Architecture 1 :
Circuits Numériques et Éléments d'Architecture
Partiel de mi-semestre

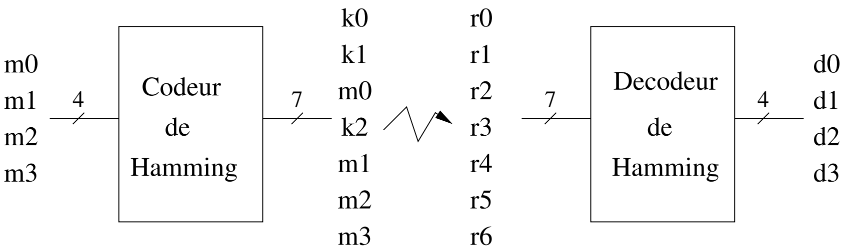
ENSIMAG 1A

Année scolaire 2009–2010

- Durée : 2h. Tous documents et calculatrices autorisés.
- Le barème est donné à titre indicatif.
- Les exercices sont indépendants et peuvent être traités dans le désordre.

Ex. 1 : Circuits pour un codeur/décodeur de Hamming (6pts)

Le code de Hamming est utilisé en transmission de données pour détecter et corriger des erreurs.



Les données à transmettre sont groupées par blocs de 4 bits appelés messages m de 4 bits $(m_0 \cdots m_3)$. Pour permettre la correction, on ajoute à chaque bloc, 3 bits de redondance k $(k_0 k_1 k_2)$ obtenus par :

$$\begin{cases} k_0 = & (m_0 + m_1 + m_2) \mod 2 \\ k_1 = & (m_0 + m_2 + m_3) \mod 2 \\ k_2 = & (m_1 + m_2 + m_3) \mod 2 \end{cases}$$

Le bloc issu du codeur est un vecteur de 7 bits agencé comme suit : $(k_0 k_1 m_0 k_2 m_1 m_2 m_3)$. Ce vecteur pouvant subir des erreurs de transmission. On appellera $R = (r_0 \cdots r_6)$ le vecteur reçu par le décodeur et $D = (d_0 \cdots d_3)$ le message décodé.

Question 1 Rappeler la table de vérité de l'addition à deux entrées modulo 2 et la porte élémentaire correspondante.

XOR

Question 2 Construire un circuit implantant le codeur de Hamming présenté avec des portes à deux entrées de la question précédente.

5 portes XOR à 2 entrées. 6 est sous-optimal.

Question 3 Pour réaliser un décodeur de Hamming, il est nécessaire de calculer le syndrome S du vecteur reçu par le décodeur définit par les équations suivantes :

$$\begin{cases} s_0 = & (r_0 + r_2 + r_4 + r_6) \mod 2 \\ s_1 = & (r_1 + r_2 + r_5 + r_6) \mod 2 \\ s_2 = & (r_3 + r_4 + r_5 + r_6) \mod 2 \end{cases}$$

Construire un circuit implantant le syndrome toujours avec les mêmes portes à 2 entrées.

8 portes XOR à 2 entrées, 5 sur l'étage 1, 3 sur l'étage 2. Pénaliser si plus de 8 XOR et si plus de 2 étages.

Question 4 Par construction, le syndrome renvoie la position de l'erreur s'il y en a une et 0 sinon[1]. En supposant que le vecteur reçu par le décodeur ait au plus une erreur, on a :
si $S = 0$, R est correct et $D = (r_2 r_4 r_5 r_6)$
si $S = i + 1$, $R' = (r_0 \cdots \bar{r}_i \cdots r_6)$ est correct et $D = (r'_2 r'_4 r'_5 r'_6)$
En utilisant le bloc syndrome de la question précédente, construire un circuit implantant le décodeur.

1 MUX pour chacune des sorties prenant en entrée r_i ou $\bar{r_i}$. La commande du MUX est obtenue avec des portes ET. Par exemple, pour r_2, il faut détecter S=3 soit $\bar{s_2}.s_1.s_0$. On pourra valoriser les étudiants qui simplifie le mux en XOR.

Ex. 2 : Gestion de la température par hysteresis (8pts)

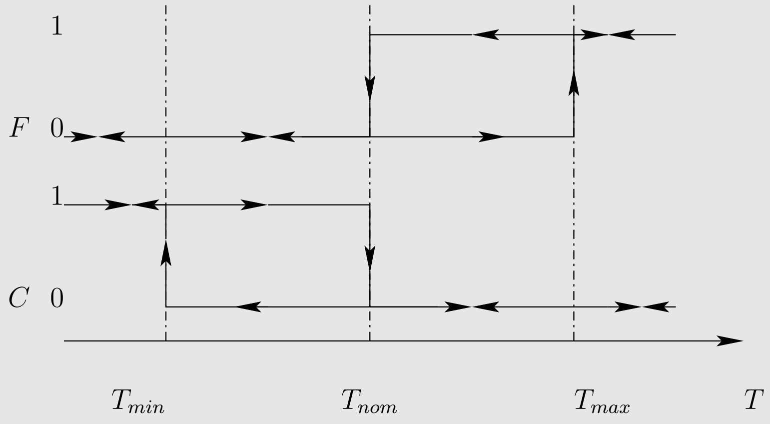
On cherche à réaliser un système permettant d'assurer le maintien à une température quasi-constante d'un poulailler industriel, afin d'assurer aux volatiles une durée de vie compatible avec les normes Européennes. Ce système repose sur la disponibilité d'un climatiseur pouvant produire du chaud, du froid, ou ne rien faire, en fonction de la température courante de l'entrepôt (qui dépend de la température externe et du degré d'agitation des gallinacées).

Le système est basé sur le principe de l'hysteresis, c'est-à-dire que le comportement lors de l'accroissement de la température est différent du comportement lorsque celle-ci baisse. Trois températures de référence sont nécessaires au fonctionnement du système : $T_{min} < T_{nom} < T_{max}$. La production du chaud ou du froid, indiquée respectivement par un signal C à 1 ou F à 1, se passe comme suit :

- lorsque la température T devient inférieure à T_{min}, il y a production de chaud : $C \leftarrow 1$;
- la production de chaud s'arrête, *i.e.* $C \leftarrow 0$, lorsque la température devient supérieure ou égale à la valeur nominale T_{nom} ;
- lorsque la température devient supérieure ou égale à T_{max}, il y a production de froid $F \leftarrow 1$;
- la production de froid cesse lorsque la température devient inférieure à la valeur nominale T_{nom}.

Question 1 Peut-on représenter les valeurs de C et F en fonction de la température sous la forme de tables de vérité ? Justifiez.

[1] Le code permet de corriger au plus une erreur.



Le système exploite un capteur de température qui fournit une information encodée sur 2 bits comme précisé ici :

b_1	b_0	cas
0	0	$T < T_{min}$
0	1	$T_{min} \leq T < T_{nom}$
1	1	$T_{nom} \leq T < T_{max}$
1	0	$T_{max} \leq T$

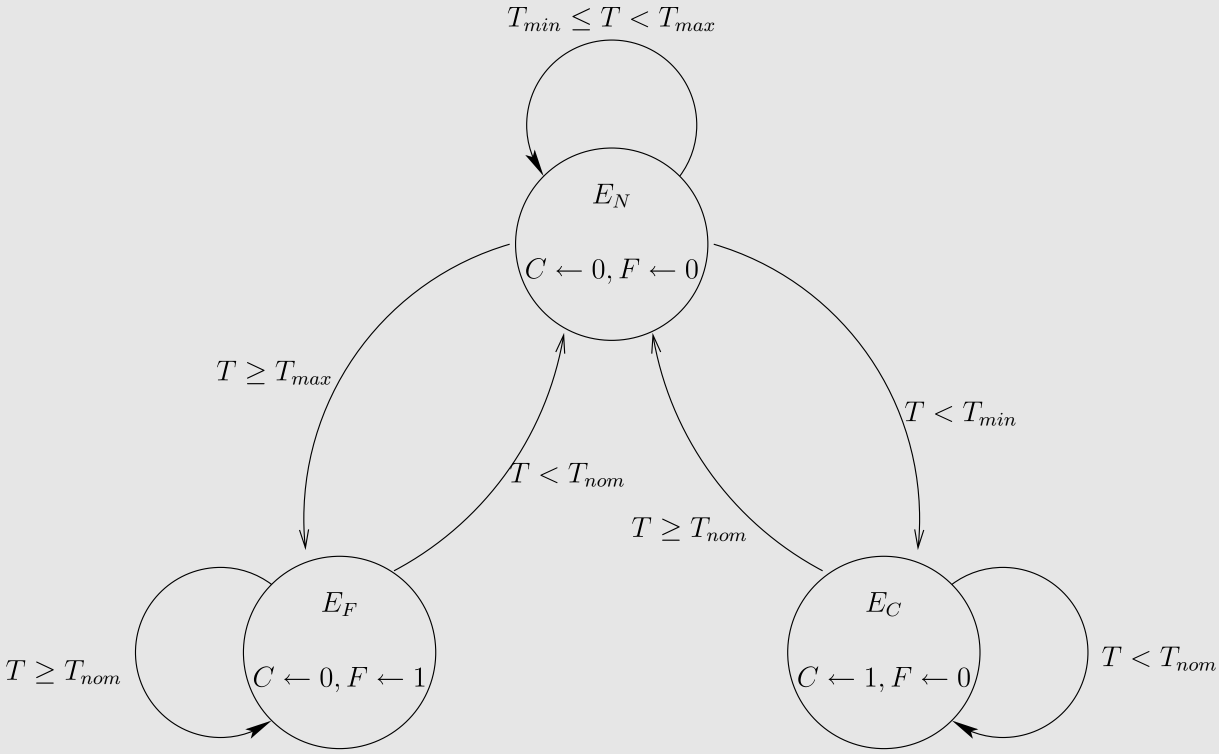
On cherche à formaliser le comportement du système comme un automate de Moore, en faisant l'hypothèse que la période de l'horloge de l'automate est très inférieure au temps qu'il faut pour que la température T du poulailler passe de T_{min} à T_{max} ou inversement, à cause de l'inertie thermique.

Question 2 Précisez quelles sont les entrées et les sorties de l'automate.

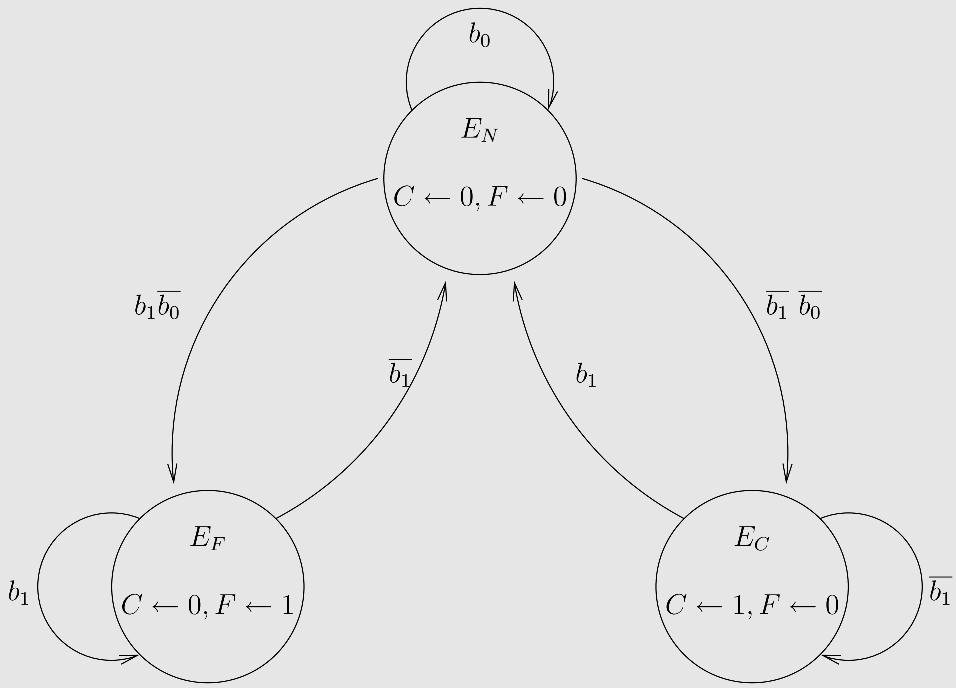
Entrées : b_1 et b_0 ;
Sorties : C et F.

Question 3 Donnez le nombre d'états de l'automate et représentez le graphe de l'automate en y incluant les conditions de transitions sous forme *d'intervalle de températures* sur les arcs et les valeurs des sorties dans les états. Transformez ensuite les intervalles de températures en condition booléennes.

L'automate doit produire 3 paires de valeurs différentes : $C \leftarrow 0, F \leftarrow 0$ lorsque l'on se trouve à une température intermédiaire, $C \leftarrow 1, F \leftarrow 0$ lorsque l'on « vient » du froid, et $C \leftarrow 0, F \leftarrow 1$ lorsque l'on « vient » du chaud. Il y a donc 3 états. Cela donne, en température :



Cela donne, avec les conditions booléennes, et en s'assurant de respecter les conditions de complétude et d'orthogonalité :



Question 4 Proposez un codage logarithmique des états de façon à ce que les expressions des sorties C et F soient simples et donnez la table de transition qui précise les sorties et l'état futur en fonction des entrées et de l'état courant. On note Q_i les sorties du registre représentant l'état courant et D_i les entrées du registre, représentant l'état futur.

Codage des états : il suffit de choisir le code proposé par les sorties.

état	Q_1	Q_0
E_N	0	0
E_F	0	1
E_C	1	0

Table de transitions :

état courant	entrées	état futur	sorties
Q_1Q_0	b_1b_0	D_1D_0	CF
00	00	10	00
	01	00	
	10	01	
	11	00	
01	0x	00	01
	1x	01	
10	0x	10	10
	1x	00	

Avec ce choix, la fonction de génération devient 2 fils.

Question 5 Donnez les expressions simplifiées de C, F et des D_i, et faites le schéma en portes correspondant.

$D_0 = b_1.(Q_0 + \bar{b_0}.\bar{Q_1})$
$D_1 = \bar{b_1}.(Q_1 + \bar{b_0}.\bar{Q_0})$
$C = Q_1 et F = Q_0$

Ex. 3 : Construction algorithmique du décaleur à barillet (6pts)

On s'intéresse ici à la généralisation du décaleur à barillet vu en TD, opérateur qui réalise le décalage d'un nombre x codé sur n bits par un nombre $p < n$ vers la gauche ou vers la droite.

On prend n égal à une puissance entière de 2 ($n = 2^k$) et donc p est codé sur k bits. Le décaleur à barillet est alors composé de k étages de n cellules. Comme vu en TD, une cellule est un MUX à 2 entrées si on fait un décalage à gauche ou à droite, et un ensemble de MUX quand on offre des décalages plus compliqués. L'étage $j, 0 \leq j \leq k-1$ effectue le décalage de 2^j positions si l'entrée $p_j = 1$ et de 0 position sinon. La $i^{\text{ème}}$ cellule génère le bit de poids i du résultat du décalage. Il s'agit de trouver la formule analytique précisant la cellule (i, j) (cellule de rang i de l'étage j du décaleur à barillet) dont la sortie est utilisée pour attaquer les entrées des cellules de l'étage suivant en fonction du type de décalage que l'on cherche à faire.

Question 1 On s'intéresse au décalage vers la gauche. Pour mémoire, le schéma d'un décaleur à gauche pour $n = 8$ est fourni en annexe, tous les MUX étant placées avec l'entrée 1 en bas et 0 en haut comme indiqué sur la cellule de coordonnées $(0, 0)$. Chaque cellule est composée d'un MUX1b ($2 \rightarrow 1$). En fonction de n, i et j, spécifiez les coordonnées de la cellule dont la sortie est connectée à l'entrée 0 ainsi que celles de la cellule dont la sortie est connectée à l'entrée 1 de la cellule (i, j).

$$E0 : (i, j-1) E1 : (i-2^j, j-1)$$

avec $E1 = 0$ si $i < 2^j$ et $(i, -1) = x_i$ pour $0 \leq i < n$

Question 2 On s'intéresse au décalage logique vers la droite. Chaque cellule est composée d'un MUX1b ($2 \rightarrow 1$). En fonction de n, i et j, spécifiez les coordonnées de la cellule dont la sortie est connectée à l'entrée 0 ainsi que celles de la cellule dont la sortie est connectée à l'entrée 1 de la cellule (i, j).

$$E0 : (i, j-1) E1 : (i+2^j, j-1)$$

avec $E1 = 0$ si $i + 2^j \geq n$ et $(i, -1) = x_i$ pour $0 \leq i < n$

Question 3 A partir des résultats des deux questions précédentes, on va traiter les décalages logiques à gauche et à droite en fonction d'une entrée d telle que $d = 1$ pour le décalage à droite. Donnez le schéma de la cellule avec les coordonnées des cellules dont la sortie attaque ses entrées.

MUX1 commandé par d :

$$E0 : (i-2^j, j-1) et E1 : (i+2^j, j-1)$$

avec $E1 = 0$ si $i + 2^j \geq n$ et $E0 = 0$ si $i < 2^j$
MUX2 commandé par p_j :

$$E0 : (i, j-1) et E1 : MUX1$$

avec $(i, -1) = x_i$ pour $0 \leq i < n$

Question 4 On veut également traiter le décalage arithmétique à droite (entrée a = 1 si on veut effectuer un décalage arithmétique quand d = 1). Donnez le schéma de la cellule avec les coordonnées des cellules dont la sortie attaque ses entrées.

décalage arithmétique :

$$E0 : (i, j-1) E1 : (i+2^j, j-1)$$

avec $E1 = x_{n-1}$ si $i + 2^j \geq n$ et $(i, -1) = x_i$ pour $0 \leq i < n$

Ne diffère du cas logique que lorsque $i + 2^j \geq n$
Donc on insère des MUXs pour ces cellules :
MUX3 commandé par a :

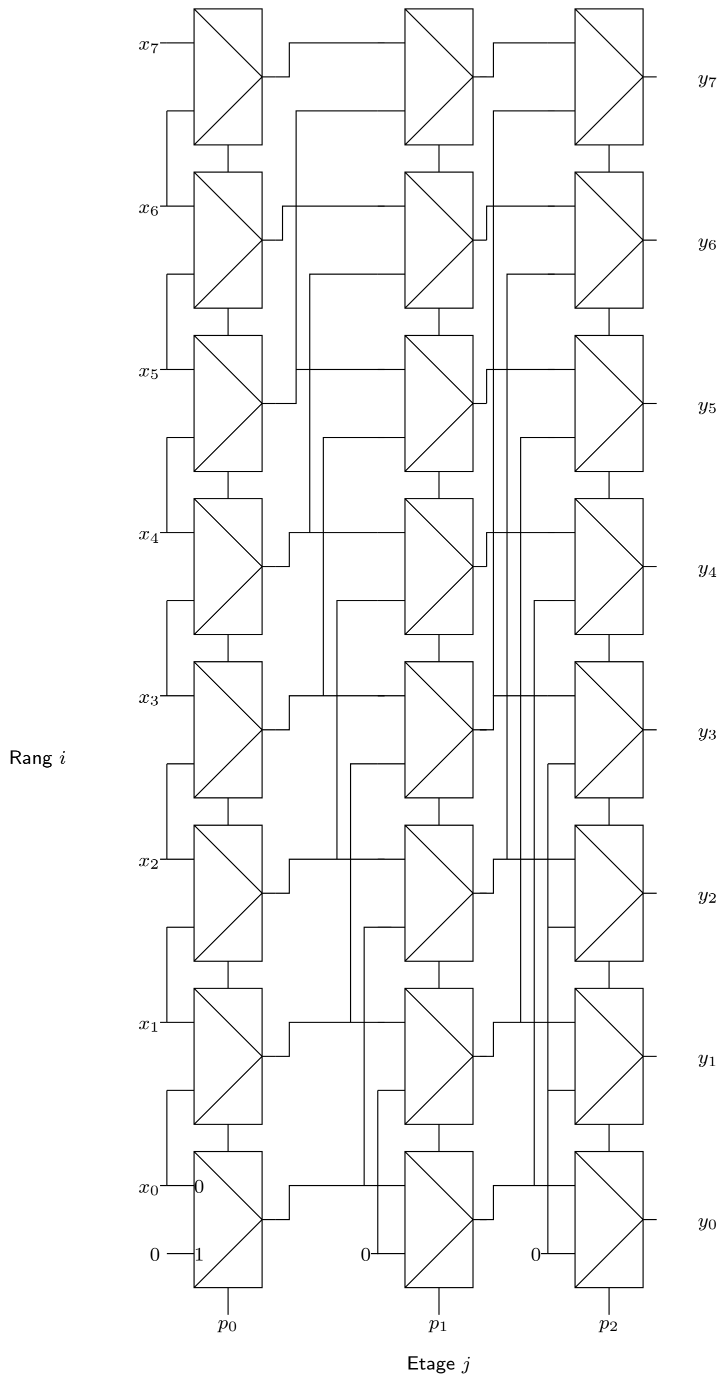
$$E0 : 0 et E1 : x_{n-1}$$

MUX1 commandé par d :

$$E0 : (i-2^j, j-1) et E1 : (i+2^j, j-1)$$

MUX2 commandé par p_j :

$$E0 : (i, j-1) et E1 : MUX1$$



x_7
x_6
x_5
x_4
x_3
x_2
x_1
x_0
0
1
0
0
0
y_7
y_6
y_5
y_4
y_3
y_2
y_1
y_0
p_0
p_1
p_2
Rang i
Etage j