ES102/PC8 : énoncé

1) De spécifications comportementales en diagrammes d'état

*Maîtrisant désormais les étapes **2** (encodage d'états) et **3** (implantation) de la synthèse d'un circuit séquentiel synchrone, on pratique ici l'étape **1** : l'élaboration du diagramme d'état. Celui d'une unité de commande découle facilement du diagramme algorithmique. Mais en général, il faut travailler avec des spécifications comportementales bien moins formalisées…*

1a) Soit un barreau de 3 LEDs, commandées respectivement de gauche à droite par les signaux g, m et d (m=1 allume la diode du **m**ilieu). Proposer un diagramme d'état permettant de voir un point lumineux osciller entre les deux extrémités, ce point sautant d'une LED à sa voisine à chaque top d'horloge.

1b) Le code Morse représente les lettres de l'alphabet sous la forme de chaînes de points (·) et de traits (-), de longueurs variées. Ainsi, la lettre A est codée par un point suivi d'un trait (·-).

L'arbre ci-contre montre le code de chaque lettre. Une arête représente respectivement un point ou un trait selon qu'elle descend vers la gauche (trait fin) ou vers la droite (trait épais). Le message « SOS » s'écrit donc : ··· --- ···.

Des espaces, appelés séparateurs, sont nécessaires entre codes de lettre (sinon on ne saurait pas si le message « SOS » débute par E, I ou S).

Il existe aussi un séparateur de mots, mais qu'on ignorera ici. On considère un signal Morse *m* synchrone (1 bit reçu à chaque top d'horloge) simplifié par rapport au standard, tel que :

- un point est représenté par un 1 isolé reçu entre deux 0 : 010 ;
- un trait est représenté par deux 1 consécutifs reçus entre deux 0 : 0110 ;
- au sein d'une même lettre, les 0 sont partagés entre points/traits successifs, donc isolés
- les lettres sont séparées par au moins deux 0 consécutifs (séparateur) ;
- la réception de plus de deux 1 consécutifs constitue une erreur.

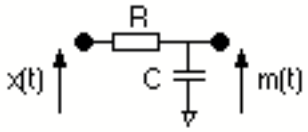
Un message « SOS » seul serait donc reçu comme …00000101010011011011001010101000…

Proposer un diagramme d'état de circuit séquentiel ayant comme signal d'entrée *m* et comme signaux de sortie *p*, *t*, *s* et *e*. Ces derniers doivent indiquer respectivement l'apparition d'un point, d'un trait, d'un séparateur ou d'une erreur, en passant à 1 sur une unique période d'horloge et dès que possible. Un tel circuit constituerait un premier étage d'interprétation du signal *m*, à l'attention d'un deuxième reconnaissant les lettres elles-mêmes.

2) Additionneur à retenues propagées (ARP) *pipeliné*

Pipeliner un ARP pour pouvoir doubler (enfin presque) sa puissance de calcul.

3) Système détecteur d'instabilité : 2 versions



Le petit montage analogique ci-contre est régi par l'équation différentielle (*ED*) **dm/dt=(x-m)/RC**. Il réalise un filtre passe-bas entre les signaux *x* et *m* ; et entre les signaux *x* et *x-m* (tension aux bornes de la résistance), un filtre passe-haut. Ce dernier permet de repérer des discontinuités affectant le signal *x*. Analytiquement, le signal *m* résulte de la convolution du signal *x* par un noyau positif unilatéral de type exponentiel décroissant : *m* est ainsi une *moyenne mobile* de *x*.

Une version en temps discret de l'ED ci-dessus est la loi d'évolution $\mathbf{m^+ = m + \varepsilon(x-m)}$ où ε, appelé « coefficient d'actualisation », est positif et typiquement petit devant 1. Les valeurs des signaux *x* et *m* peuvent aussi être discrétisées, sans altérer les propriétés de filtrage. On va donc les représenter sous forme numérique, avec d'éventuels chiffres après la virgule, indispensables en tout cas pour exprimer ε.

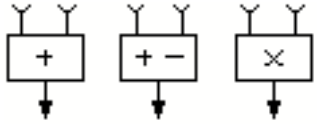
Un processus physique présentant un danger d'instabilité est observé via un capteur qui fournit à une fréquence f_{capt} la mesure numérique *x* d'une grandeur physique caractéristique. L'instabilité se manifeste par des hausses relatives soudaines de *x*, alors que ce signal varie lentement en temps normal. On va justement détecter ces instabilités :

- en tenant à jour une moyenne *m* de *x* grâce à la loi d'évolution précédente, avec $\boldsymbol{\varepsilon \approx 5\%}$;
- en signalant le système observé comme instable chaque fois que $\mathbf{(x-m) > \beta m}$, avec $\boldsymbol{\beta \approx 0{,}3}$.

Ceci demande d'effectuer des calculs à chaque nouvelle mesure fournie par le capteur, c'est-à-dire à la fréquence f_{capt}, d'où un caractère nativement séquentiel.

3a) On dispose d'un signal d'horloge CK_{capt} de fréquence f_{capt} et *x* est supposé synchrone par rapport à celui-ci. Rappeler ce que cela signifie.

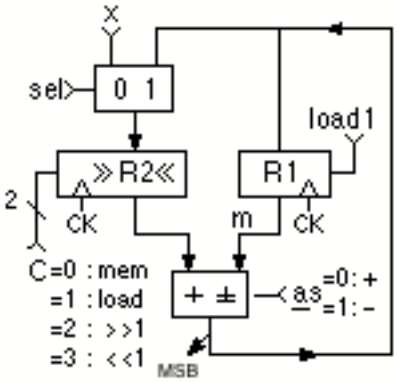
3b) Proposer un système séquentiel répondant directement au besoin exprimé, à un niveau numérique. Se servir notamment des briques combinatoires ci-contre (dont un multiplieur) en faisant comme si ε et β étaient des entiers.



3c) Mais ε et β ne sont pas des entiers. Utiliser des décaleurs *n* bits (>>n) et des additionneurs pour réaliser approximativement les multiplications correspondantes.

3d) Quel est le coût en transistors de ce détecteur d'instabilité si *x* et *m* sont sur 32 bits ?

Ci-dessus, les calculs relatifs à la détection d'instabilité ont été réalisés de façon combinatoire, l'horloge CK_{capt} servant seulement à mettre à jour la moyenne mobile *m* peu après chaque arrivée d'une nouvelle mesure *x*. On envisage désormais de réaliser ces calculs sur le chemin de données (CD) ci-contre, pour gagner en flexibilité. Ceci implique une séquentialisation plus fine en temps, par une horloge CK dont la fréquence *f* sera un multiple assez grand de f_{capt} (supposée modérée). Ce CD comporte deux registres cadencés par CK : R1 (entrée *load1*) servira à mémoriser la moyenne mobile *m*, tandis que R2 sera au cœur des calculs, équipé de décaleurs 1 bit vers la droite ou la gauche (cf. CM7/P7) avec commande C sur 2 bits (voir détails sur la figure). L'additionneur/soustracteur permet en outre de calculer au choix \$R2+\$R1 si as=0 ou \$R2-\$R1 si as=1 (*a* pour **a**ddition, *s* pour **s**oustraction). Le résultat peut être dirigé vers R1 (load1=1) ou R2 (sel=1 et C=1). Ce petit CD et son UC pourront être *embarqués* à bord du capteur pour en faire un *capteur intelligent*.



3e) A priori, pourquoi R1, porteur de *m*, ne peut-il se réduire à une simple bascule D numérique ?

3f) Ce chemin de données coûte-t-il plus ou moins cher que la solution combinatoire précédente ?

3g) Partant d'un top d'horloge où une nouvelle valeur de *x* est fournie par le capteur, établir la séquence d'opérations nécessaire pour implanter la loi d'évolution de *m*.

3h) Poursuivre la « programmation » du CD pour pouvoir détecter s'il y a instabilité.

3i) A quoi ressemblerait le diagramme d'état de l'UC nécessaire pour piloter le CD considéré ?