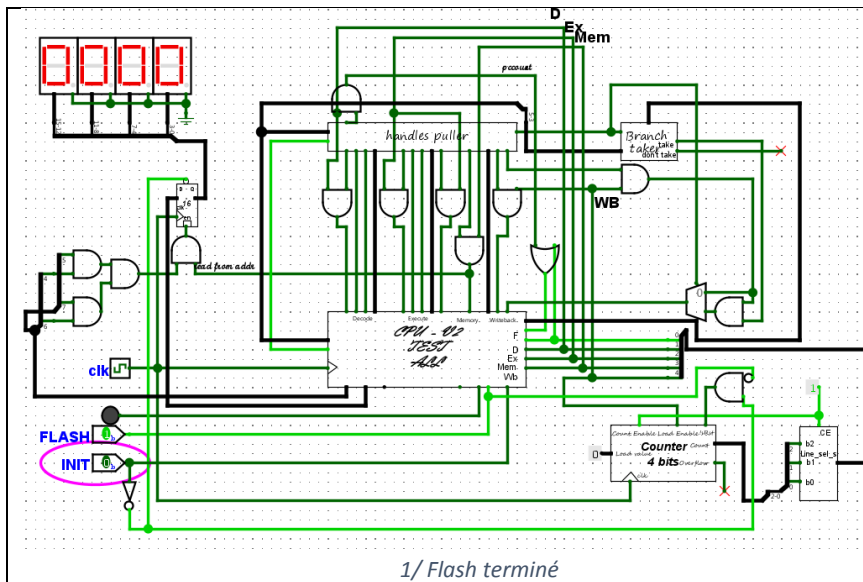
8. Afficher la cadence d'horloge actuelle de la simulation : Fichier → Préférences → onglet Fenêtre → cocher « Afficher la fréquence des tics »



9. Régler la fréquence des tics demi-cycle dans le menu déroulant « *Simulation* » Nota : on règle la cadence en demi-cycle mais Logisim affiche la cadence en cycles complets, d'où le facteur 2.
10. Basculer **INIT** ver '**0**' : le programme est copié sur la mémoire.



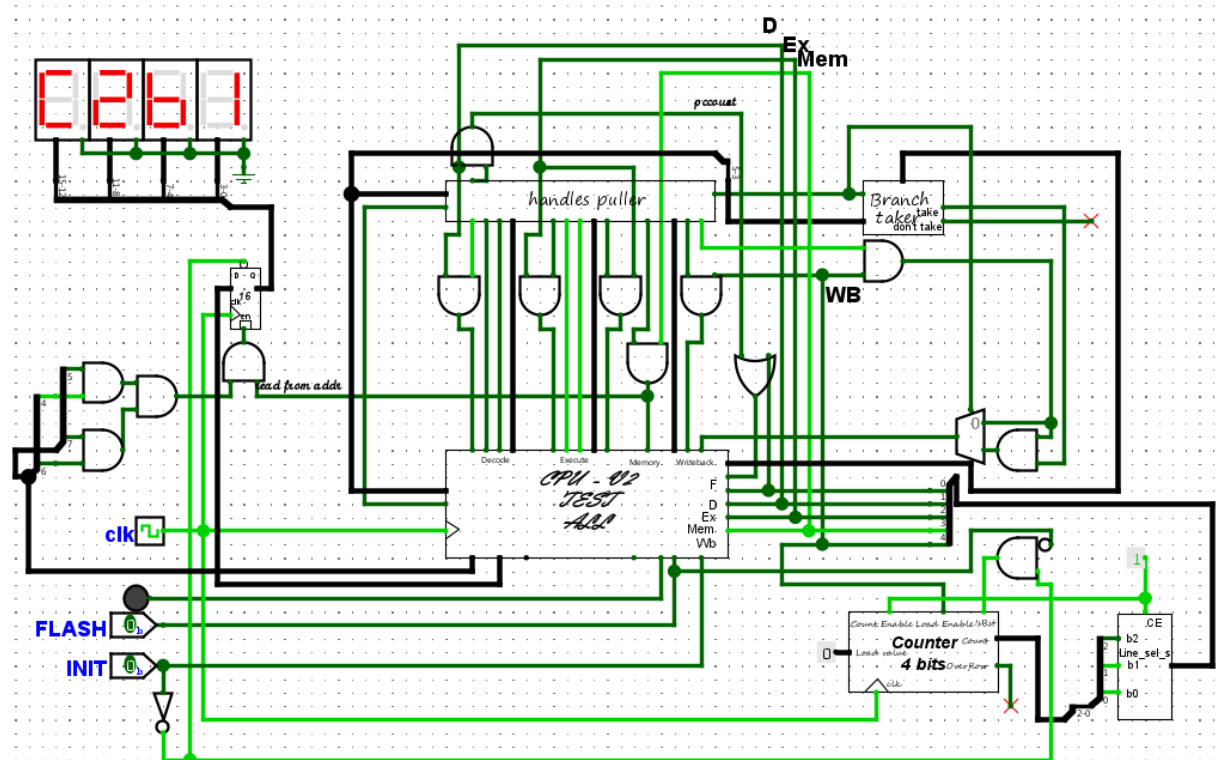
11. Quand la lumière rouge s'éteint, basculer la broche **FLASH** vers '0 pour voir le programme s'exécuter.



12. L'écran 7-segments affiche un décompte qui se termine par la valeur du calcul de $A \times B$, affichée en hexadécimal.

```
24 ; ## DATA segment : Global Variables ##
25 .org 0x80
26 .word 253 ; must be <256
27 .word 197 ; must be <256. Put the lowest here.
28
```

A value
B value



```
>>> 253*197
49841
>>> hex(_)
'0xc2b1'
```