

Circuits et architectures logiques

Électronique numérique

2022 – durée 1h30 – Matthieu Denoual

CORRIGÉ

Consignes : les documents et calculatrices sont autorisés. Les réponses seront données sur ces feuilles à l'intérieur des espaces prévus à cet usage.

Partie 1 : Numération et codage [5]

Exercice 1 [2.5] : Convertissez en base 2 (**16 bits** virgule fixe **Q12,4**) le nombre : -517,69. Expliquez votre démarche et calculez l'**erreur de représentation**.

Nombre négatif, on traite d'abord la valeur absolue :

$$517.69 \cdot 2^4 = 8283,04 \rightarrow \text{arrondi } 8283$$

↓

205B h

↓

0010 0000 0101 1011

Complément à 1 : 1101 1111 1010 0100

Ajout bit à '1' à droite : 1101 1111 1010 0101

$$\text{Erreur} = |0.69 - (0.5 + 0.125 \cdot 0.0625)| = 0.0025$$

On vérifie que l'erreur est inférieure à l'erreur maximum.

$$\text{Erreur max} = q/2 = q^{-4}/2 = 2^{-5} = 0.03125$$

* les nombres binaires seront représentés en complément à deux sur **16 bits virgule fixe Q12,4**.

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

Exercice 2 [2.5] : Codez la valeur A en format 32 bits virgule flottante selon la norme IEEE 754 rappelée en bas de page. $A = -876\,085,248$

$$A = (-1)^1 \cdot 1.671 \cdot 2^{19}$$

$$E - 127 = 19 \rightarrow E = 146 \rightarrow 1001\,0010$$

0,671	0,736	0,776	0,416	0,656	0,496
x 16	x 16	x 16	x 16	x 16	x 16
10,736	11,776	12,416	6,656	10,496	7,936
1010	1011	1100	0110	1010	0111
					100 Arrondi

S	E7	E0	F22	F0
A	1 1001 0010 1010 1011 1100 0110 1010 100			

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 : $\underbrace{S}_{\text{signe}} \underbrace{e_7 e_6 \dots e_1 e_0}_E \underbrace{f_{22} f_{21} \dots f_2 f_1 f_0}_F$; E et F sont codés en binaire non signé.

Partie 2 : synthèse combinatoire [/7]

Exercice 3 [/5] – Transcodeur binaire vers biquinaire

L'objectif de cet exercice est la synthèse logique combinatoire d'un transcodeur binaire vers biquinaire. Le tableau 1 présente le code biquinaire.

Tableau 1: code biquinaire.
La ligne (1 0 8 6 4 2 0) représente les poids du code biquinaire

Chiffre	Code biquinaire						
	1	0	8	6	4	2	0
	q ₆	q ₅	q ₄	q ₃	q ₂	q ₁	q ₀
0	0	1	0	0	0	0	1
1	1	0	0	0	0	0	1
2	0	1	0	0	0	1	0
3	1	0	0	0	0	1	0
4	0	1	0	0	1	0	0
5	1	0	0	0	1	0	0
6	0	1	0	1	0	0	0
7	1	0	0	1	0	0	0
8	0	1	1	0	0	0	0
9	1	0	1	0	0	0	0

Question 3.1 [/3]

Quelle propriété caractérise le code biquinaire et quelle peut être son application ?

Il y a seulement 2 bits à '1' pour chaque code. Cela permet de repérer des erreurs de transmission si l'on constate un code avec un nombre de '1' différent de 2.

Question 3.2 [/2]

Établir les équations logiques d'un transcodeur permettant de passer du code binaire au code biquinaire.

Table de vérité du transcodeur binaire vers biquinaire

	Code binaire				Code biquinaire							
	8 b ₃	4 b ₂	2 b ₁	1 b ₀	1	0	8	6	4	2	0	
	q ₆	q ₅	q ₄	q ₃	q ₂	q ₁	q ₀					
0	0	0	0	0	0	1	0	0	0	0	1	
1	0	0	0	1	1	0	0	0	0	0	1	
2	0	0	1	0	0	1	0	0	0	1	0	
3	0	0	1	1	1	0	0	0	0	1	0	
4	0	1	0	0	0	1	0	0	1	0	0	
5	0	1	0	1	1	0	0	0	1	0	0	
6	0	1	1	0	0	1	0	1	0	0	0	
7	0	1	1	1	1	0	0	1	0	0	0	
8	1	0	0	0	0	1	1	0	0	0	0	
9	1	0	0	1	1	0	1	0	0	0	0	

Les tables de Karnaugh ne sont pas nécessaires dans ce cas, les équations peuvent être obtenues directement à partir de la table de vérité.

$$q_6 = b_0$$

$$q_5 = \bar{b}_0$$

$$q_4 = b_3$$

$$q_3 = b_2 \cdot b_1$$

$$q_2 = b_2 \cdot \bar{b}_1$$

$$q_1 = \bar{b}_2 \cdot b_1$$

$$q_0 = \bar{b}_3 \cdot \bar{b}_2 \cdot \bar{b}_1$$

Exercice 4 [/2] – Porte logique CMOS

Rappel : 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.

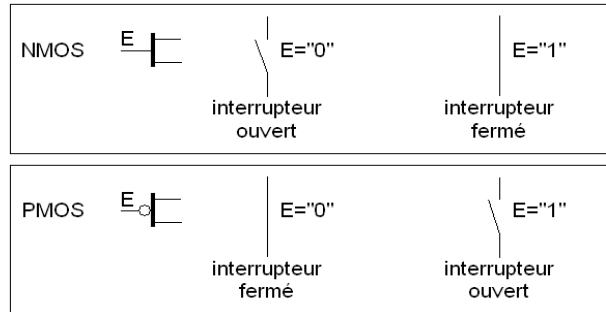
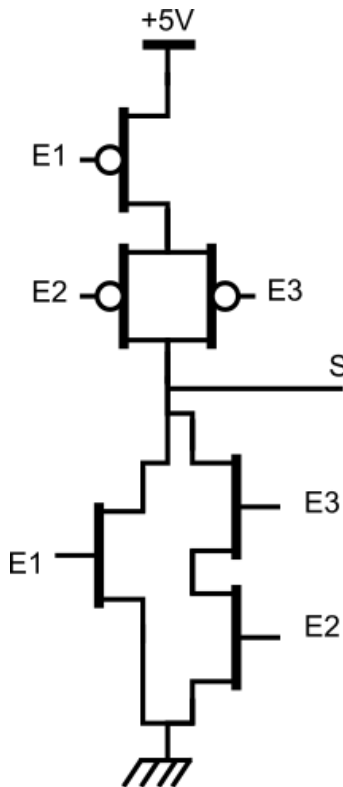


Figure 1: Modèles des transistors MOS (E entrée de commande)

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



S Fonction logique à 3 entrées

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

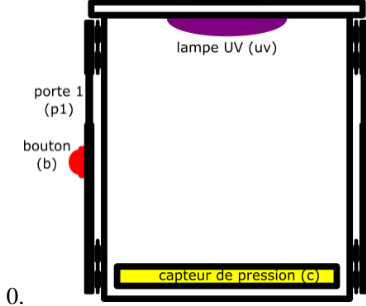
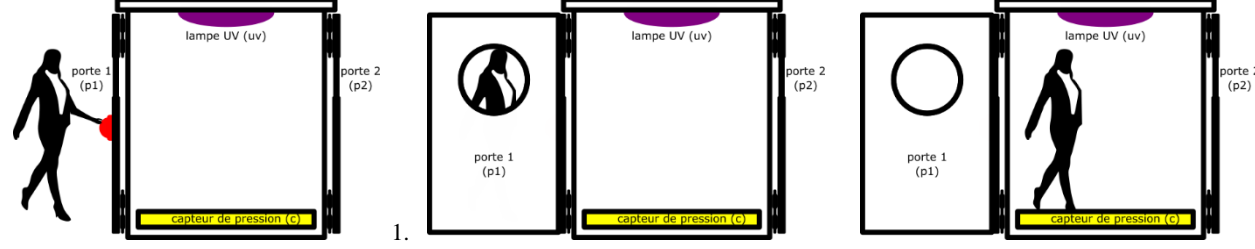
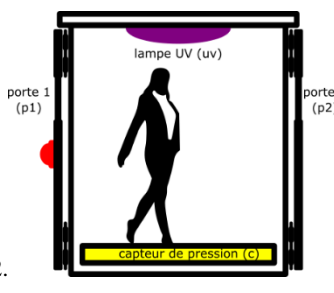
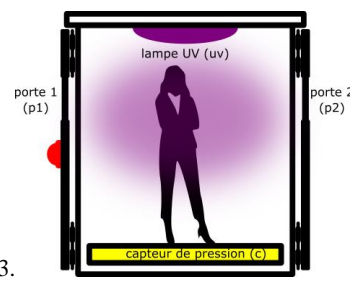
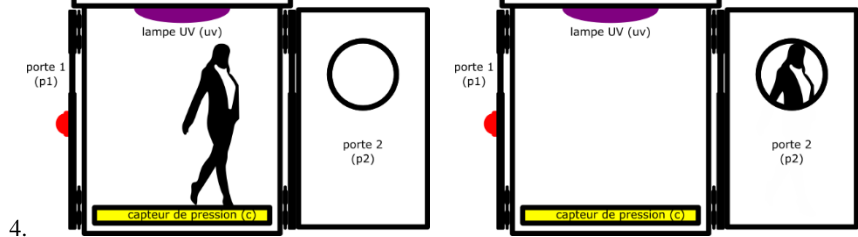
$S = \overline{E_1}(\overline{E_3} + \overline{E_2})$

Partie 3 : Synthèse logique séquentielle

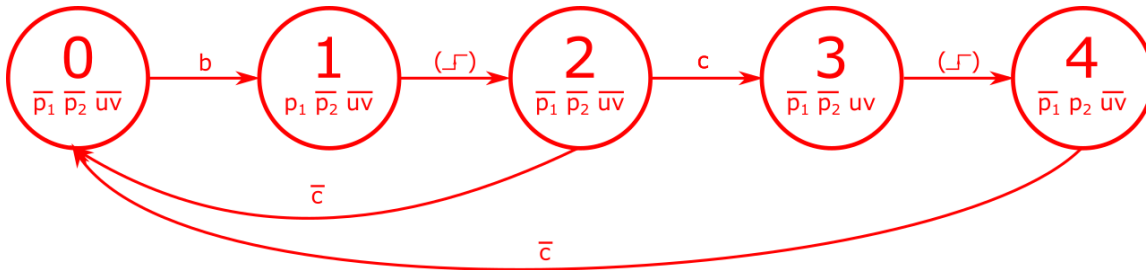
Exercice 5 [/8] Machine à états finis pour la gestion d'un sas de décontamination.

Le tableau 2 illustre le fonctionnement du sas de décontamination. Un temps de cycle identique est considéré pour toutes les étapes.

Tableau 2 : Illustration du fonctionnement du sas de décontamination.

<p>0. Par défaut le sas ne fait rien et attend l'arrivée d'une personne. Les portes (p1 et p2) sont fermées et la lumière UV (uv) est éteinte.</p>	 <p>0.</p>		
<p>1. Si une personne presse le bouton d'entrée (b), la porte 1 (p1) s'ouvre, laisse le temps à la personne d'entrer.</p>			
 <p>1.</p>			
<p>2. Une fois la personne entrée, la porte 1 (p1) se referme.</p>	 <p>2.</p>	<p>3. Si une personne est détectée dans le sas par le capteur de pression (c), la décontamination UV (uv) est effectuée. La décontamination UV prend un temps de cycle. Si aucune personne n'est détectée, le sas retourne à l'état par défaut.</p>	 <p>3.</p>
<p>4. Une fois la décontamination terminée, la porte 2 (p2) s'ouvre et reste ouverte tant que la personne est détectée par le capteur de pression (c).</p>	 <p>4.</p>		
<p>Finalement, la porte 2 (p2) se referme quand le capteur de pression (c) ne détecte plus personne dans le sas et le sas est de retour à la situation par défaut.</p>			

Question 5.1 : Traduisez le cahier des charges sous forme d'un diagramme d'état. Précisez si nécessaire les hypothèses ou conventions que vous adoptez.



.p₁ : porte d'entrée 1 (1 porte ouverte, 0 porte fermée)

.p₂ : porte de sortie 2 (1 porte ouverte, 0 porte fermée)

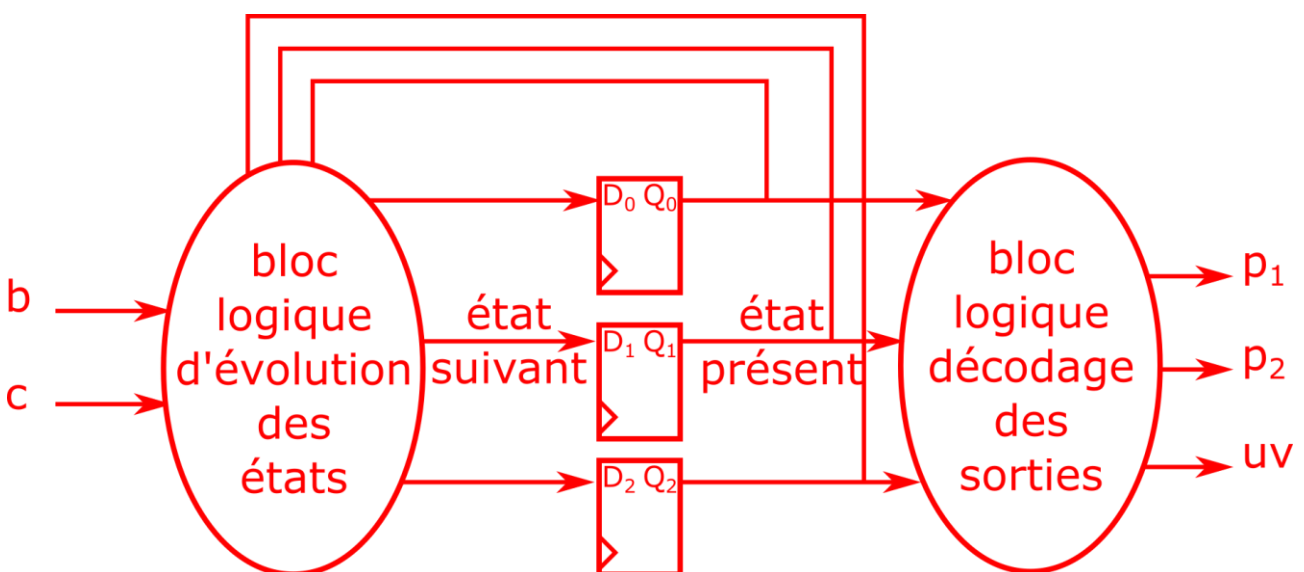
.uv : lumière UV (1 lumière UV ON, 0 lumière UV OFF)

5 états donc 3 bascules

2 entrées b (bouton) et c (capteur personne, capteur pression)

3 sorties : p₁, p₂, uv

Question 5.2 : Dessinez la structure globale de la machine à états finis de type Moore qui implémentera le diagramme d'état de la question 5.1.



Question 5.3 : Synthétisez la machine d'état. La synthèse ira jusqu'aux expressions algébriques. Veillez à décrire votre démarche et vos résultats intermédiaires.

Table de vérité du bloc logique d'évolution des états.

Les états 5, 6 et 7 sont non prévus dans le fonctionnement initial. Ils peuvent correspondre à un dysfonctionnement. Pour garantir qu'une personne ne reste pas coincée à la suite d'un dysfonctionnement conduisant la machine d'état dans un de ces états (5, 6 et 7), on propose de renvoyer vers l'état 4 qui ouvre la porte p2 avant de retourner à l'état initial une fois qu'aucune personne n'est détectée dans le sas.

Entrées système		Entrées				Sorties			
b	c	État présent				État suivant			
		Q2	Q1	Q0		D2	D1	D0	
0	X	0	0	0	0	0	0	0	0
1	X	0	0	0	0	1	0	0	1
X	X	0	0	1	1	2	0	1	0
X	0	0	1	0	2	0	0	0	0
X	1	0	1	0	2	3	0	1	1
X	X	0	1	1	3	4	1	0	0
X	0	1	0	0	4	0	0	0	0
X	1	1	0	0	4	4	1	0	0
X	X	1	0	1	5	4	1	0	0
X	X	1	1	0	6	4	1	0	0
X	X	1	1	1	7	4	1	0	0

$$D_0 = b \cdot \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0} + c \cdot \overline{Q_2} \cdot Q_1 \cdot \overline{Q_0}$$

$$D_1 = \overline{Q_2} \cdot \overline{Q_1} \cdot Q_0 + \overline{Q_2} \cdot Q_1 \cdot \overline{Q_0}$$

$$D_2 = c \cdot Q_2 \cdot \overline{Q_1} \cdot \overline{Q_0} + Q_2 \cdot Q_1 + Q_2 \cdot Q_0 + Q_1 \cdot Q_0$$

Table de vérité du bloc logique de décodage des sorties.

Entrées : état présent				Sorties		
Q2	Q1	Q0		p1	p2	uv
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	0	2	0	0	0
0	1	1	3	0	0	1
1	0	0	4	0	1	0
1	0	1	5	0	1	0
1	1	0	6	0	1	0
1	1	1	7	0	1	0

Porte p2 ouverte en cas de problème (états non prévus 5, 6 et 7), au passage, cela permet de simplifier l'expression de p2.

$$p_1 = \overline{Q_2} \cdot \overline{Q_1} \cdot Q_0$$

$$p_2 = Q_2$$

$$uv = \overline{Q_2} \cdot Q_1 \cdot Q_0$$