

Séance TD n° 7 : Le séquenceur:

Exo 1:

I - Supposant que les instructions sont de format court:

1/ Dérouler les instructions suivantes et détailler les micro-commands:

* ADD Adr (Addition en mode Direct):

Remarque:

Lect/Ecrit toujours dans le début de ligne (on les lance après le traitement de la sortie et l'entrée)

| Étape | Micro instructions | Micro commandes |
|--|---|---|
| Préface de l'instruction | (CO) → RAM Lecture, $CO \leftarrow (CO) + 1$ (RIM) → RI | SCO, ERAM LECTURE, INCO SRIM, ERI |
| Décodage | DECODAGE | DEC |
| Instructions effectif de l'instruction | (RI @) → RAM Lecture (RIM) → UAL1, (ACC) → UAL2, ADD, RES → ACC IF RES → ACC | SPI, ERAM LECTURE SRIM, EVAL1, EVAL2, ADD, EACC C... |

* RGM ADR (rangement en mode Direct du contenu de l'ACC dans le mot mémo d'@ ADR):

| | Micro instructions | Micro commandes |
|---|---|---|
| 1 | (CO) → RAM Lecture, $CO \leftarrow (CO) + 1$ (RIM) → RI | SCO, ERAM LECTURE, INCO SRIM, ERI |
| 2 | DECODAGE | DEC |
| 3 | (RI @) → RAM, (ACC) → RIM Ecriture | SPI, ERAM, SACC, ERIM ECPITURE |

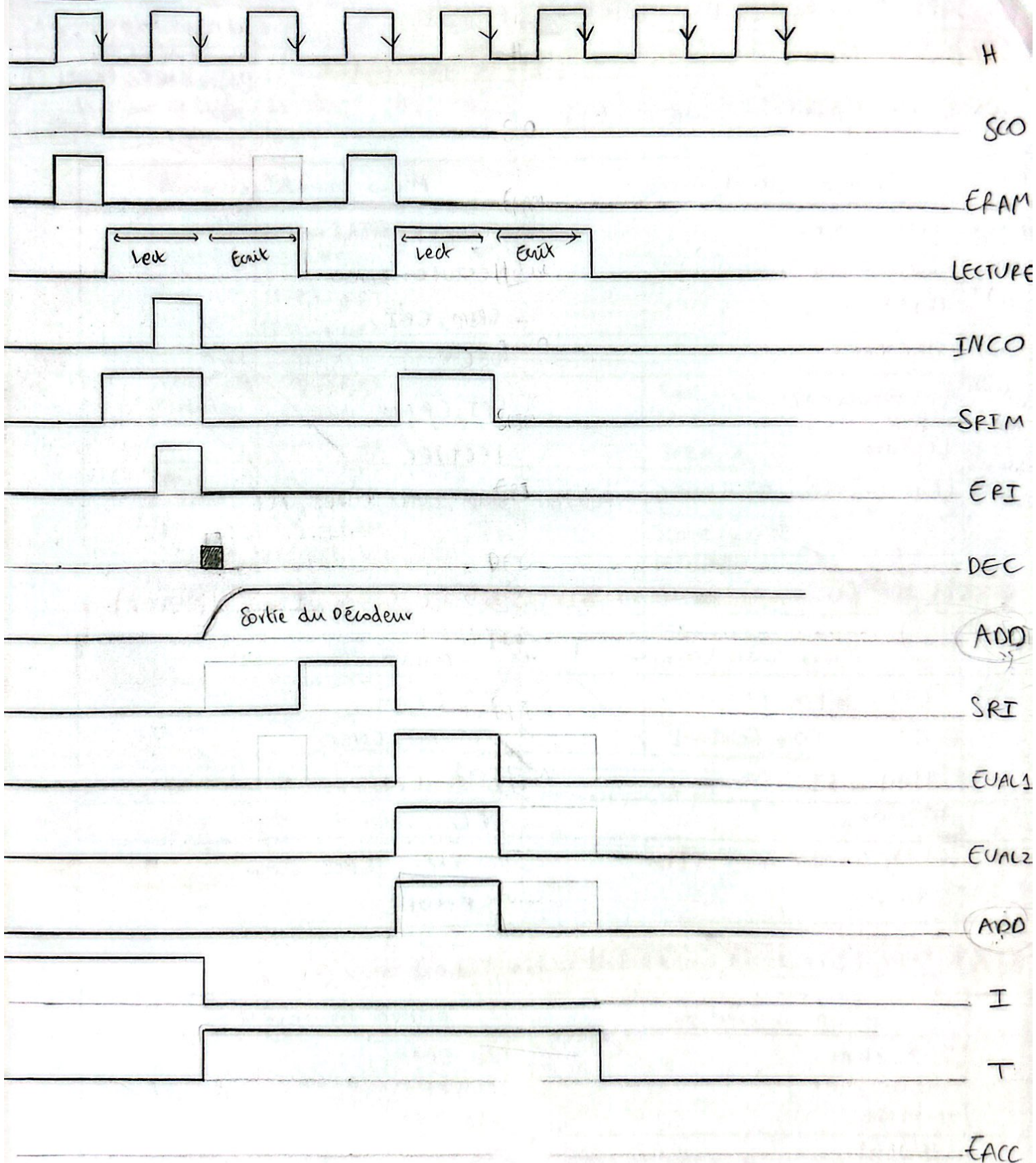
* RI *ADR (Branchement inconditionnel en mode indirect à une @):

| | Micro instructions | Micro commandes |
|---|---|---|
| 1 | (CO) → RAM Lecture, $CO \leftarrow (CO) + 1$ (RIM) → RI | SCO, ERAM LECTURE, INCO SRIM, ERI |
| 2 | DECODAGE | DEC |
| 3 | (RI @) → RAM Lecture (RIM) → CO | SPI, ERAM LECTURE SRIM, TDA, ECO |

2/ Tracer les chronogrammes des instructions :

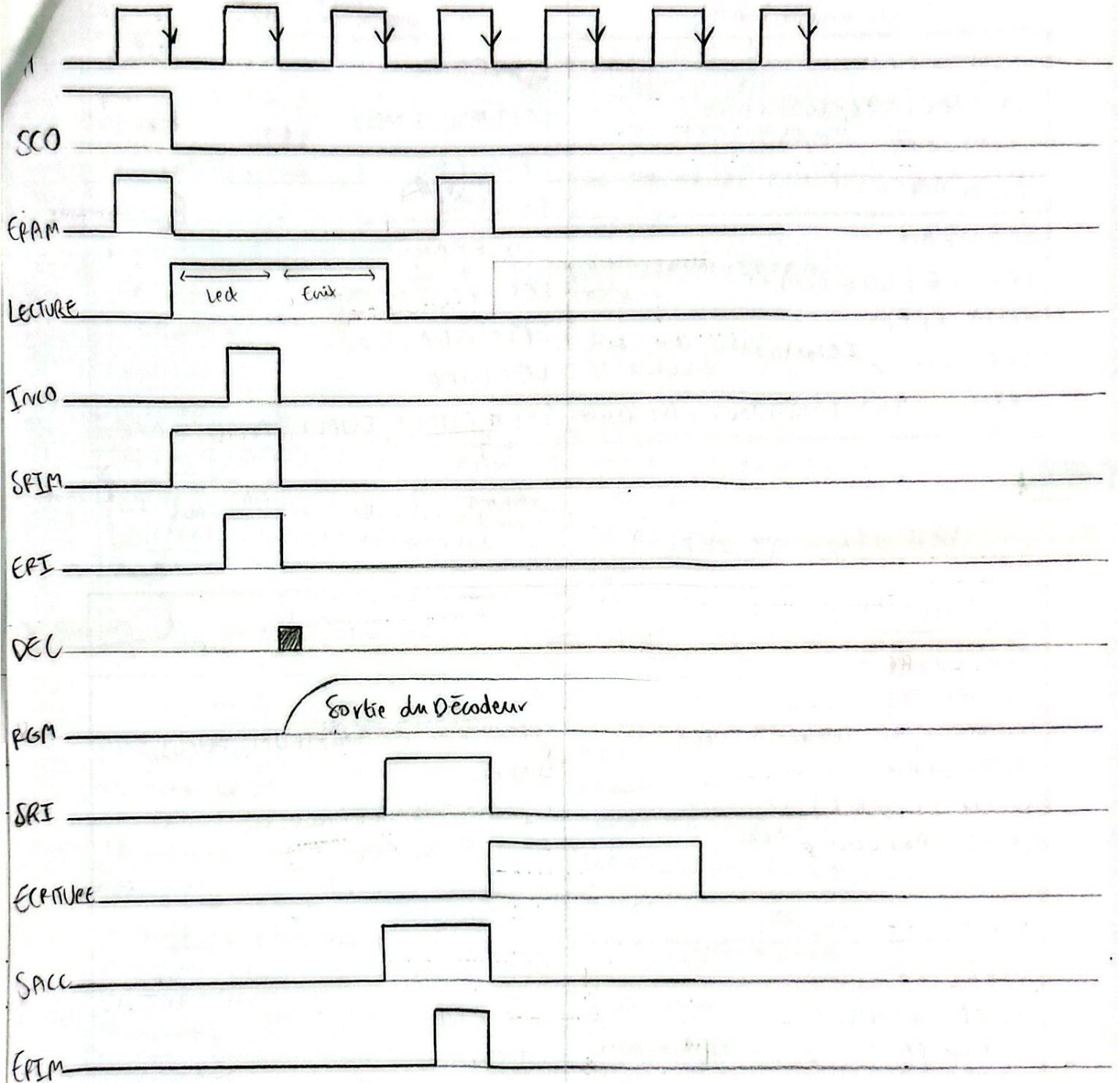
* ADD Adv :

→ PC (M) ADD P :



M ADR:

RGM ADR



3

II - AND *ADP (ET logique en mode indirect entre l'ACC et le mot pointé par l'@).

| Micro instructions | Micro commandes |
|---|---|
| (CO) → PAM LECTURE, CO ← (CO) + 1 (PI) → PI | SCO, EAM LECTURE, INCO SPIM, EPI |
| DECODAGE | DEC |
| (CO) → RAM LECTURE, CO ← (CO) + 1 (PI) → RAM LECTURE \leftarrow $\begin{matrix} \text{Erase to add a:} \\ \text{(PI) → RAM} \\ \text{Lecture} \end{matrix}$ (PI) → UAL1, (AQ) → UAL2, AND, EACC | SCO, EAM LECTURE, INCO SPIM, TDA, EAM LECTURE SPIM, EUAL1, EUAL2, AND, EACC |

En02:

1/ Addition entre P1 et P2: ADD P1, P2

Remarque: quand on veut mettre la valeur de UAL₁ dans l'ACC, on fait "CHM"

| Micro Instructions | Micro commandes |
|--|--|
| PAMREG ← 0 LECTURE BREG (PIMBREG) → RAM, (PIMBREG) → UAL1, CHM (ACC) + 1 → ACC Lecture MC, (ACC) → PIMBREG (PI) → PI, Ecriture BREG | PAZ LEC SBREG, TRMC, EAM, TRUAL, EUAL1, CHM 4 ACC L, SACC, TRUAL, EBREG SRIM, EPI, ECR |
| DECODAGE | DEC |
| (PI, RA) → PAMREG Lecture BREG (PIMBREG) → UAL1, $\begin{matrix} \text{CHM} \\ \text{ACC} \leftarrow \text{UAL1} \end{matrix}$ (PI, RB) → PAMREG Lecture BREG (PIMBREG) → UAL2, (ACC) → UAL1, ADD | SPA, ENRG LEC <u>SBREG, TRUAL, EUAL1, CHM</u> <u>SRB, ENRG</u> LEC SBREG, TRMC, TAD1, EUAL2, SACC, EUAL1 ADD |

Ask the teacher if it is possible to just leave the content of RA dans l'UAL1
sans le changer dans l'ACC, pour ensuite le remettre dans l'UAL1?

answer: You can't, because it'll fade away and you'll lose the info ...

4

addition entre un registre et un mot mémoire :

ADD Ri, ADR

| Micro Instructions | Micro Commandes |
|--|---|
| $RAMREG \leftarrow 0$ LECTURE BREG $(RIMBREG) \rightarrow RAM$ $(RIMBREG) \rightarrow UAL1, ACC \leftarrow (UAL1)$ $ACC \leftarrow (ACC) + 1$ $RIMBREG \leftarrow (ACC)$ ECRITURE BREG LECTURE MC $(RIM) \rightarrow PI$ | RAZ LEC SBREG, TRMC, ERAM SBREG, TRUAL, EUAL1, CHM +1ACC SACC, TRUAL, EBREG ECR L SPIM, EPI |
| DECODAGE | DEC |
| $(PI.RA) \rightarrow RAMREG, (PI.ADR) \rightarrow RAM$ LECTURE BREG, LECTURE MC $(RIMBREG) \rightarrow UAL1, (RIM) \rightarrow UAL2$ ADD | SRA, ENRG, SADR, ERAM LEC, L SBREG, TRUAL, EUAL1, SPIM, EUAL2 ADD |

3/ Branchement à un sous-programme avec indirection et sauvegarde de l'@ de retour dans R1 :

| Micro Instructions | Micro Commandes |
|---|---|
| $RAMREG \leftarrow 0$ LECTURE BREG $(RIMBREG) \rightarrow RAM$ $(RIMBREG) \rightarrow UAL1, ACC \leftarrow (UAL1)$ $ACC \leftarrow (ACC) + 1$ $(ACC) \rightarrow RIMBREG, RAMREG \leftarrow (RAMREG) + 1$ ECRITURE BREG LECTURE MC $(RIM) \rightarrow PI$ | RAZ LEC SBREG, TRMC, ERAM SBREG, TRUAL, EUAL1, CHM +1ACC SACC, TRUAL, EBREG, +1 ECR L SPIM, EPI |
| DECODAGE | DEC |
| $(PI.ADR) \rightarrow RAM$ LECTURE MC $(RIM) \rightarrow RAM$ LECTURE MC $(RIM) \rightarrow RIMBREG$ $RAMREG \leftarrow 0$ ECRITURE BREG | SADR, ERAM L SPIM, TDA, ERAM L SPIM, TOA, TRMC, EBREG RAZ ECR |

9 BI * ADR

5

En03:

1/ Microprogramme :

| μ -programme | Étape | μ -Instructions | μ -Commandes |
|------------------|--------------------------------------|---|--|
| BI ADR | Pedache de l'instruction | $(CO) \rightarrow RAM$ LECTURE, $(CO \leftarrow (CO) + 1)$ $(RIM) \rightarrow RI$ | SCO, ERAM LECTURE, INCO SPIM, ERI |
| | Décodage | DEC | DEC |
| | Traitement effectif de l'instruction | $(CO) \rightarrow RAM$ LECTURE, $(CO \leftarrow (CO) + 1)$ $(RIM) \rightarrow RT$ $(RT) \rightarrow CO$ | SCO, ERAM LECTURE, INCO SPIM, ET1 ST1, ECO |
| BC *ADR | Pedache de l'instruction | $(CO) \rightarrow RAM$ LECTURE, $(CO \leftarrow (CO) + 1)$ $(RIM) \rightarrow RI$ | SCO, ERAM LECTURE, INCO SPIM, ERI |
| | Décodage | DEC | DEC |
| | Traitement effectif de l'instruction | $(CO) \rightarrow RAM$ LECTURE $(RIM) \rightarrow RT$ $(RT) \rightarrow RAM$ LECTURE $(RIM) \rightarrow RT$ $(RT) \rightarrow CO$ | SCO, ERAM LECTURE SPIM, ET1 ST1, ERAM LECTURE SPIM, ET1 ST1, ECO |

Remarque : BI et BC se font de la même manière, c'est pas important ici de savoir si la condition s'est réalisée ou non.

2/ Les chronogrammes :

6