

Nom :
 Prénom :
 N° de place :

ENSICAEN
 1^{ère} année
 informatique

Examen de circuits logiques 2013-2014

durée : 90 minutes, tout document autorisé, calculatrice autorisée
 Les réponses seront données sur ces feuilles à l'intérieur des espaces prévus à cet usage.

1- Conversion numérique

- ◆ Complétez le tableau ci-dessous

Base 2 (12 bits)*	Base 10	démarche et/ou commentaires
	112,36	
	-76,071	
	131,25	
	-0,123	

* les nombres binaires seront représentés en complément à deux sur 12 bits en codage virgule fixe $Q_{8,4}$.

Rappel représentation $Q_{m,k}$ sur N bits: $b_{m+k-1}b_{m+k-2} \dots b_k b_{k-1} \dots b_2 b_1 b_0$; $N=m+k$

- ◆ Codez la valeur suivante en virgule flottante suivant la norme IEEE 754, présentez vos résultats intermédiaires.

A = 43253,76

Rappel : représentation en virgule flottante suivant la norme IEEE 754.

La valeur X est représentée suivant la forme : $X = (-1)^S \cdot 2^{E-127} \cdot 1, F$

X s'écrit alors en binaire virgule flottante : $\underbrace{e_7 e_6 \dots e_1}_{\text{signe}} \underbrace{e_0 f_{22} f_{21} \dots f_2 f_1 f_0}_E$; E et F sont codés en binaire non signé.

2- Bloc logique combinatoire d'un décodeur DCB/7-segments

L'objectif de cet exercice est la synthèse du bloc logique combinatoire d'un décodeur DCB¹/7segments. Ce bloc logique peut servir d'interface entre un compteur à sortie DCB et un afficheur 7-segments comme dans l'exemple illustré en Figure 1. Dans un premier temps vous devez établir les équations logiques exprimant les sorties de ce bloc logique en fonction des entrées. Dans un second temps, ces équations seront implémentées sur une structure PAL.

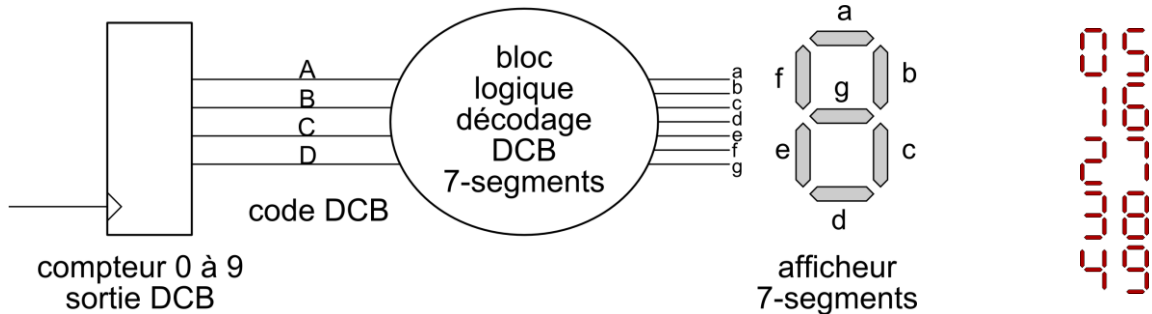


Figure 1: à gauche, : bloc logique décodeur DCB 7-segments entre un compteur et un afficheur 7-segments (D poids fort, A poids faible ; exemple 5 décimal se code 0101 (D à « 0 », C à « 1 », B à « 0 », A à « 1 »). A droite, représentation des chiffres avec les 7 segments.

Q2.1. Etablir les équations du bloc logique combinatoire pour les 3 premières sorties du bloc (a,b et c), la démarche est identique pour les 4 autres. Présentez vos résultats intermédiaires et votre démarche.

¹ DCB décimal codé binaire, BCD binary coded decimal

Q2.2. Implémentez les équations précédentes sur la structure PAL ci-dessous.

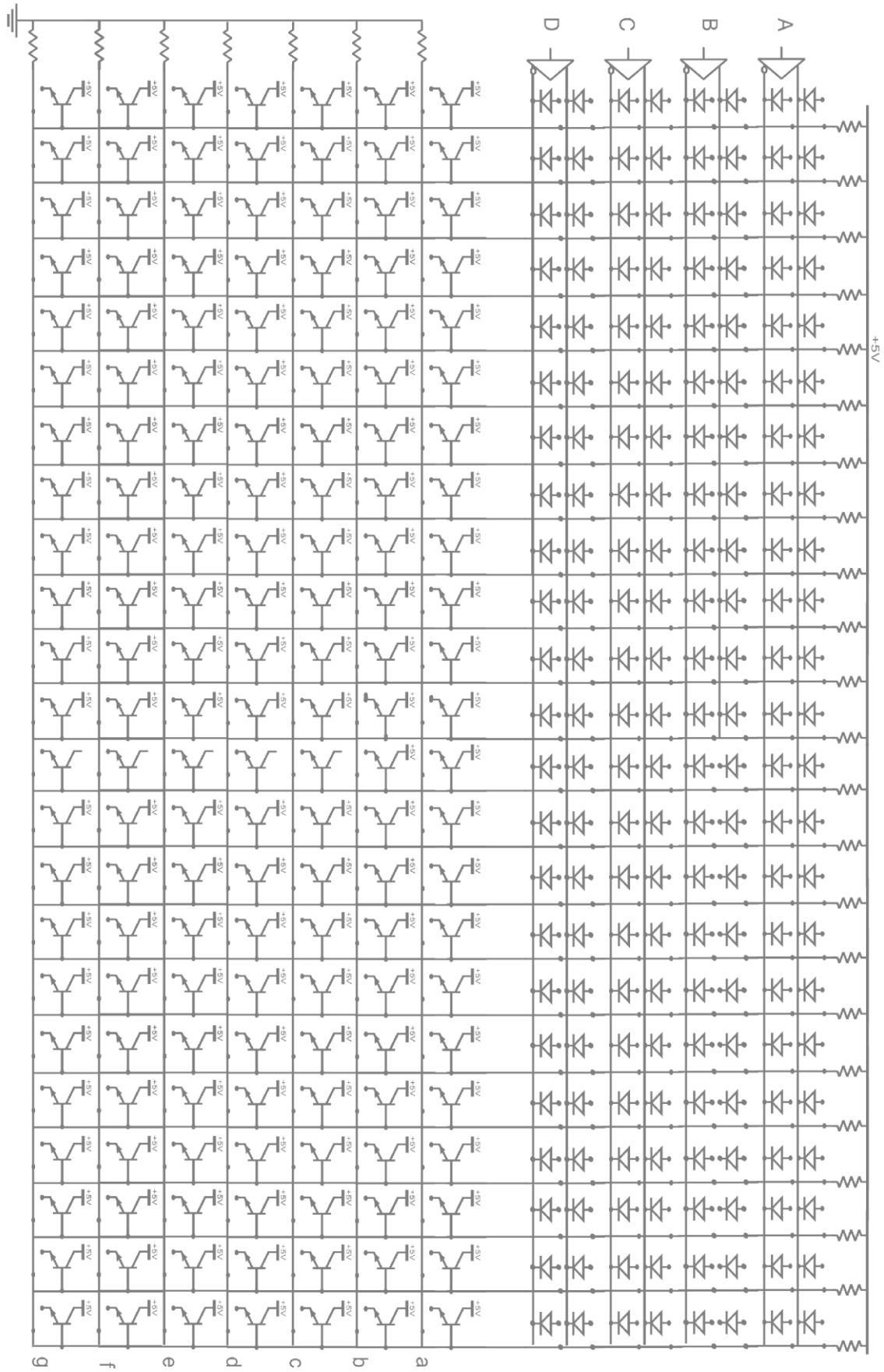


Figure 2 : Structure PAL pour l'implémentation du décodeur

3- Implémentation d'une division

L'objectif de cet exercice est la conception d'une machine à états finis contrôlant les étapes d'un algorithme de division implémentée de façon matérielle (hardware). La division de nombres de 4 bits est implémentée sur l'unité de traitement de la Figure 3 en suivant l'algorithme sans restauration (*Non-Restoring Division Algorithm*) présenté et illustré ci-dessous.

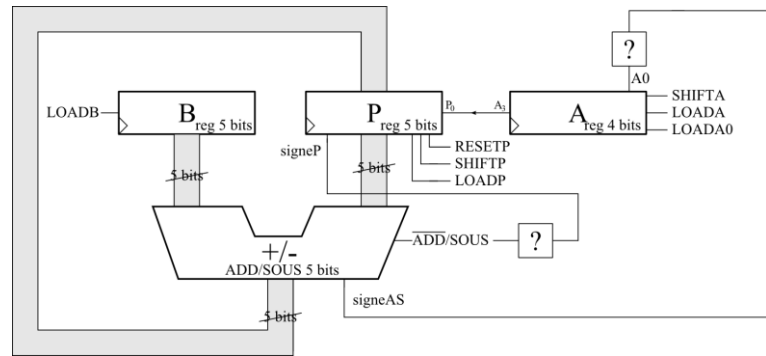


Figure 3: Architecture de l'unité de traitement implémentant l'algorithme de division A/B. Pour la division du contenu du registre A (dividende) par le contenu du registre B (diviseur). Le résultat de la division (quotient) se trouve dans le registre A à la fin de l'opération. P est un registre de travail initialisé à zéro et stockant les résultats intermédiaires d'addition de soustraction et de décalage.

Algorithme de division sans restauration :

- Décalage des registres P et A de 1 bit vers la gauche.
 - Si le contenu du registre P est négatif :
 - on additionne le contenu du registre B à P
 - Si le contenu du registre P est positif :
 - on soustrait le contenu du registre B à P
 - Si le résultat chargé dans P est négatif on met le bit de poids faible de A (A0) à « 0 » sinon à « 1 ».
- Ces étapes sont répétées N fois, N étant le format des données, ici N=4.

Opération élémentaire	P	A	Commentaires
	P ₄ P ₃ P ₂ P ₁ P ₀	A ₃ A ₂ A ₁ A ₀	
Chargement des registres	00000	0111	Initialisation des registres. Chargement du dividende (A) et du diviseur (B)
décalage	00000	111-	décalage P,A de 1 bit à gauche
-B	11110	111-	P ≥ 0, on soustrait le diviseur B (P+(-B)).
A0 ≤ 0	11110	1110	résultat < 0, A0 mis à « 0 ».
décalage	11101	110-	décalage P,A de 1 bit à gauche
+B	00010	110-	P < 0, on additionne le diviseur B (P+B).
A0 ≤ 0	11111	1100	résultat < 0, A0 mis à « 0 ».
décalage	11111	100-	décalage P,A de 1 bit à gauche
+B	00010	100-	P < 0, on additionne le diviseur B (P+B).
A0 ≤ 1	00001	1001	résultat > 0, A0 mis à « 1 ».
décalage	00011	001-	décalage P,A de 1 bit à gauche
-B	11110	001-	P ≥ 0, on soustrait le diviseur B (P+(-B)).
A0 ≤ 1	00001	0011	résultat > 0, A0 mis à « 1 ».
	reste	quotient	

Tableau 1: Algorithme de division sans restauration. Exemple de la division de 7 (0111) dans le registre A par 2 (0010) dans le registre B. Remarque : la soustraction de 2 est représentée dans le tableau par l'addition de -2 (11110). Le quotient obtenu 0011 (3) et le reste 00001 (1) correspondent au résultat de la division de 7 par 2.

Les commandes ADD/SOUS et la valeur du bit A_0 sont déterminés par les signes du contenu du registre P et du résultat de l'additionneur-soustracteur. Les sorties signe sont à « 1 » pour un nombre négatif et à « 0 » pour un nombre positif ou nul.

Q3.1. Exprimez ADD/SOUS et la valeur de A_0 à charger en fonction des bits de signe de P et du résultat de l'additionneur/soustracteur.

Q3.2. Comment sont obtenues les sorties signe de P et du résultat de l'additionneur/soustracteur ?

Q3.3. Concevez la machine à états finis contrôlant les étapes de l'algorithme. La conception devra conduire à l'expression des tables de vérité de la machine à états finis. Les équations logiques et le dessin au niveau portes ne sont pas demandés. Vous pouvez vous appuyer sur le tableau décrivant l'évolution des signaux de commande de l'unité de traitement et/ou le diagramme d'état proposés.

Le signal DIVstart démarre l'algorithme. Les registres P et A conservent les valeurs du reste et du quotient respectivement à la fin de l'algorithme. Une sortie DIVend indique quand le résultat est disponible.

Les commandes LOAD effectuent le chargement des valeurs présentes en entrée des registres (B, P, A et A_0). Les commandes SHIFT décalent le contenu des registre P et A d'un bit à gauche. La commande RESETP met à 00000 le contenu du registre P.

On considère que la période de l'horloge de cadencement, clk, de la machine d'état est supérieure au temps de calcul de l'additionneur/soustracteur.

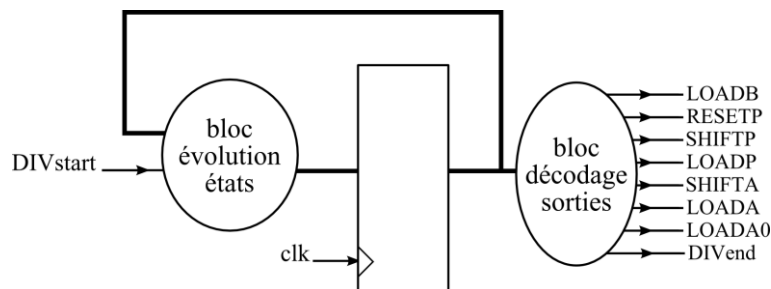


Figure 4: Schéma bloc de la machine à états finis contrôlant l'algorithme de division

description	état/cycle	LOADB	RESETP	SHIFTP	LOADP	SHIFTA	LOADA	LOADA0	DIVend
division terminée, attente DIVstart	0	0	0	0	0	0	0	0	1
début division, chargement registres	1	1	1	0	0	0	1	0	0

La machine à états finis se comporte comme un compteur 10 états avec le passage de l'état 0 à l'état 1 conditionné par la valeur du signal d'entrée DIVstart. Ensuite le passage d'un état au suivant est indépendant du signal DIVstart.

