Informatique

Notions fondamentales

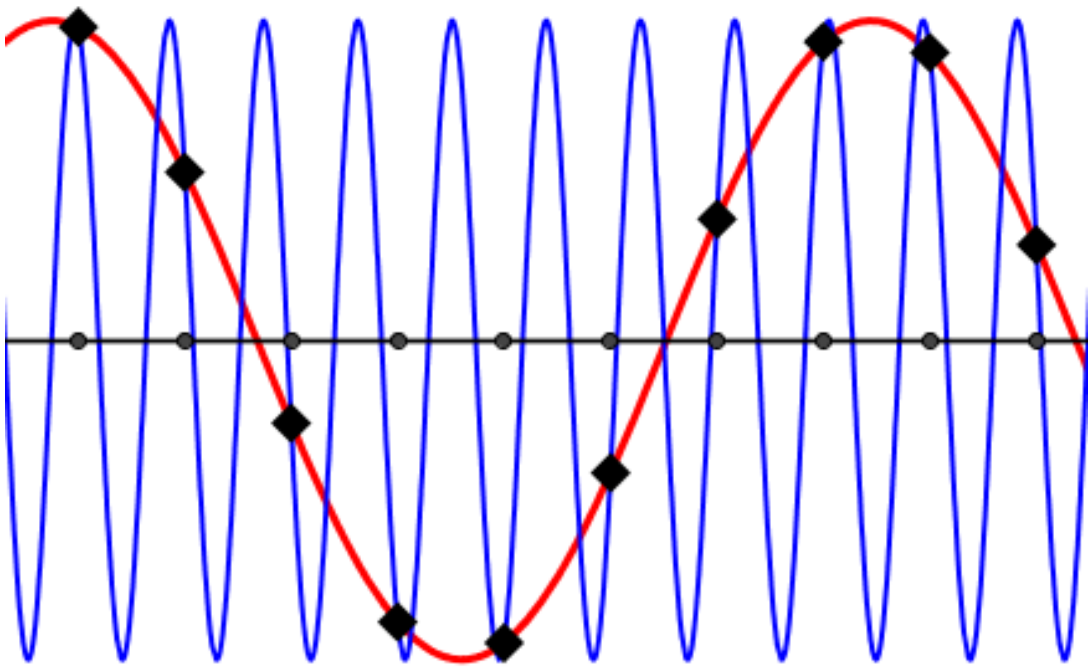
Christian NGUYEN

Département d'informatique
Université de Toulon

Plan

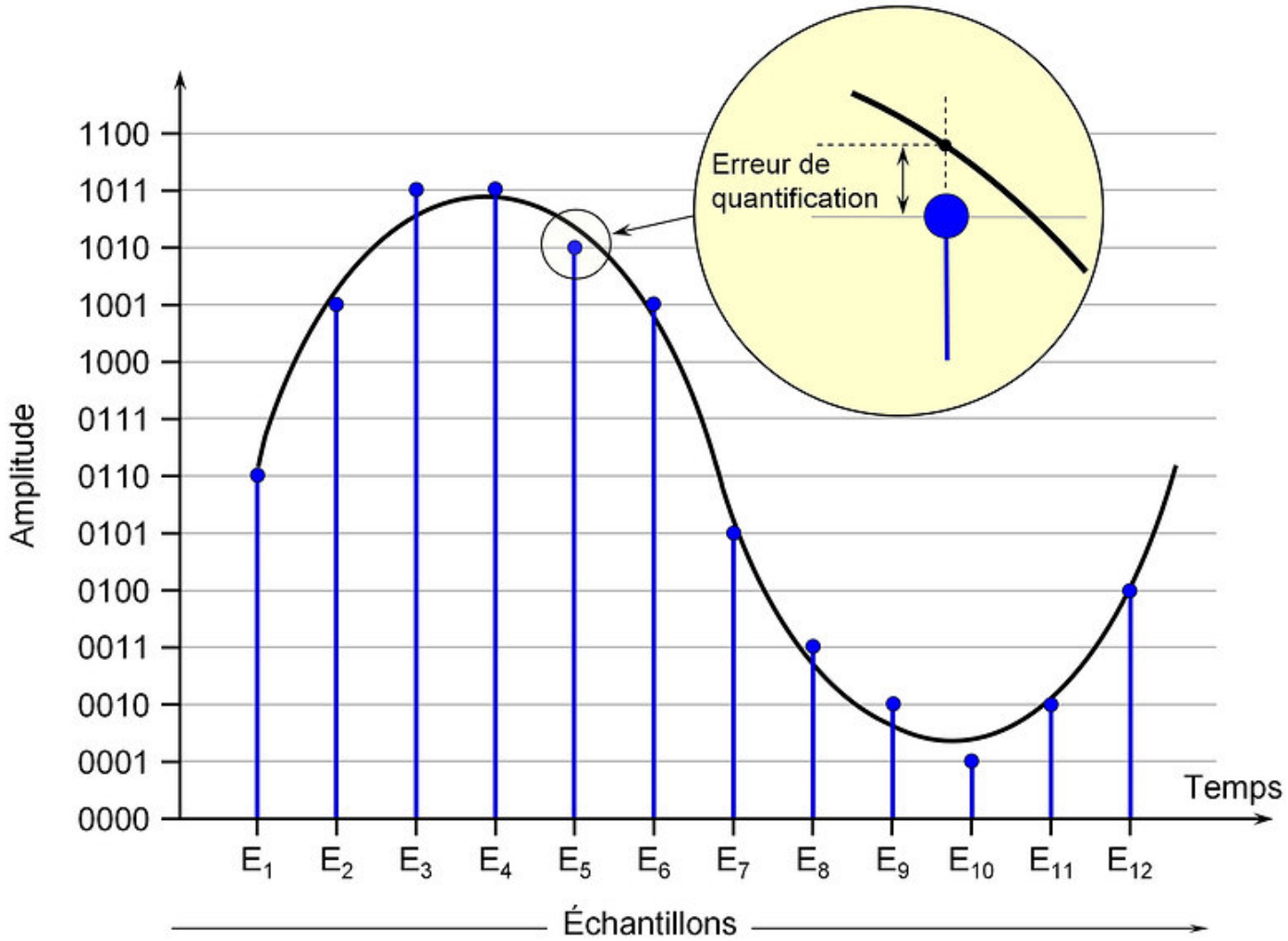
Informatique, science de l'information

Information : ensemble de connaissances réunies sur un sujet déterminé.



numérisation de l'information

Informatique, science de l'information



quantification : échantillonnage et codage

Décidabilité

Logique : étude des règles formelles que doit respecter toute argumentation correcte.

-330 : Aristote, syllogisme (exemple : « Tous les hommes sont mortels, or Socrate est un homme ; donc Socrate est mortel »).

1903 : Russell, fondement des mathématiques (exemple de paradoxe : « Je mens », Épiménide -556)).

1928 : Hilbert, « mécanisation » des mathématiques.

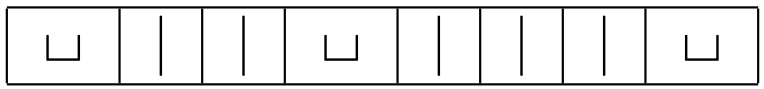
1931 : Gödel, existence de vérités mathématiques impossible à démontrer par une suite d'opérations logiques (décidabilité logique).

1936 : Turing, décidabilité en arithmétique.

Machine de Turing

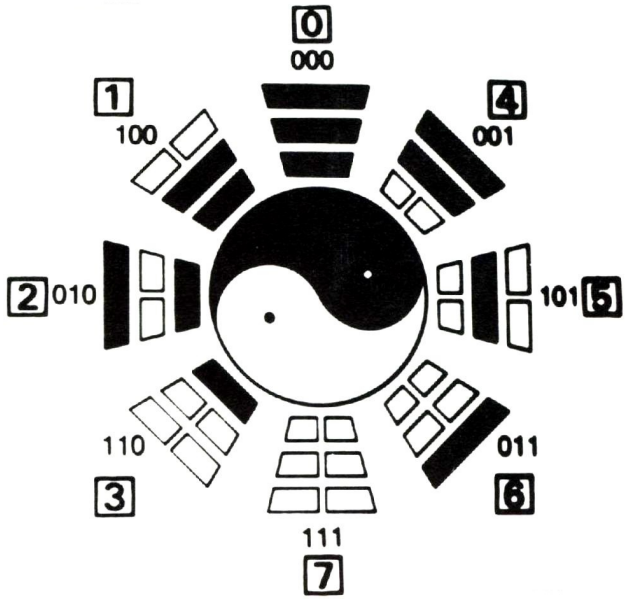
Etats	vide (␣)	bâton (|)
1	avancer	état 2
2	écrire et état 3	avancer
3	reculer et état 4	avancer
4	arrêt	effacer

Table de décision



ruban initial

Système de numération binaire



Chine -3000, octogone à trigramme

1699 : Leibniz, opérations arithmétiques en binaire.

1854 : Boole, un processus logique est décomposable en opérations logiques appliquées sur *deux états*.

1938 : Shannon fait le parallèle entre les circuits électriques et l'algèbre booléenne et définit le bit.

Unités

Unité	Abréviation	Signification
1 bit (ou logon)	bit	0 ou 1
1 octet	o	8 bits
1 kibioctet	Kio	2^{10} o soit 1024 octets
1 mébioctet	Mio	2^{20} o soit 1024 Kio
1 gibioctet	Gio	2^{30} o soit 1024 Mio
1 tébioctet	Tio	2^{40} o soit 1024 Gio

norme de la Commission Électrotechnique Internationale (1998)

Données

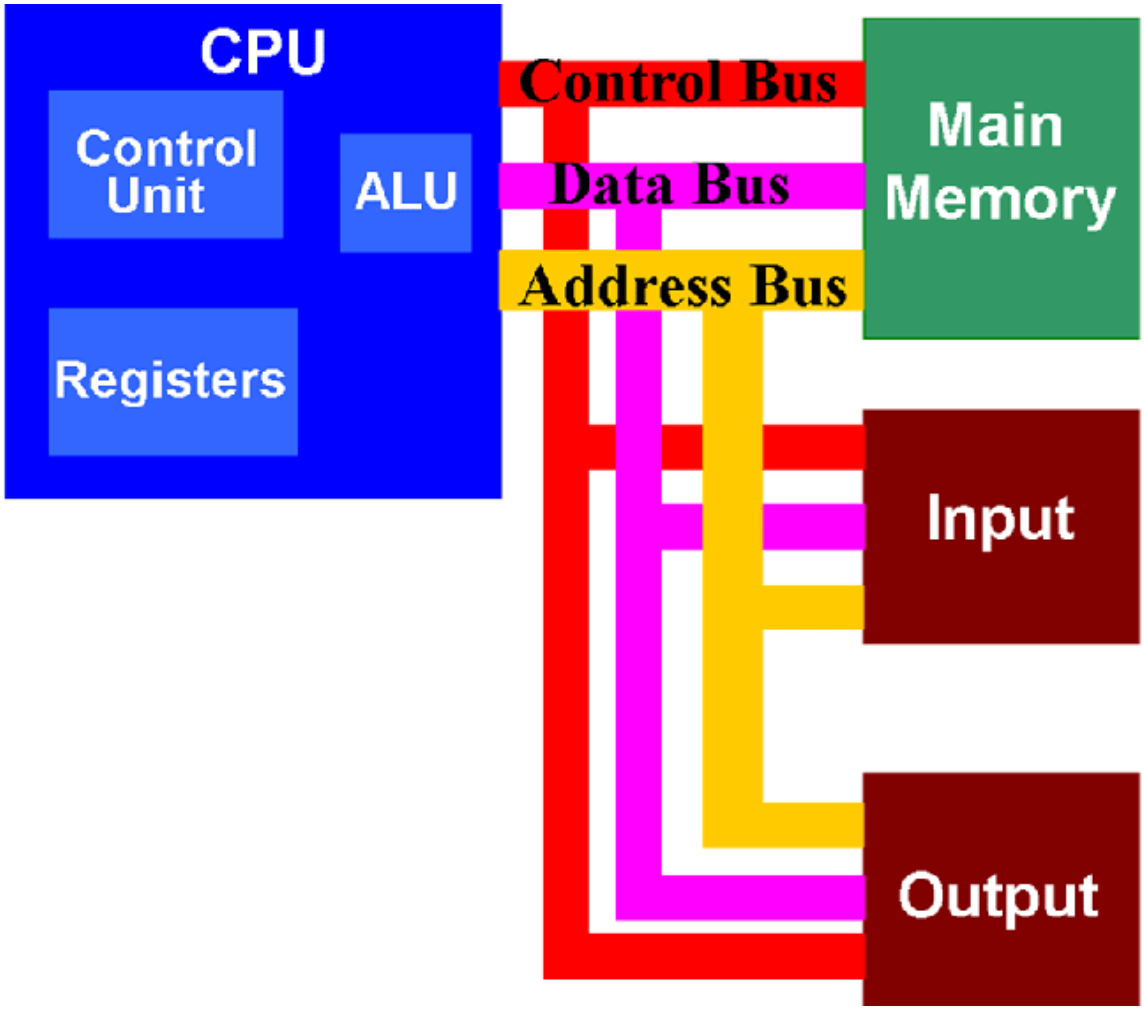
Le traitement d'un volume de données en progression constante nécessite d'optimiser chacune des étapes liées à son traitement :

- variété : donnée structurée ou non structurée,
- volume (plus de données en une journée qu'il n'en a été produit entre les débuts de l'humanité et l'an 2000),
- vélocité (1 ordinateur de bureau typique de 2022 de 1000 gigaFLOPS[1] $\approx$ 88 superordinateurs Deeper Blue de 1997 de 11,4 gigaFLOPS).

1. FLoating-point Operations Per Second

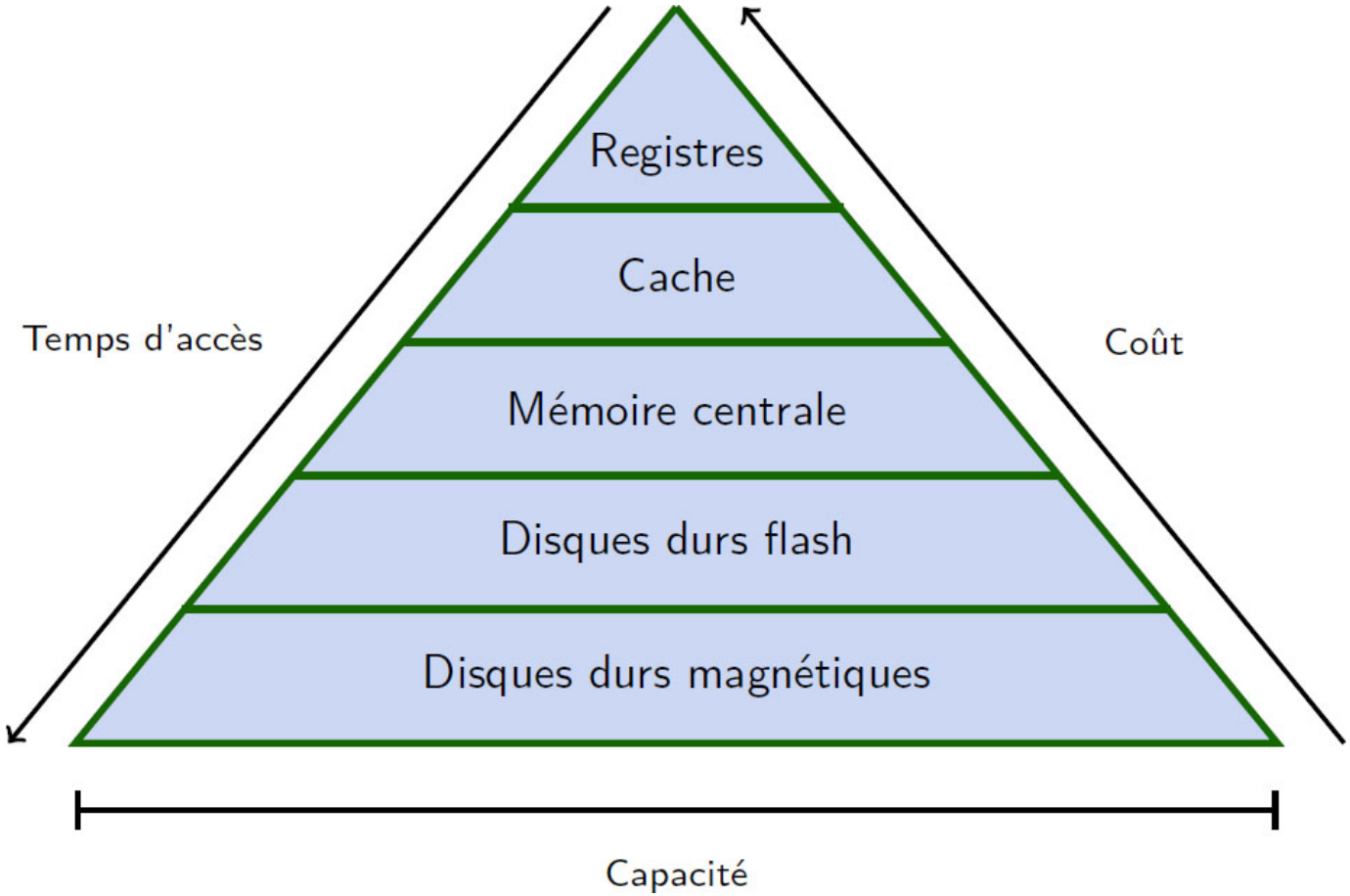
Plan

Organisation d'un ordinateur



architecture de von Neumann

Mémoires



propriétés des mémoires

Classes des jeux d'instructions

instruction	opérande	commentaire
PUSH	B	empiler B
PUSH	C	empiler C
ADD		dépiler Cpuis B, calculer $B + C$, empiler
POP	A	dépiler et mémoriser dans A

architecture zéro adresse

instruction	opérande	commentaire
LOAD	B	charger B dans l'ACC
ADD	C	ajouter C au contenu de l'ACC
STORE	A	mémoriser le contenu de l'ACC dans A

architecture à accumulateur

Classes des jeux d'instructions

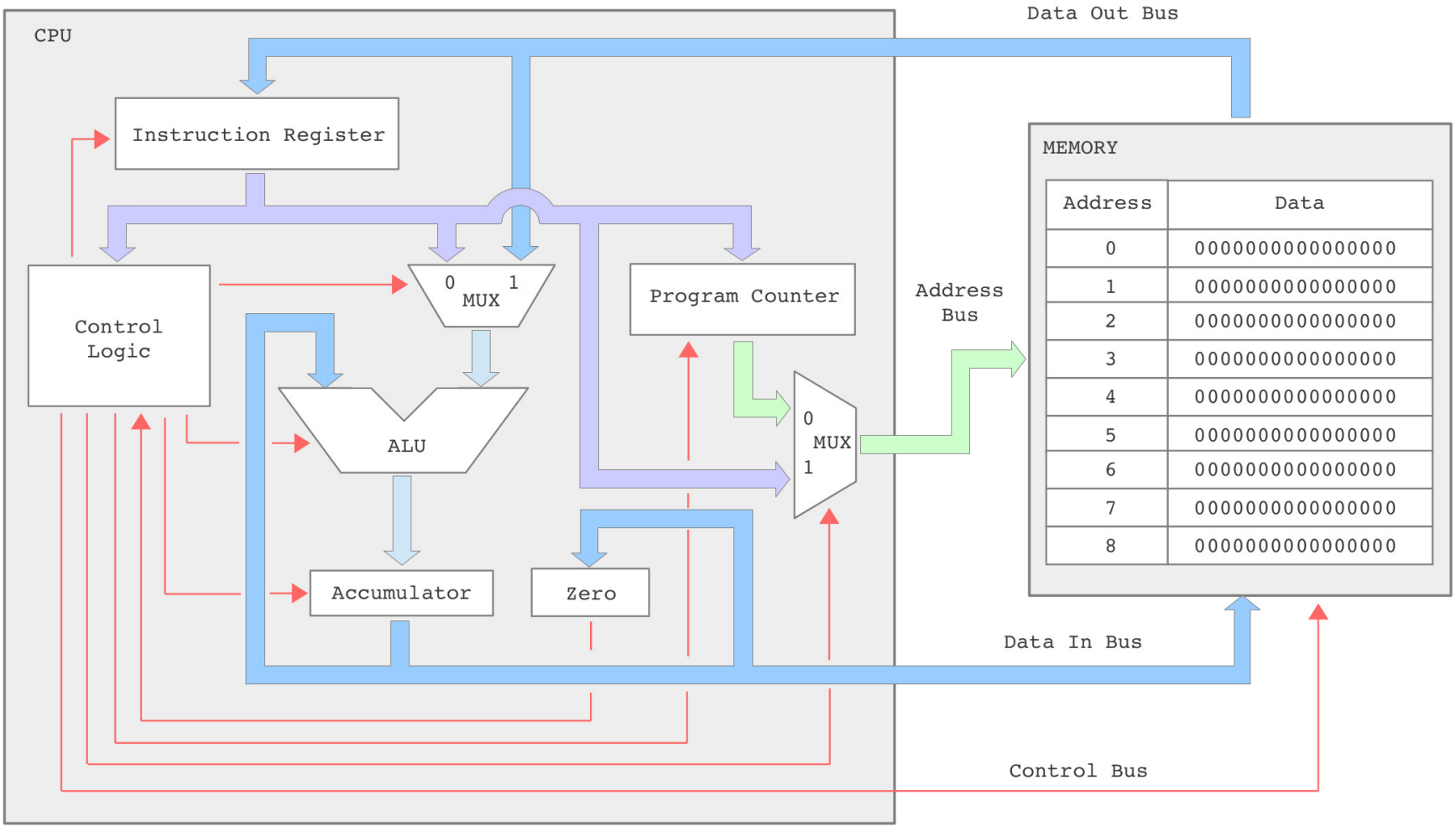
instruction	opérandes	commentaire
LOAD	R_0, B	charger B dans le registre R_0
ADD	R_1, R_0, C	additionner C à R_0 et mémoriser le résultat dans R_1
STORE	R_1, A	mémoriser le contenu du registre R_1 dans A

architecture registre-mémoire

instruction	opérandes	commentaire
LOAD	R_0, B	charger B dans le registre R_0
LOAD	R_1, C	charger C dans le registre R_1
ADD	R_2, R_1, R_0	additionner R_1 à R_0, mémoriser le résultat dans R_2
STORE	R_2, A	mémoriser le contenu du registre R_2 dans A

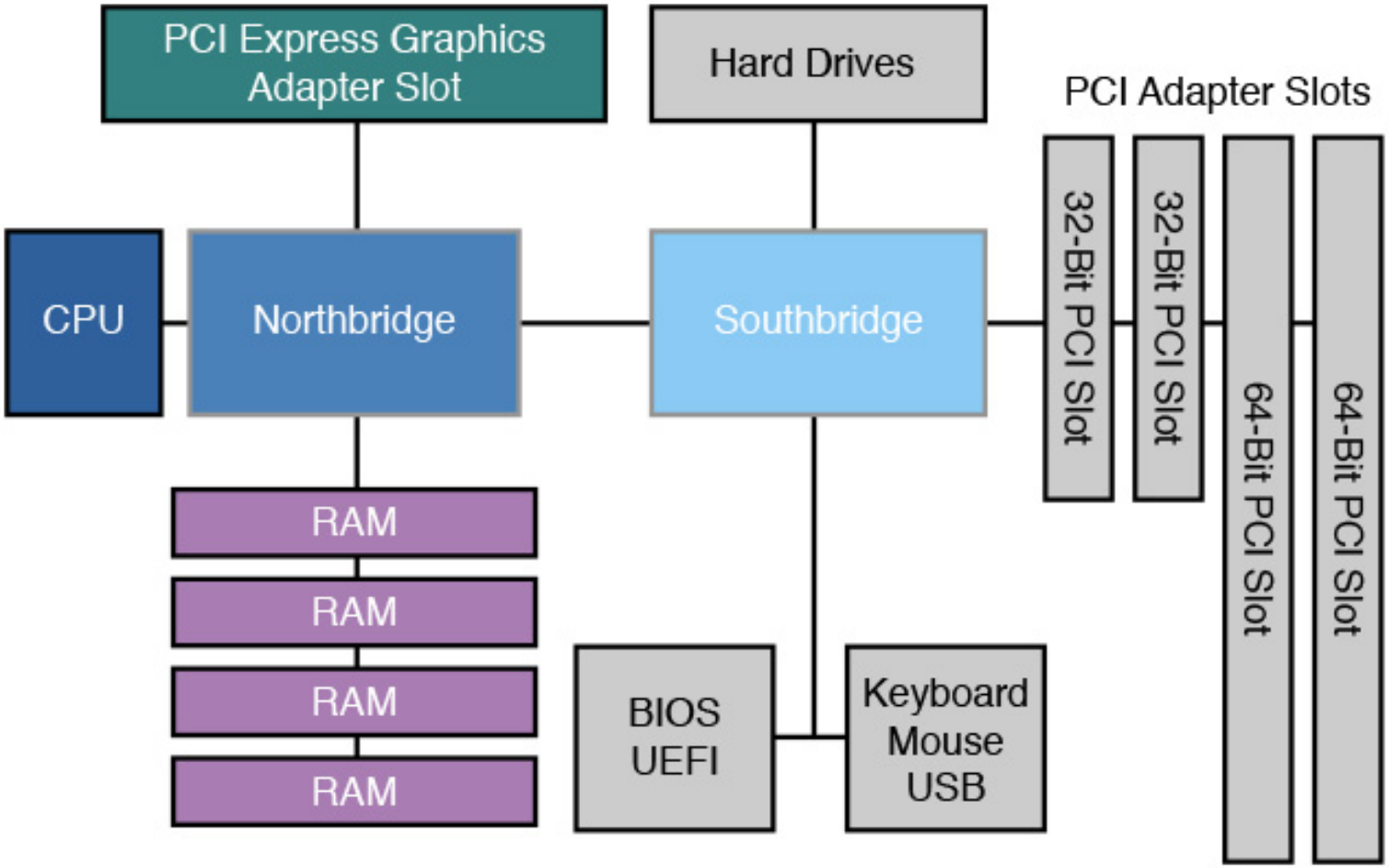
architecture registre-registre

Rôle et fonctionnement du CPU



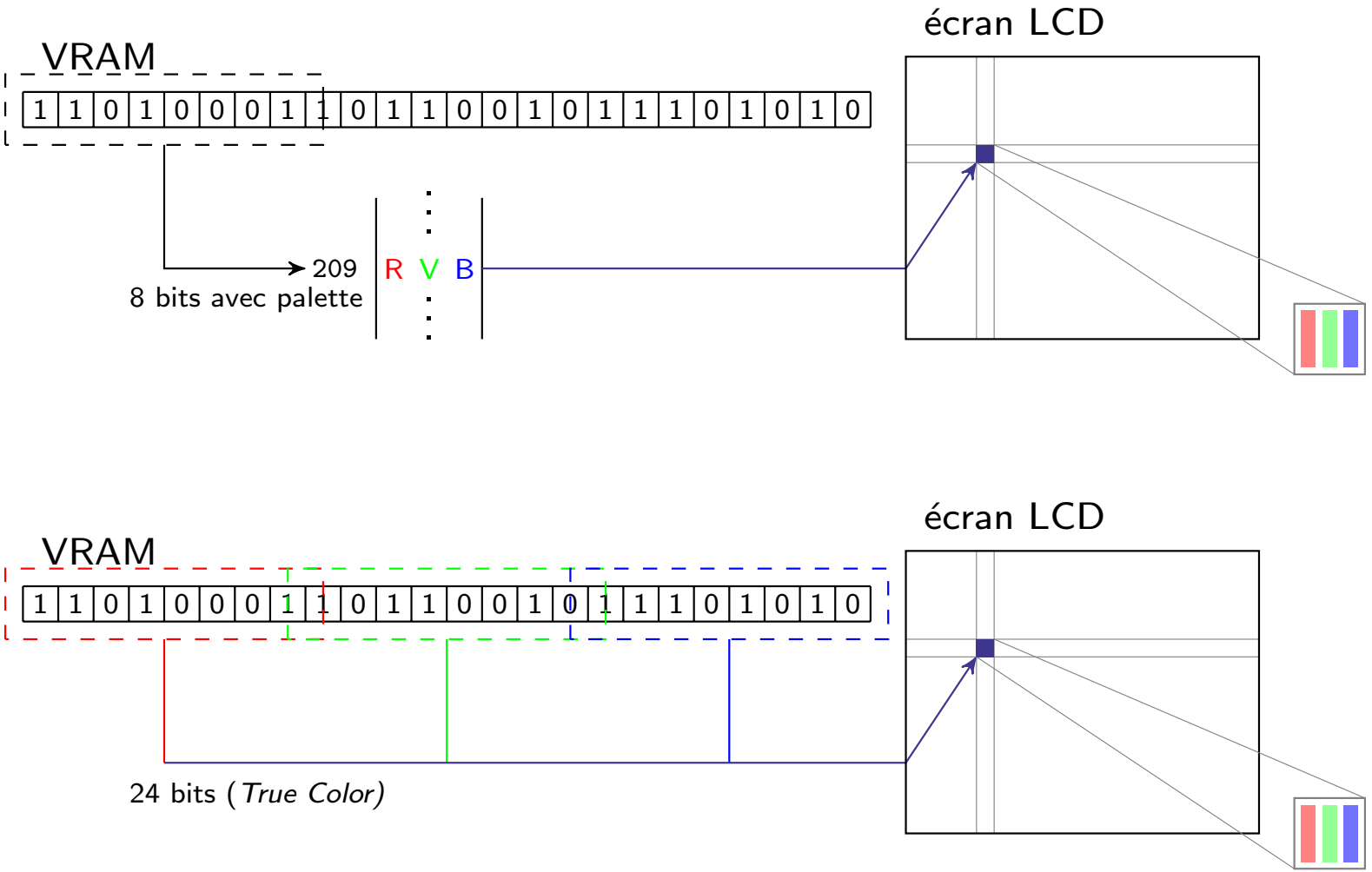
architecture simplifiée du CPU

Bus



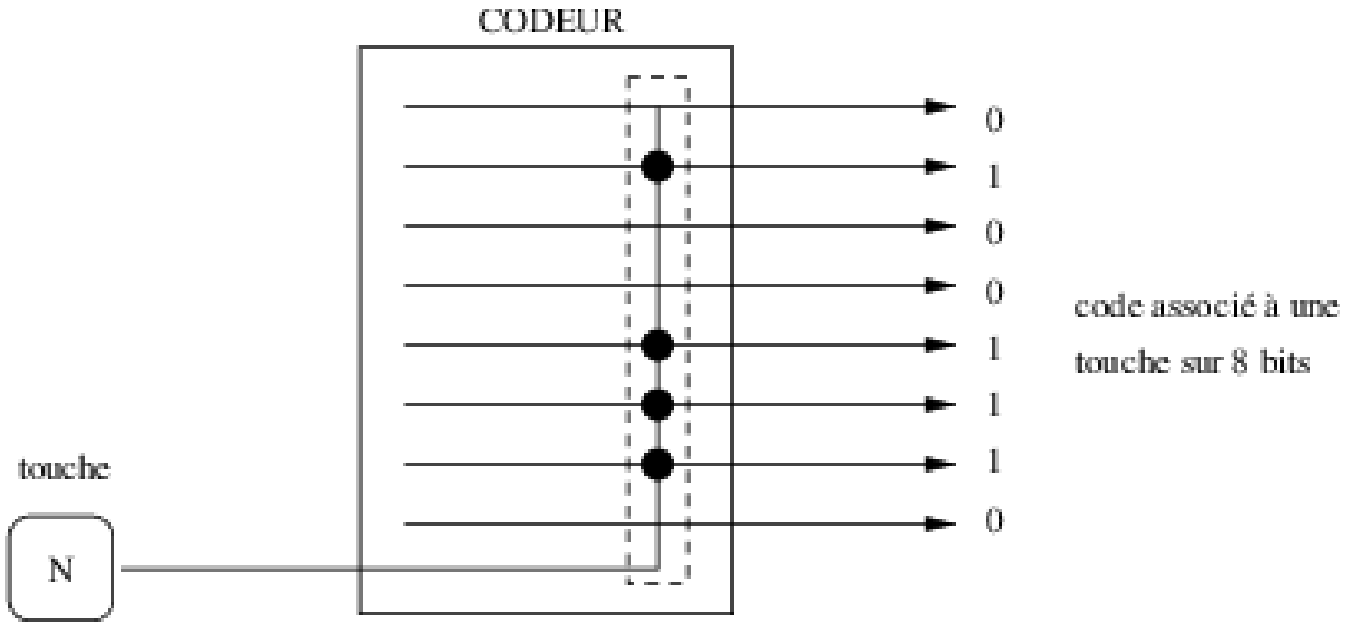
différents canaux de données

Écrans



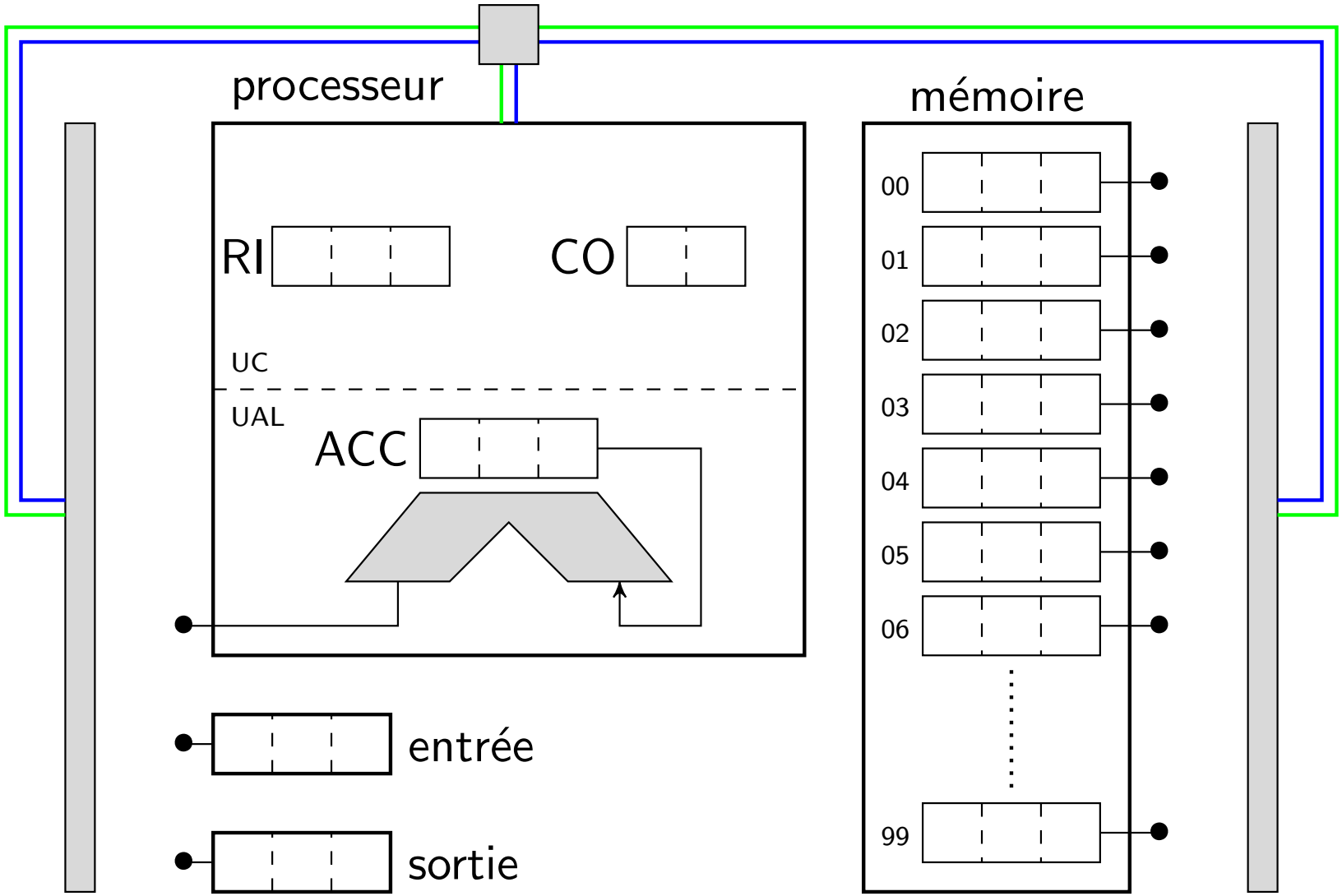
affichage d'un pixel à l'écran

Claviers

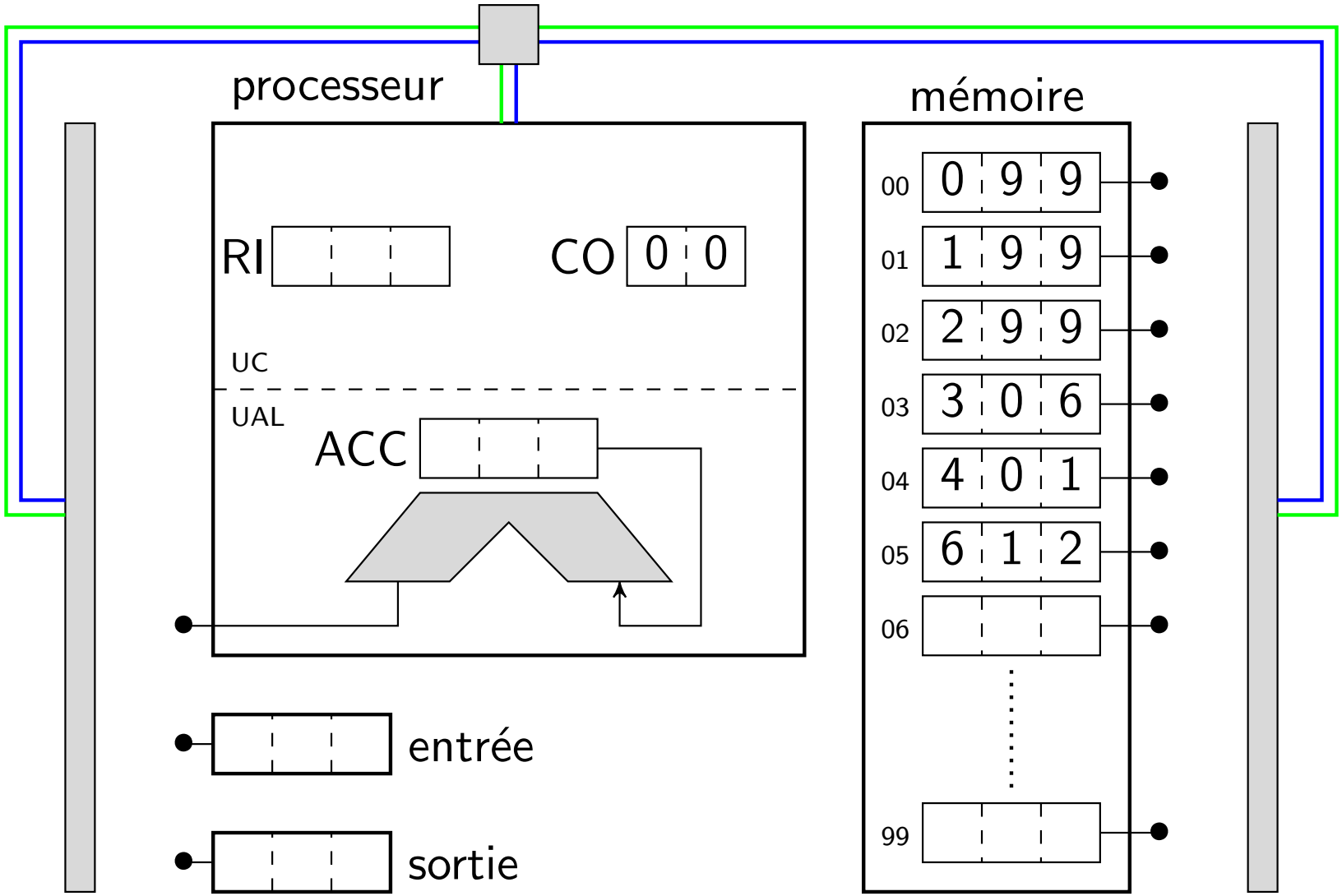


principe de fonctionnement d'un clavier

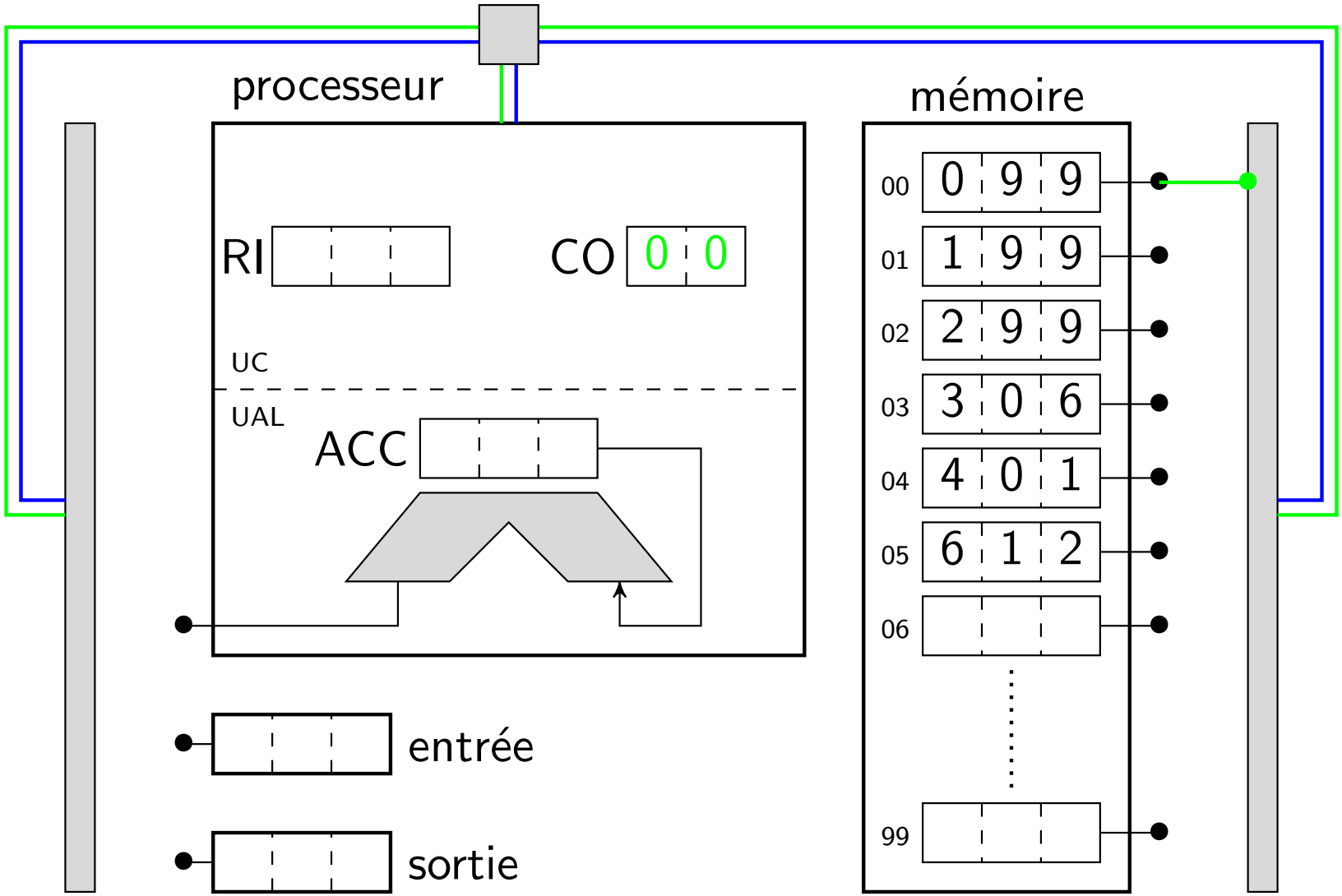
Ordinapoche



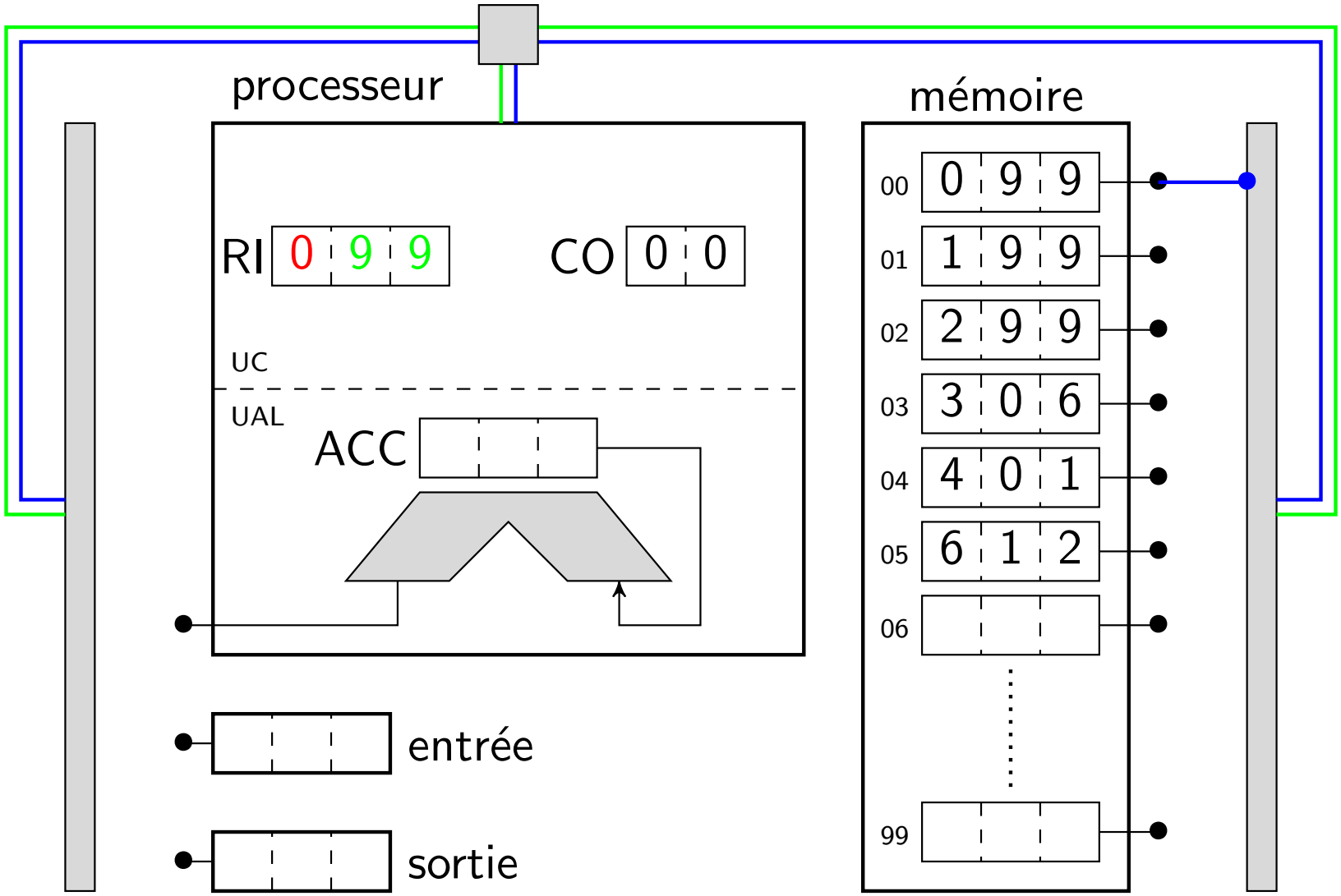
Ordinapoche - input (code 0, assembleur INP)



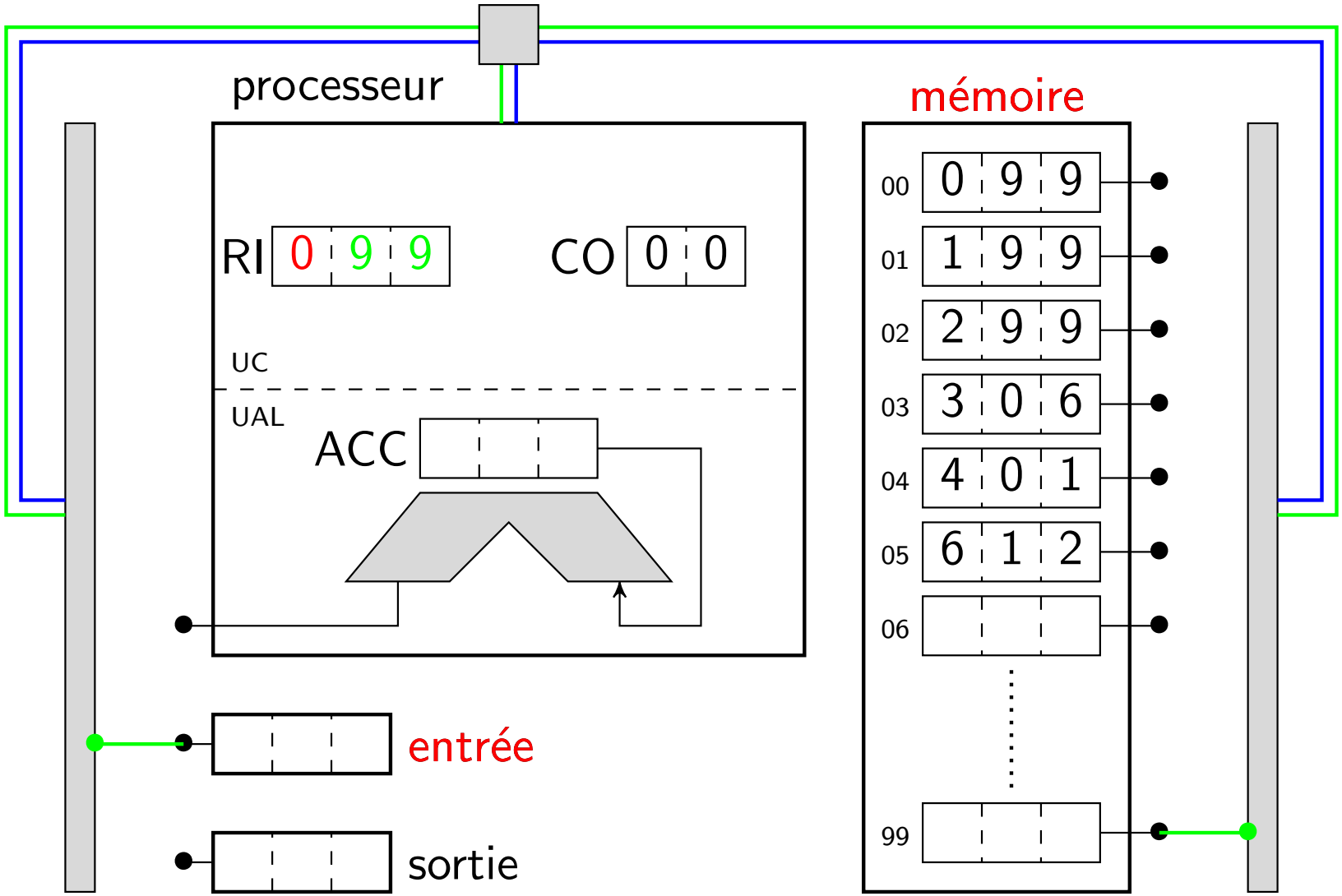
Ordinapoche - input (code 0, assembleur INP)



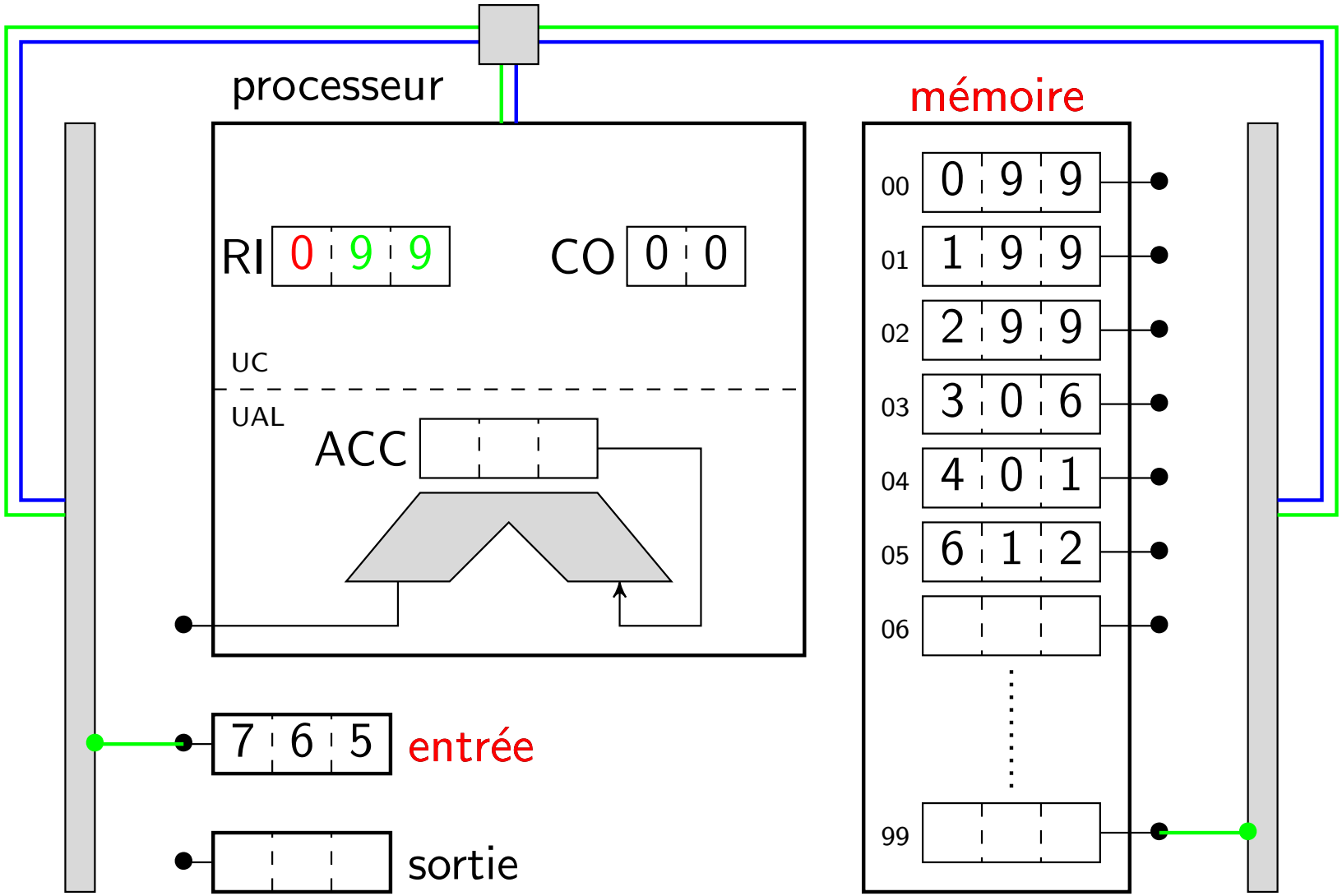
Ordinapoche - input (code 0, assembleur INP)



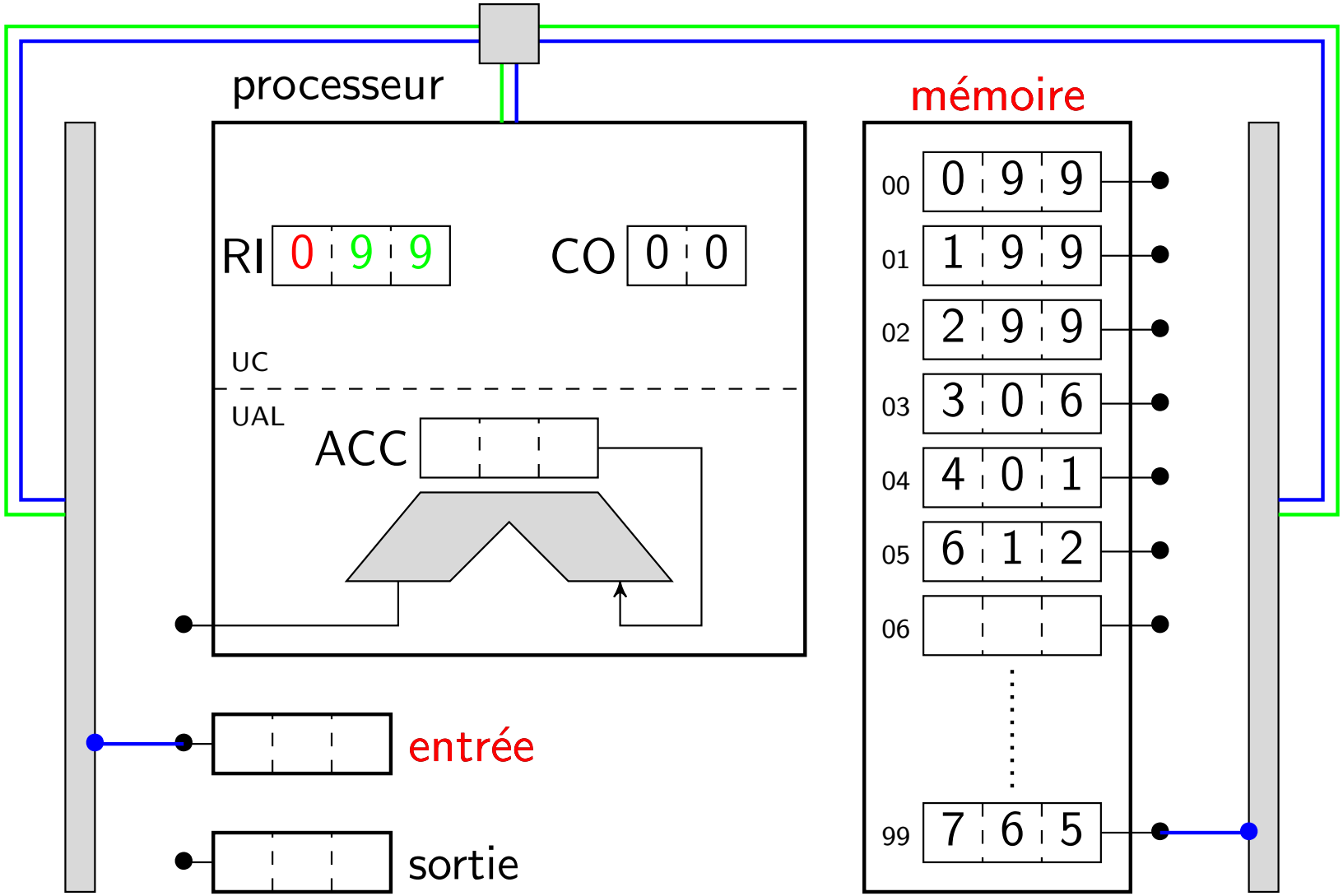
Ordinapoche - input (code 0, assembleur INP)



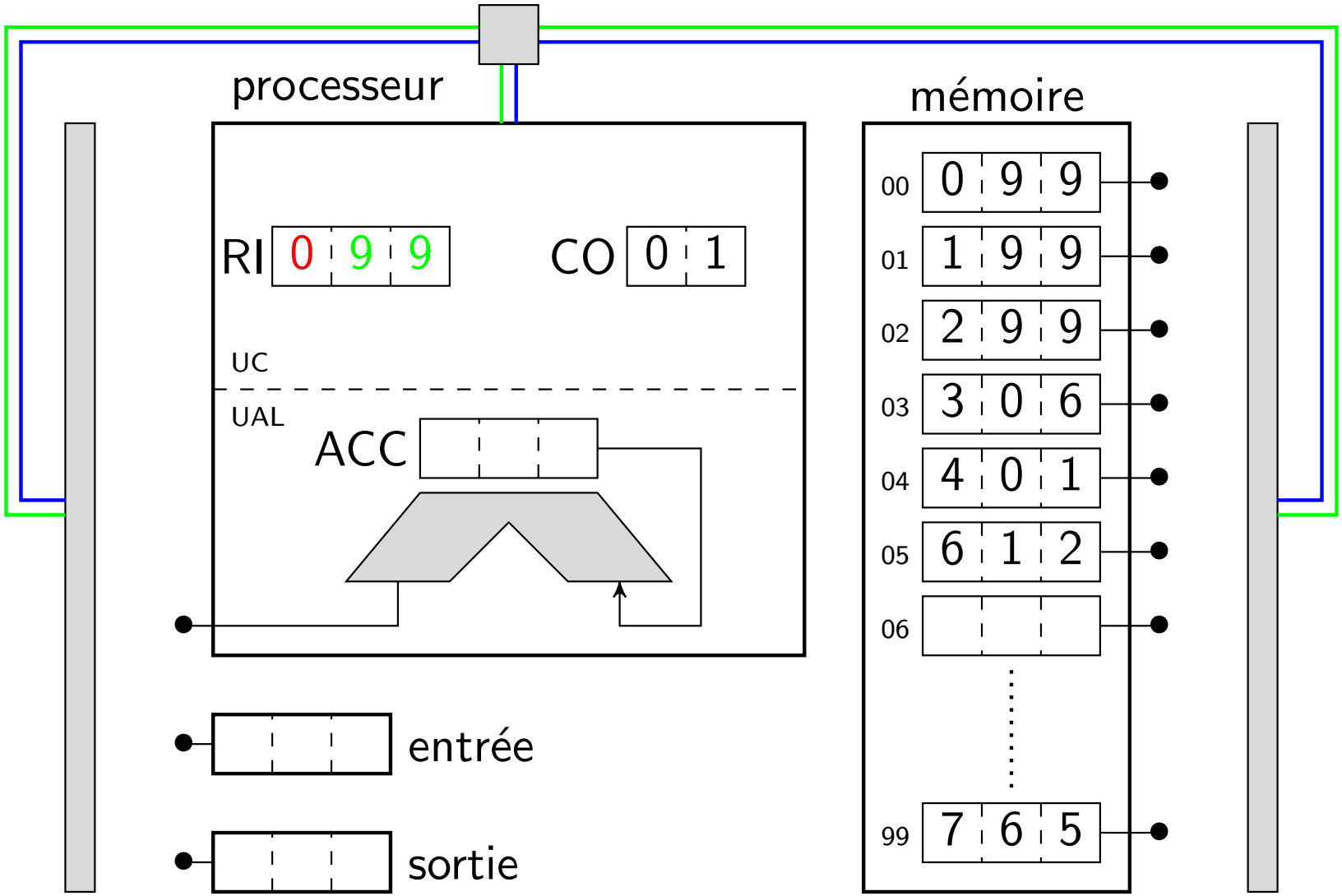
Ordinapoche - input (code 0, assembleur INP)



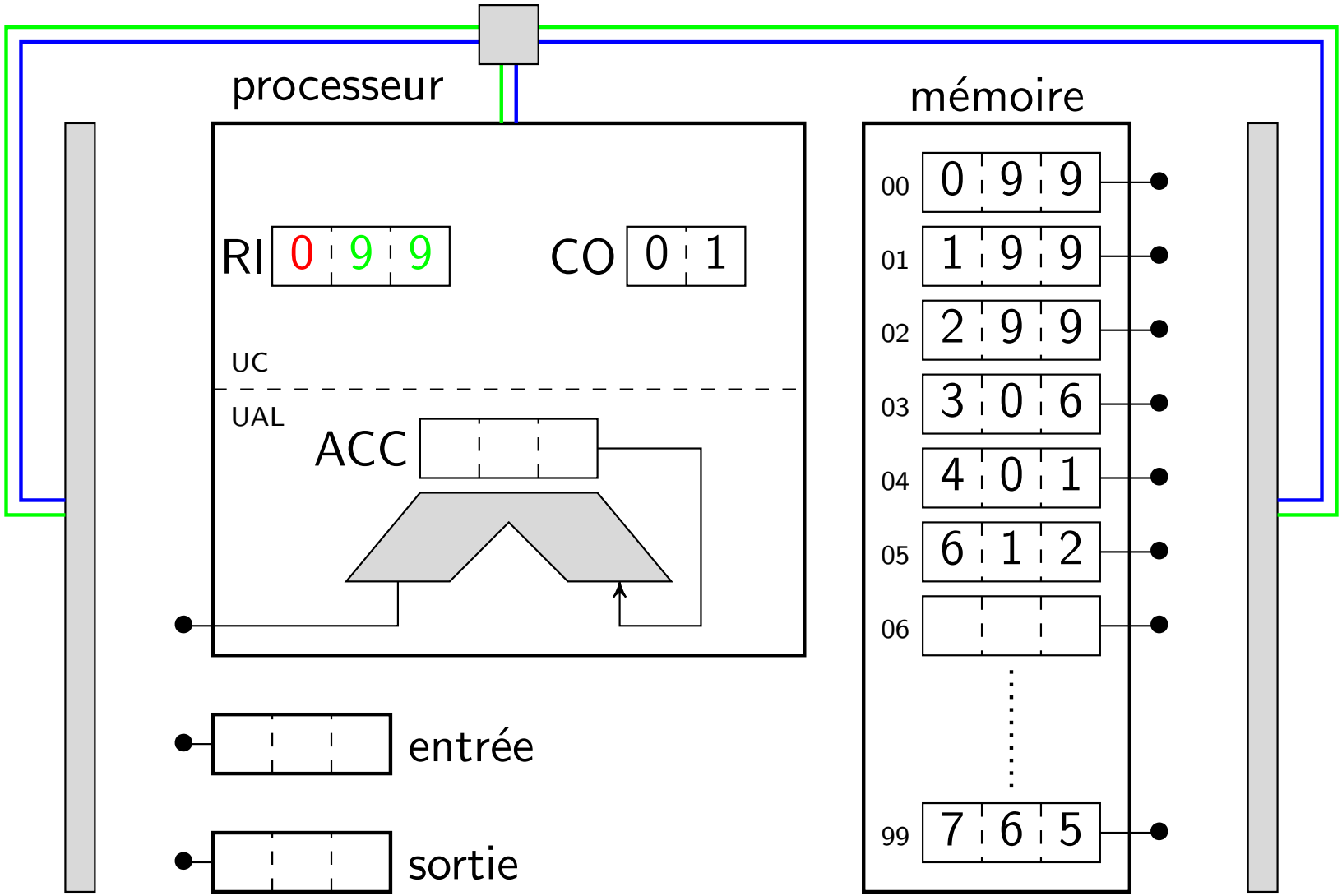
Ordinapoche - input (code 0, assembleur INP)



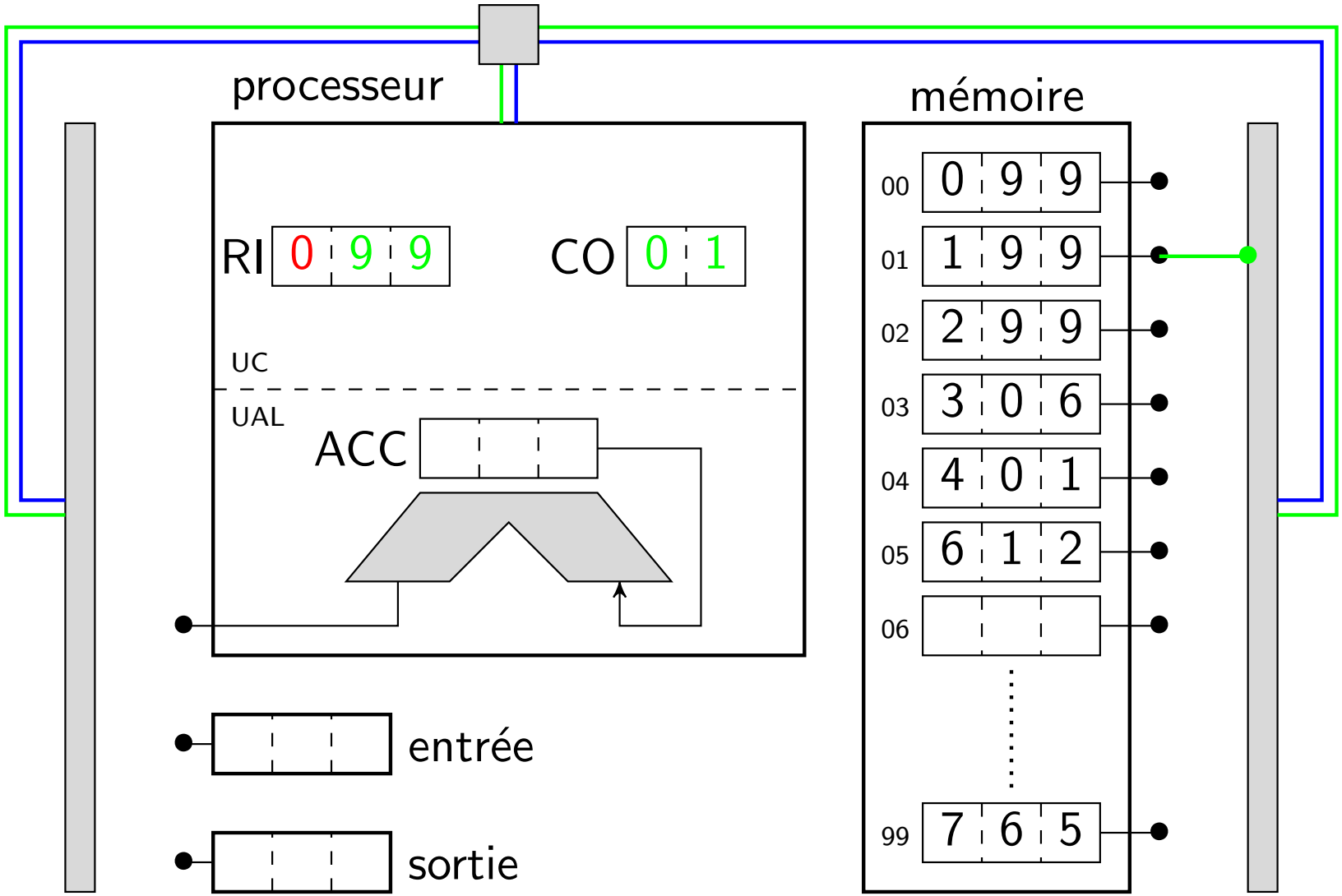
Ordinapoche - input (code 0, assembleur INP)



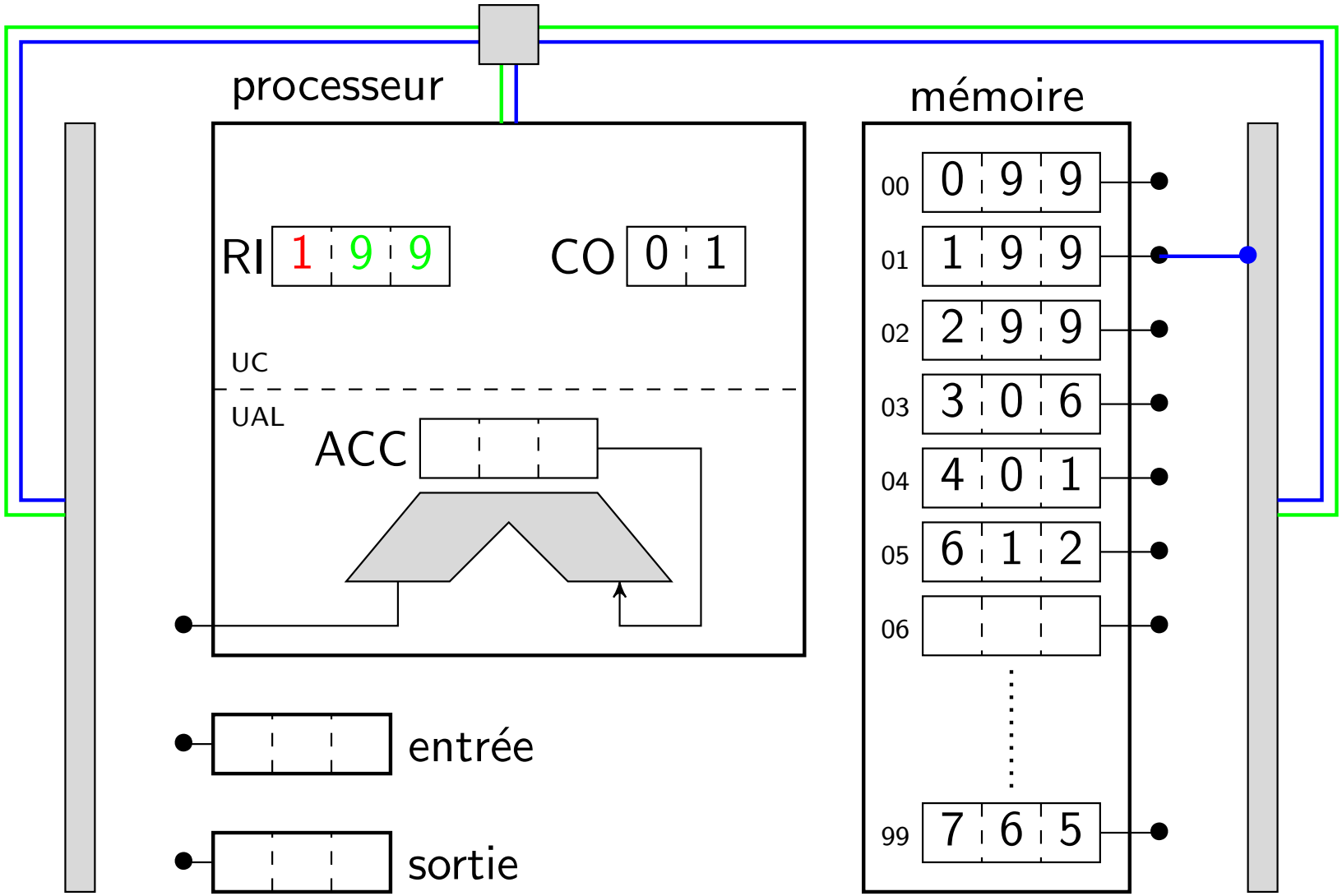
Ordinapoche - output (code 1, assembleur OUT)



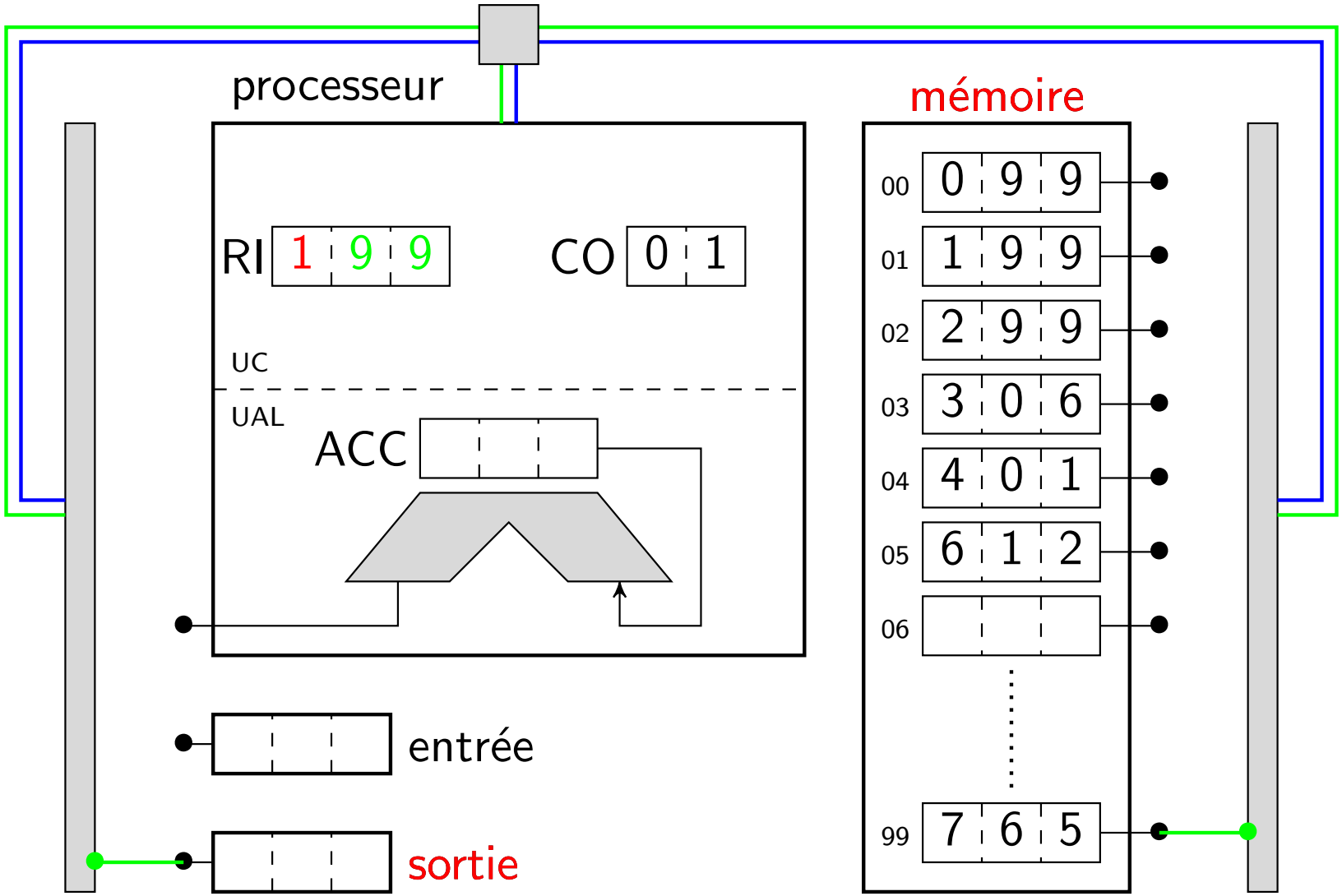
Ordinapoche - output (code 1, assembleur OUT)



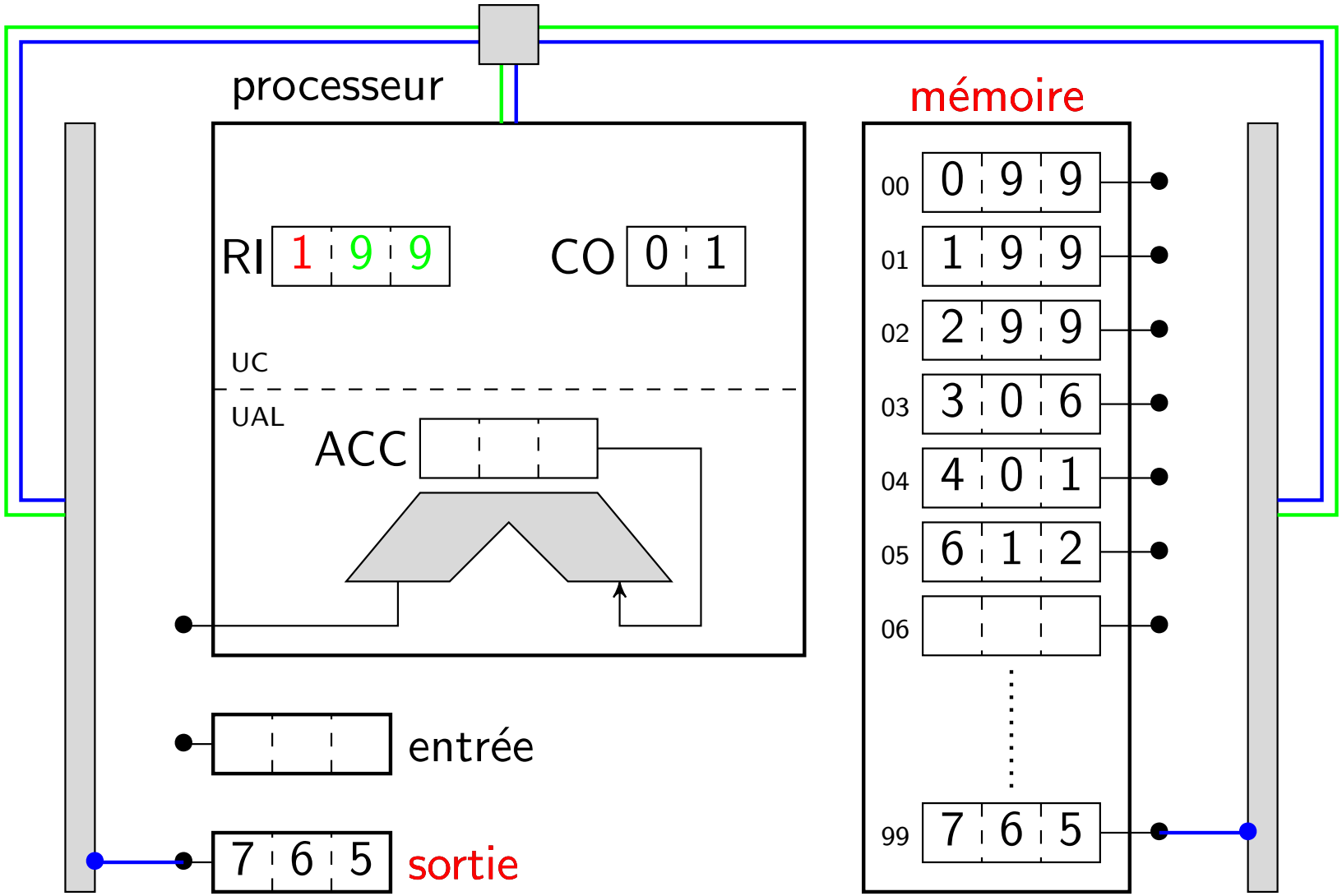
Ordinapoche - output (code 1, assembleur OUT)



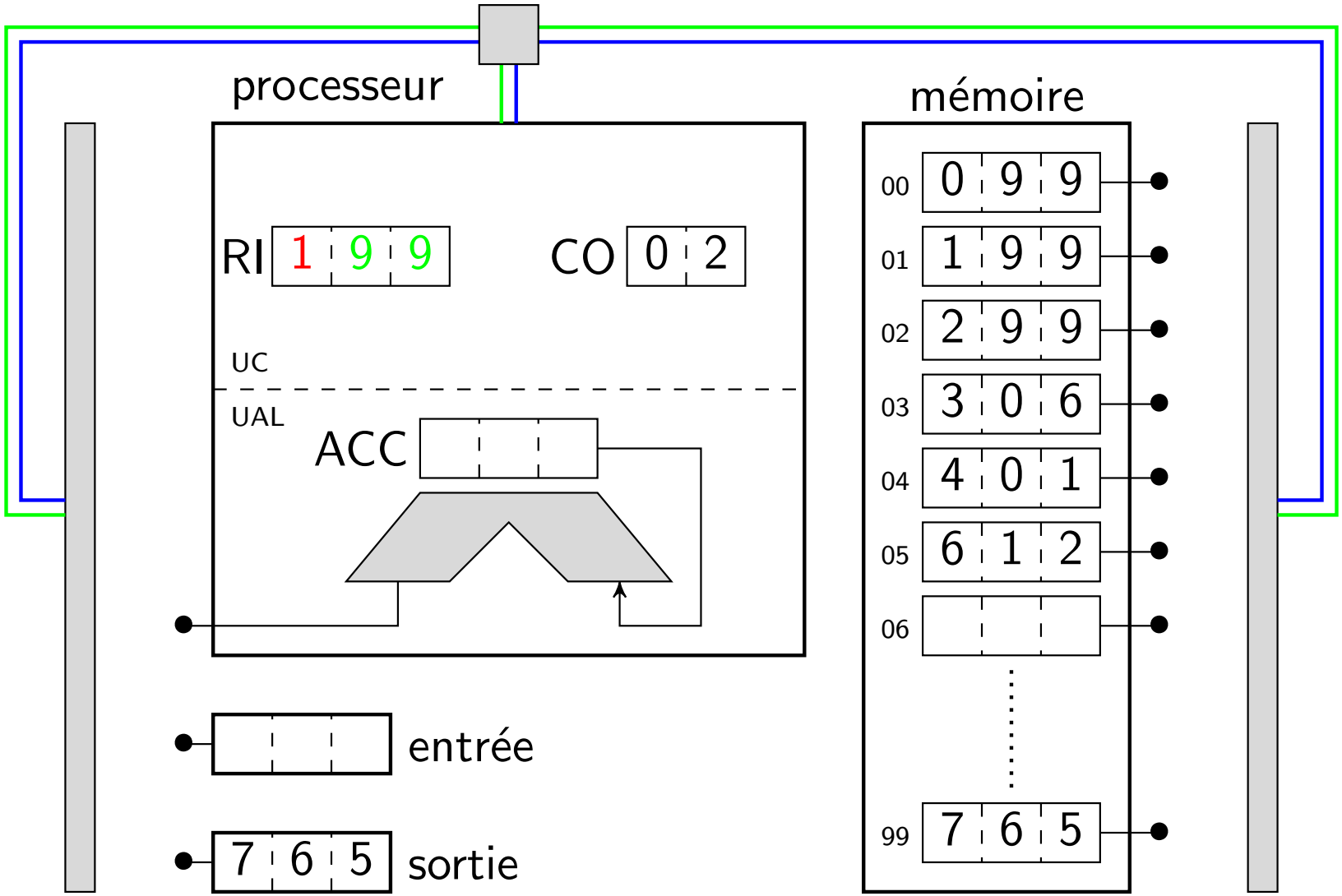
Ordinapoche - output (code 1, assembleur OUT)



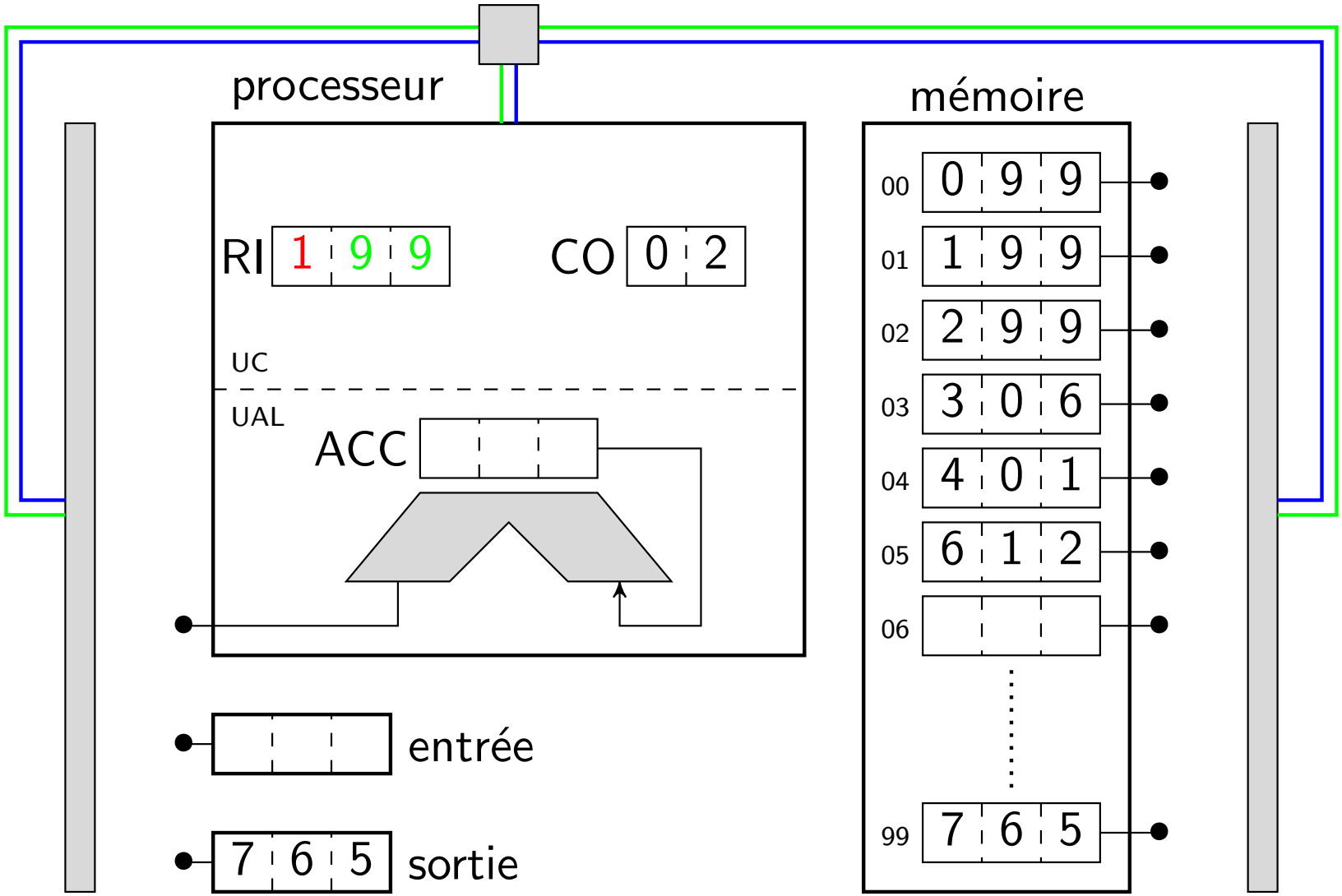
Ordinapoche - output (code 1, assembleur OUT)



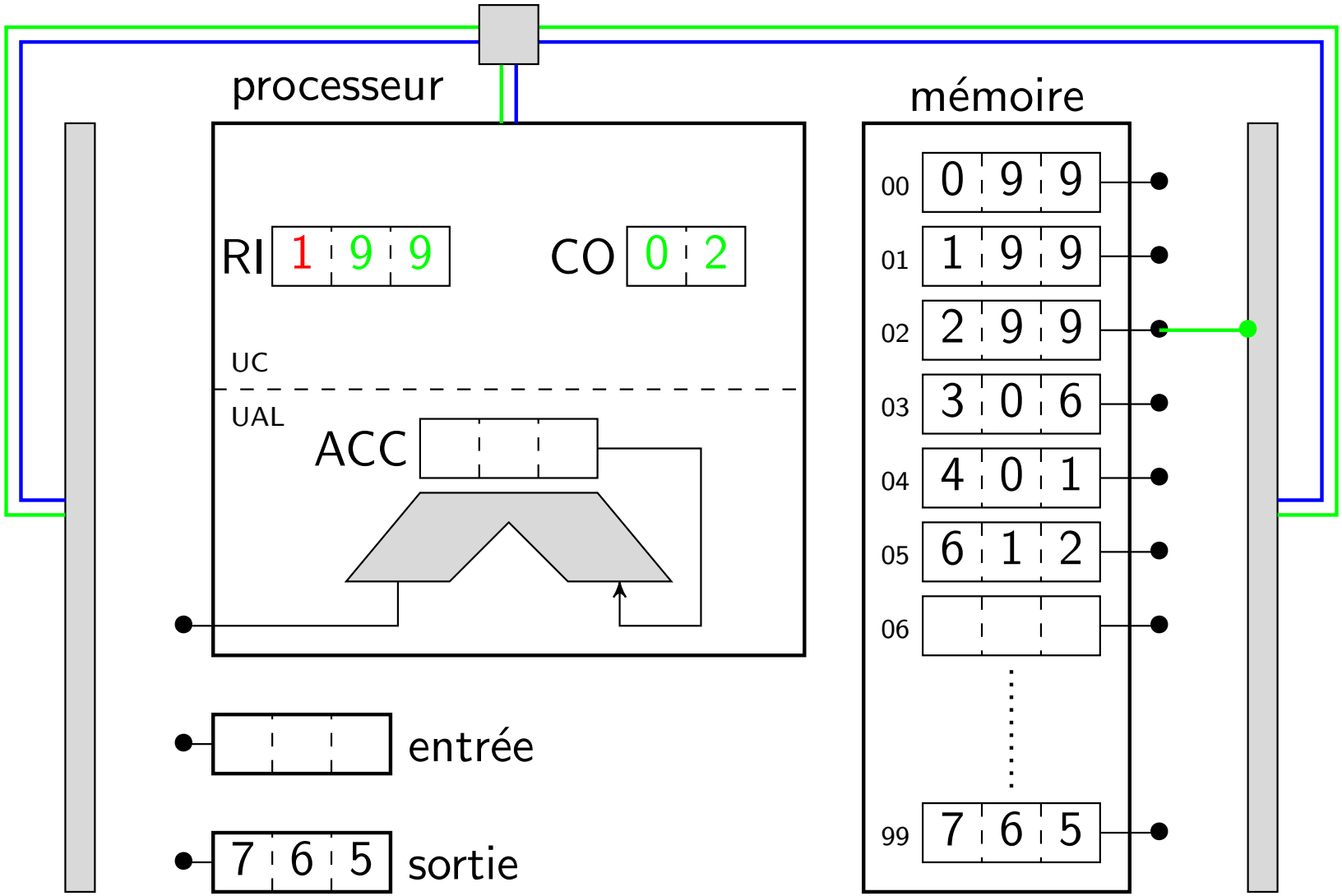
Ordinapoche - output (code 1, assembleur OUT)



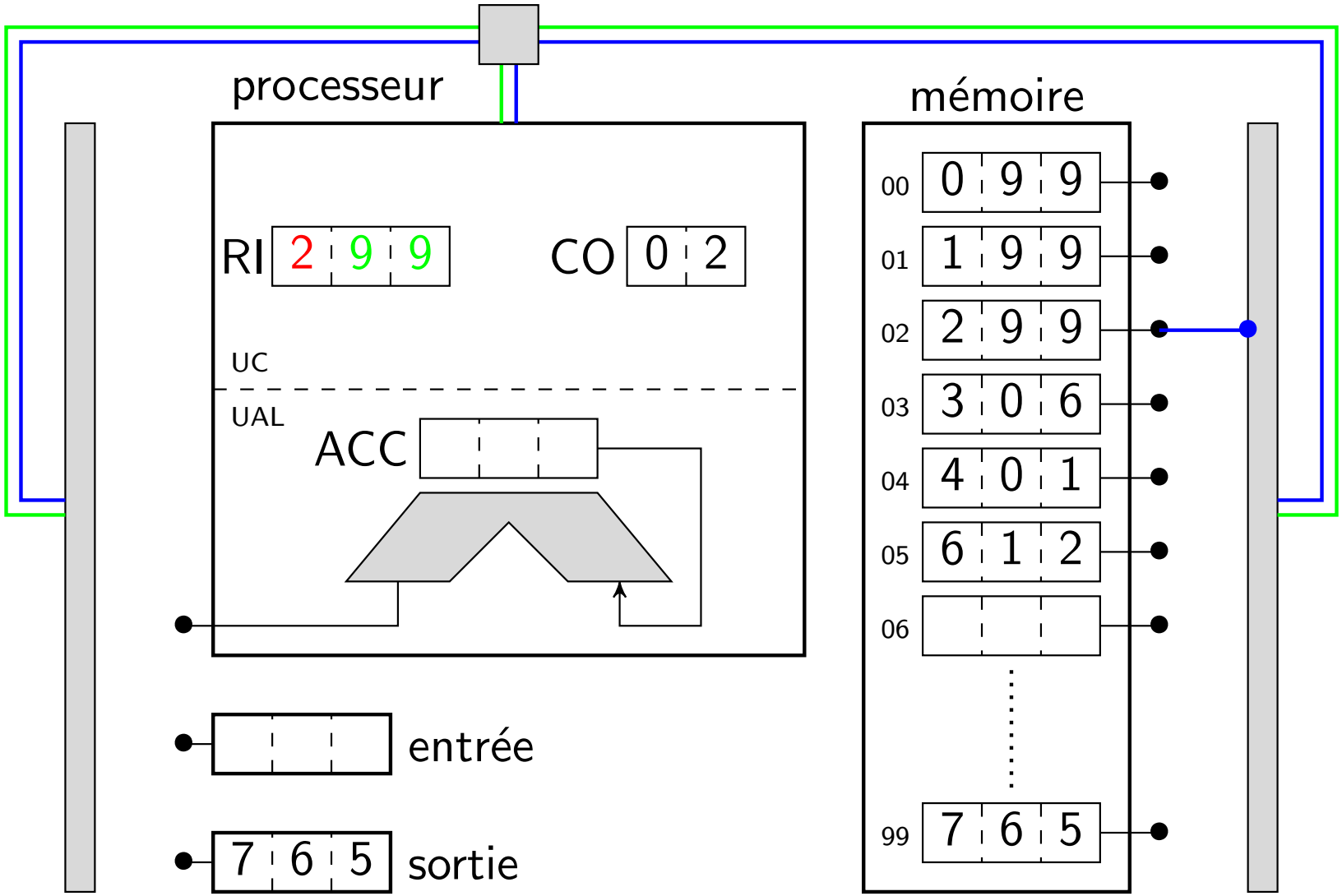
Ordinapoche - clear and add (code 2, assembleur CLA)



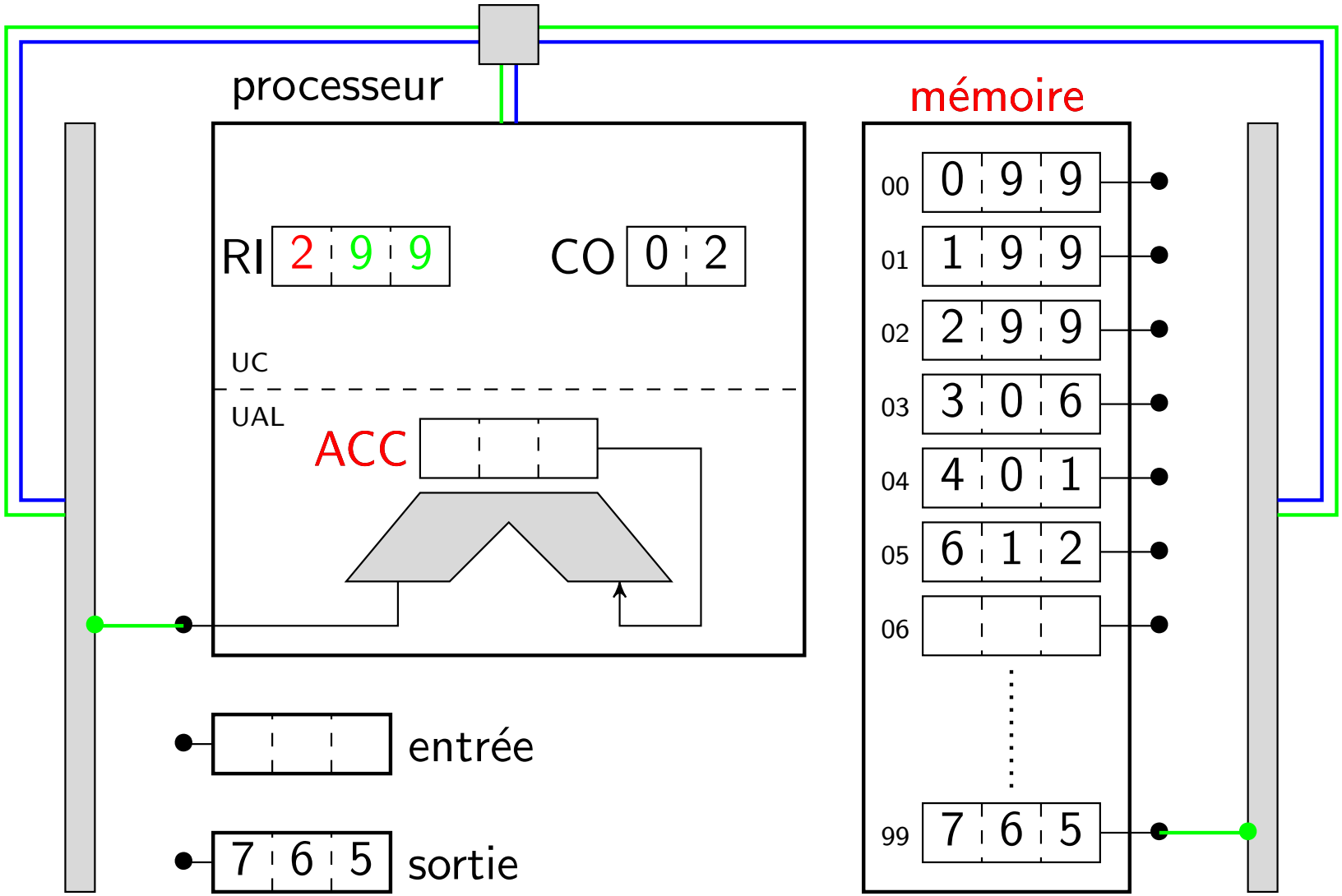
Ordinapoche - clear and add (code 2, assembleur CLA)



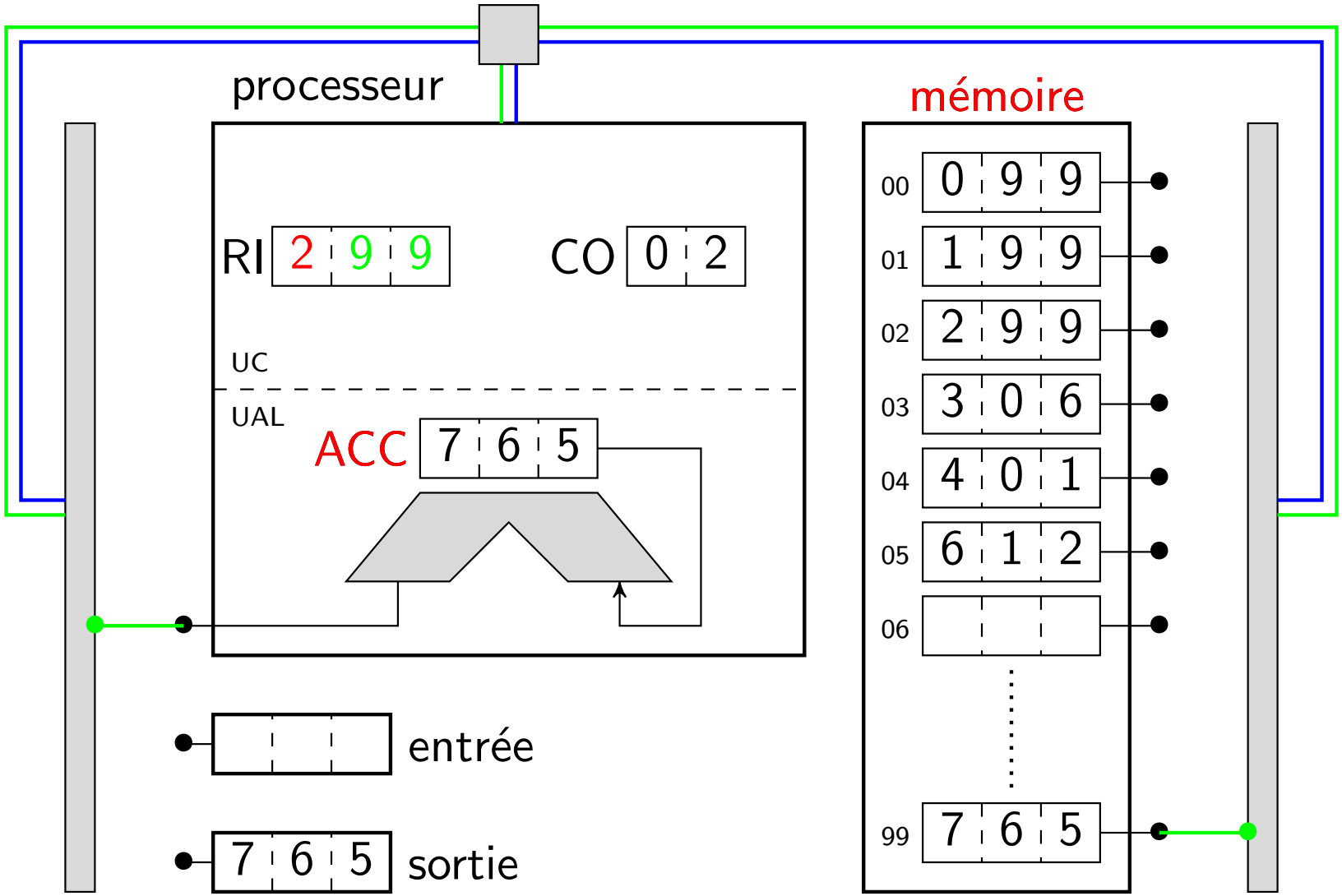
Ordinapoche - clear and add (code 2, assembleur CLA)



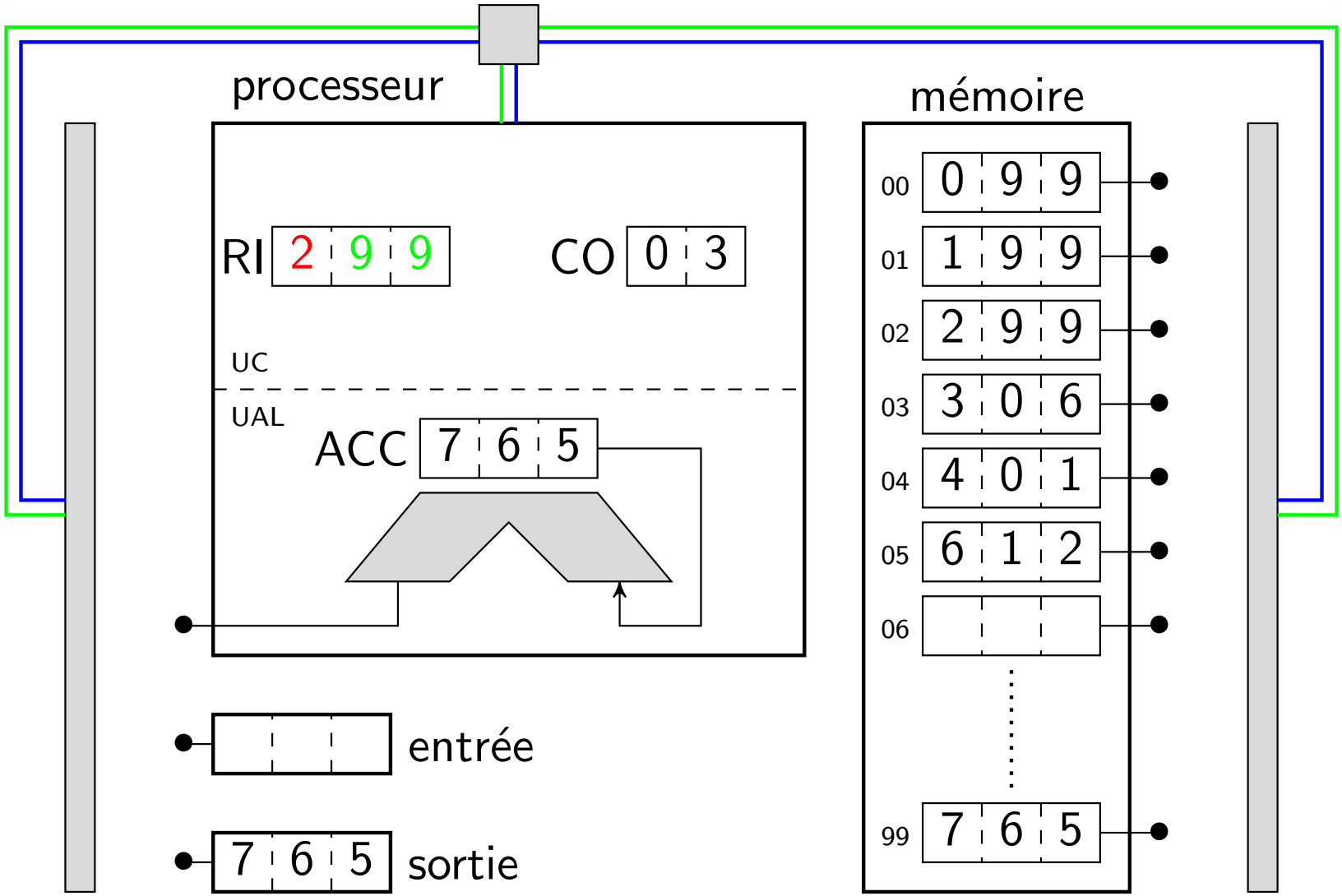
Ordinapoche - clear and add (code 2, assembleur CLA)



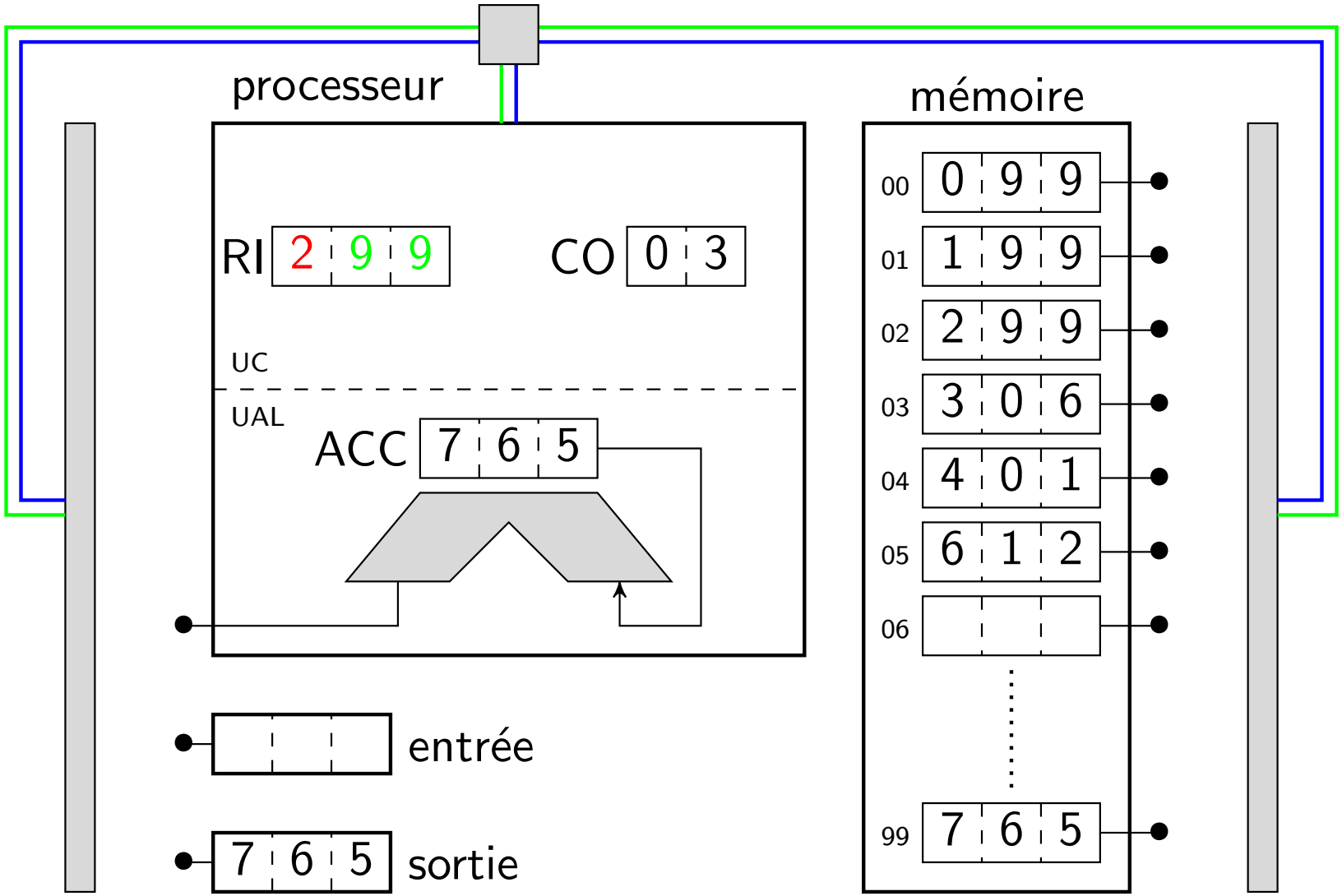
Ordinapoche - clear and add (code 2, assembleur CLA)



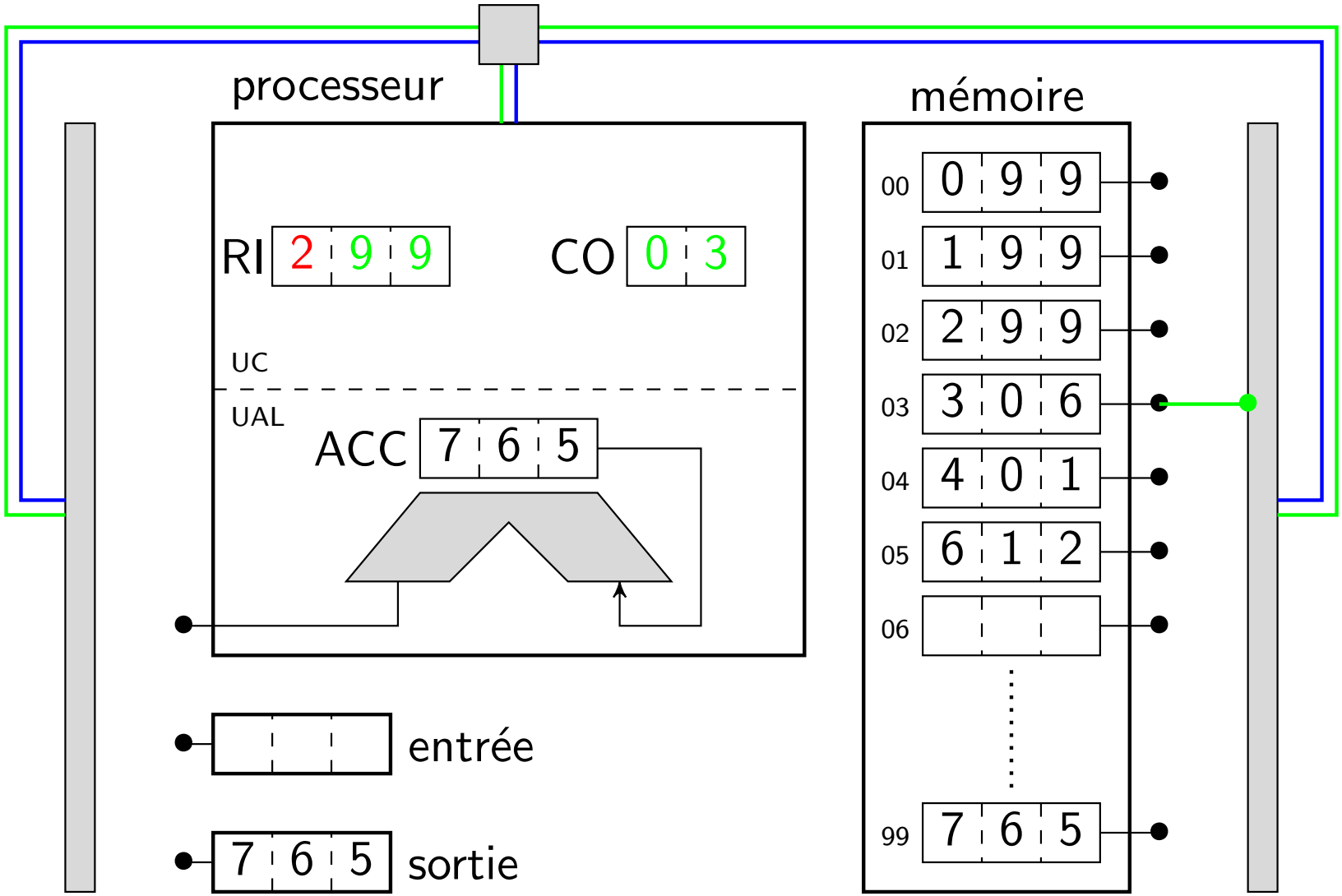
Ordinapoche - clear and add (code 2, assembleur CLA)



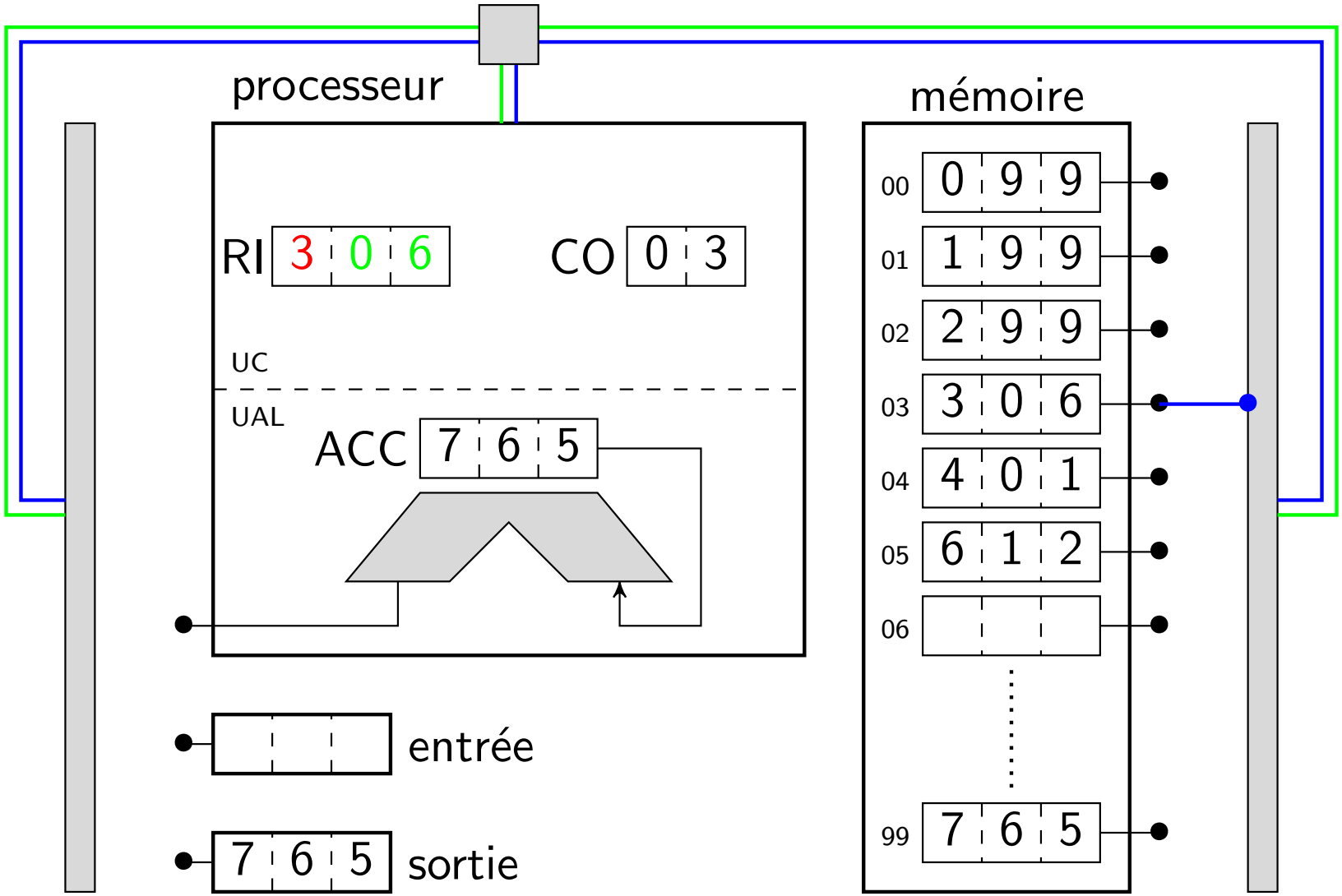
Ordinapoche - store (code 3, assembleur STO)



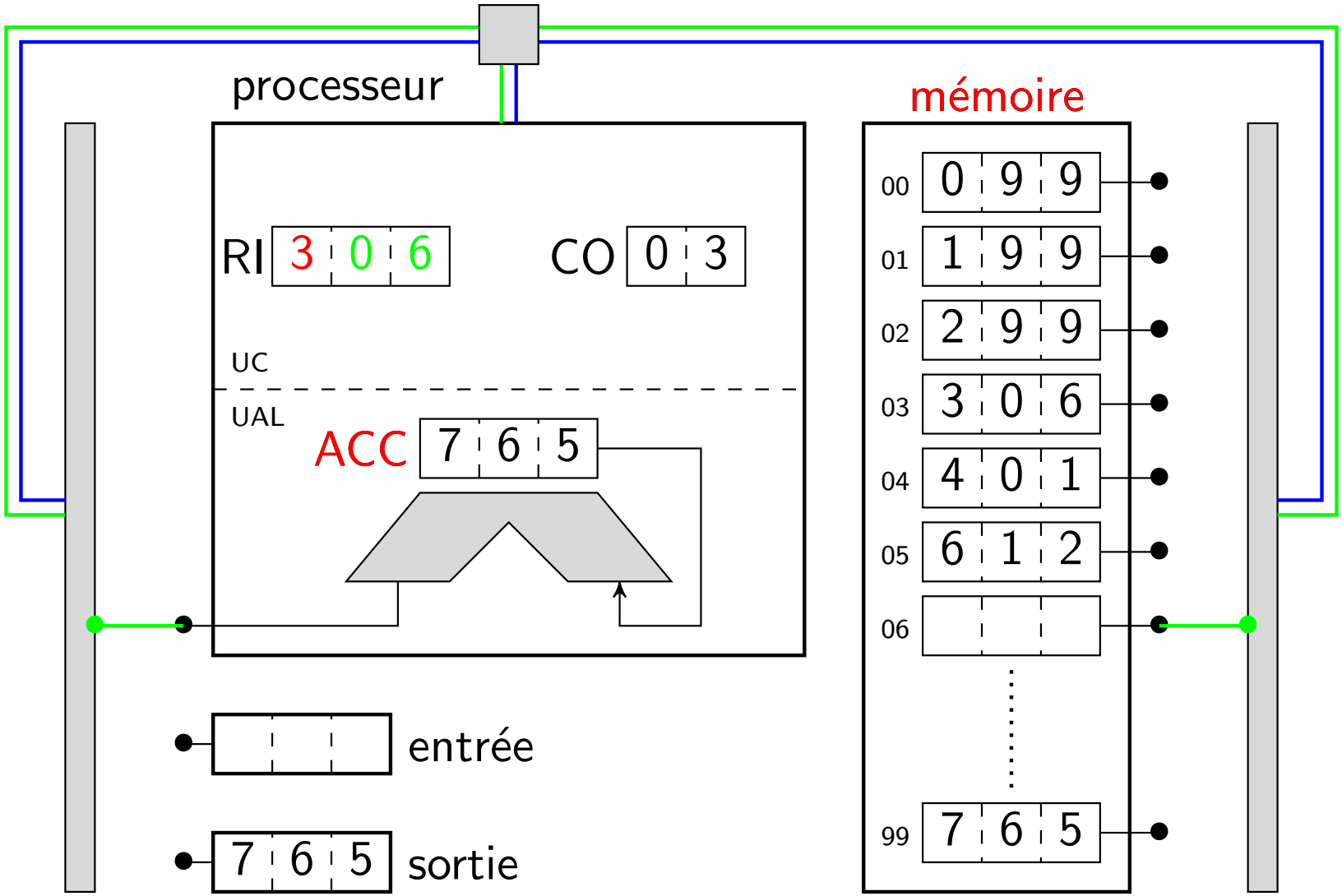
Ordinapoche - store (code 3, assembleur STO)



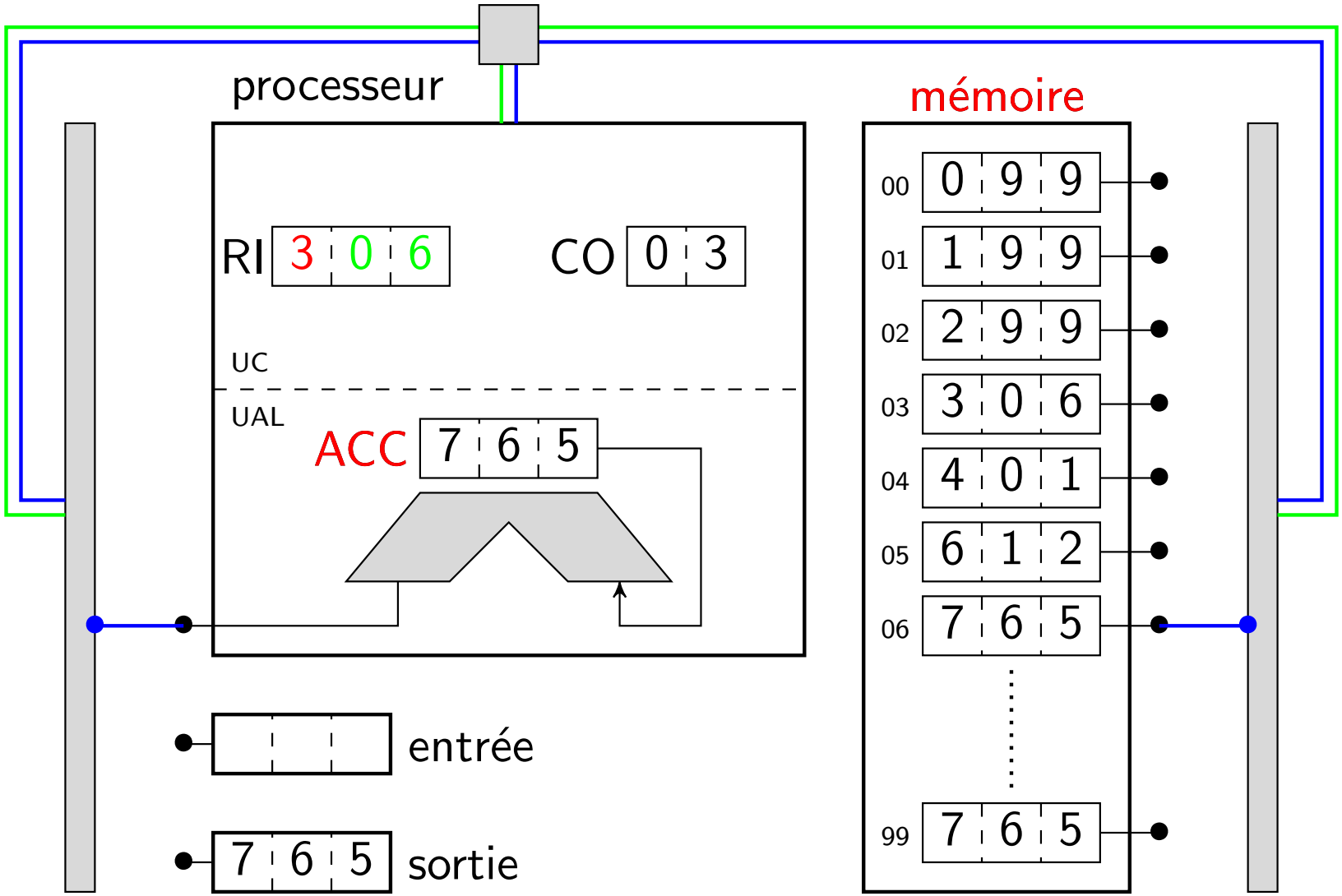
Ordinapoche - store (code 3, assembleur STO)



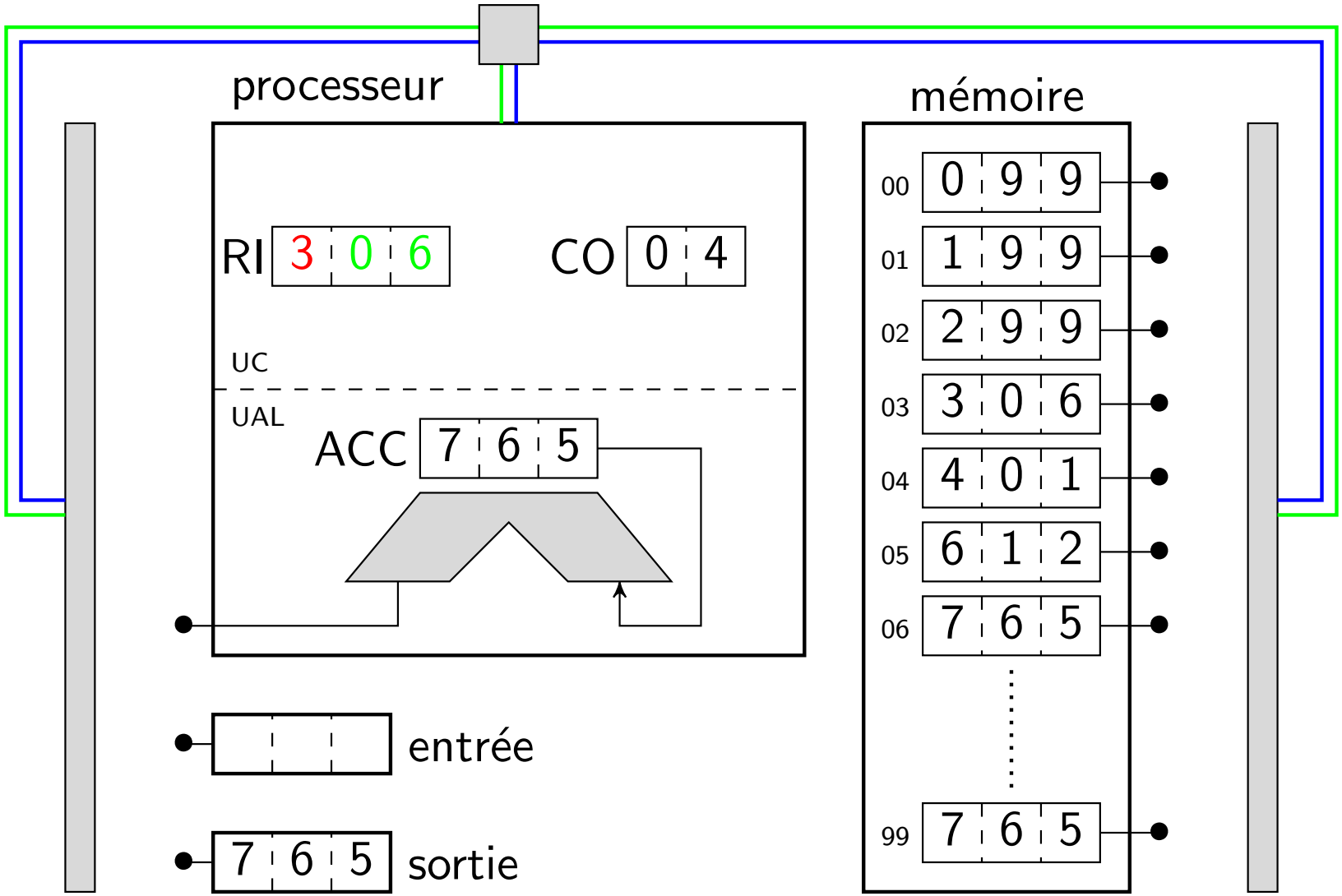
Ordinapoche - store (code 3, assembleur STO)



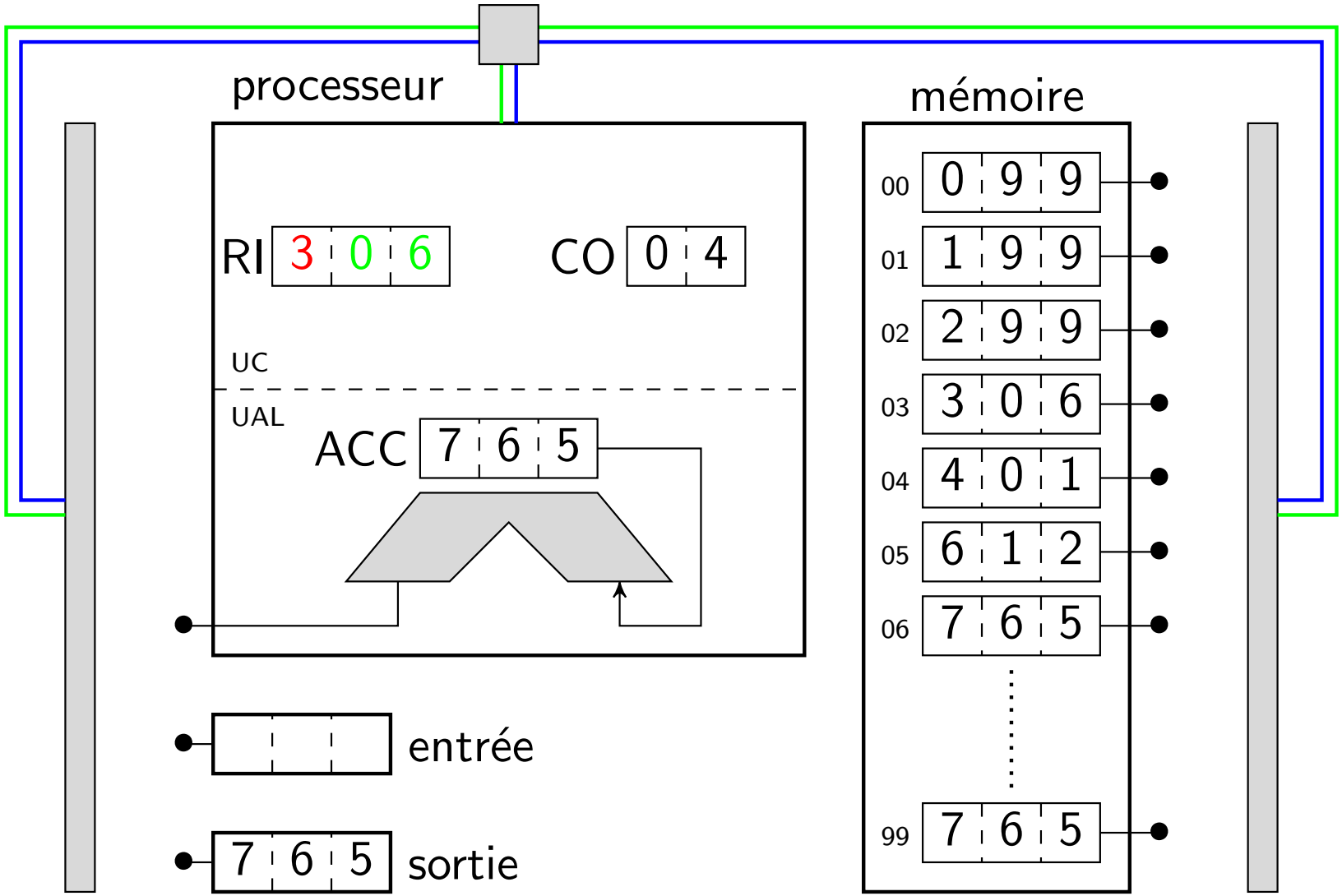
Ordinapoche - store (code 3, assembleur STO)



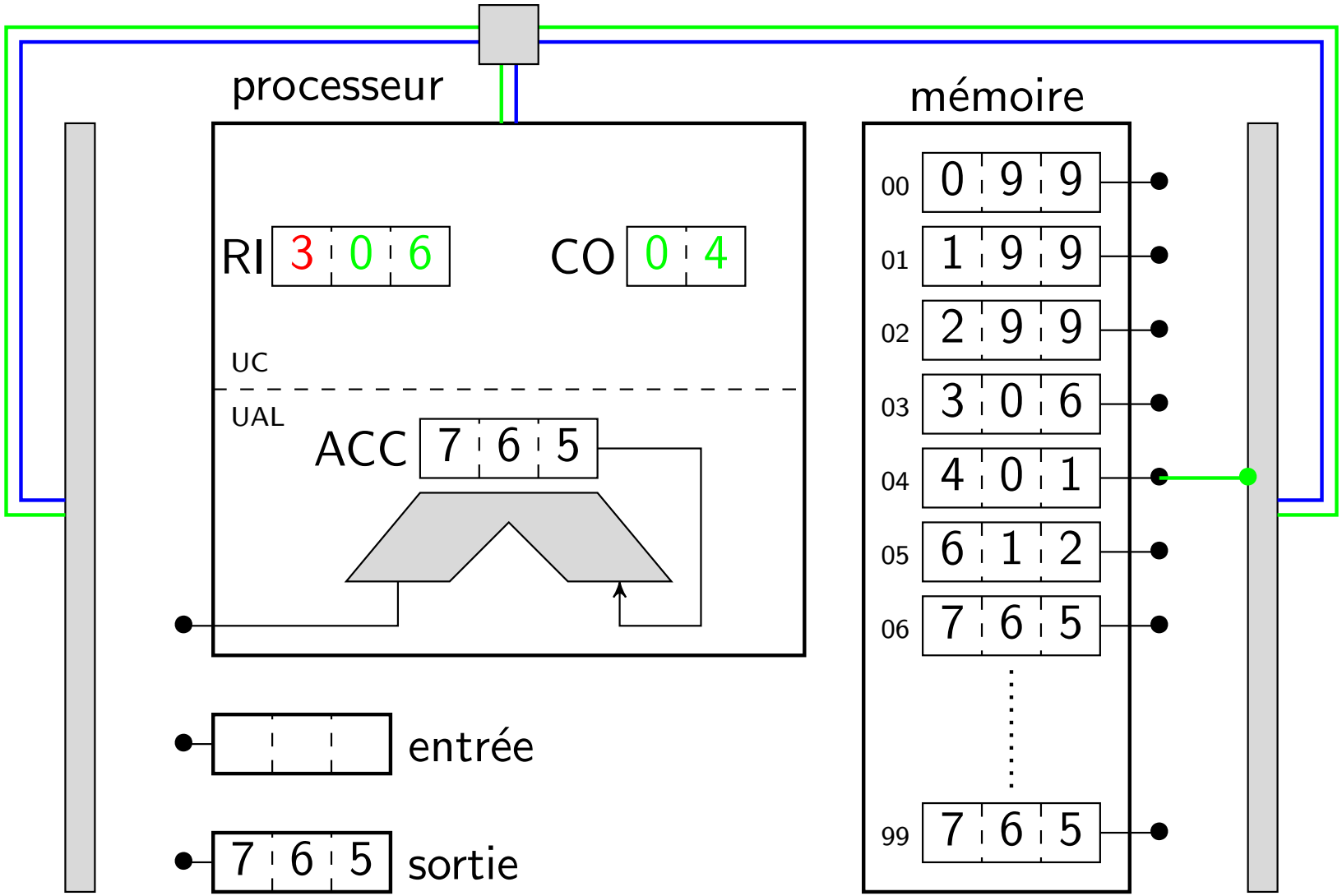
Ordinapoche - store (code 3, assembleur STO)



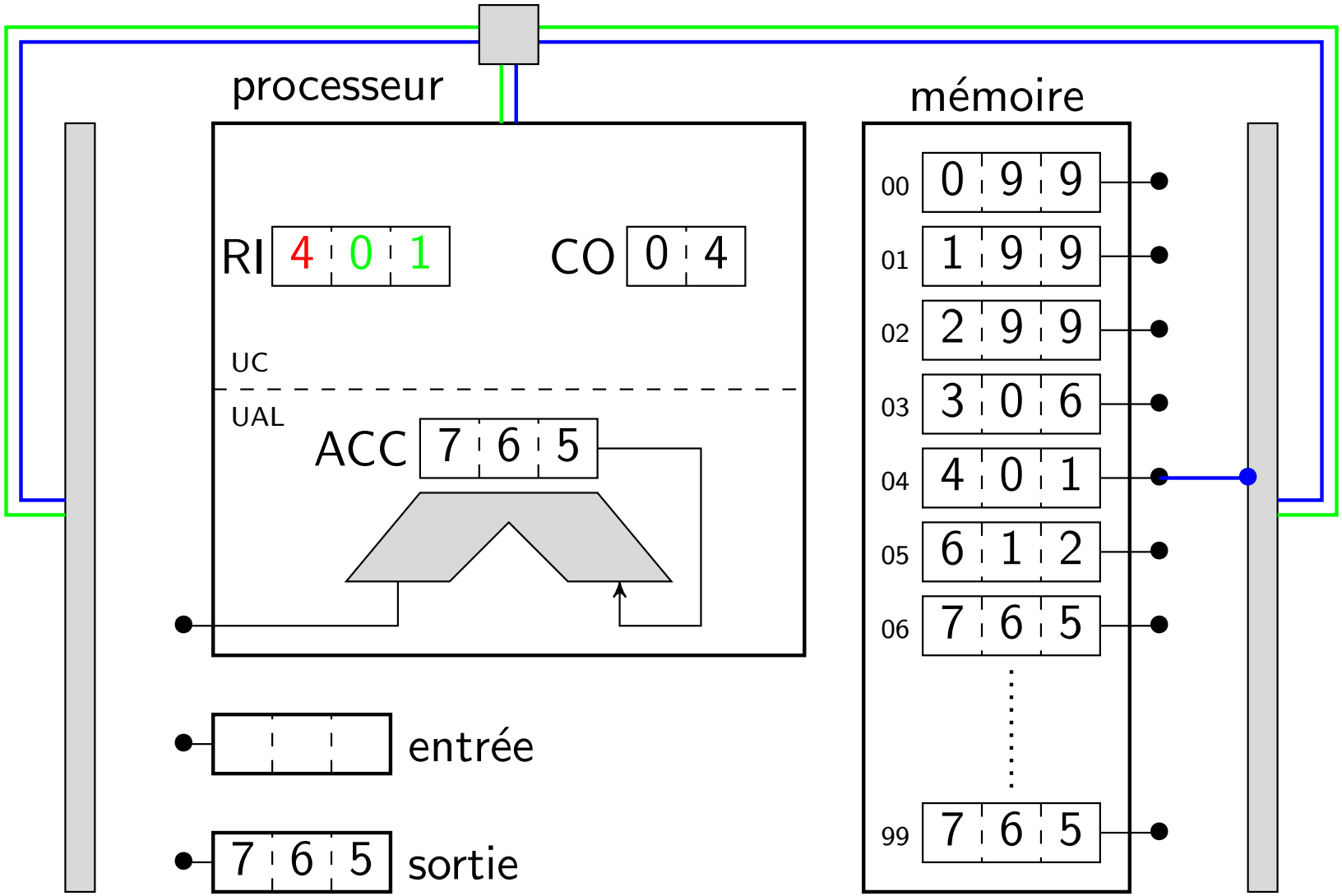
Ordinapoche - add (code 4, assembleur ADD)



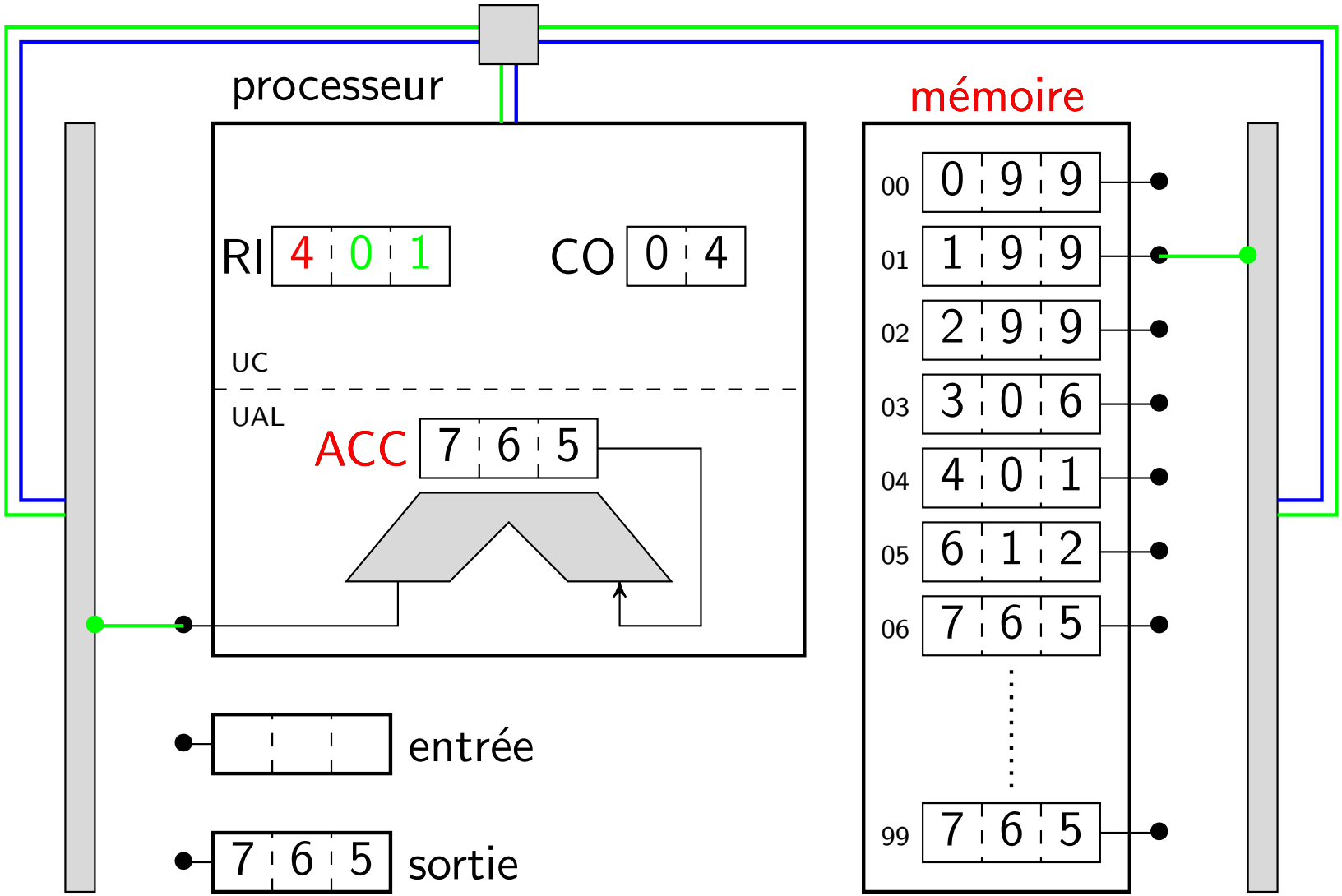
Ordinapoche - add (code 4, assembleur ADD)



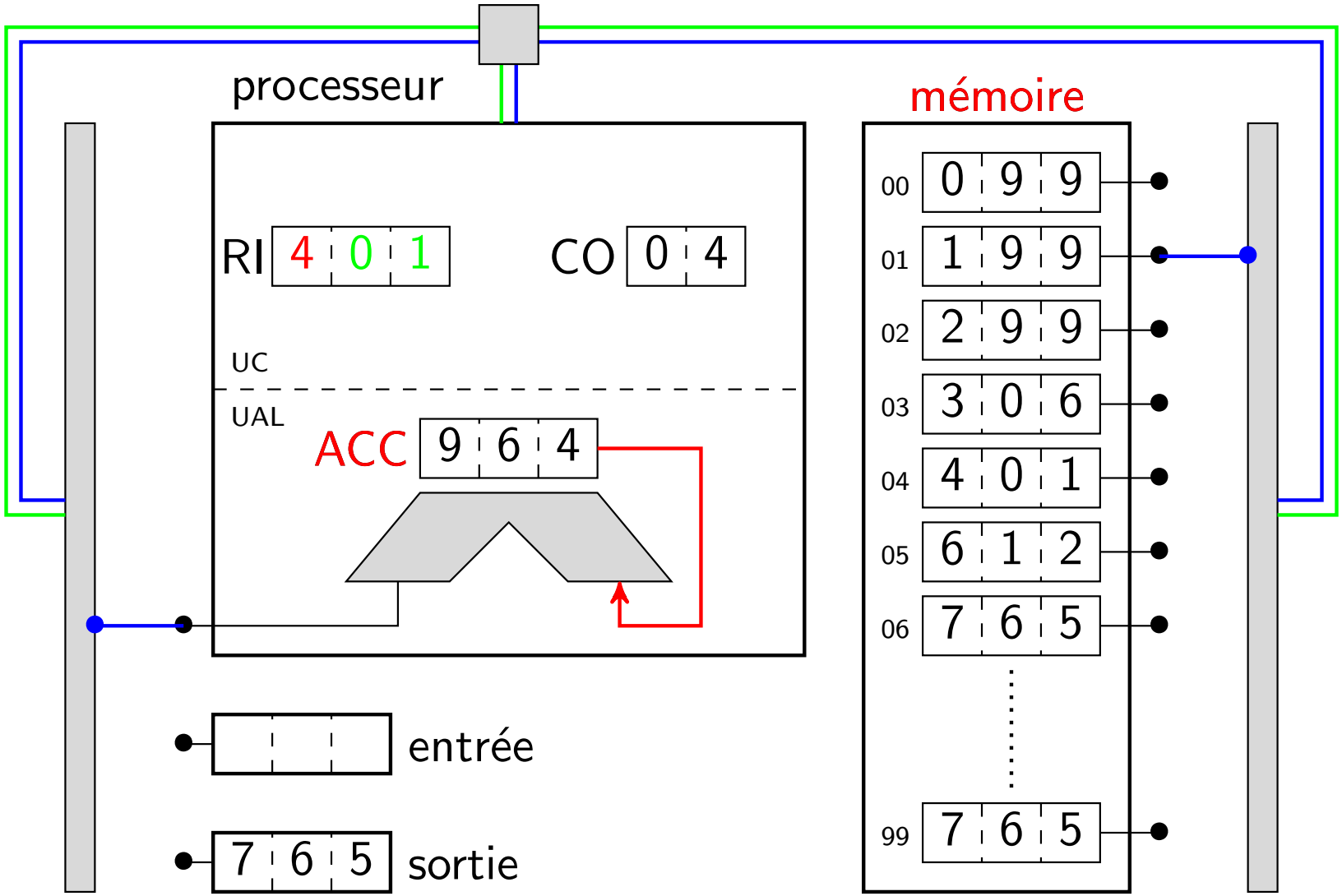
Ordinapoche - add (code 4, assembleur ADD)



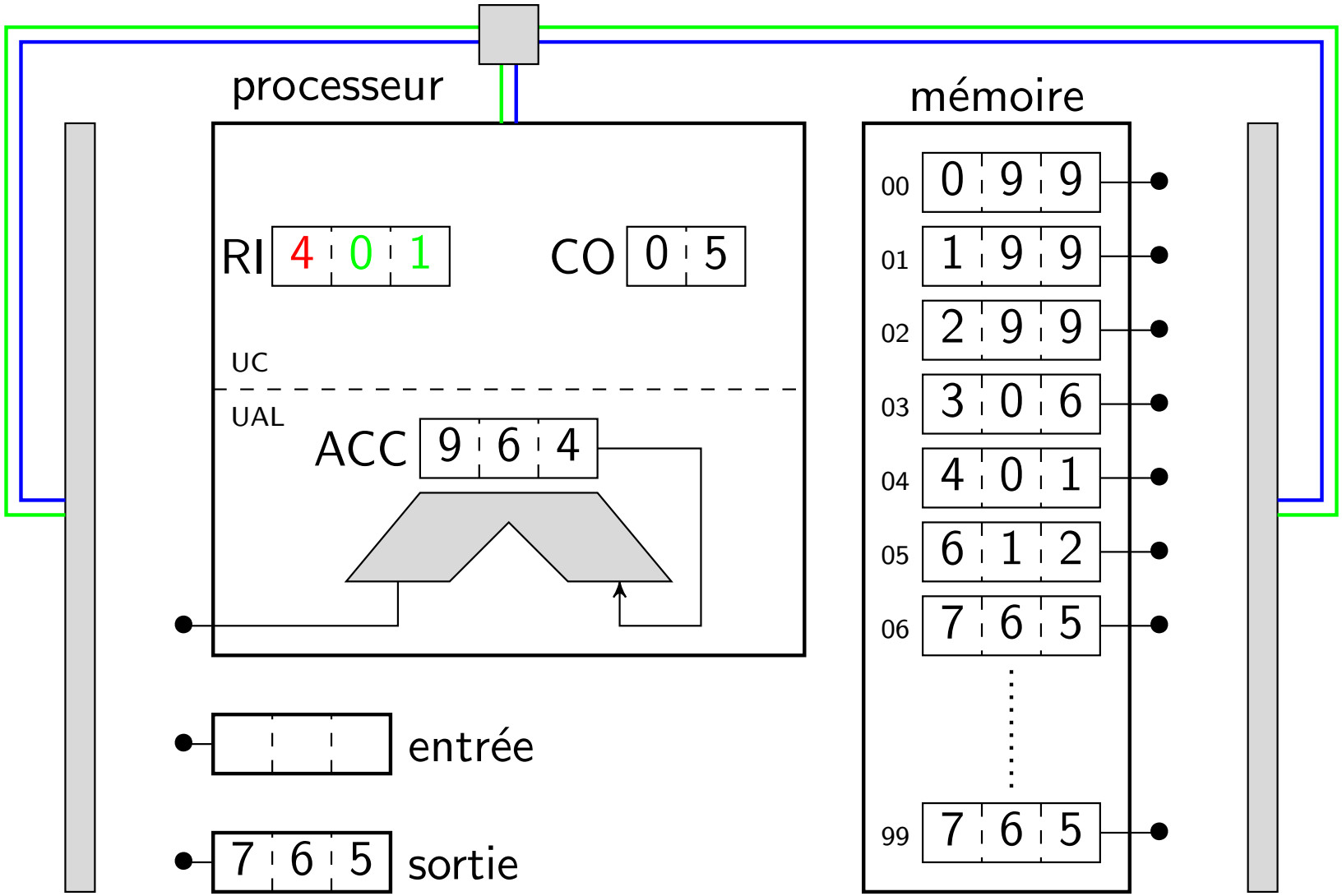
Ordinapoche - add (code 4, assembleur ADD)



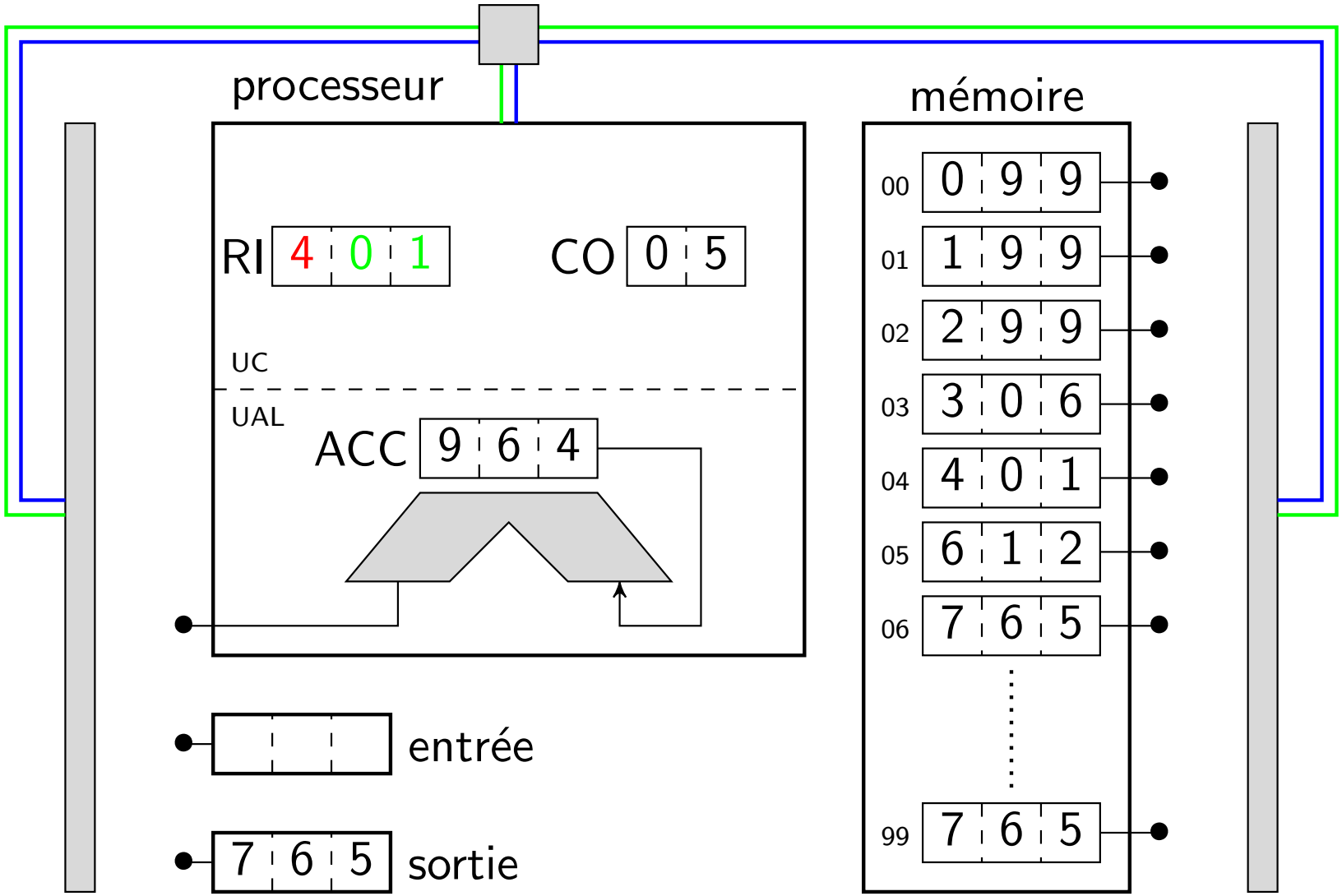
Ordinapoche - add (code 4, assembleur ADD)



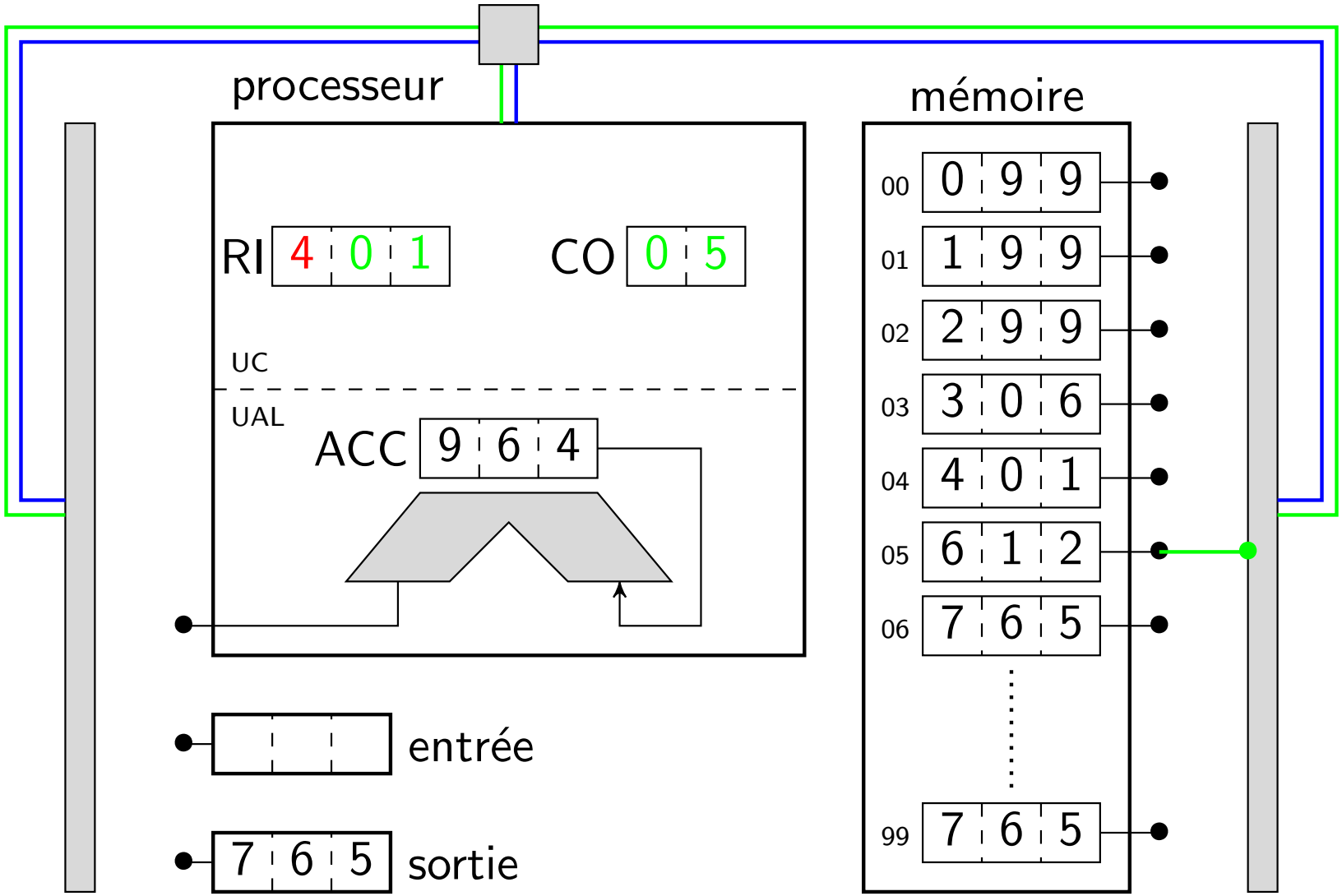
Ordinapoche - add (code 4, assembleur ADD)



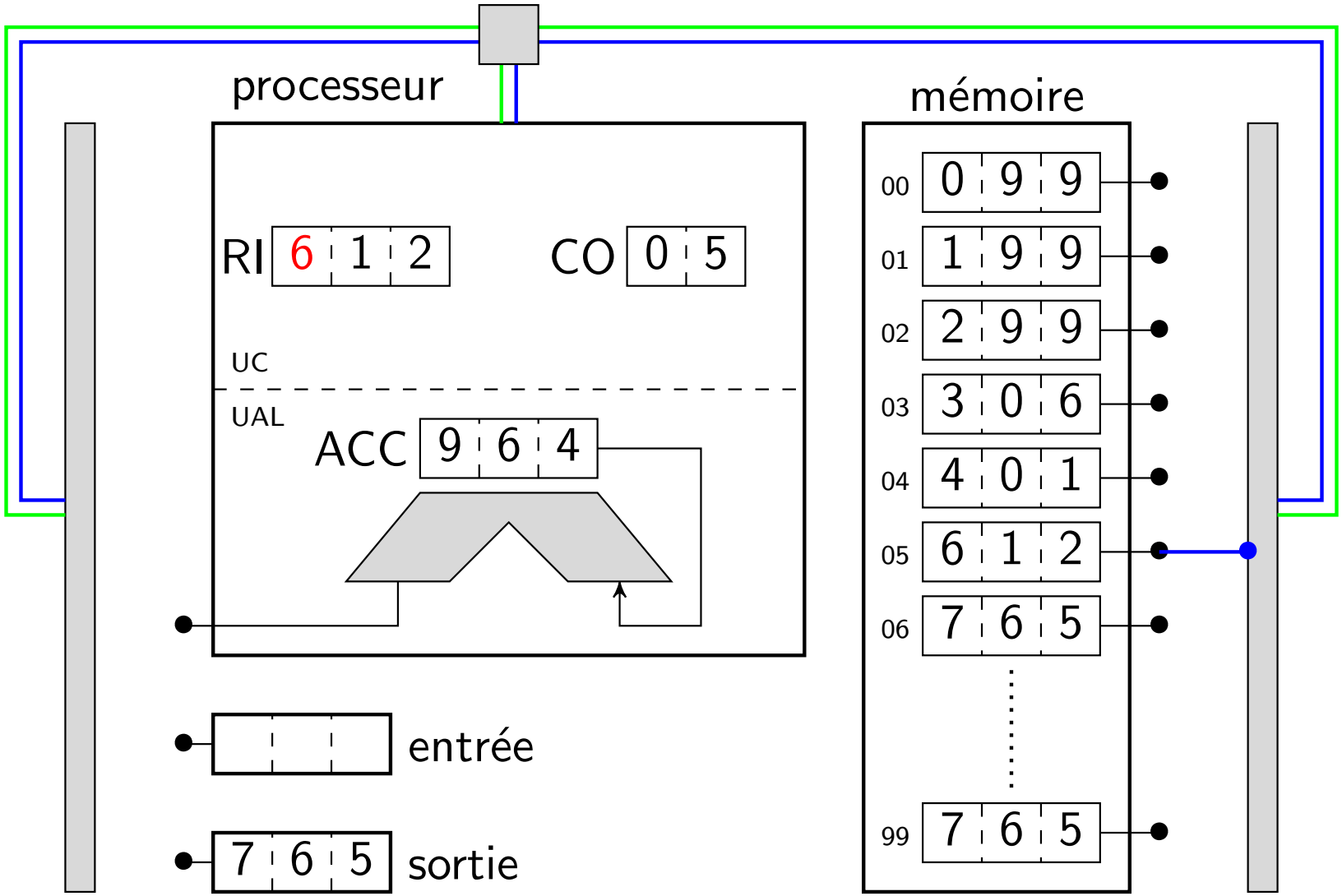
Ordinapoche - shift (code 6, assembleur SHT)



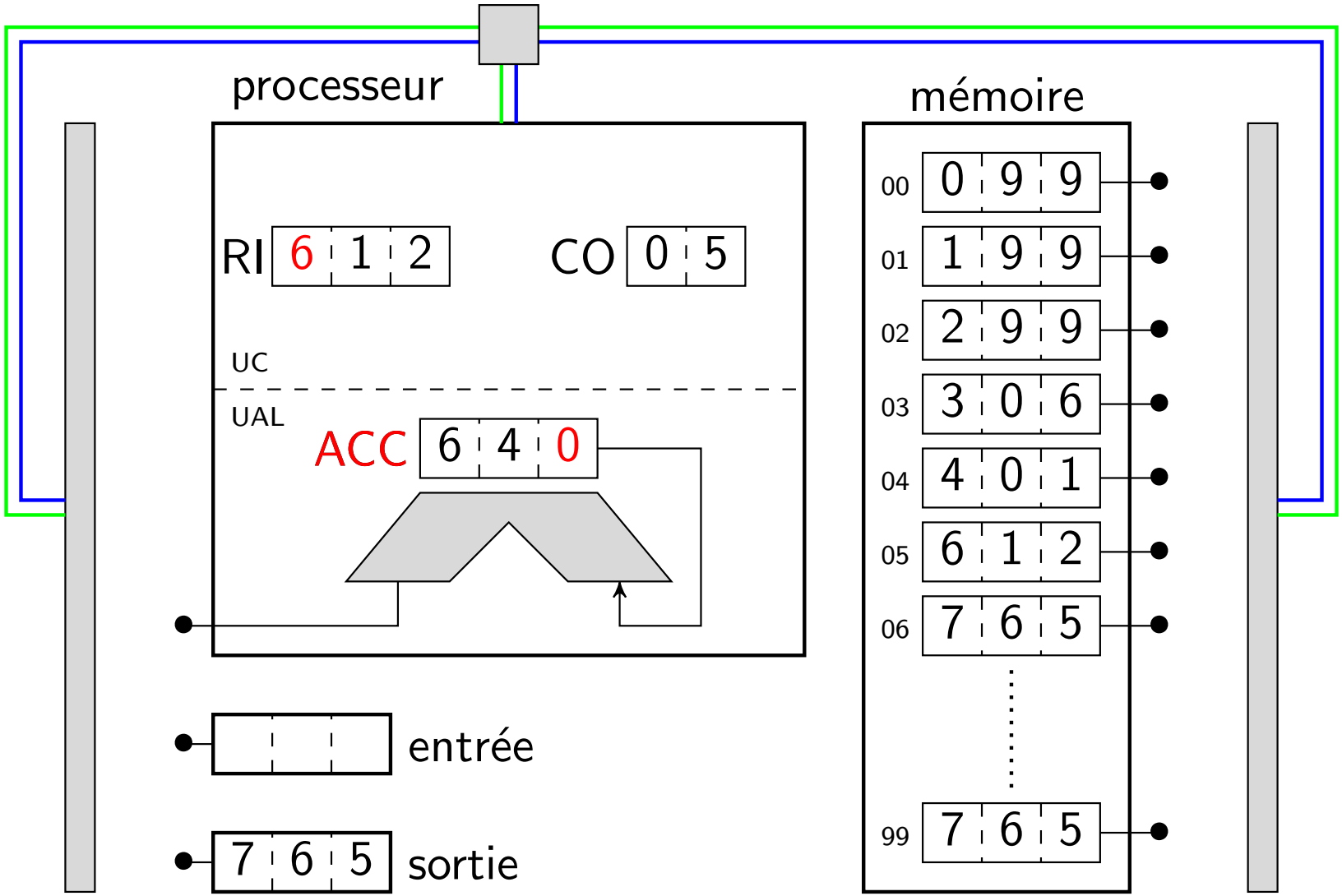
Ordinapoche - shift (code 6, assembleur SHT)



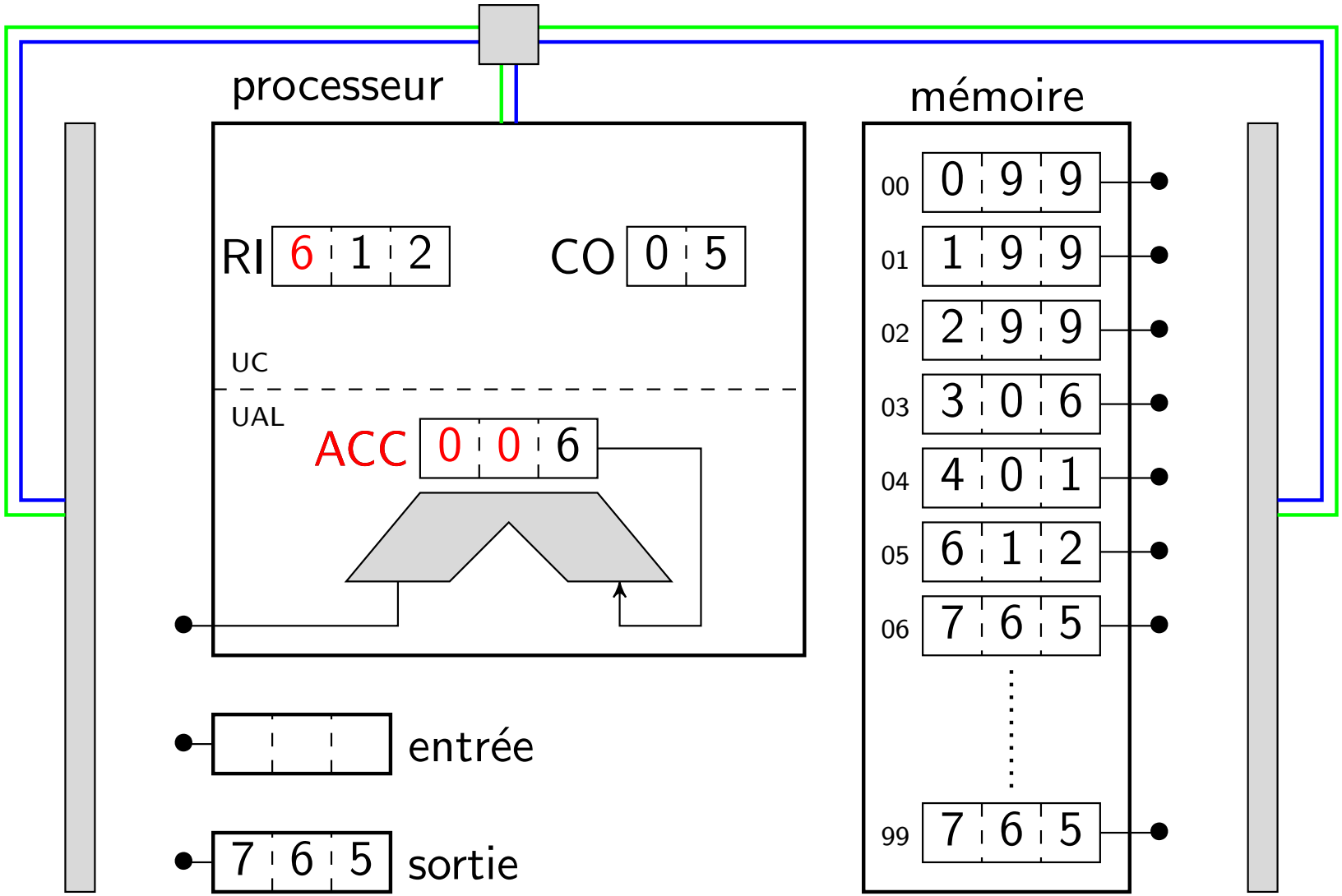
Ordinapoche - shift (code 6, assembleur SHT)



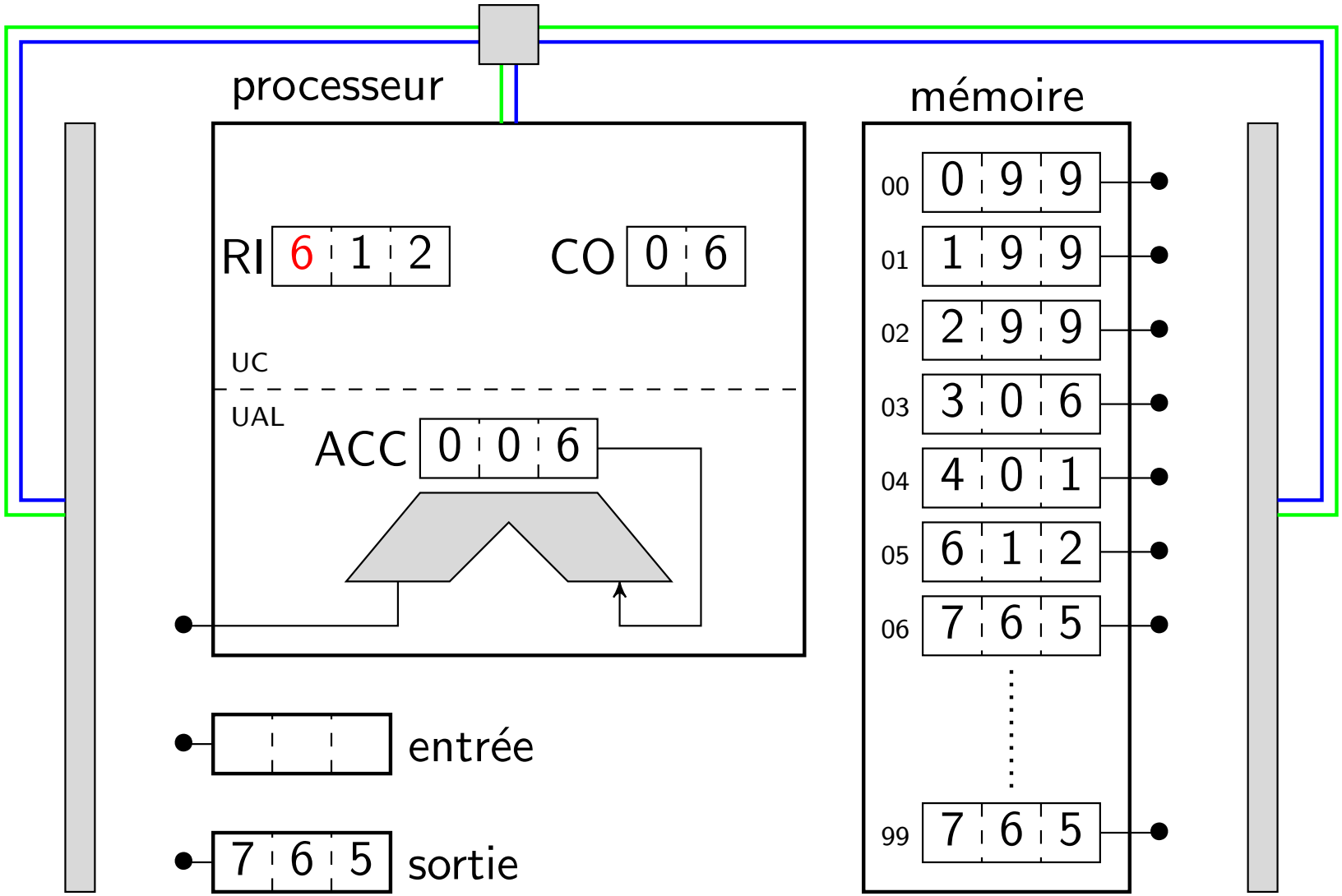
Ordinapoche - shift (code 6, assembleur SHT)



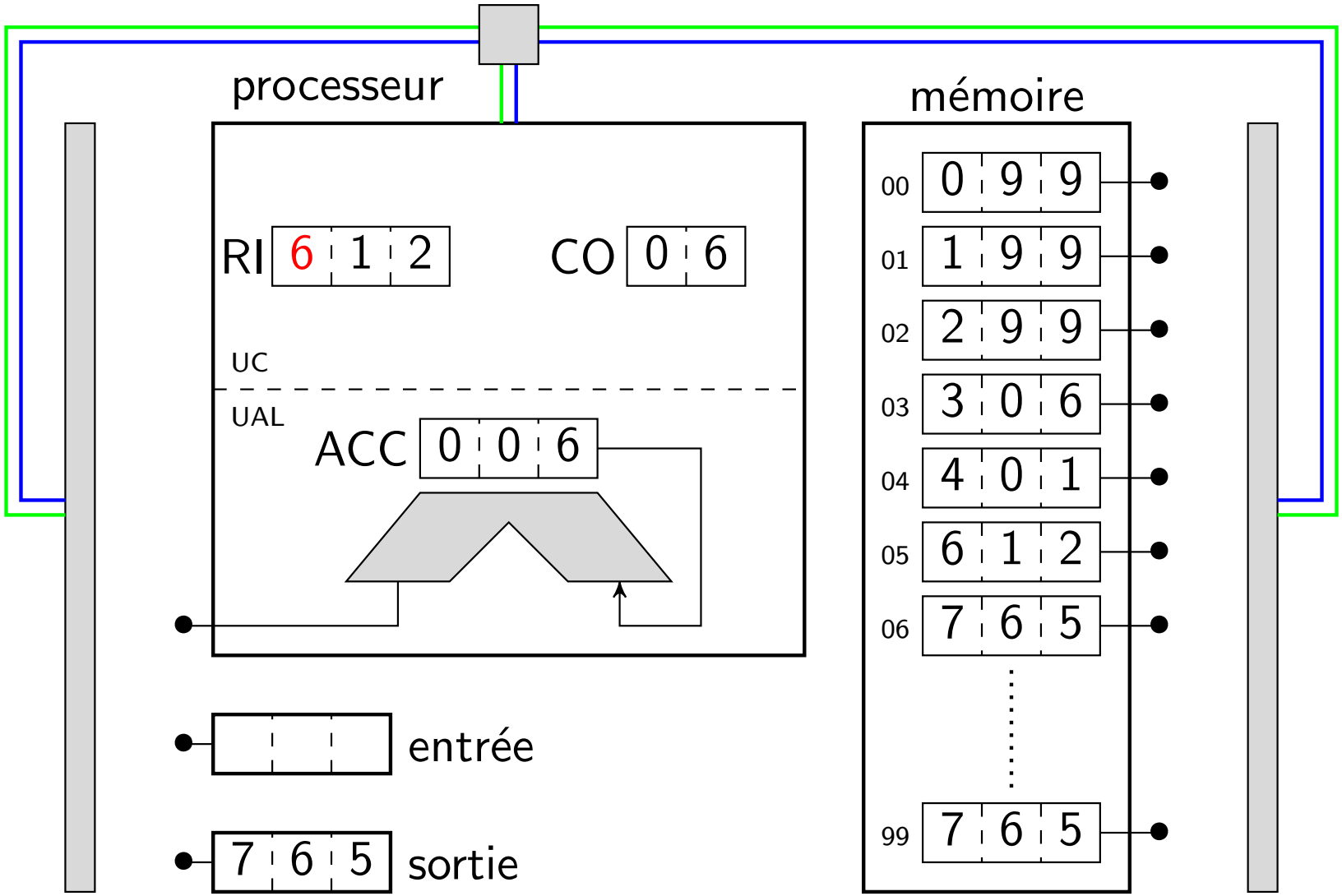
Ordinapoche - shift (code 6, assembleur SHT)



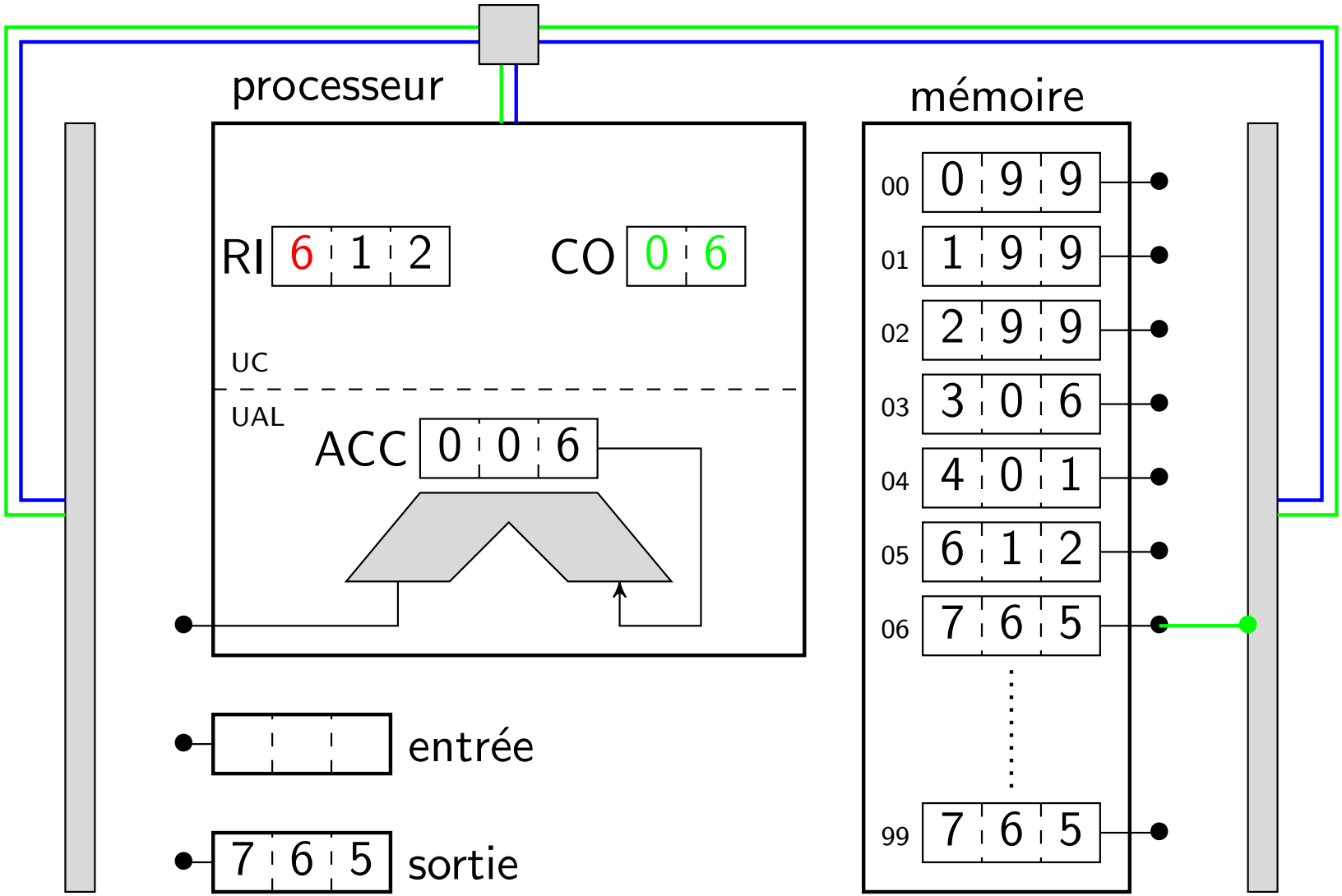
Ordinapoche - shift (code 6, assembleur SHT)



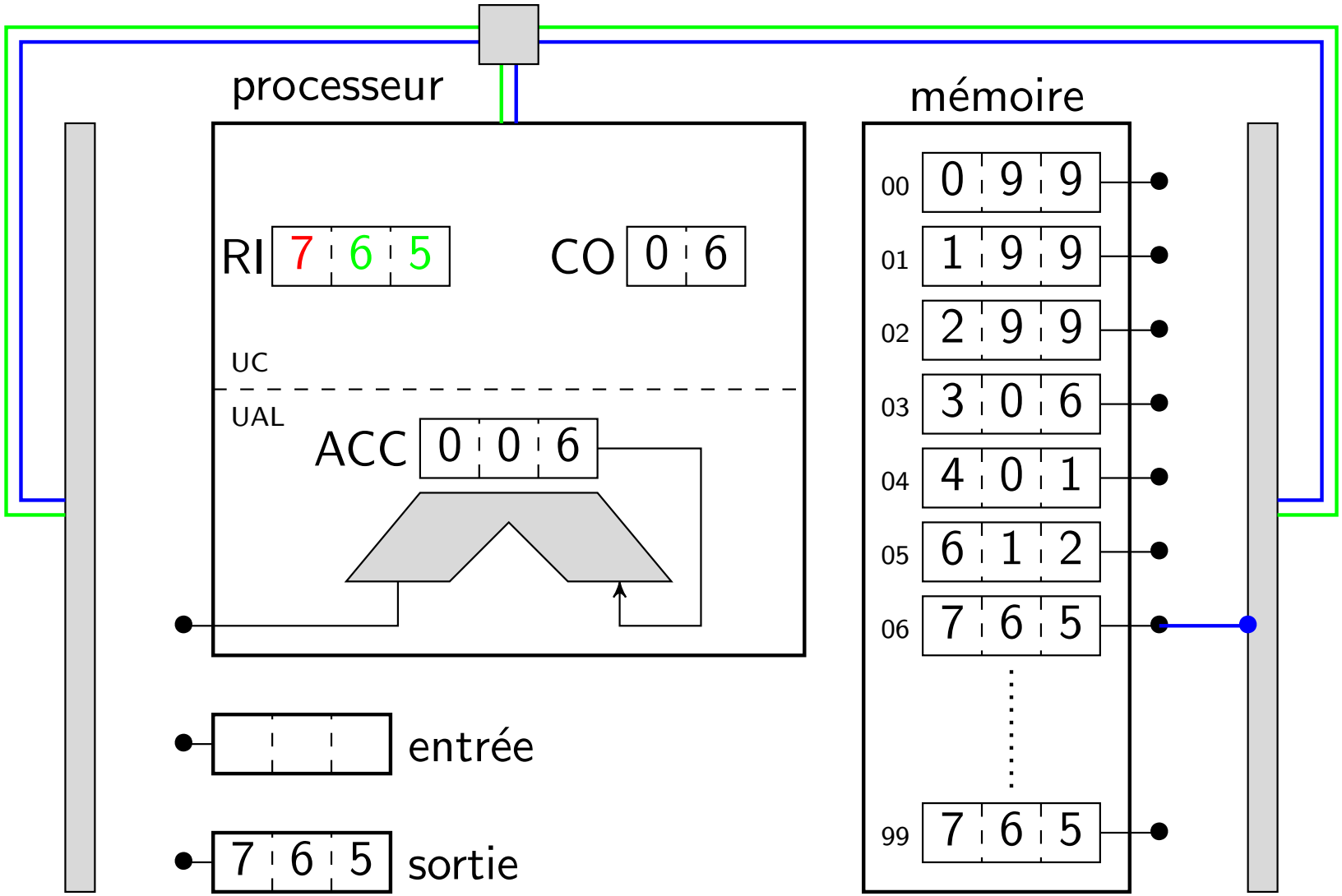
Ordinapoche - jump (code 7, assembleur JMP)



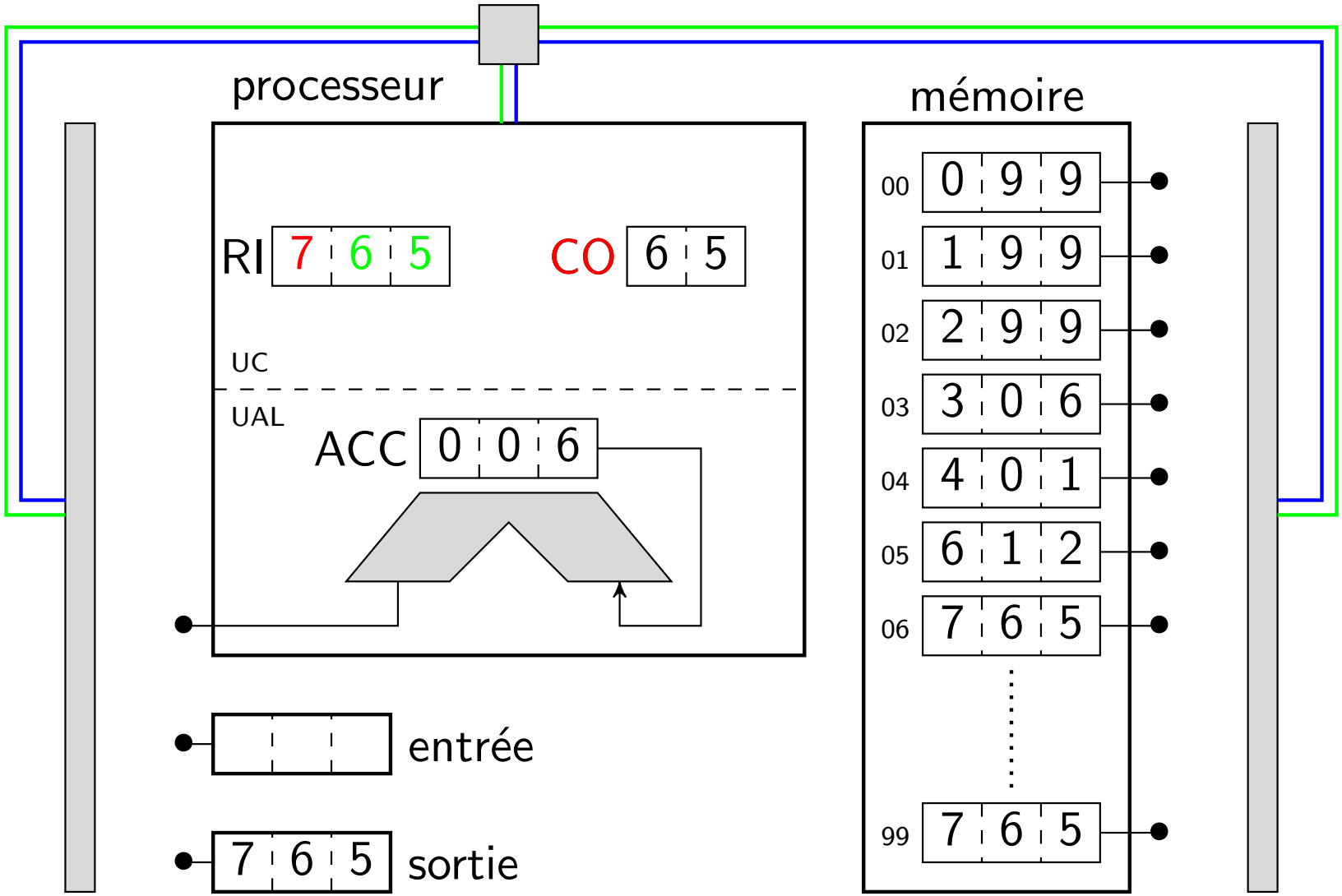
Ordinapoche - jump (code 7, assembleur JMP)



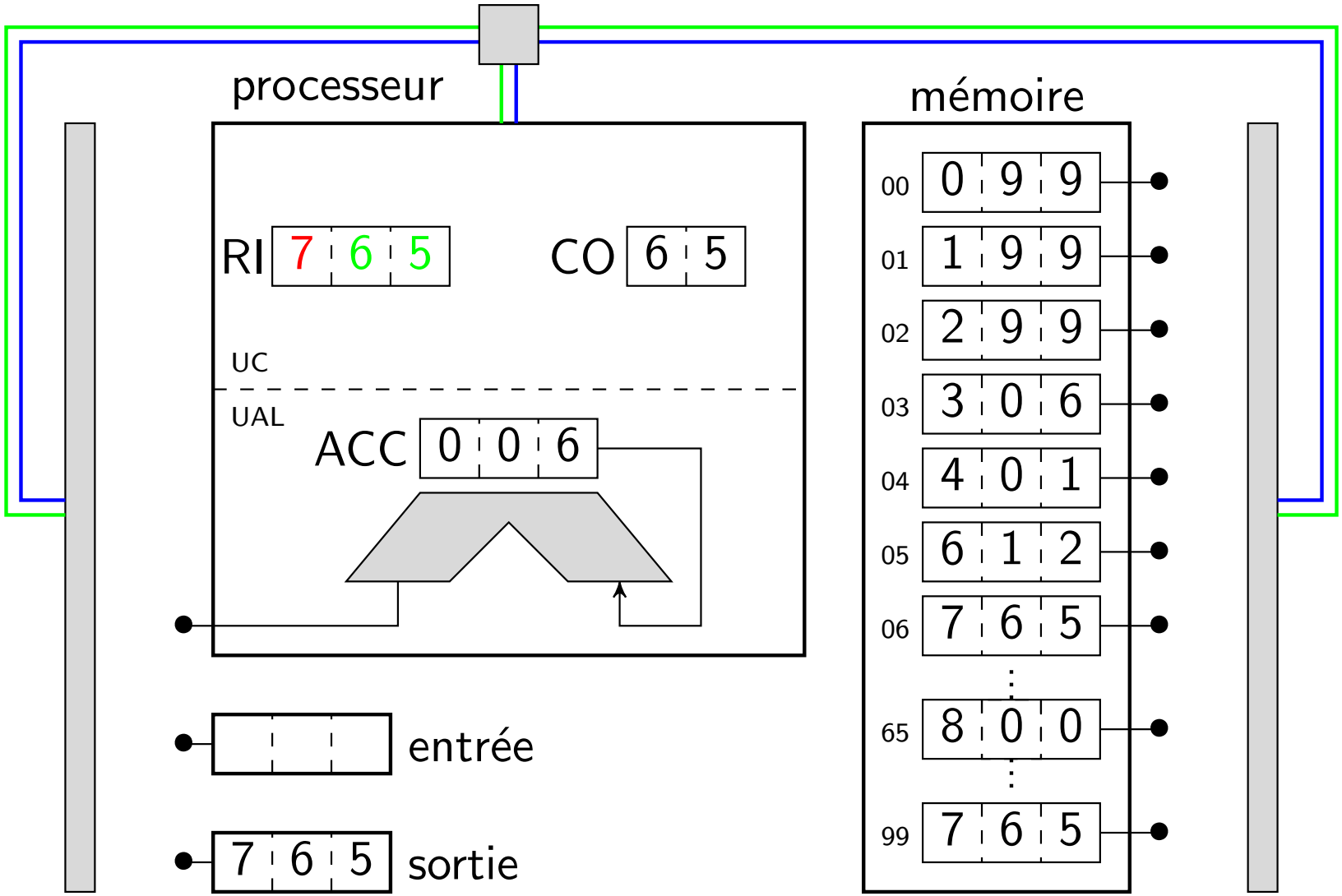
Ordinapoche - jump (code 7, assembleur JMP)



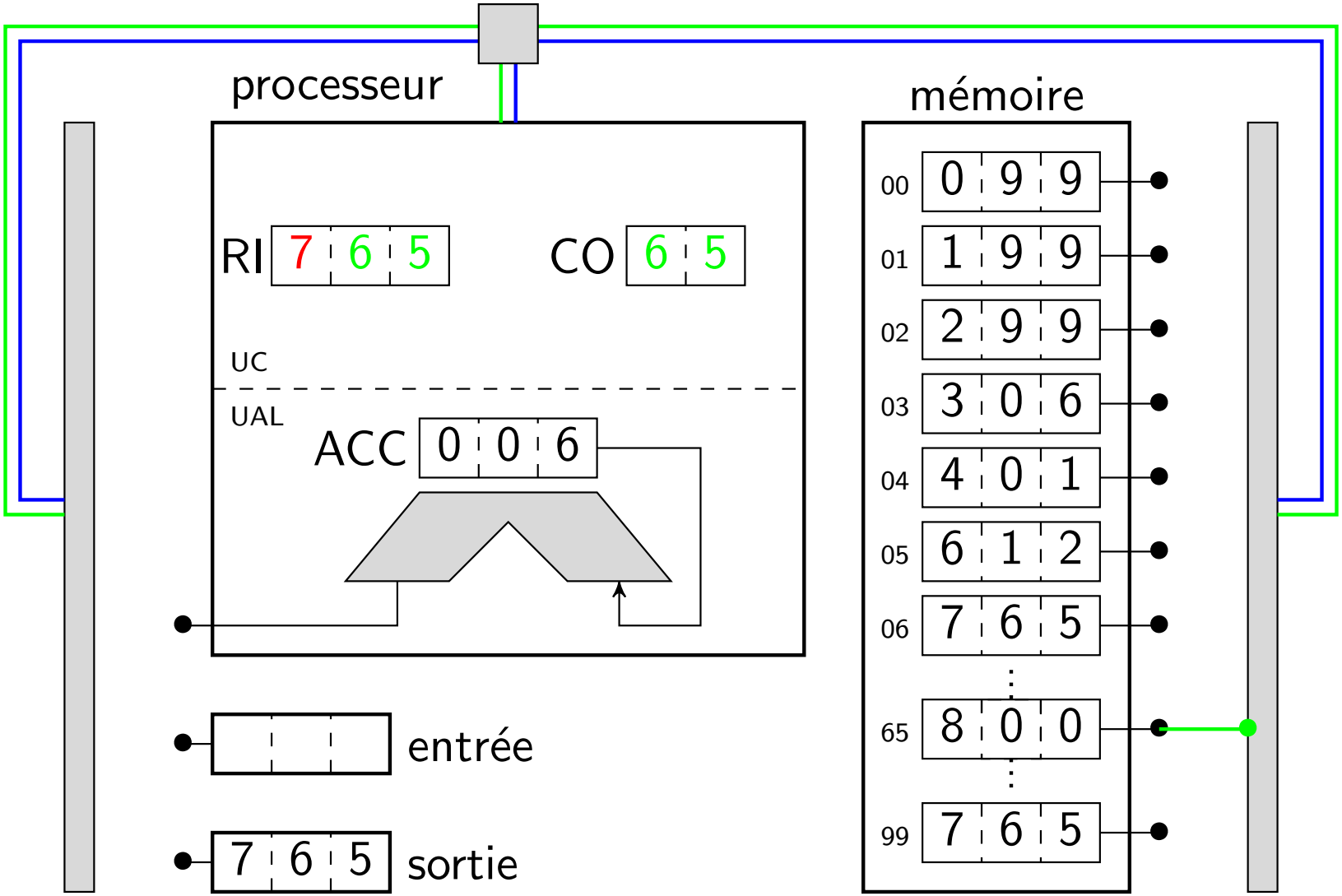
Ordinapoche - jump (code 7, assembleur JMP)



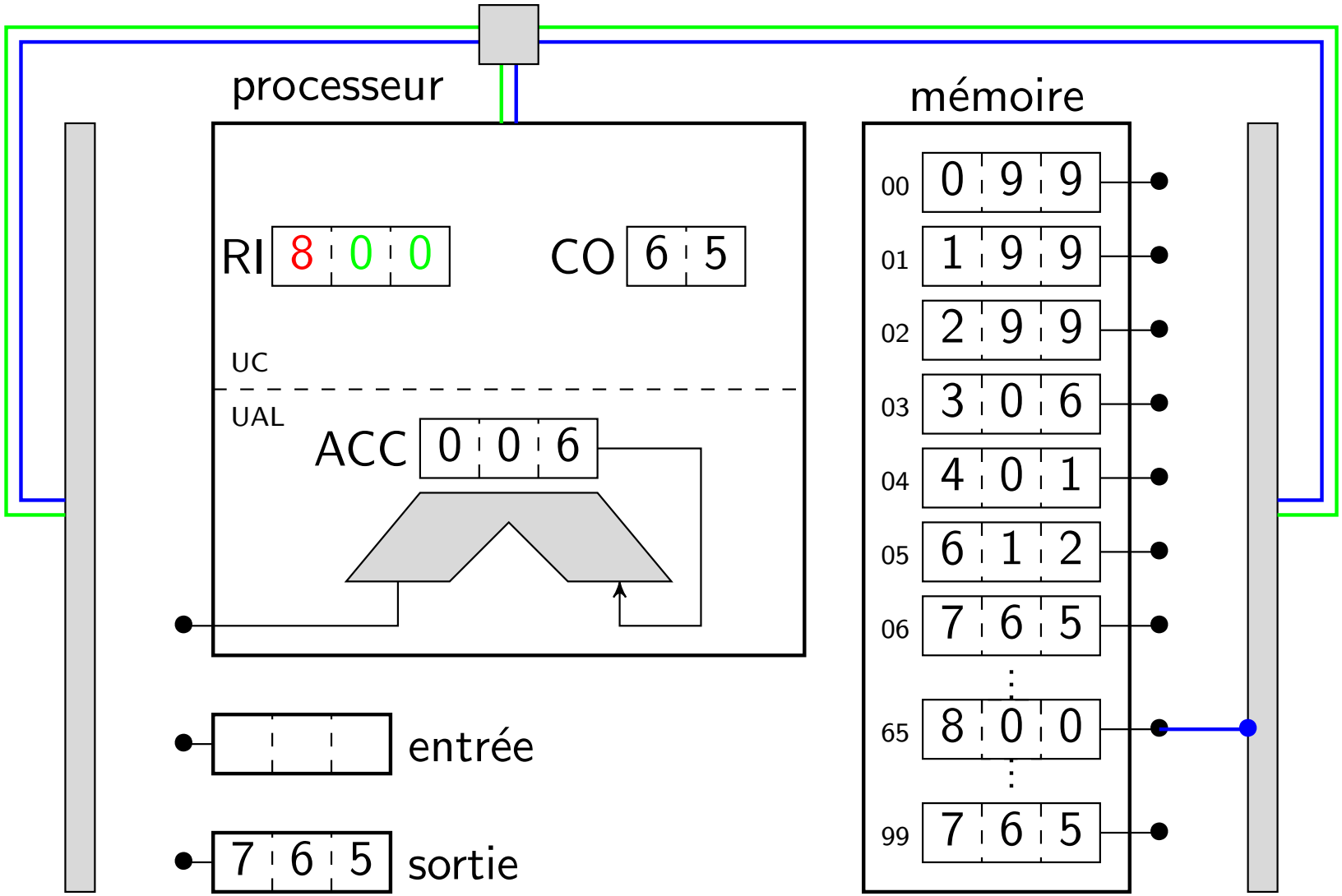
Ordinapoche - test acc. content (code 8, assembleur TAC)



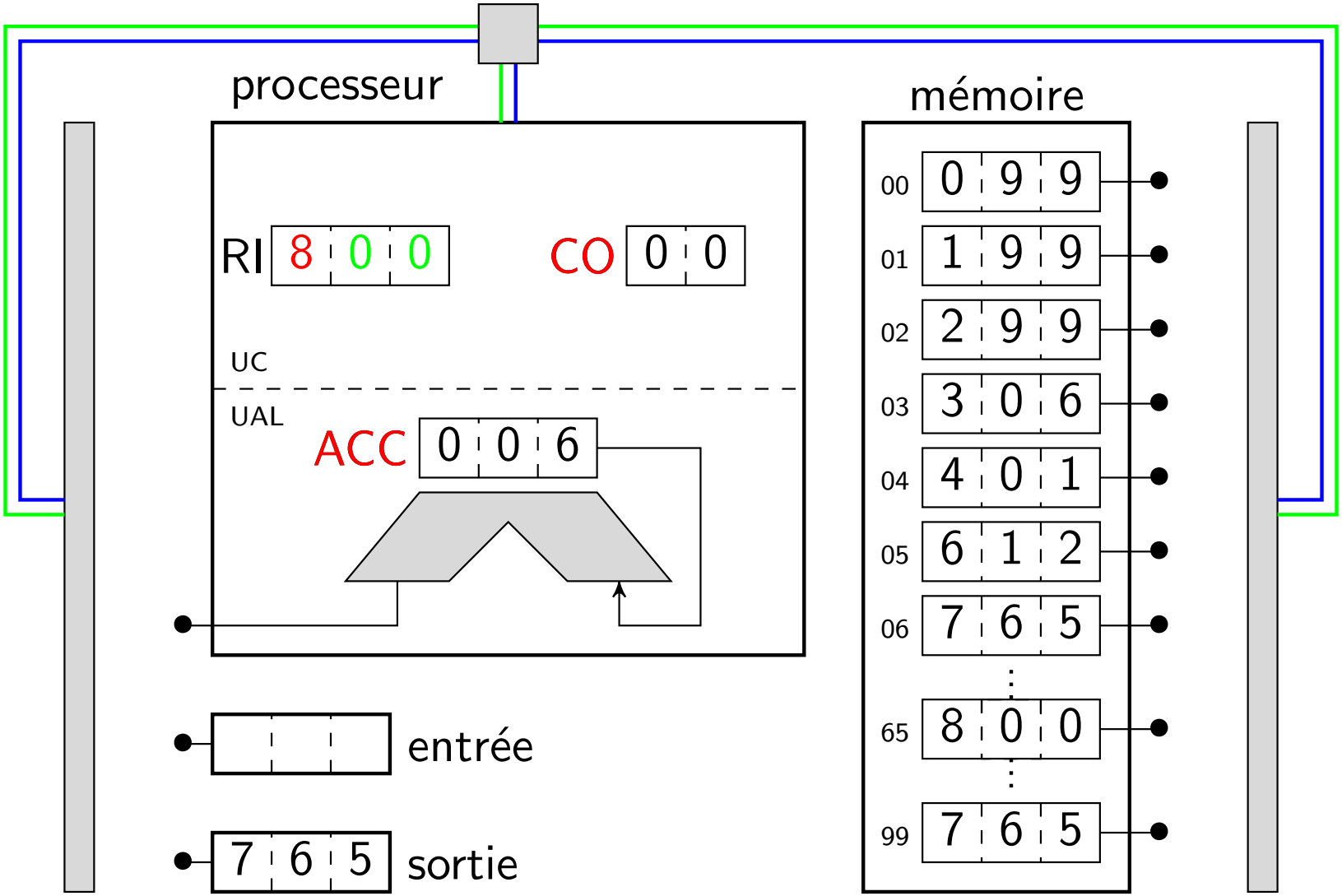
Ordinapoche - test acc. content (code 8, assembleur TAC)



Ordinapoche - test acc. content (code 8, assembleur TAC)



Ordinapoche - test acc. content (code 8, assembleur TAC)



Ordinapoche - halt and reset (code 9, assembleur HRS)

Instruction de fin de programme.

ATTENTION

Un programme sans cette instruction est a priori faux.

Ordinapoche - un premier programme

Adresse	Contenu	Commentaires
00	010	lire A
01	011	lire B
02	210	initialiser l'ACC et additionner A
03	411	additionner B
04	312	mémoriser le résultat dans S
05	112	afficher S
06	900	arrêter et remettre à l'état initial
⋮	⋮	⋮
10	?	donnée A
11	?	donnée B
12	?	donnée S