

CC1 - LIF6 Printemps 2015

Durée prévue : 1h15

Aucun document autorisé. Calculatrices, téléphones et ordinateurs portables interdits.

N'oubliez pas de noter votre nom, votre prénom et votre numéro d'étudiant sous le cache rabattable. Débrouillez vous pour que le cache soit rabattu et collé !

Justifiez brièvement et précisément vos réponses, tout en respectant les emplacements prévus.

Le barème est donné à titre indicatif.

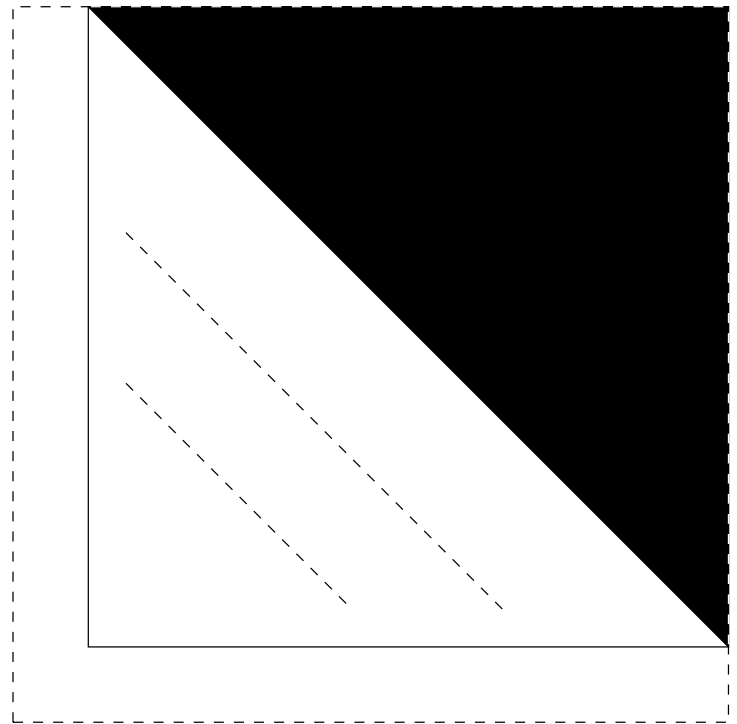
Ne rien écrire dans ce cadre.

Partie 1 :

Partie 2 :

Partie 3 :

Partie 4 :



1 Calculs avec les nombres flottants binaires (7 pts)

Dans cet exercice, on note $x_1 = (0,3)_{10}$, $x_2 = (1,1001010)_2$ et $x_3 = (1,1011101)_2$. **Détaillez soigneusement vos calculs.** On travaille sur une machine où les flottants binaires sont codés sur 12 bits, selon le codage suivant :

$c =$	11	10	9	8	7	6	5	4	3	2	1	0
	s	b				m						

Le nombre représenté est $f(c) = (-1)^s \times (1, m)_2 \times 2^e$, avec $e = (b)_2 - 2^3$ si $1 \leq (b)_2 \leq 14$, (on ne se préoccupe pas du codage des sous-normaux ni des valeurs exceptionnelles ici).

1. (2 points) Convertissez le nombre x_1 en *notation positionnelle à virgule en binaire*. Indiquez bien la période.

.....

.....

.....

.....

.....

.....

.....

2. (2 points) Comment est représenté x_1 dans le format flottant binaire décrit ci-dessus ? Vous effectuerez un arrondi au plus proche, et exprimerez votre résultat sous la forme d'un code binaire, puis en *hexadécimal*.

.....

.....

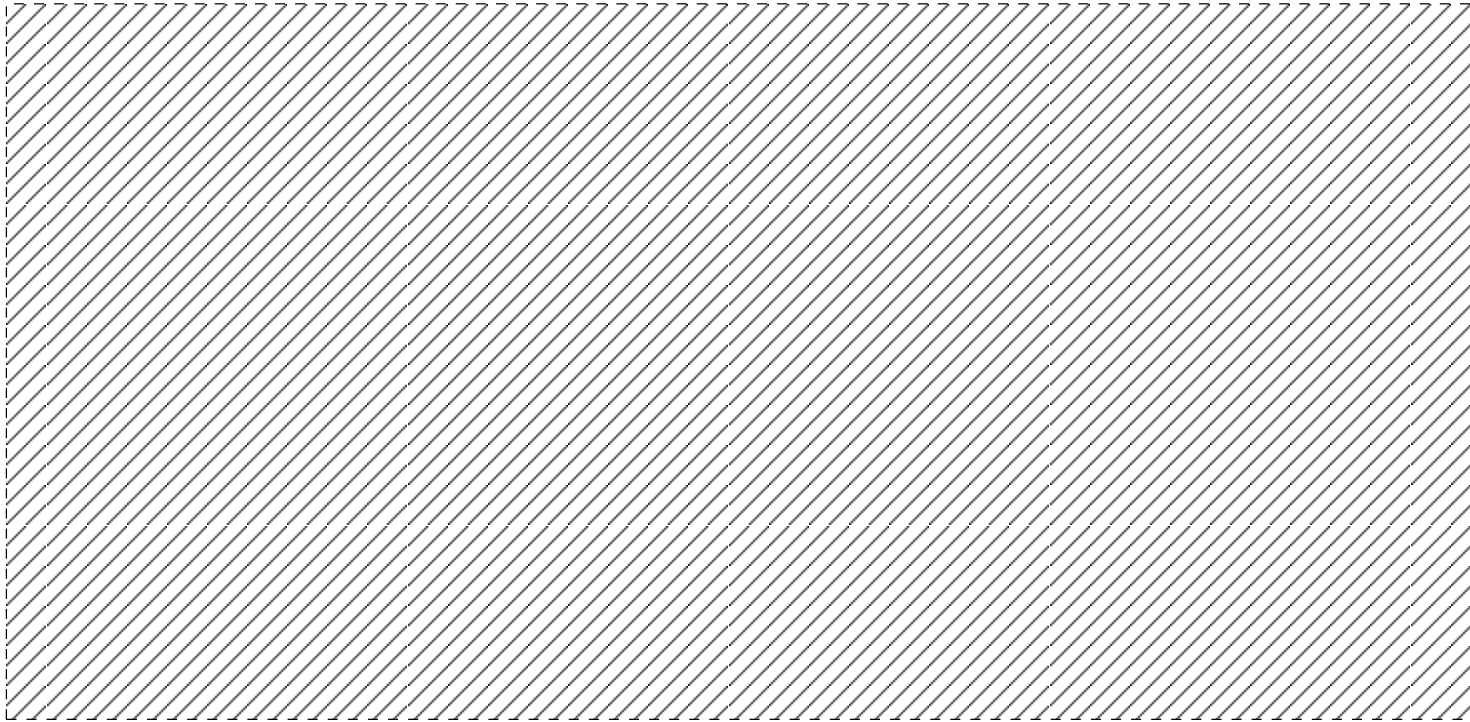
.....

.....

.....

.....

.....



3. (1 point) Posez l'addition $x_2 + x_3$; on note x_4 le résultat de l'opération.

.....

.....

.....

.....

.....

4. (2 points) Donnez la représentation de x_4 dans le format flottant binaire décrit ci-dessus. Vous effectuerez un arrondi au plus proche, et exprimerez votre résultat sous la forme d'un code binaire, puis en *hexadécimal*.

.....

.....

.....

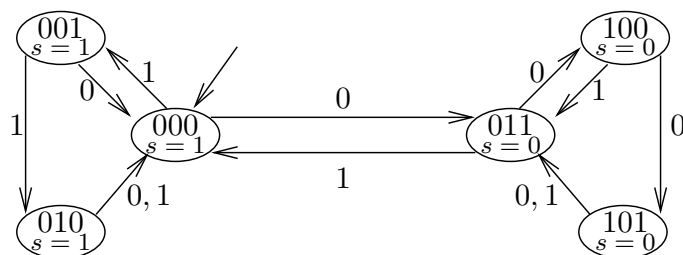
.....

.....

.....

2 Circuits séquentiels (7 pts)

On considère l'automate fini représenté ci-dessous. On veut mettre en place un circuit séquentiel pour l'implanter : le circuit présentera une entrée e (en plus du signal d'horloge h) et une sortie s . On note $(q_2q_1q_0)$ l'état courant de l'automate, et $(q_2^+q_1^+q_0^+)$ l'état suivant. Les transitions se produisent en fin de cycle, sur un front montant de l'horloge. On note $F : (q_2q_1q_0) \mapsto (q_2^+q_1^+q_0^+)$ la fonction de transition, et $f : (q_2q_1q_0) \mapsto s$ la fonction de sortie.



1. (1 point) Complétez la table de vérité vérité suivante.

état courant			entrée	état suivant			sortie
q_2	q_1	q_0	e	q_2^+	q_1^+	q_0^+	s
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				

2. (1 point) Montrez que : $q_1^+ = \overline{q_1} \cdot (\overline{q_2} \cdot (\overline{q_0} \oplus e) + q_2 \cdot (q_0 + e))$.

.....

.....

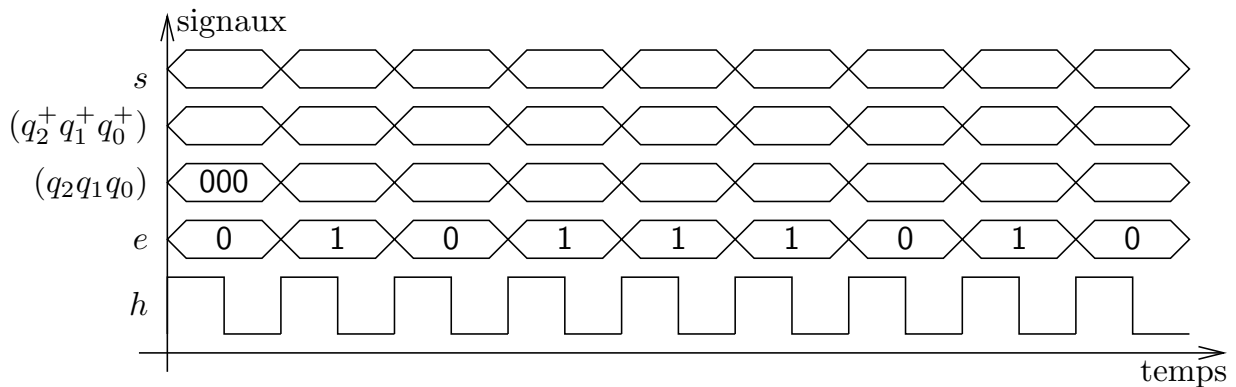
.....

.....

.....

.....

3. (2 points) Complétez le chronogramme suivant.



4. (3 points) Supposons que l'on dispose déjà d'un registre 3 bits, et de circuits combinatoires réalisant la fonction de transition F et celle de sortie f (pas besoin de les détailler). Donnez un circuit séquentiel implantant l'automate fini considéré. **Annotez votre schéma** (nom des circuits, des signaux, tailles en nombre de bits).

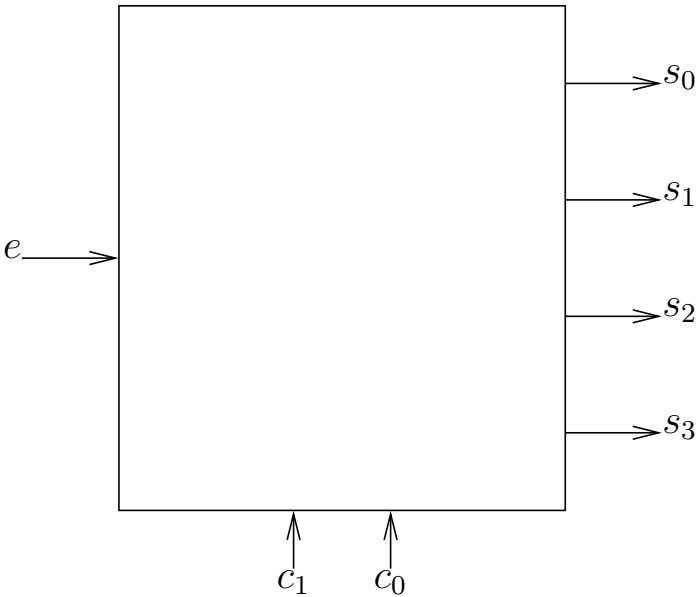
3 Circuits combinatoires (3 pts)

On donne la définition d'un *démultiplexeur* 1 vers 2^n : il s'agit d'un circuit combinatoire qui présente n entrées de commande $c_{n-1}, \dots, c_0$, une entrée e à démultiplexer, et 2^n sorties $s_{2^n-1}, \dots, s_0$, telles que $s_{(c_{n-1} \dots c_0)_2} = e$, et $s_i = 0$ si $i \neq (c_{n-1} \dots c_0)_2$.

1. (1 point) Complétez la table suivante dans le but de réaliser un démultiplexeur 1 vers 4.

c_1	c_0	s_0	s_1	s_2	s_3
0	0				
0	1				
1	0				
1	1				

2. (2 points) Donnez un circuit logique combinatoire pour un démultiplexeur 1 vers 4, en utilisant 4 portes AND à 3 entrées, et des inverseurs.



4 Programmation en assembleur (3 pts)

On travaille sur un processeur qui dispose de 4 registres, nommés R0, ..., R3. **Vous ne pouvez utiliser que les deux instructions suivantes :**

- ADD RD,RS1,RS2 qui effectue $RD \leftarrow RS1 + RS2$,
- LHS RD,RS qui effectue $RD \leftarrow 2 \times RS$.

Donnez un code en langage d'assemblage pour multiplier par 21 le contenu de R1, et ranger le résultat dans R0¹.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

1. La solution naïve qui implique de recopier une vingtaine de fois la même instruction ne se verra attribuer que très peu de points... Vous pouvez commenter votre programme pour aider le correcteur à suivre votre solution, mais sans verbosité excessive : cela pourriez avoir pour effet de lui faire perdre patience...