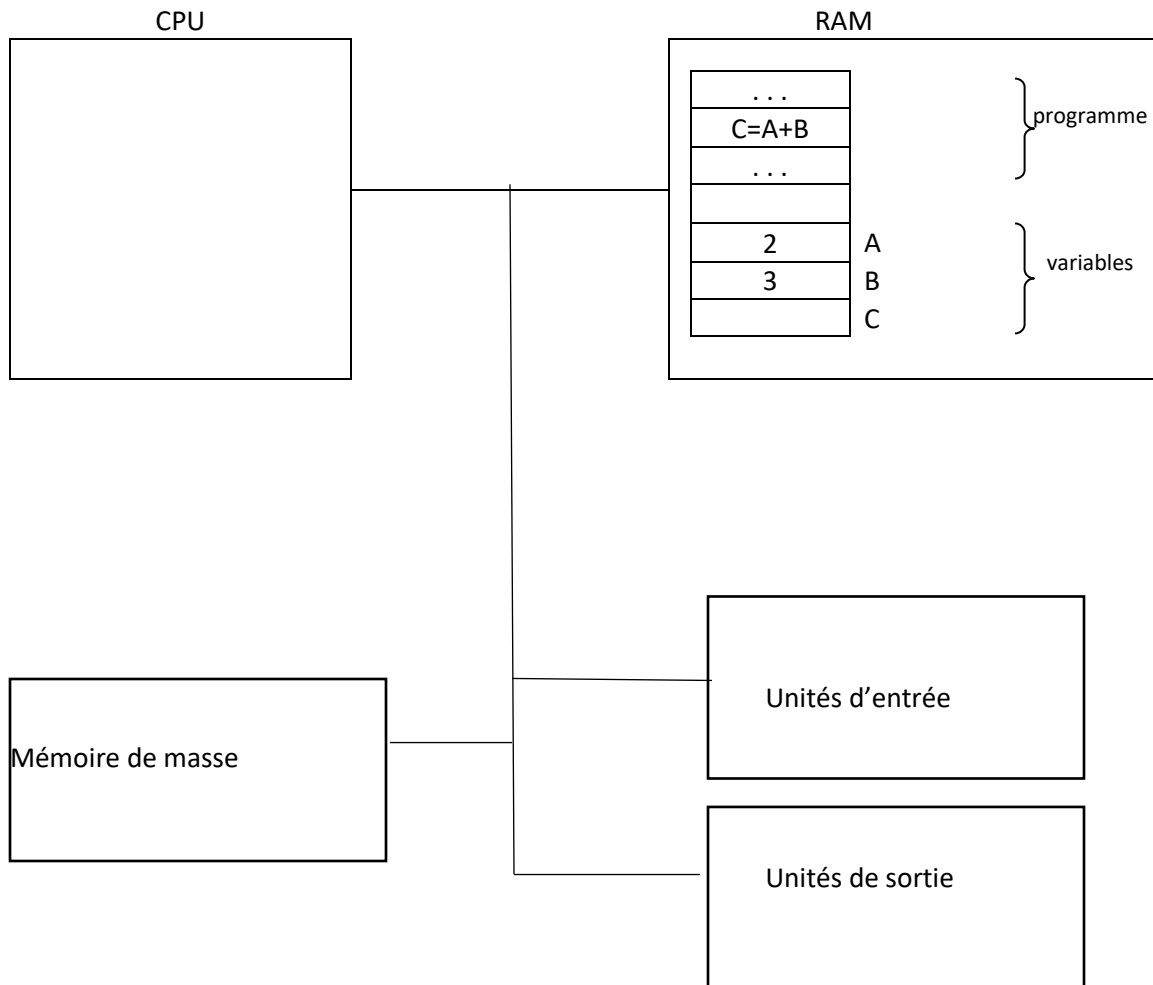
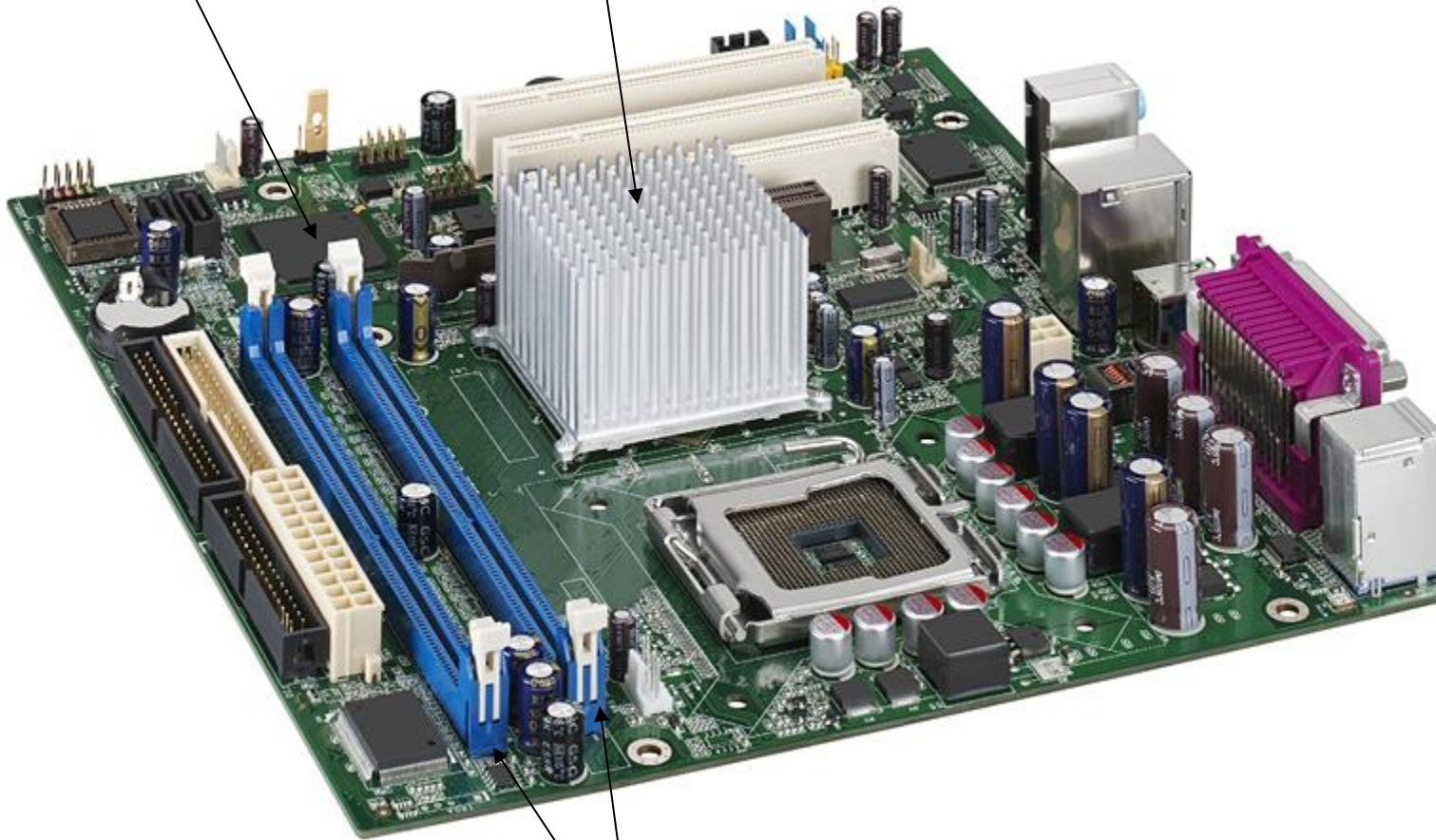


Présentation du micro-ordinateur (Partie 1)

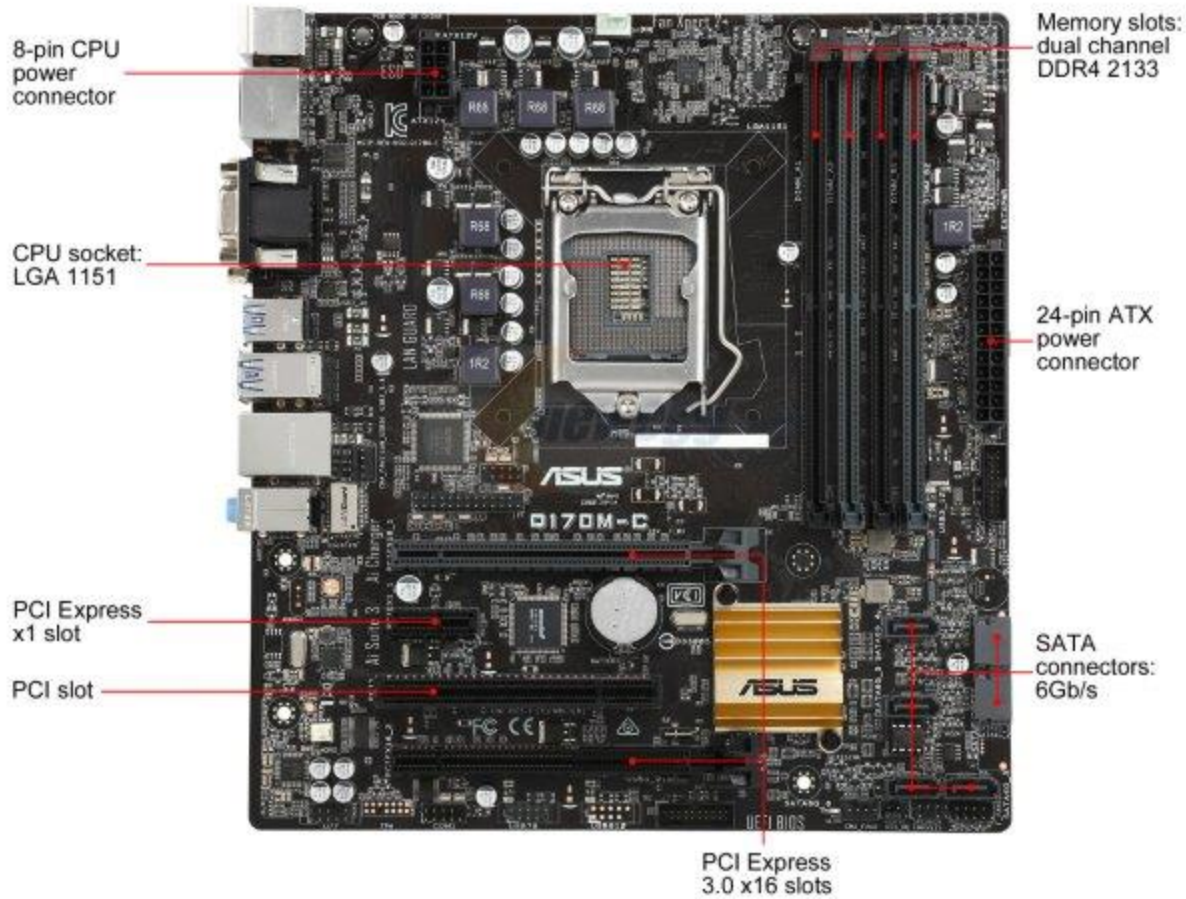


Composant secondaire
du chipset

Composant principal
du chipset (recouvert d'un dissipateur de chaleur)



Connecteurs de mémoire pour barrettes DIMM (en bleu)



Le CPU

Composante principale du micro-ordinateur. (On dira aussi processeur ou micro-processeur).

C'est la composante la plus rapide.

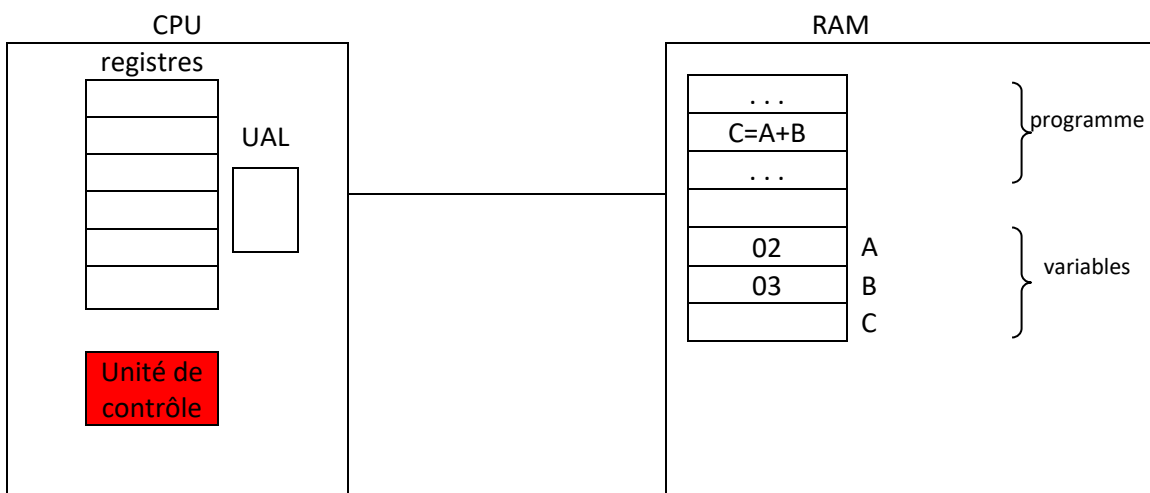
Tâches effectuées par le CPU : exécute les instructions des programmes et gère les communications avec ses périphériques.

1.Exécute les instructions des programmes

À remarquer : le lien important : CPU-RAM

Notes : --pour raison de commodité : pour les premiers exemples, le contenu de la RAM ne sera pas écrit en binaire comme c'est en réalité, mais en base 16

--RAM : 1 case=1 octet



Registres : cases mémoire à l'intérieur du CPU

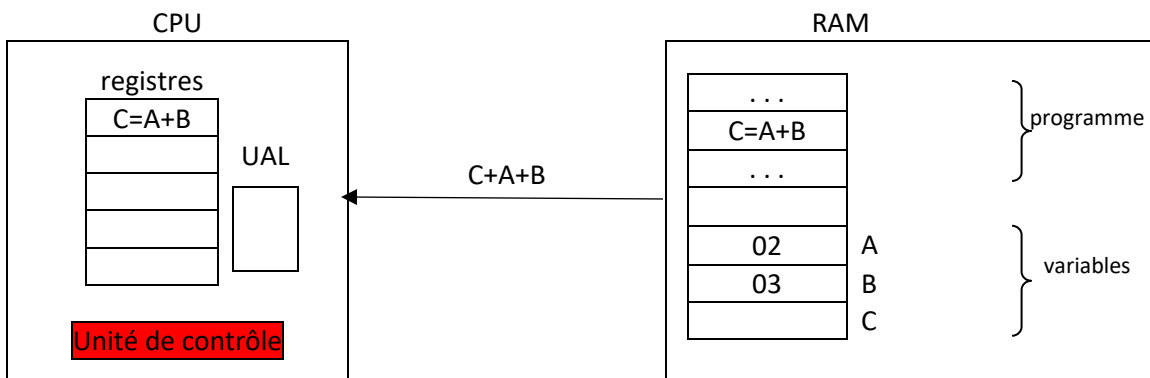
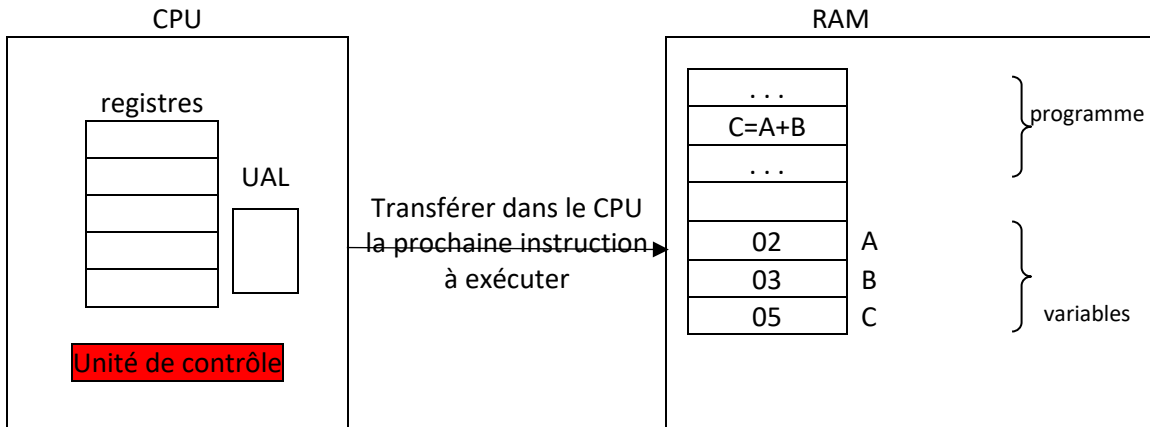
UAL : Unité arithmétique et logique (pour les calculs)

Unité de contrôle : composante qui dirige les opérations à l'intérieur du CPU

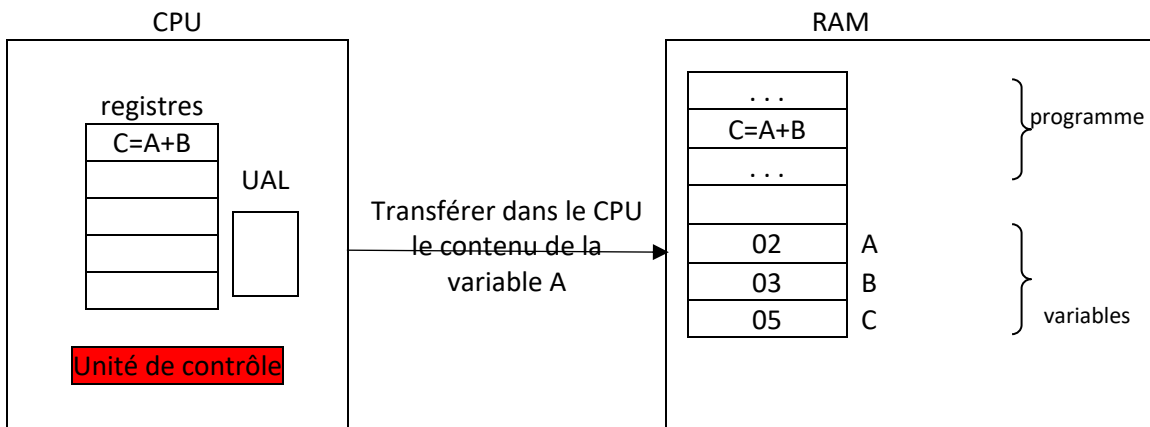
Exécution d'une instruction :

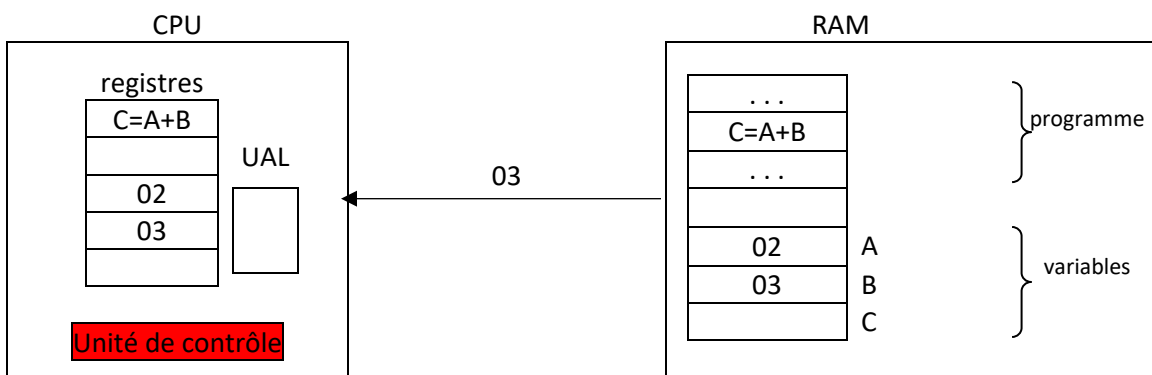
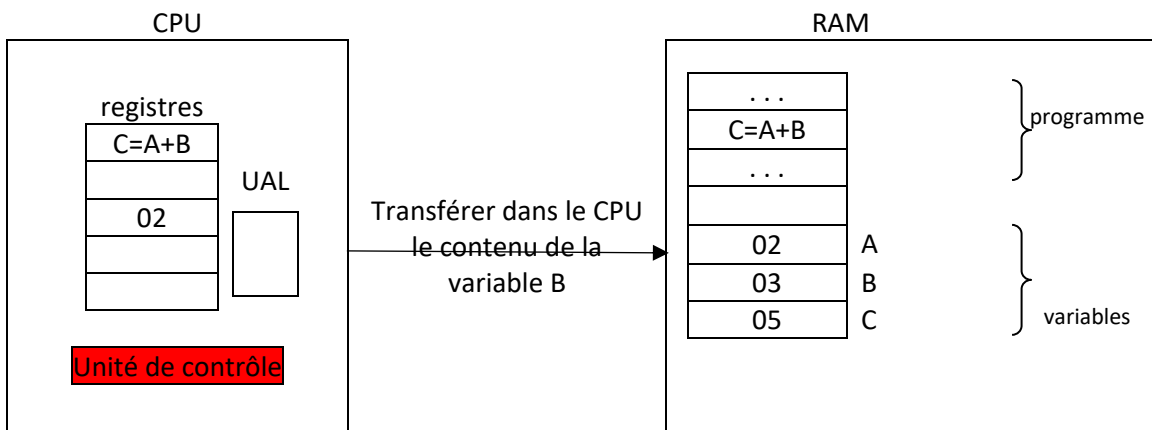
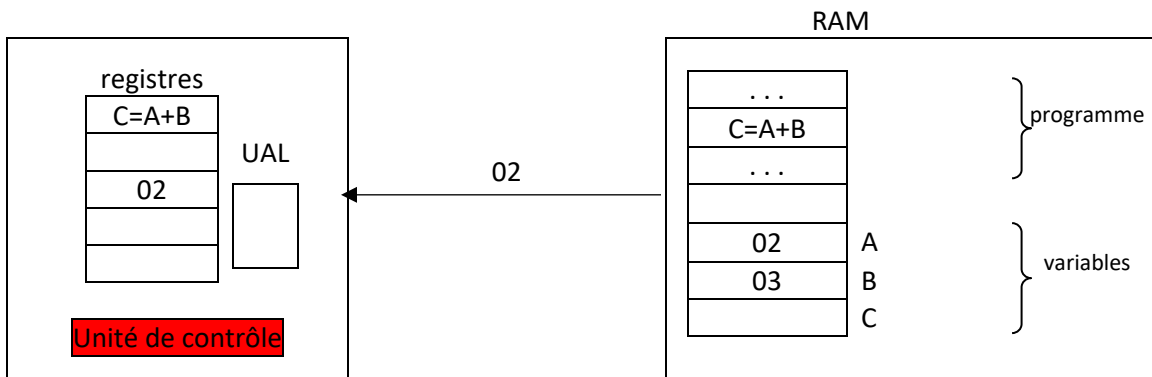
Étape 1 : L'unité de contrôle demande à la RAM de lui envoyer la prochaine instruction qui doit être exécutée (ici : $C=A+B$). L'instruction est placée dans un des registres du CPU.

Note : c'est une copie de l'instruction qui est envoyée au CPU.

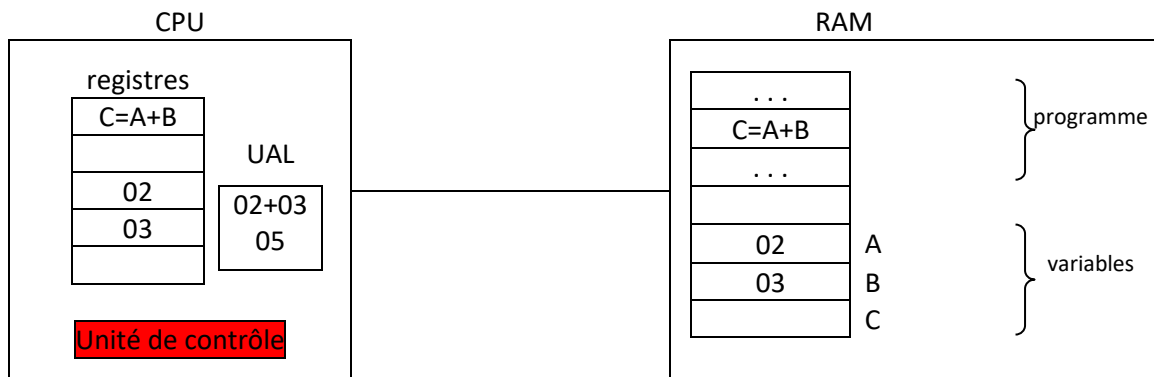


Étape 2 : L'unité de contrôle demande ensuite à la RAM de lui envoyer le contenu des variables A et B. Ces contenus sont aussi placés dans des registres du CPU (une copie aussi, comme pour les instructions).

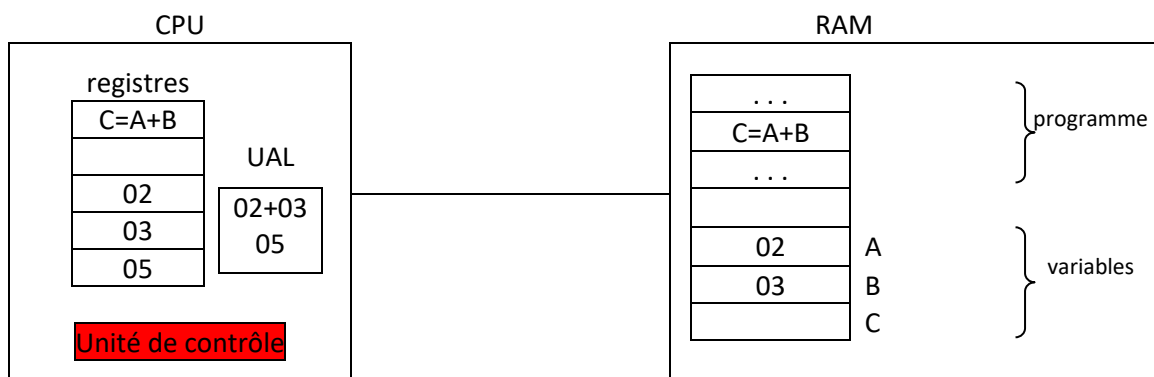




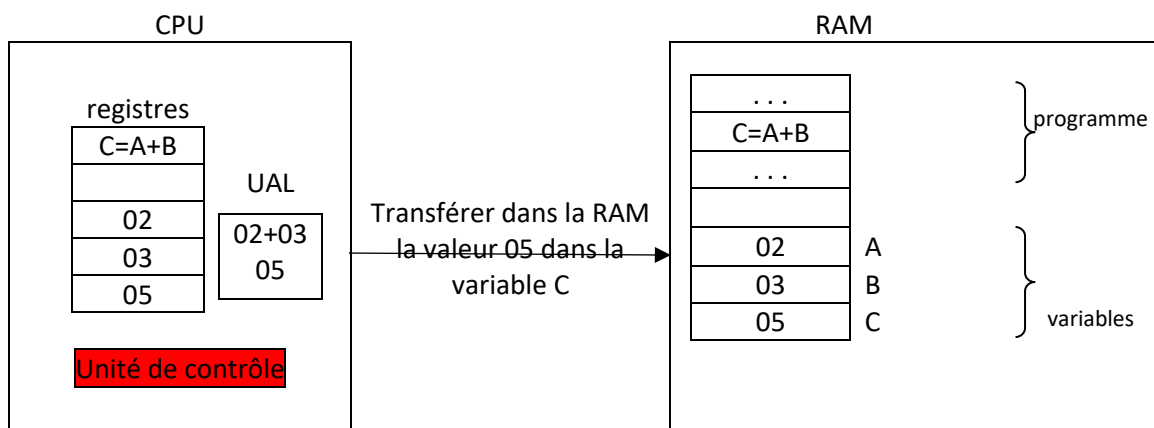
Étape 3 : Le CPU a maintenant tout ce qu'il faut pour exécuter l'instruction. L'unité de contrôle commande à l'UAL de faire l'opération arithmétique.



Étape 4 : Le résultat de l'opération arithmétique est placé dans un autre registre du CPU.



Étape 5 : L'unité de contrôle demande à la RAM de placer le résultat dans la variable C



L'unité de contrôle demande ensuite à la RAM de lui envoyer l'instruction suivante.

Notion d'adresses en RAM

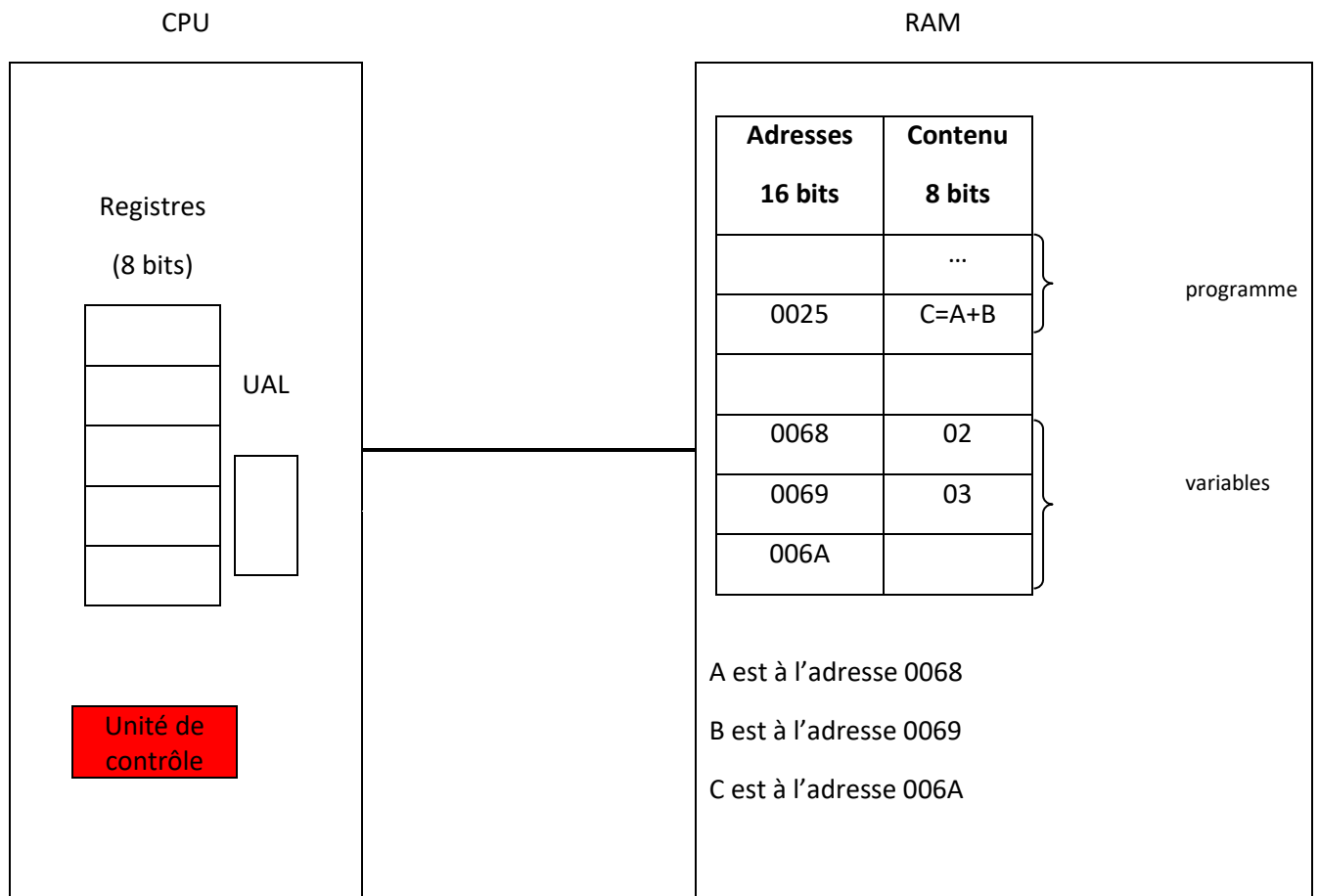
Jusqu'à présent, nous avons utilisé des noms de variables pour identifier les cases mémoires qui contiennent les données et les résultats. En fait, pendant l'exécution du programme, ce sont des adresses qui sont utilisées. Chaque case mémoire (donc octet) est numéroté. On appelle ces numéros des adresses.

Les noms de variables sont pour le programmeur et les adresses sont utilisées par l'ordinateur. Avant l'exécution d'un programme, il y a une correspondance qui est faite entre les variables et les adresses en RAM. Cette opération est transparente à l'utilisateur.

Reprenons l'exemple vu précédemment : (le programme a été placé en RAM par le système d'exploitation)

Si A est associé à l'adresse 0068, B à l'adresse 0069, C à l'adresse 006A on aura en RAM :

Adresses (16 bits)	Contenu (8 bits)	
	...	
0025	C=A+B	Instruction C=A+B du progr
	...	
0068	02	A
0069	03	B
006A	05	C

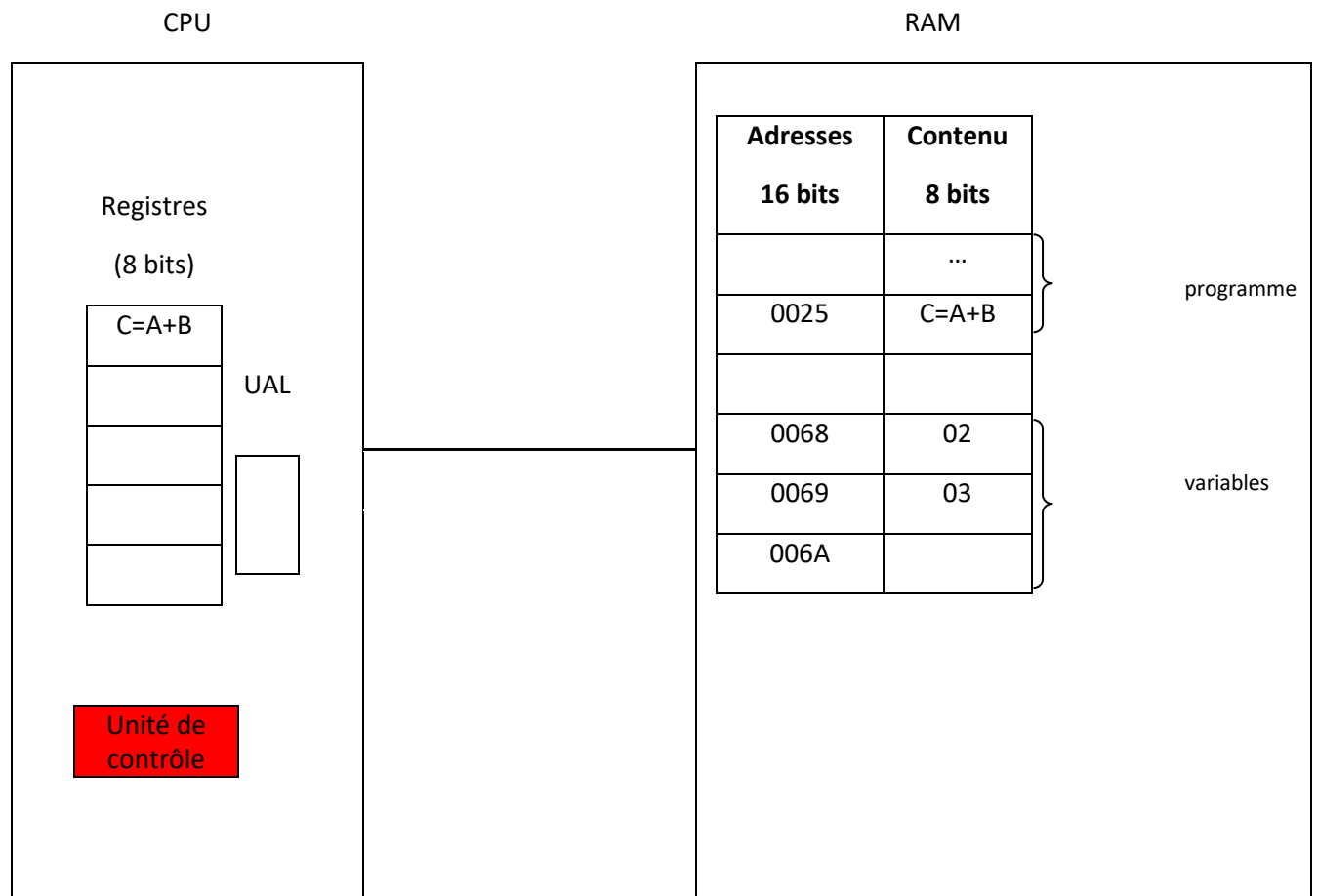


L'étape 2 de la page 3 était :

Étape 2 : L'unité de contrôle demande ensuite à la RAM de lui envoyer le contenu **des variables A et B**. Ces contenus sont aussi placés dans des registres du CPU (une copie aussi, comme pour les instructions).

Elle devient :

Étape 2 : L'unité de contrôle demande ensuite à la RAM de lui envoyer le contenu des **ADRESSES 0068 et 0069**. Ces contenus sont aussi placés dans des registres du CPU (une copie aussi, comme pour les instructions).



Notion de bus

Les informations qui circulent entre le CPU et la RAM sont de 3 types : commandes du CPU à la RAM, adresses et contenu de la RAM (revoir l'exécution d'une instruction pages 3 à 5).

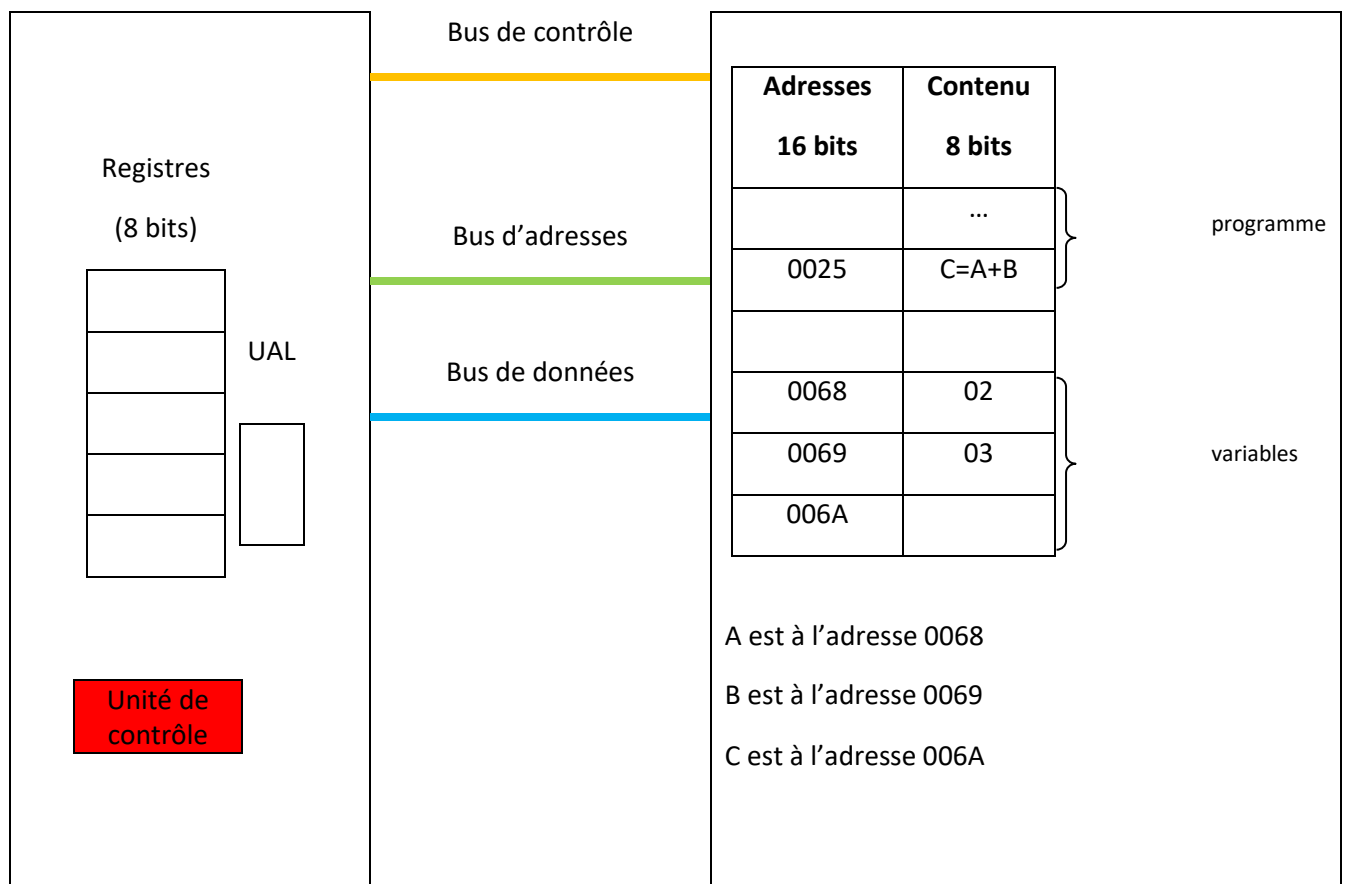
Il existe des voies de communication entre toutes les composantes de l'ordinateur. Ce sont les bus. Dans les exemples précédents, j'ai utilisé une seule ligne pour représenter toutes les voies de communication. En fait, il en existe 3 : bus de contrôle, bus d'adresses et bus de données, selon les informations qui y circulent.

Le bus de contrôle sert entre autres à véhiculer les commandes du CPU (ici vers la RAM)

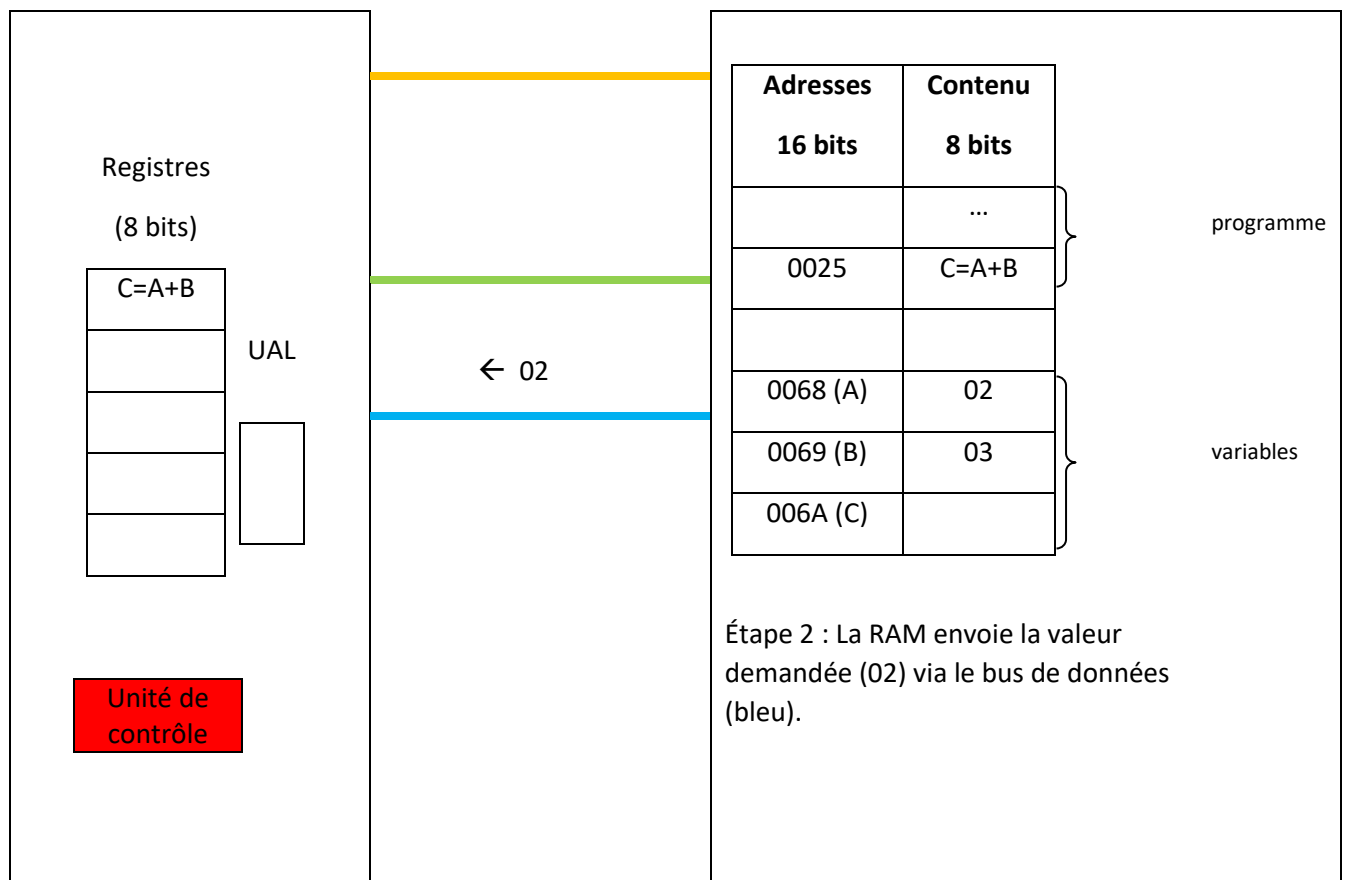
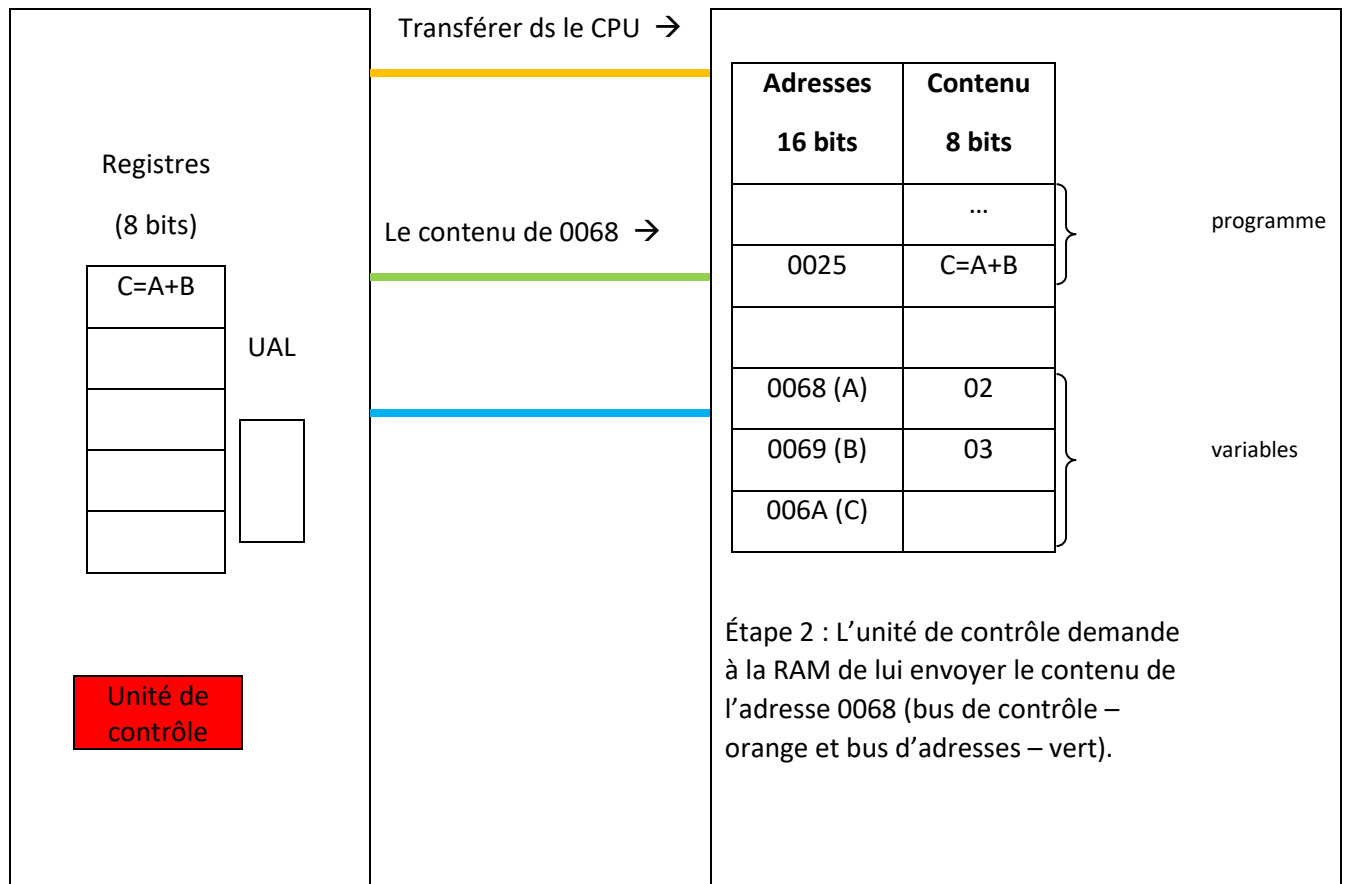
Le bus d'adresses sert à véhiculer les adresses (ici, les adresses de la RAM)

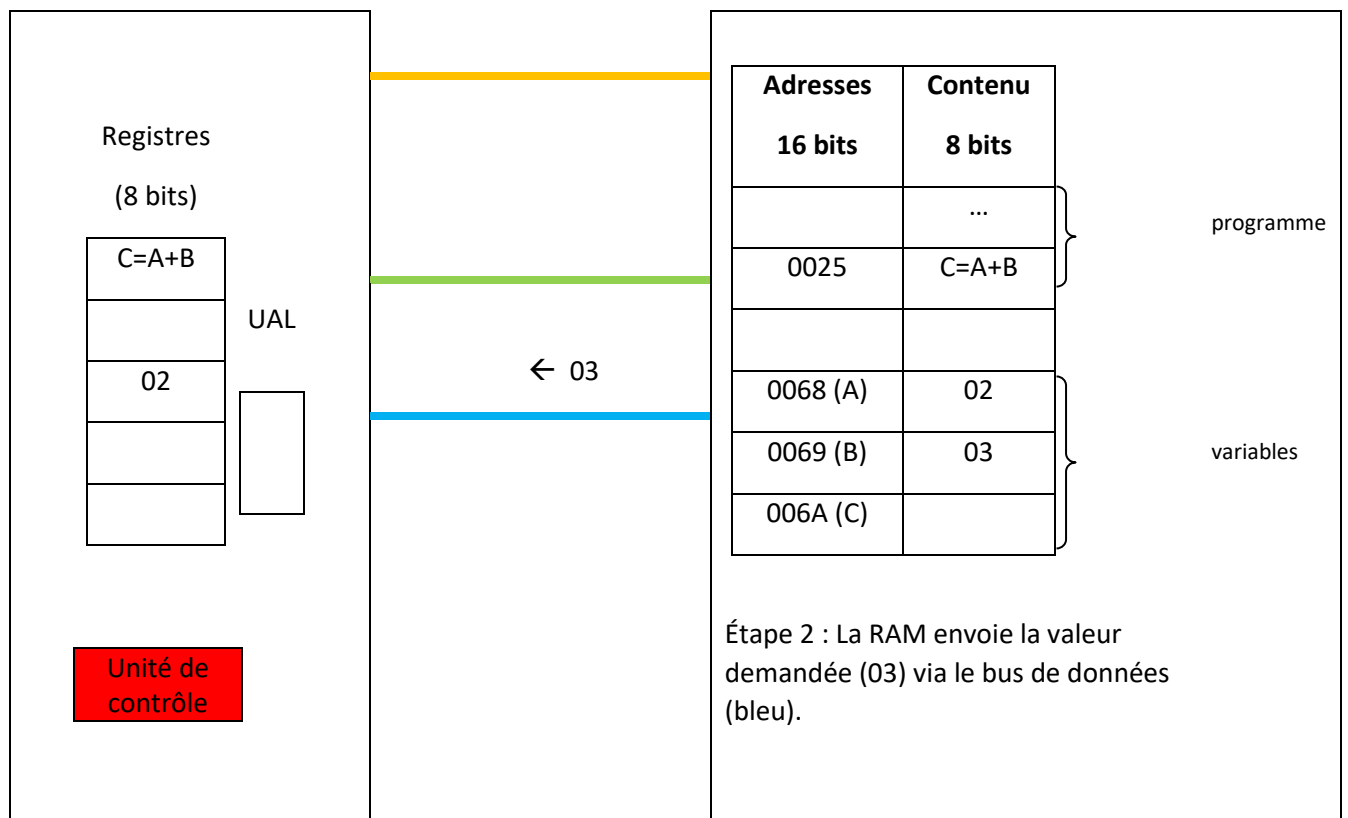
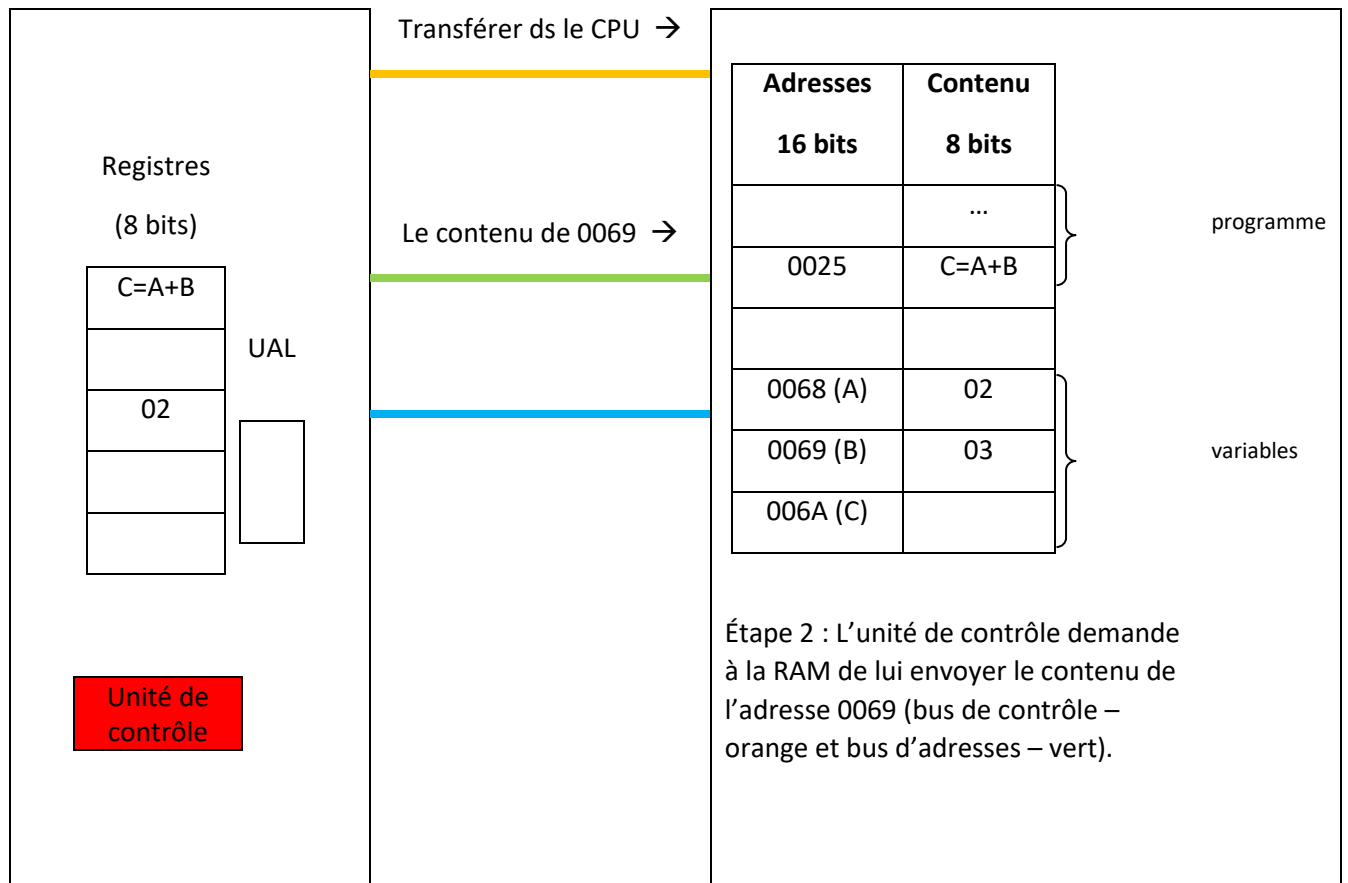
Le bus de données sert à véhiculer les contenus associés aux adresses (ici, le contenu de la RAM)

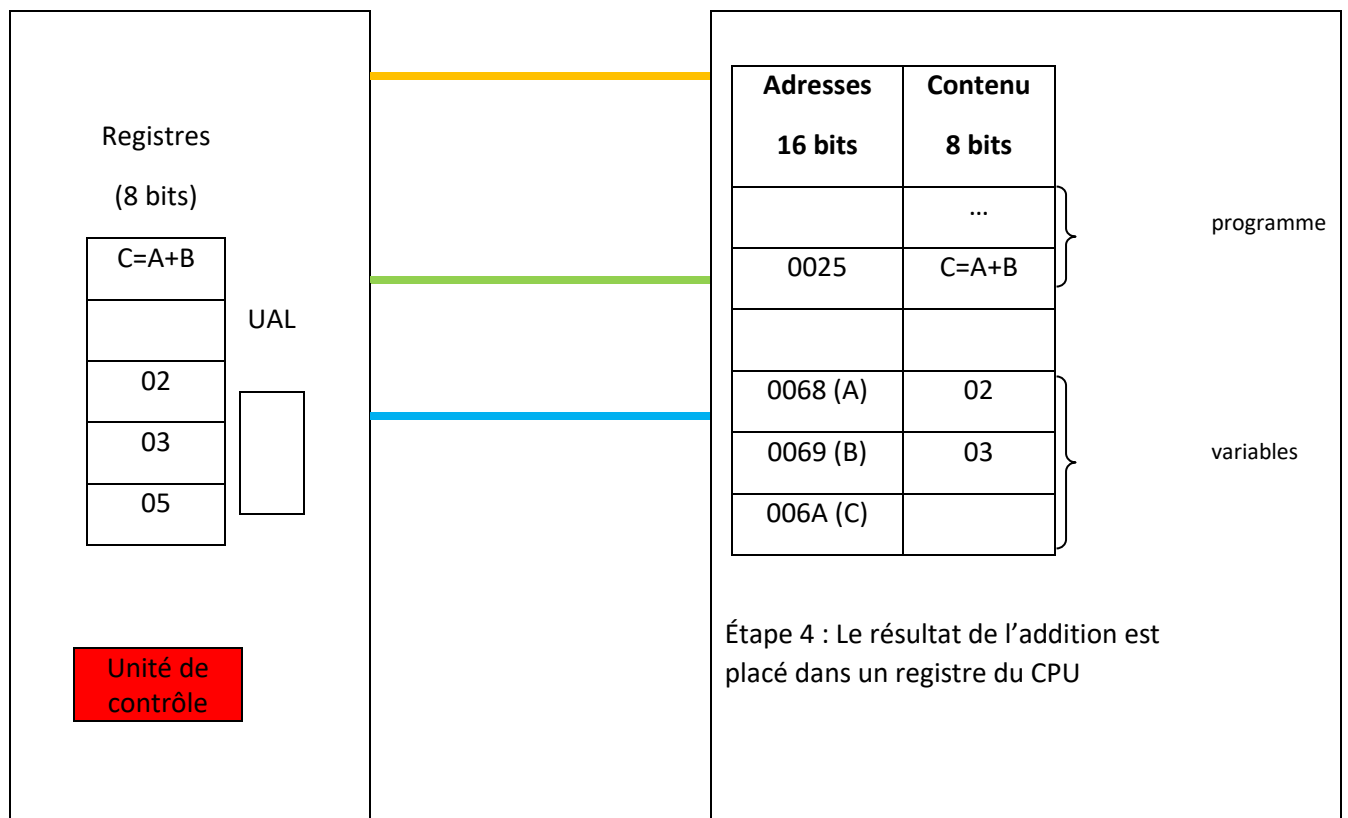
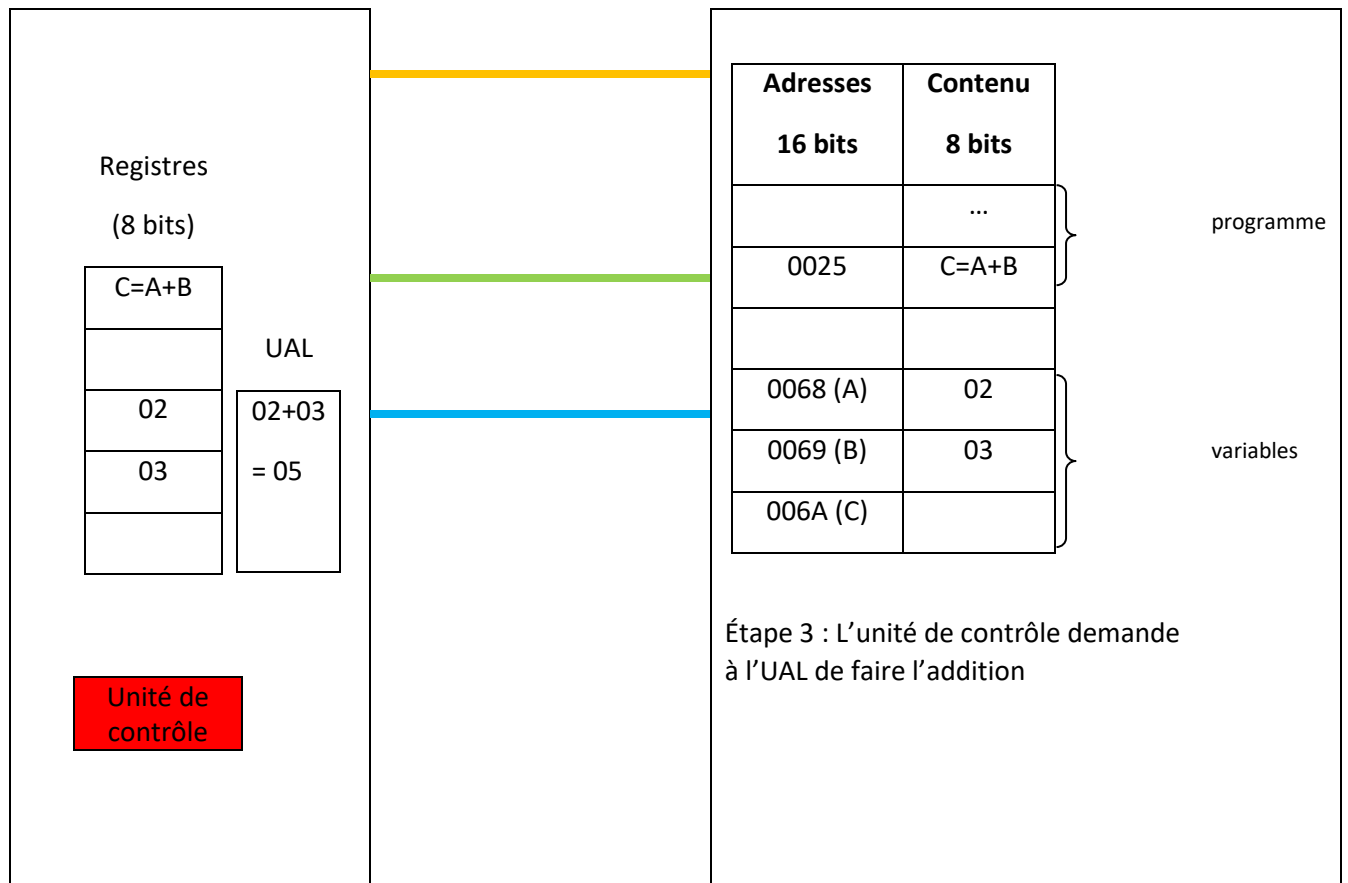
Si on indique les 3 types de bus, lien entre le CPU et la RAM devient :

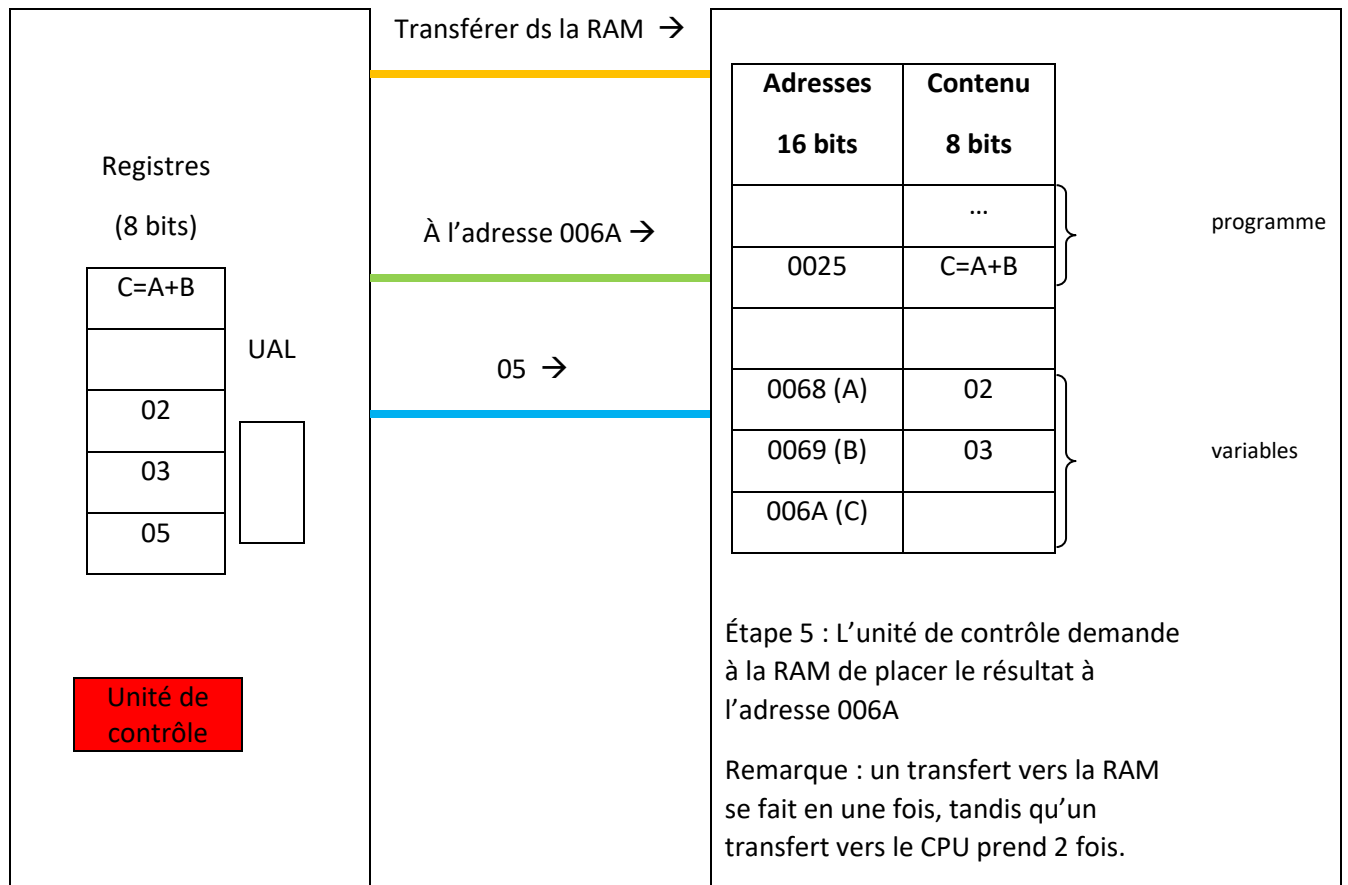


Reprenons l'exemple des pages 3 et 4 avec les 3 types de bus :









Résultat, l'instruction est exécutée :

Adresses 16 bits	Contenu 8 bits
	...
0025	C=A+B
0068 (A)	02
0069 (B)	03
006A (C)	05

programme

variables

En binaire maintenant :

Un bus est un ensemble de fils de cuivre (1 bit par fil). Un bus de 8 bits a donc 8 fils.

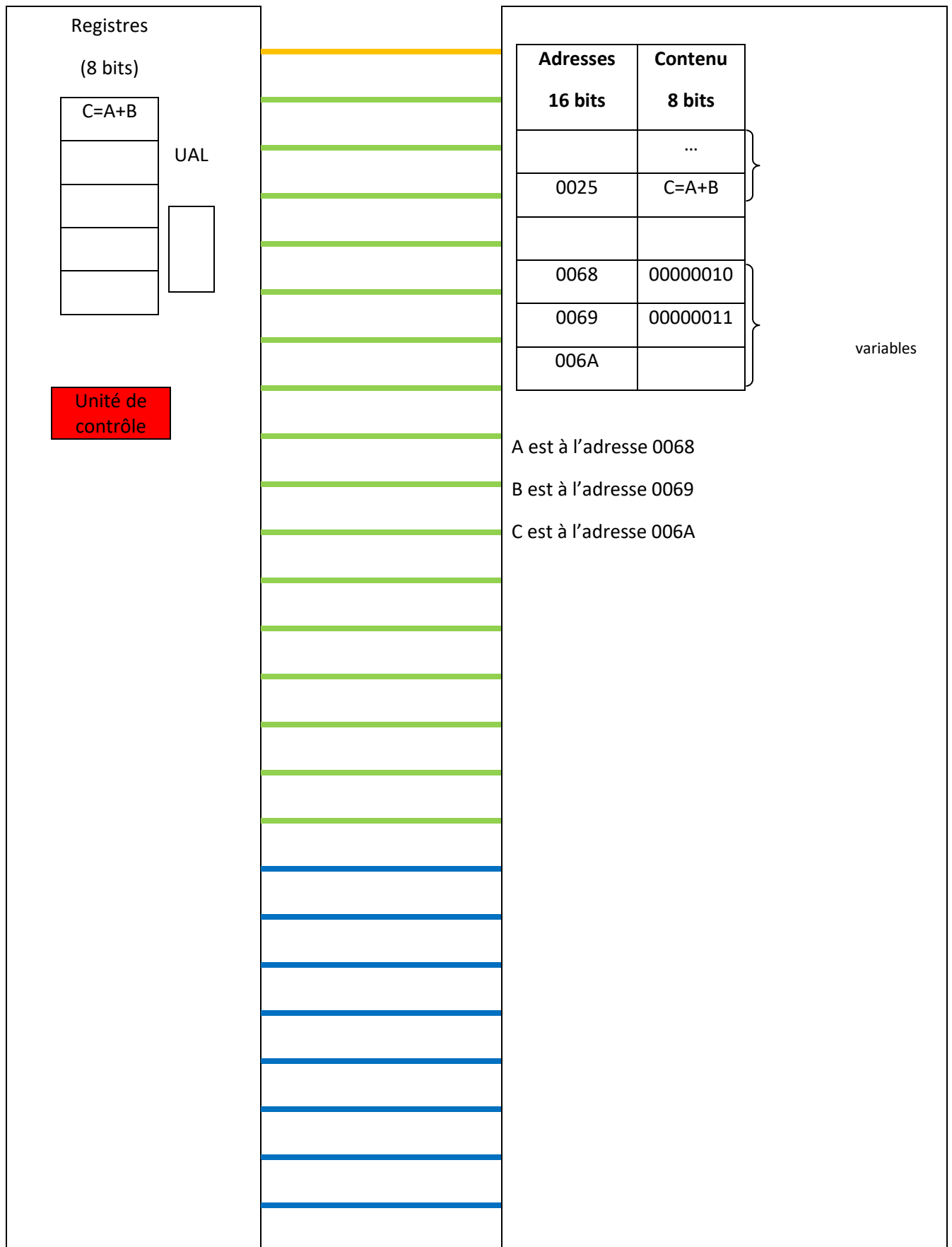
Pour notre exemple :

Le bus d'adresses (bus vert) sert à transporter des adresses de 16 bits. Il aura donc 16 fils.

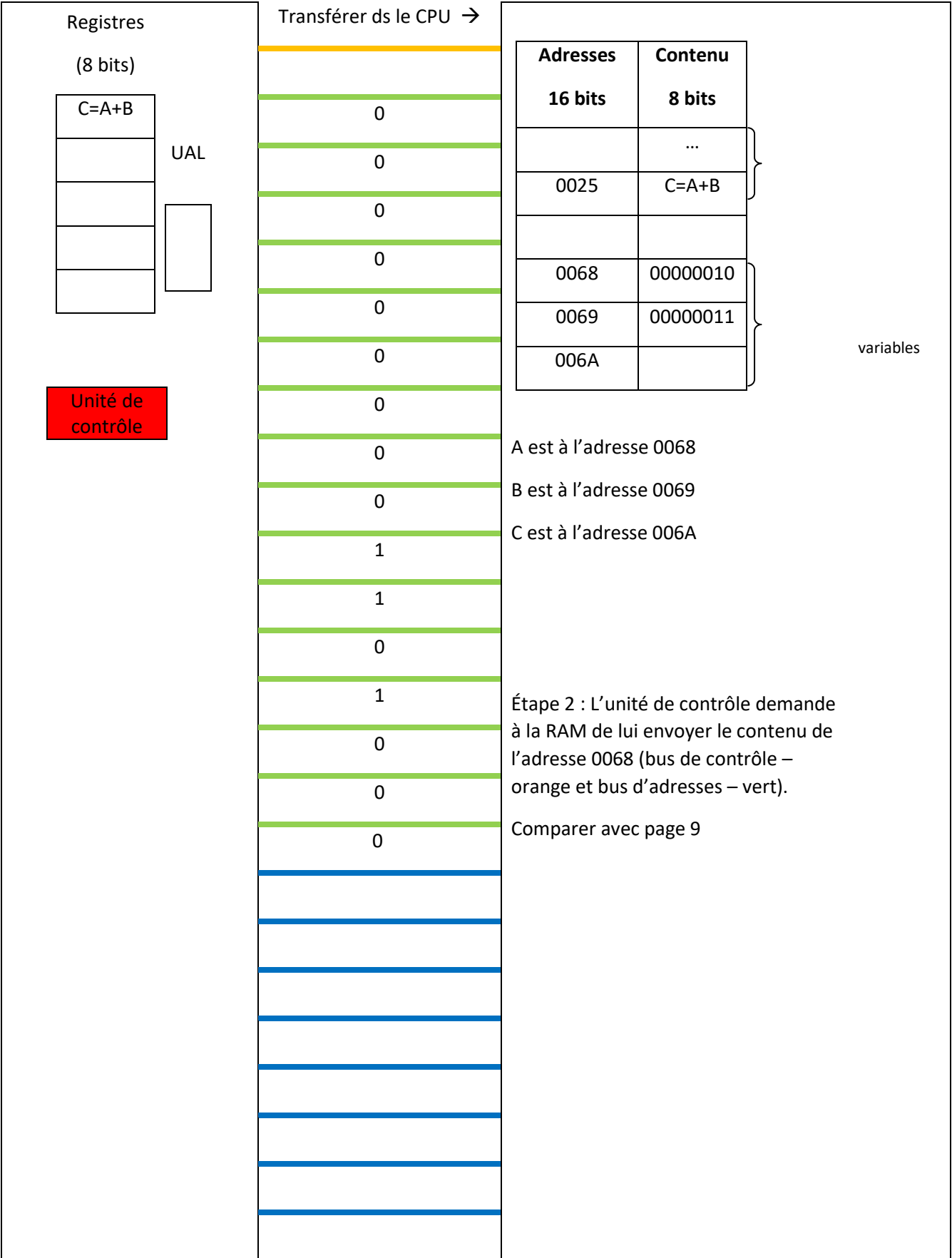
Le bus de données (bus bleu) sert à transporter des octets. Il aura donc 8 bits.

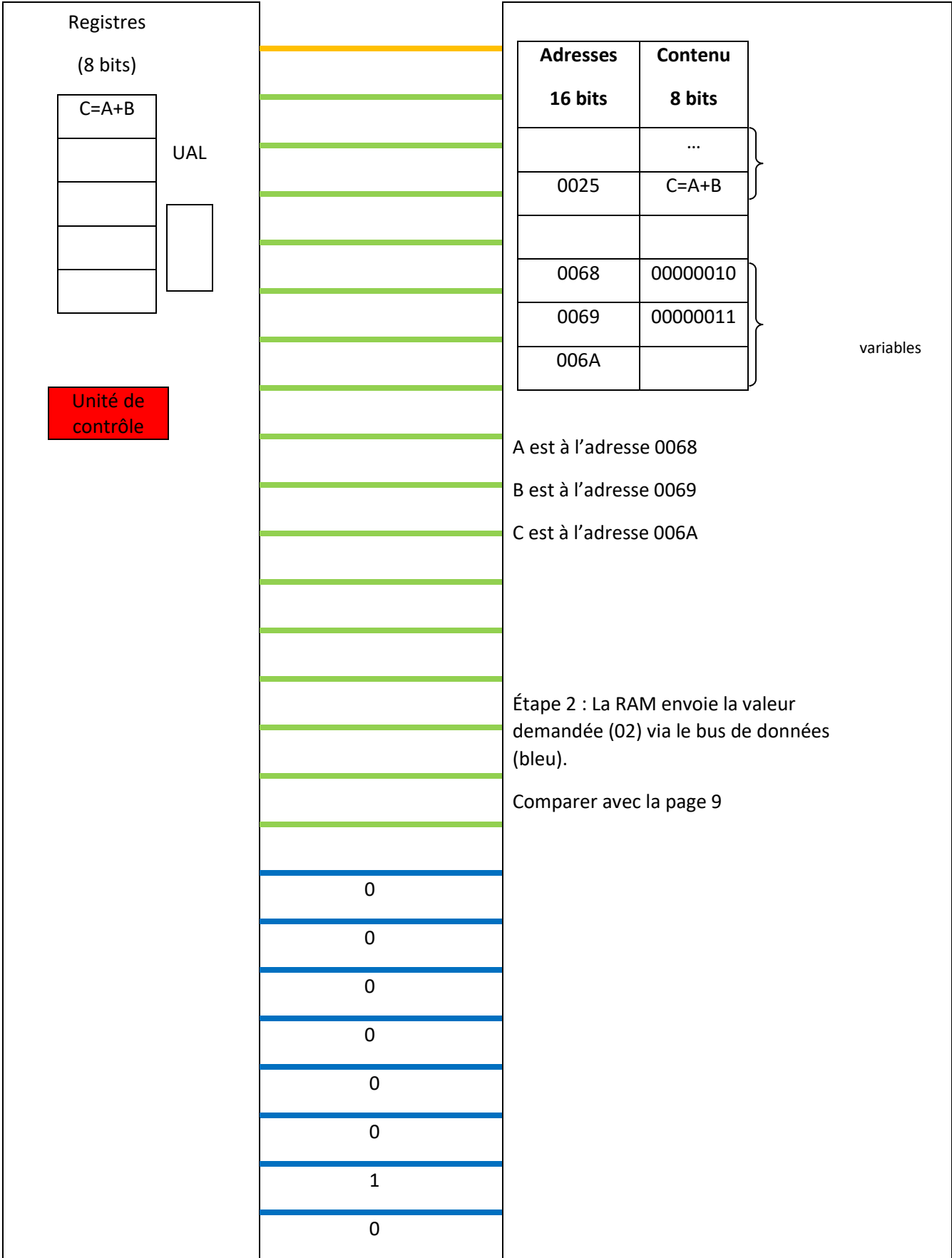
Nous ne parlerons pas du bus de contrôle (ou très peu...)

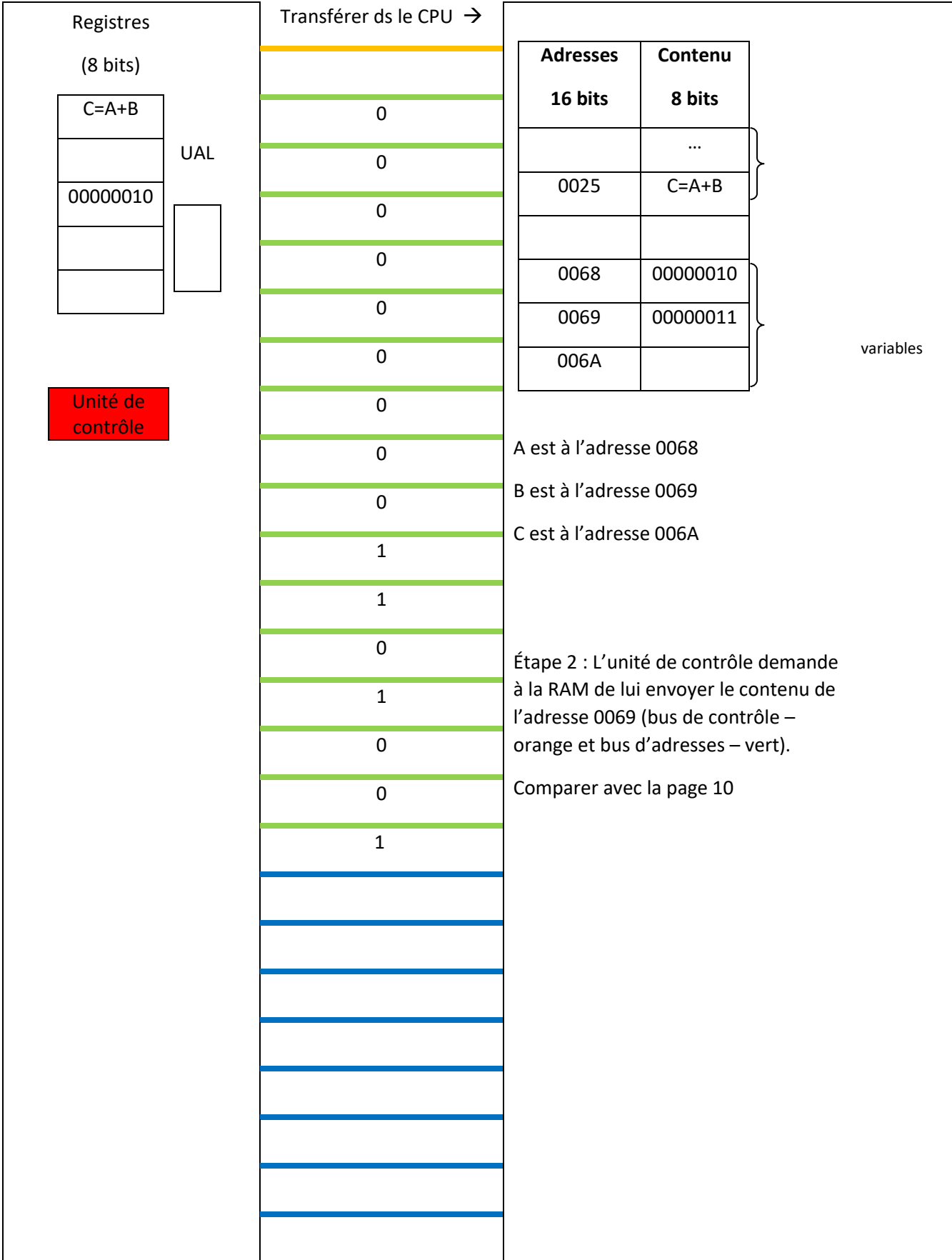
Si on complète le schéma de la page précédente avec les bus détaillés, on a :

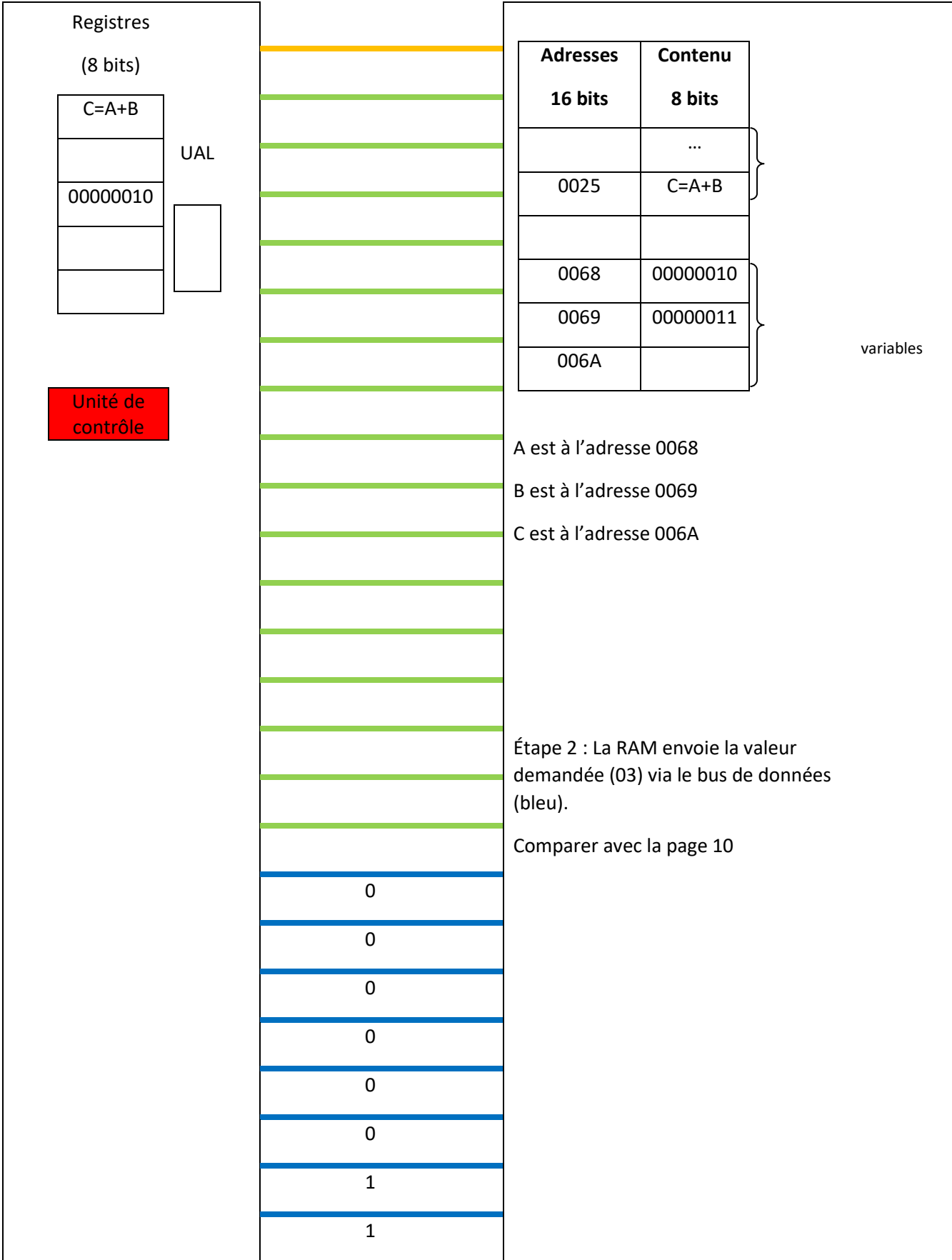


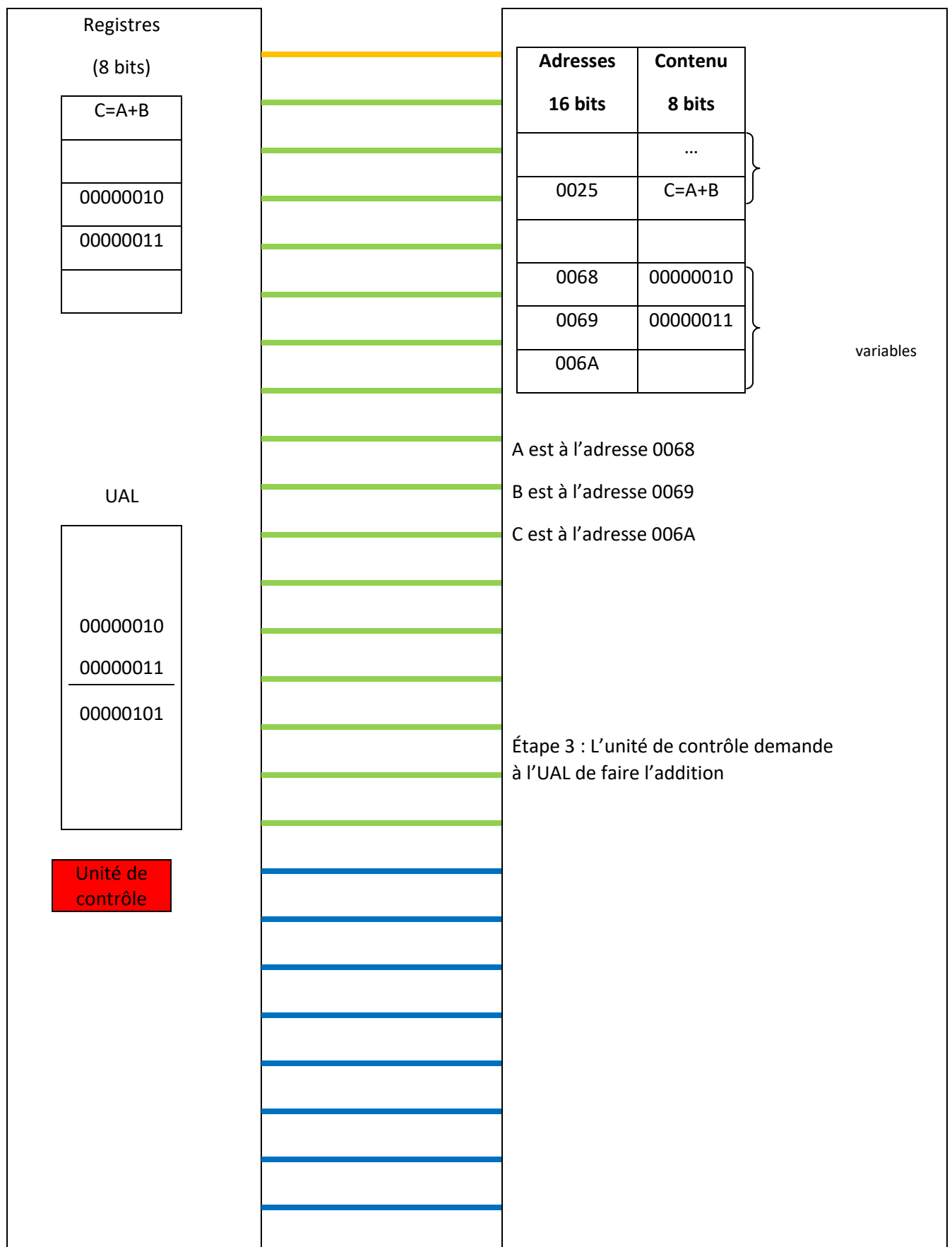
L'exemple de l'exécution de l'instruction C=A=B devient : (à partir de l'étape 2)

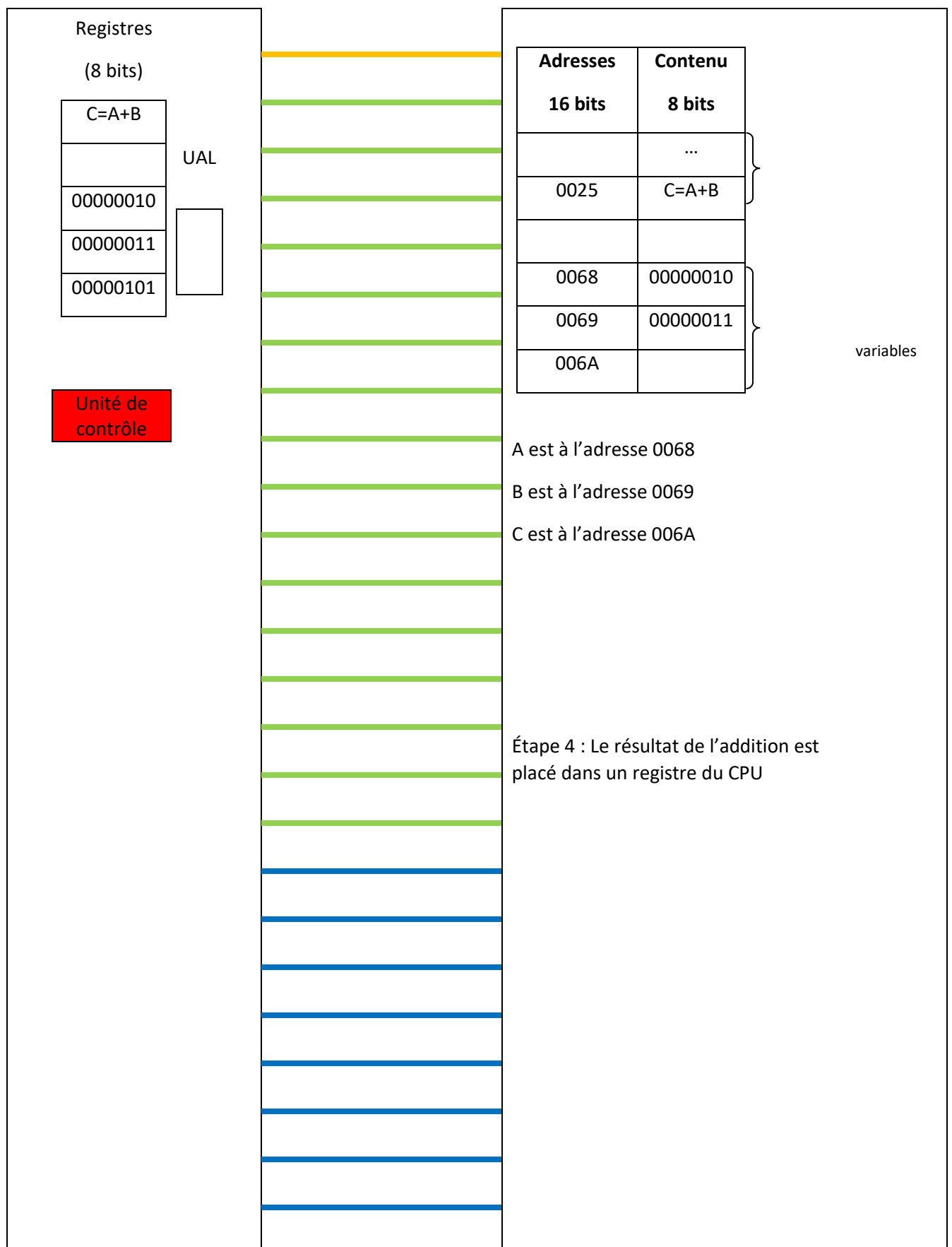


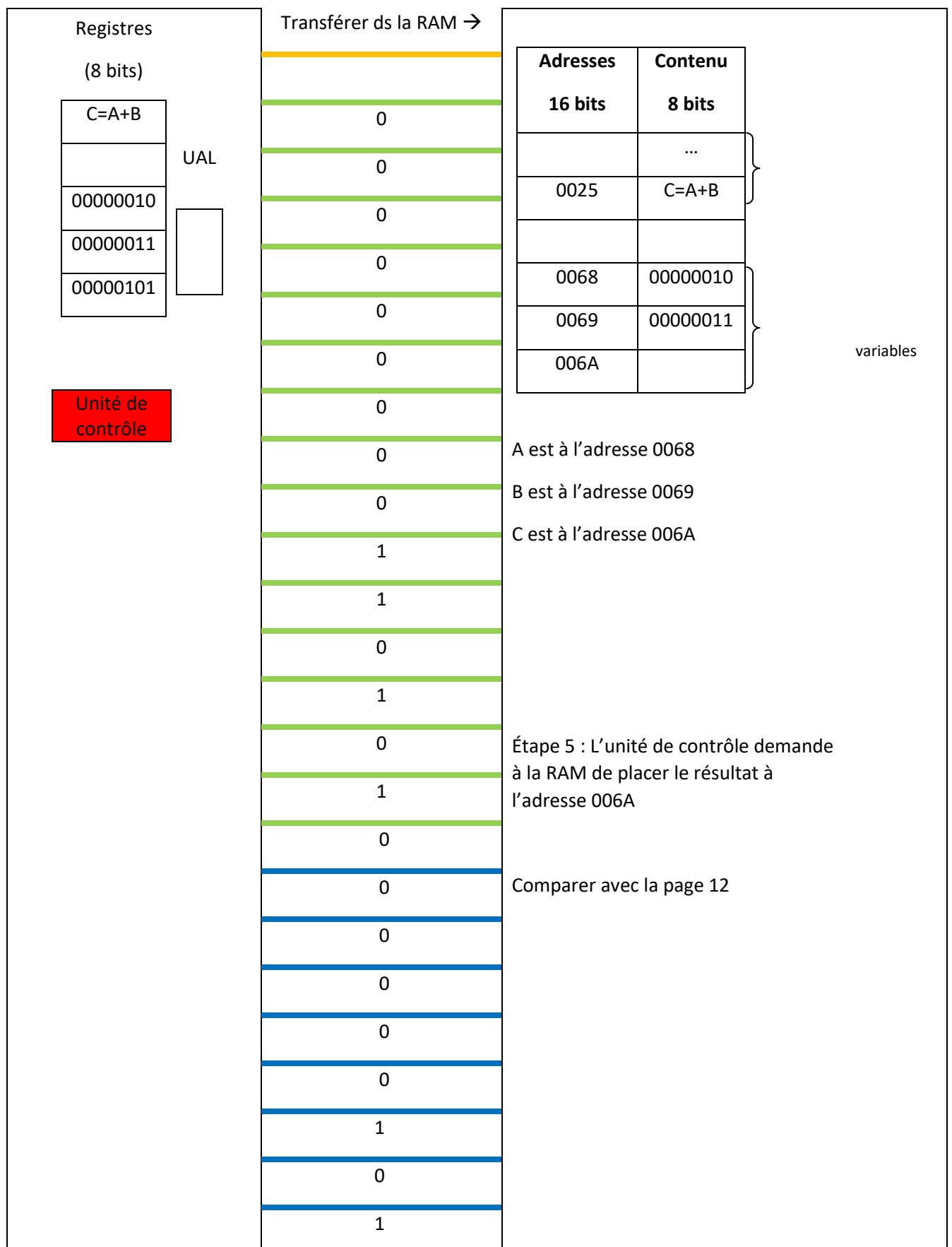


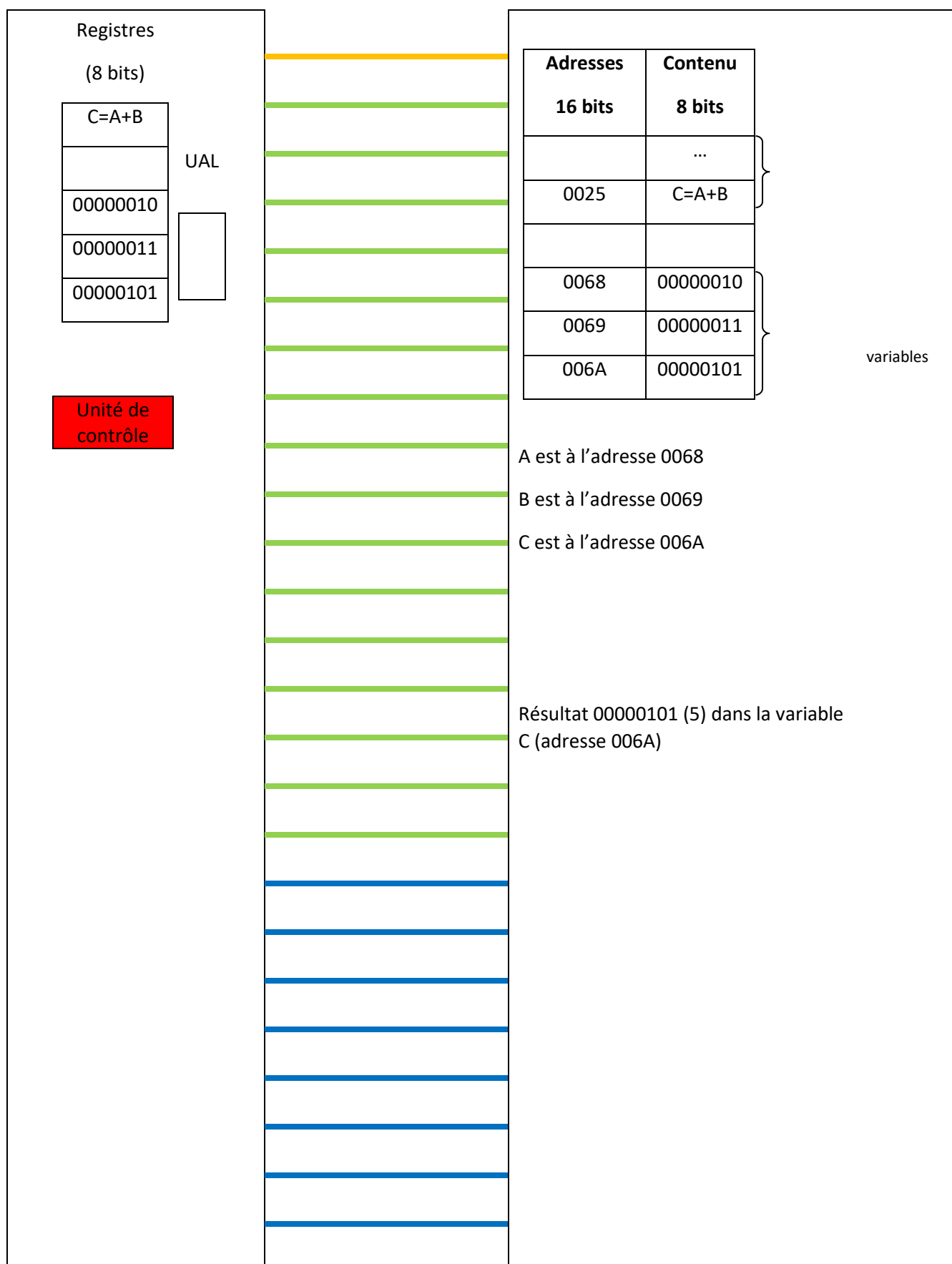












Exercice : écrire ce qui se passe lors des étapes de l'exécution de l'instruction suivante : $S = N1 + N2 + N3$

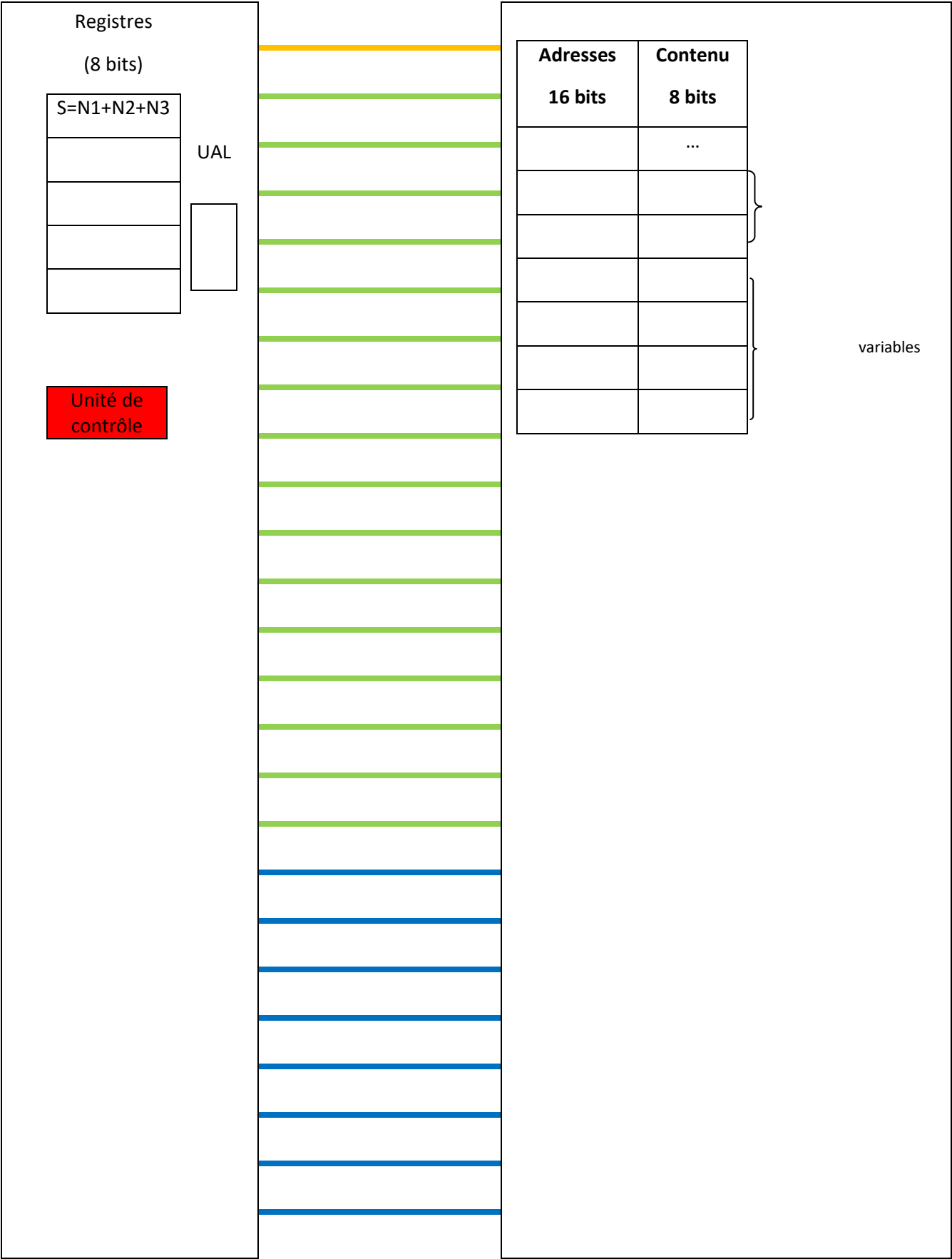
Cette instruction est placée en RAM à l'adresse 1A00.

Les variables N1, N2, N3 et S prennent chacun 1 octet signé et sont placées en RAM respectivement aux adresses 1A61, 1A62, 1A63 et 1A64.

Les valeurs lues pour N1, N2 et N3 sont respectivement : 12_{10} , -22_{10} et 6_{10}

Commencer à partir de l'étape 2, une fois que l'instruction est dans le CPU.

Faire des copies du gabarit de la page suivante tant que nécessaire...



Solution

