

Nom :
Prénom :
N° d'ordre :

ENSICAEN
1^{ère} année
informatique

Examen de circuits logiques 2010

durée : 90 minutes

Les documents et calculatrices ne sont pas autorisés.

Les réponses seront données sur ces feuilles à l'intérieur des cadres prévus à cet usage.

1- Conversion numérique

- ◆ Compléter le tableau ci-dessous

Base 2 (12 bits)*	Base 10 **	Base 16 **	DCB
		3D.6	
	-130		
		-12.4	

* les nombres binaires seront représentés en complément à deux sur 12 bits et les valeurs non entières en virgule fixe Q_3 . (Rappel représentation Q_k sur N bits: $b_{N-1}b_{N-2}...b_{k+1}b_k...b_2b_1b_0$)

** les nombres non entiers en base 10 avec 2 chiffres significatifs derrière la virgule et les nombres non entiers en base 16 avec 1 chiffre significatif.

- ◆ Coder les valeurs suivantes sur 10 bits virgule flottante (E sur 4 bits ; F sur 5 bits) suivant la norme IEEE 754.

A = 0.1875

B = -26

Rappel : représentation en virgule flottante norme IEEE 754. La valeur X est représentée suivant la forme :

$$X = (-1)^S \cdot 2^{E-7} \cdot 1.F \quad ; \quad X \text{ s'écrit alors en binaire virgule flottante : } \underbrace{S}_{\text{signe}} \underbrace{e_3 e_2 e_1 e_0}_E \underbrace{f_4 f_3 f_2 f_1 f_0}_F$$

E et F sont codés en binaire non signé.

- ◆ Quelles sont le plus petit et le plus grand positifs pouvant être représenté par ce code 10 bits virgule flottante (E sur 4 bits ; F sur 5 bits) ?

2- Questions concernant le projet (10 min maximum)

- ◆ Dans quel groupe de projet étiez-vous ?
- ◆ Quel était le sujet (type d'opérations, nombres de bits, format des données...)?
- ◆ Quelle(s) partie(s) avez-vous développée(s) et réalisée(s) ?
- ◆ Comment était réparti le travail au sein du groupe de projet ?
- ◆ Quelles difficultés avez-vous rencontrées ?

3- Synthèse logique

On souhaite réaliser la fonction logique F exprimée par la table de vérité ci-dessous :

- ◆ Grâce à une table de Karnaugh, exprimer la fonction logique F.
- ◆ Transformer l'expression de F pour une implémentation avec des portes NAND.

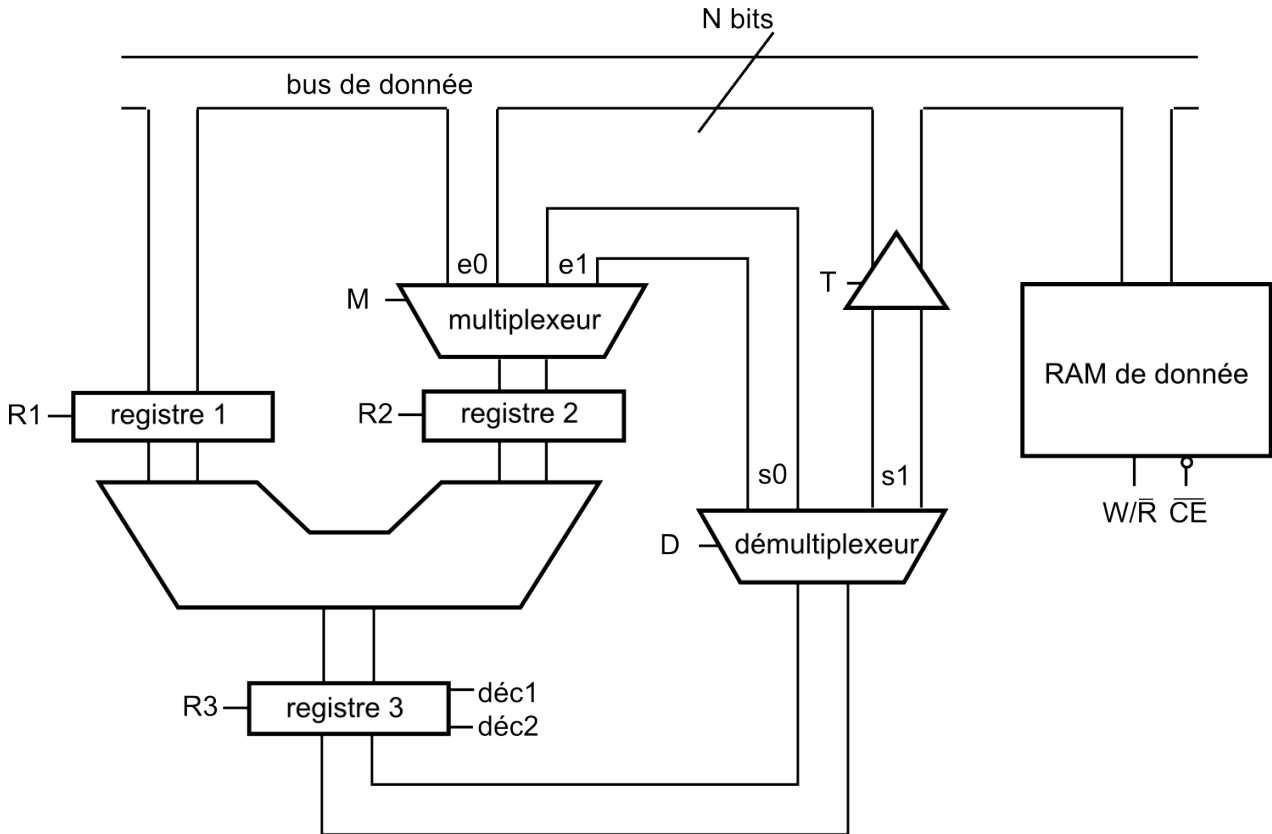
d	c	b	a	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0

table de vérité de F

4- Machine à états finis

L'objectif est de dimensionner l'unité de contrôle sous forme de machine à états finis permettant d'effectuer la moyenne de 4 éléments, x_i à x_{i+3} , contenus dans la RAM (x_i donnée stockée en mémoire RAM et y_i résultat du calcul devant être mémorisé dans la RAM) sur l'unité de traitement présentée ci-dessous.

$$y_i = \frac{1}{4} \sum_{j=i}^{j=i+3} x_j$$



Remarques :

L'addition en prend un temps de cycle d'horloge.

Il faut que la donnée soit présente sur le bus avant de pouvoir être chargée dans le registre ou écrite en mémoire. On supposera que les adresses mémoire pour la lecture et l'écriture sont gérées automatiquement.

Les signaux R_i commandent le chargement des registres lorsqu'ils sont à l'état 1. Les sorties du *tristate* sont connectées au bus de données lorsque T_1 est à l'état 1, elles en sont isolées sinon.

Le registre 3 possède deux commandes de décalage actives à l'état haut : *déc1* permet de décaler d'un bit le contenu du registre sur la droite et *déc2* permet de décaler de deux bits le contenu du registre. Il faut que le contenu du registre soit chargé avant de pouvoir être décalé.

Questions :

1. Remplir le tableau d'évolution du système pour opérer le calcul.
2. En déduire la taille le machine d'état nécessaire.
3. Synthétiser la machine d'état avec une registre d'état à base de bascule D à front.

Machine d'état