

Nom :
Prénom :
N° de place :

ENSICAEN
1^{ère} année

Examen de circuits logiques 2014-2015

durée : 90 minutes, documents et calculatrice autorisés
Les réponses seront données sur ces feuilles à l'intérieur des espaces prévus à cet usage.

1- Conversion numérique (5 points)

◆ Complétez le tableau ci-dessous

Base 2 (12 bits), Q _{8,4} , cplt à 2*	Base 10	Démarche. Erreur de représentation
01101110,0000	109,99	$109 = 64 + 32 + 8 + 4 + 1$ $\begin{array}{r} 0,99 \quad 0,84 \\ \times 16 \quad \times 16 \\ \hline 15,84 \quad 13,44 \\ \hline 1111 \leftarrow 1 \end{array}$ $01101101,1111$ $01101110 \Rightarrow 110$ erreur = $0,01 \leq 2^{-5}$ $\left(\frac{9}{2}\right)$ $0,03125$
10101000,0100	-87,72	$87 = 64 + 16 + 4 + 2 + 1 \Rightarrow 01010111$ $\begin{array}{r} 0,72 \quad 0,52 \\ \times 16 \quad \times 16 \\ \hline 11,52 \quad 8,32 \\ \hline > 8 \end{array}$ $01010111,1100$ $10101000,0011 + 1$ $10101000,0100$ arrondi à 12 erreur = $\left(\frac{12}{16} - 0,72\right) = 0,03 \leq \frac{9}{2}$

* les nombres binaires seront représentés en complément à deux sur 12 bits en 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 A sur 32 bits virgule flottante (E sur 8 bits) suivant la norme IEEE 754.

A = -0,2235 = $(-1) \cdot 1,788 \times 2^{-3}$

$\Rightarrow 2^{-3} = 2^{E-127}$

$0,788$	$0,608$	$0,728$	$0,648$	$0,368$	$0,888$
$\times 16$	$\times 16$	$\times 16$	$\times 16$	$\times 16$	$\times 16$
$12,608$	$9,728$	$11,648$	$10,368$	$5,888$	$14,208$
1000	1001	1011	1010	0101	1110

E = 124
E = $(0111100)_2$

A = $\underbrace{1}_S \underbrace{0111100}_E \underbrace{11001001101101010111}_F$

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 : $\underbrace{S}_{\text{signe}} \underbrace{e_7 \dots e_2 e_1 e_0}_{E} \underbrace{f_{22} \dots f_2 f_1 f_0}_{F}$; E et F sont codés en binaire non signé.

2- Synthèse logique combinatoire (5 points)

Q2.1 Comparateur de deux nombres de 1 bit A0 et B0 en binaire naturel

Synthétisez un bloc logique combinatoire permettant la comparaison de deux nombres positifs de 1 bit A0 et B0 codés en binaire naturel.

Le bloc logique doit générer :

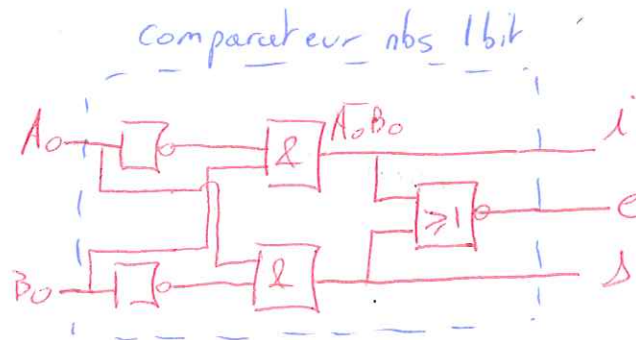
- un signal « s » actif si A0 est supérieur à B0,
- un signal « e » actif si A0 est égal à B0,
- un signal « i » actif si A0 est inférieur à B0.

A ₀	B ₀	s	e	i	table de vérité
0	0	0	1	0	
0	1	0	0	1	
1	0	1	0	0	
1	1	0	1	0	

$$s = A_0 \bar{B}_0 ;$$

$$e = A_0 \oplus B_0 ;$$

$$i = \bar{A}_0 B_0$$



Q2.2 Comparateur de deux nombres de 2 bits A1A0 et B1B0 en binaire naturel

Synthétisez un bloc logique combinatoire permettant la comparaison de deux nombres de 2 bits positifs codés en binaire naturel, A1A0 et B1B0.

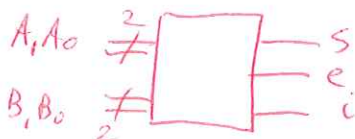
Le bloc logique doit générer :

- un signal « s » actif si A1A0 est supérieur à B1B0,
- un signal « e » actif si A1A0 est égal à B1B0,
- un signal « i » actif si A1A0 est inférieur à B1B0.

table de Karnaugh pour S

A ₁ A ₀ \ B ₁ B ₀	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

$$s = A_1 A_0 \bar{B}_0 + A_0 \bar{B}_1 \bar{B}_0 + A_1 \bar{B}_1$$



A ₁ A ₀	B ₁ B ₀	s	e	i	table de vérité
00	00	0	1	0	
00	01	0	0	1	
00	10	0	0	1	
00	11	0	0	1	
01	00	1	0	0	
01	01	0	1	0	
01	10	0	0	1	
01	11	0	0	1	
10	00	1	0	0	
10	01	1	0	0	
10	10	0	1	0	
10	11	0	0	1	
11	00	1	0	0	
11	01	1	0	0	
11	10	1	0	0	
11	11	0	1	0	

e B_1, B_0 table de Karnaugh pour e

A_1, A_0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

$e = \bar{A}_1 \bar{A}_0 \bar{B}_1 \bar{B}_0 + \bar{A}_1 A_0 \bar{B}_1 B_0 + A_1 A_0 B_1 B_0 + A_1 \bar{A}_0 \bar{B}_1 \bar{B}_0$
 Equation de e

i B_1, B_0 table de Karnaugh pour i

A_1, A_0	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

$i = \bar{A}_0 B_1 B_0 + \bar{A}_1 \bar{A}_0 B_0 + \bar{A}_1 B_1$

3- Portes logiques et transistors (2 points)

D'un point de vue physique, les portes logiques sont réalisées à partir de transistors. Plusieurs types de transistors peuvent être employés, mais le plus couramment utilisé est celui des transistors Métal Oxyde Semiconducteur plus connu sous le sigle transistor MOS. La fabrication des portes logiques s'appuie alors sur l'utilisation de deux types de transistors, les transistors NMOS et les transistors PMOS. Ces transistors sont utilisés comme des interrupteurs et chacun de ces transistors est caractérisé par les niveaux logiques qui le rendent assimilable à un interrupteur ouvert ou bien à un interrupteur fermé. La figure 1 reprend les configurations possibles.

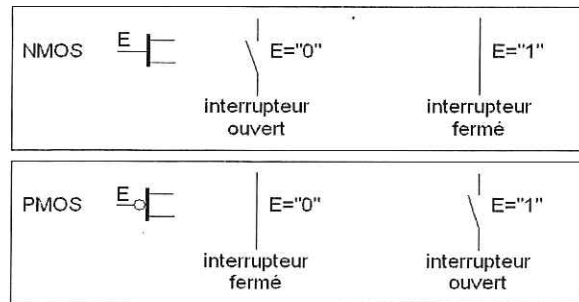


Figure 1: Modèles des transistors MOS

A partir de ces informations, déterminez pour le schéma ci-après la fonction logique réalisée.

$\bar{S} = 1$ si
 E_3 à "0" et
 E_1 ou E_2 à "0"

S à "0" (masse)
 si E_3 à "1" ou
 E_1 et E_2 à "1" $\Rightarrow \bar{S} = E_3 + E_1 E_2$

E_3	E_2	E_1	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

$S = \bar{E}_3 \cdot (\bar{E}_1 + \bar{E}_2) = \bar{E}_3 \bar{E}_1 + \bar{E}_3 \bar{E}_2$

fonction réalisée : $\bar{E}_3 \cdot (\bar{E}_1 + \bar{E}_2)$ non E_3 et (non E_1 ou non E_2)

3- Transmission numérique et codage (8 points)

Code AMI (Alternate Mark Inversion)

Dans le contexte de la communication série sur fil conducteur, le code AMI ou bipolaire simple consiste à coder l'information logique « 1 » par une alternance de niveau +V et -V et l'information logique « 0 » par une tension nulle 0V. La figure 2 illustre un exemple de code AMI associé à des données numériques. L'intérêt de ce codage est sa valeur électrique moyenne nulle.

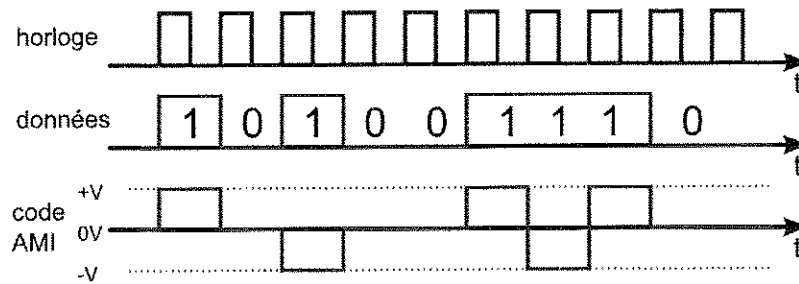


Figure 2: exemple de chronogramme avec une série de données et le code AMI correspondant.

On propose de réaliser un codeur AMI en suivant la structure proposée dans la figure 3.

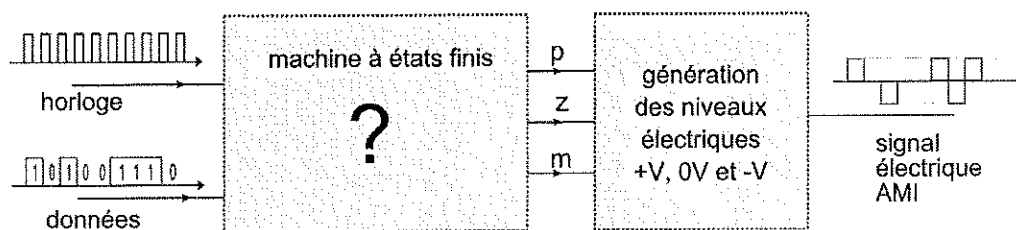


Figure 3: structure proposée pour le codeur AMI

Le codeur se compose d'un bloc de génération de niveaux électriques et d'une machine à états finis. Le bloc de génération des niveaux électriques a pour rôle d'imposer sur le fil de communication soit +V, soit 0V, soit -V. La machine à états finis de type Moore est chargée de délivrer les signaux p, z et m de commande du bloc de génération des niveaux électriques en fonction de l'entrée de données.

p	z	m	sortie
0	0	0	0V
0	0	1	-V
0	1	0	0V
0	1	1	0V
1	0	0	+V
1	0	1	0V
1	1	0	0V
1	1	1	0V

La table ci-contre résume le fonctionnement du bloc de génération des niveaux électriques. Le niveau +V est généré uniquement si le signal p est actif seul. Le niveau -V est généré uniquement si le signal m est actif seul. Dans tous les autres cas, un niveau électrique 0V est imposé sur le fil de communication.

Q3.1. Diagramme d'état

La figure 4 présente une proposition d'états pour le diagramme d'état en vue de la synthèse d'une machine à états finis de type Moore assurant le passage des données binaires vers le code AMI, la machine à états-finis de la figure 3.

Les états du diagramme proposé sont les suivants :

Etat 0v- : donnée à « 0 » sachant que la dernière donnée à « 1 » correspondait à -V.

Etat 0v+ : donnée à « 0 » sachant que la dernière donnée à « 1 » correspondait à +V.

Etat 1v- : donnée à « 1 » devant être codée -V.

Etat 1v+ : donnée à « 1 » devant être codée +V.

Complétez le diagramme d'état proposé.

Remarque : vous pouvez si vous le souhaitez proposer votre propre diagramme d'états.

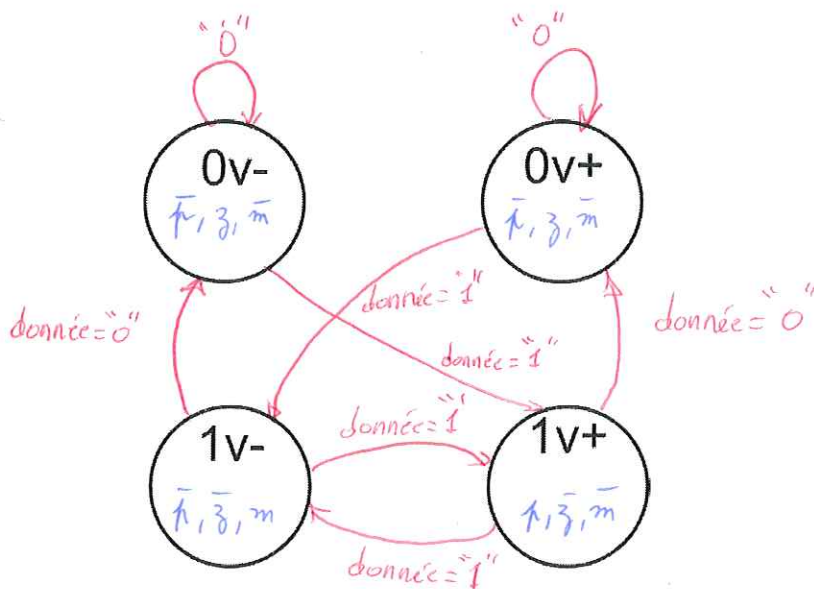
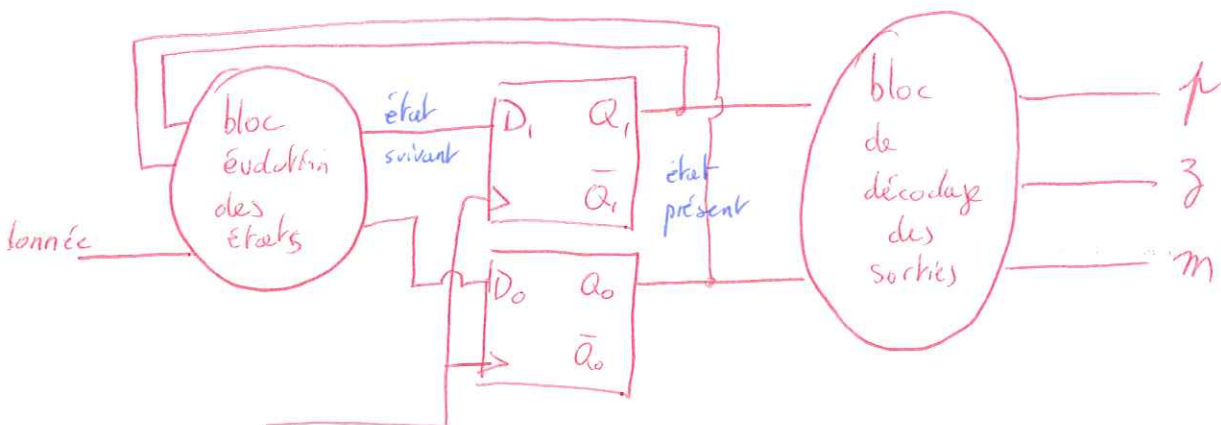


Figure 4: Diagramme d'état pour code AMI

4 états \Rightarrow 2 bascules pour le registre d'état.



Q3.2. Synthétisez la machine à états finis correspondant au diagramme d'états précédent

La synthèse ira jusqu'à l'obtention des équations des blocs logiques. Les schémas à base de portes logiques ne sont pas demandés.

Remarque : veillez à expliquer votre démarche et à présenter vos résultats intermédiaires.

Codage des états, on choisit un codage binaire.

①

$0v^-$	00
$0v^+$	01
$1v^-$	10
$1v^+$	11

③

Table de décodage des sorties

état présent			sorties		
Q_1	Q_0		f	z	m
0	0	$0v^-$	x	1	x
0	1	$0v^+$	x	1	x
1	0	$1v^-$	0	0	1
1	1	$1v^+$	1	0	0

②

Table d'évolution des états

entrée donnée	état présent		état suivant			
	Q_1	Q_0	Q_1	Q_0	D_1	D_0
0	0	0	$0v^-$	$0v^-$	0	0
0	0	1	$0v^+$	$0v^+$	0	1
0	1	0	$1v^-$	$0v^-$	0	0
0	1	1	$1v^+$	$0v^+$	0	1
1	0	0	$0v^-$	$1v^+$	1	1
1	0	1	$0v^+$	$1v^-$	1	0
1	1	0	$1v^-$	$1v^+$	1	1
1	1	1	$1v^+$	$1v^-$	1	0

$f = Q_1 Q_0$

$m = Q_1 \bar{a}_0$

$z = \overline{a_1 a_0 + a_1 \bar{a}_0}$ tous les autres cas

$z = \bar{a}_1 a_0 + \bar{a}_1$

$z = \bar{a}_1$

à partir de la table

$z = \bar{a}_1 \bar{a}_0 + \bar{a}_1 a_0 = \bar{a}_1$

$D_1 = \text{donnée}$

donnée	Q_1	Q_0
0	0	1
1	1	0

$D_0 = \text{donnée } Q_0 + \text{donnée } \bar{Q}_0$

$D_0 = \text{donnée} \oplus Q_0$

$D_0 = \text{donnée} \oplus Q_0$

Q3.3. Quel intérêt voyez-vous au fait de générer un signal de valeur moyenne nulle sur le fil électrique pour la communication de données ?

On n'utilise pas d'énergie pour une valeur moyenne ne transportant pas d'information. Cela laisse libre le canal à basse fréquence (à très basse fréquence) pour d'autres transmissions ou pour l'alimentation

Le code AMI présente un inconvénient lors de longue série de données à « 0 ». A cause du signal toujours à 0V et de l'absence de rythme, le récepteur peut se désynchroniser. Par ailleurs, le récepteur ne peut pas faire la distinction entre une perte de signal (fil électrique débranché) et une longue série de « 0 ». On se propose d'éliminer ce problème en codant certains « 0 » par des niveaux électriques non nuls lors de séries de « 0 ».

Précisément, lors d'une série d'au moins 4 bits consécutifs à « 0 », le 4^{ème} bit à « 0 » est remplacé par un bit dit « bit de viol » à un niveau non nul ne respectant pas l'alternance des niveaux +V/-V.

Un exemple est donné dans la figure ci-dessous.

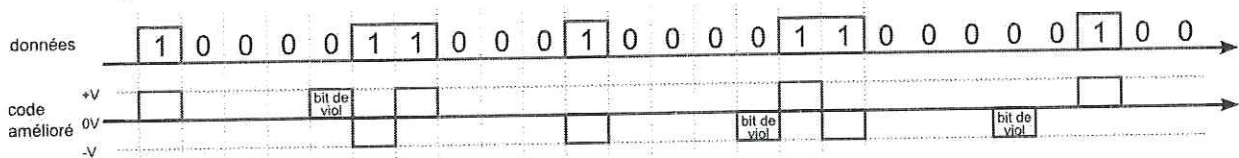


Figure 5: Chronogramme du code amélioré. Le 4^{ème} bit d'une série d'au moins 4 "0" est remplacé par un niveau non nul ne respectant pas l'alternance.

Q3.4. Proposez un diagramme d'état pour générer un signal codé avec l'amélioration proposée

