Université Grenoble Alpes
MIAGE L3 – Architectures des Ordinateurs

Examen, première session
Décembre 2022, Durée 1h
Sujet Architecture des ordinateurs sur 1 recto-verso.

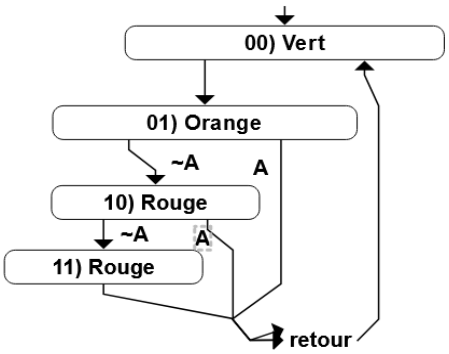
I. Automates (barème indicatif : 6 points)

L'automate ci-contre commande un feu de circulation.

Q1. Donner la table de vérité de sa fonction de transition (les états codés sur 2 bits, sont donnés sur la figure, en entrée 1 seul signal A, ~A est son complémentaire)

Q2. Donner la table de vérité de sa fonction de sortie (3 signaux : V(ert), O(range), R(ouge))

Q3. Donner la forme du circuit général et quelques éléments représentatifs de chaque partie du circuit.



II. Plus Un. (barème indicatif : 8 points)

L'exercice consiste à transformer le programme ci-après en circuit. Au choix, vous pourrez le traduire sous forme de circuit à flot de données, ou sous forme de circuit PC-PO. Si vous choisissez la réalisation d'un circuit PC-PO, après avoir donné la « partie opérative » et l'automate formel de la « partie contrôle », vous pourrez réduire le dessin du circuit lui-même à quelques éléments représentatifs de chaque partie du circuit.

```
Entrée : X entier naturel sur n bits (prendre n=8 si nécessaire)
Sortie : R entier naturel sur n bits
Début
 | Tmp ← 1
 | Tant que X & Tmp :
 |    | X ← X & Tmp
 |    | Tmp ← Tmp x 2
 |    └ FinTantQue
 | R ← X | Tmp
 └ Fin
```

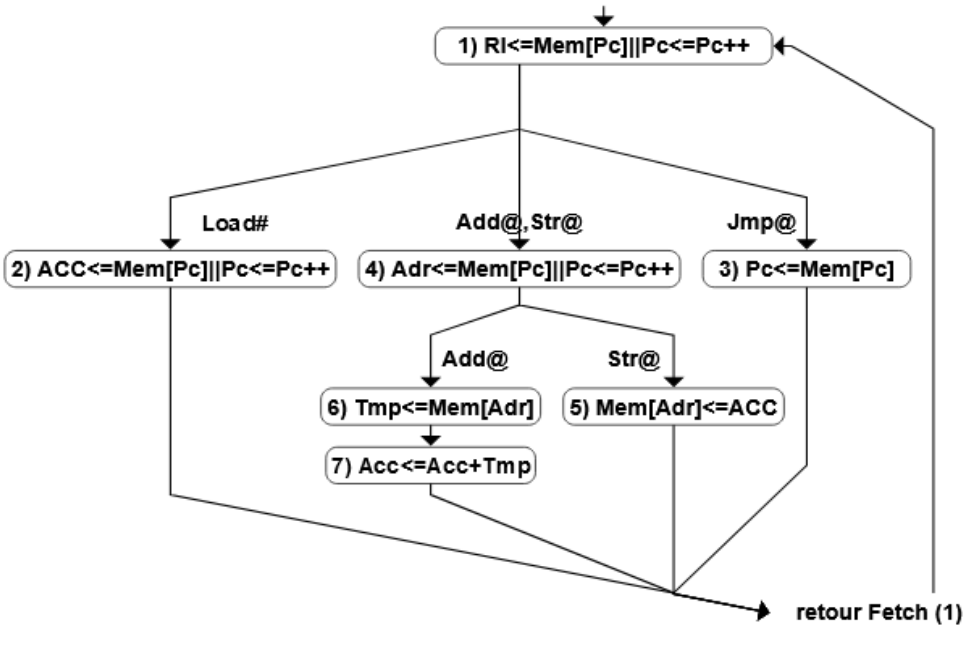
III. Machine à 4 instructions. (barème indicatif : 6 points)

L'exercice concerne une machine à accumulateur similaire à celle vue en cours. Elle comporte 4 instructions dont les codes et opérandes sont donnés sur 8 bits (code en héxadécimal ci-dessous) et dont les descriptions sont comme en cours :

- ld# vi : chargement d'une valeur vi dans l'accumulateur (adressage immédiat), code 0x01
- st@ ad : rangement de l'accumulateur à l'adresse ad (adressage direct), code 0x02
- ad@ ad : augmentation de l'accumulateur avec la valeur à l'adresse ad (adressage direct), code 0x04
- jp@ ad : saut inconditionnel à l'adresse ad (adressage absolu), code 0x08

L'UAL est capable de deux opérations (addition et soustraction), mais une seule est accessible au programmeur (l'addition) ; l'UAL fournit en sortie l'indicateur N, vrai ssi une opération donne un résultat négatif.

L'automate de contrôle est donné par le schéma ci-contre (le même qu'en cours).



Pour remplacer l'instruction de saut inconditionnel (jp) par une instruction de saut conditionnel (jn : saut si opération avec résultat négatif, sinon rien), dans un premier temps il faut sauvegarder l'indicateur N lors d'une opération (addition ou soustraction). Pour faire cela, une mémoire de 1 bit a été ajoutée à la PO du processeur, reliée à N en entrée, commandée par un signal ecrN (ecrN = 1 => sauvegarde de N) que la PC doit activer et fournissant en sortie un signal NS qui sera fourni à la PO en plus de N.

Q1. Ajouter la commande ecrN dans l'automate donné pour sauvegarder N quand une addition est faite (ad@ ad). Préciser ce qu'il faut faire au niveau du schéma de l'automate. Indiquer les changements à faire au niveau de la fonction de transition et de la fonction de sortie de la PC pour ajouter cette fonctionnalité. Dessiner la petite partie de circuit qu'il faudra ajouter à la PC pour commander la sauvegarde de N dans la PO.

Second temps : transformer l'automate pour que le saut inconditionnel (jp) soit remplacé par un saut conditionnel (jn).
Q2. Préciser ce qu'il faut faire au niveau du schéma de l'automate. Indiquer les changements à faire au niveau de la fonction de transition et de la fonction de sortie de la PC.

Éléments de correction

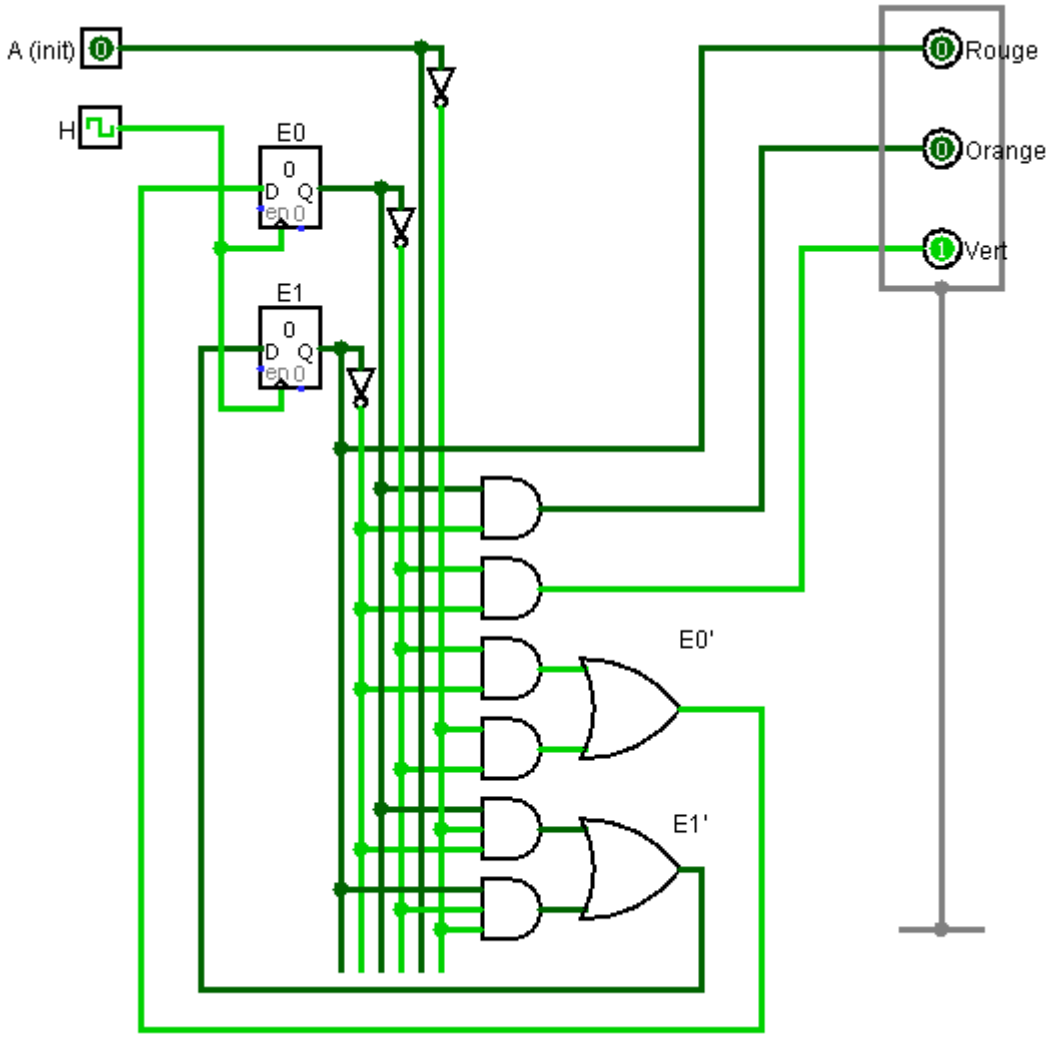
I) Q1) Transition

Etat1	Etat0	A	Etat1’	Etat0’
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

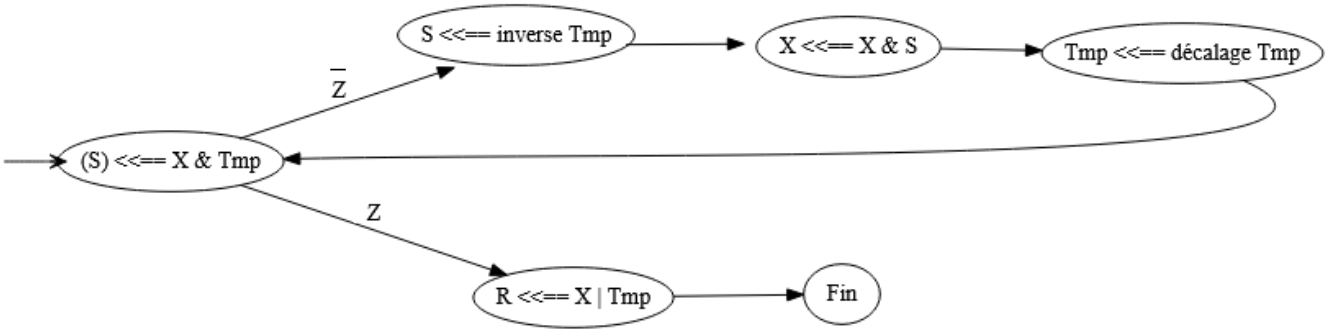
Q2) Sortie

Etat1	Etat0	V(ert)	O(range)	R(ouge)
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	0	0	1

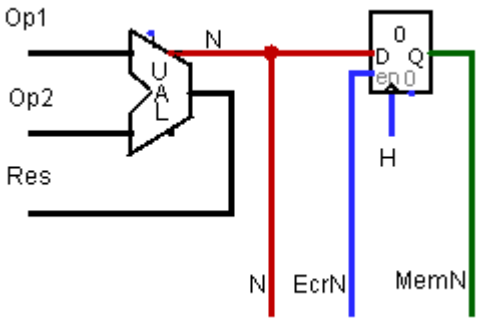
Q3) Circuit



II) Plus Un, par programme, version PC-PO
PO : prendre une PO comme en cours, avec 4 registres : X (initialisé à sa valeur initiale), Tmp initialisé à 1, R (non initialisé) et un registre supplémentaire S pour calculer l'inverse de Tmp ; et 4 opérations : et (&), ou (|), inverse, décalage à gauche (x2) ; fournissant en sortie d'UAL l'indicateur Z.
PC : l'automate formel suivant (à traduire sous forme de circuit)



III) Saut conditionnel
1) Dessin de memN à la sortie de l'UAL (PO), vers l'automate (PC)



Changement au niveau de l'automate formel : dans l'état 7, il faut activer le signal ecrN (et pas dans les autres états) ; fonction de transition : pas de modification ; fonction de sortie : il faut produire ce signal ecrN, si l'état est sauvegardé sur 3 bits, cela correspond à ecrN = E2 . E1 . E0 (à ajouter au schéma précédent), si l'état est sauvegardé sur 4 bits : ecrN = $\overline{E3}$. E2 . E1 . E0

2) Changement de l'automate formel : remplacer la transition sur jmp entre l'état 1 et l'état 3 par une transition sur (jneg & memN) toujours entre l'état 1 et l'état 3 ; ajouter une transition entre l'état 1 et un nouvel état 8 sur (jneg & $\overline{memN}$) ; dans ce nouvel état 8, faire juste l'action PC++ ; ajouter une transition entre ce nouvel état 8 et l'état 1.
Changement de la fonction de transition : modifier la transition 1=>3 ; ajouter une transition 1=8 ; ajouter une transition 8=>1.
Changement de la fonction de sortie : ajouter l'action PC++ sur l'état 8.