

# DEVOIR SURVEILLÉ – AP3

## ELECTRONIQUE

*Durée 2 heures*

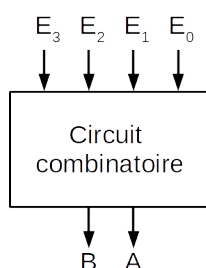
*2 pages*

*Sans document*

*Calculatrice non programmable autorisée*

### EXERCICE 1 : DÉTECTEUR DE 1

Soit un circuit combinatoire à 4 lignes d'entrée  $E_0, E_1, E_2$  et  $E_3$  ( $E_0$  étant le bit de poids faible) et 2 lignes de sorties A et B (B étant le poids faible), comme le montre la figure 1.



*Figure 1*

$EN/\overline{EN}$	$E_0 \dots E_3$
1	Fonctionnement normal
0	$B=A=0$

*Table 1*

Le fonctionnement est le suivant :

- Lorsqu'une seule ligne d'entrée parmi  $E_0, E_1, E_2$  ou  $E_3$  se trouve au niveau haut, son numéro est codé en binaire sur les sorties A et B.
- Si plusieurs lignes sont simultanément au niveau haut, le circuit code le numéro le plus élevé.
- Si toutes les lignes d'entrée sont au niveau bas, le circuit code  $AB=00$ ,

1. Donner la table de vérité du codeur.
2. Donner les expressions logiques des sorties A, B en fonction des entrées de  $E_0 \dots E_3$ .
3. En déduire le circuit logique du codeur. Vous utiliserez pour cela seulement les portes disponibles dans l'annexe.
4. Calculer le temps de propagation du circuit logique.
5. A l'aide de multiplexeurs 2 vers 1, modifiez le schéma précédant de façon à y ajouter une entrée  $EN/\overline{EN}$  dont la fonction est décrite par la Table 1.

### EXERCICE 2 : COMPTEUR/DÉCOMPTEUR

On veut réaliser un compteur/décompteur synchrone modulo 5. Un exemple de fonctionnement est présenté sur la Figure 2. Quand le signal Up / down est à 1, le système incrémente la sortie à chaque coup d'horloge. Arrivé à la valeur 4, le compteur recommence à 0. Quand le signal Up / down est à 0, le compteur décrémente sa sortie à chaque coup d'horloge. La sortie sera composée de trois signaux logiques  $Q_0, Q_1$  et  $Q_2$ , codant en binaire naturel la valeur de la sortie,  $Q_0$  étant le bit de poids faible.

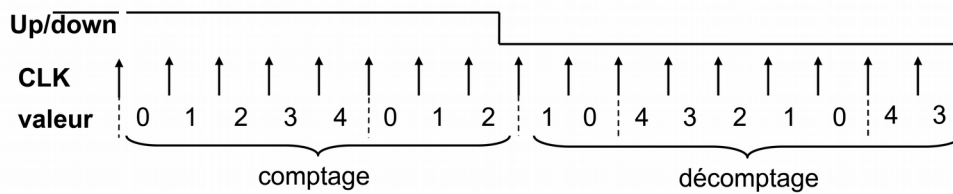


Figure 2 : Exemple de fonctionnement du compteur / décompteur modulo 5

Pour construire ce système, nous utiliserons la démarche vue en cours pour synthétiser une machine d'états.

1. Quelles sont les entrées et sorties du système ?
2. Construire le graphe d'états associé à ce système dans le cas d'une machine de MOORE.
3. Pour ce système, nous utiliserons des bascules de type D. De combien de bascules avez-vous besoin ? Justifiez votre réponse.
4. Écrire la table de transition du système complet. Pour le traitement des états non utilisés, l'état suivant sera indifférent et permettra donc de simplifier les équations logiques. Pour ces états, la valeur de sortie sera aussi sans importance.
5. Assignez des combinaisons binaires aux différents états que vous avez définis. Écrire la nouvelle table de transitions ainsi créée.
6. On appellera bloc F le bloc logique qui calcule l'état suivant à partir de l'état courant et bloc G, le bloc logique qui calcule l'état des sorties à partir de l'état courant. A partir de la table de transitions précédente, trouvez les équations simplifiées des blocs F et G (Pour la clarté des équations, le signal Up / down pourra être appelé X)
7. Faire un schéma propre de votre solution pour le système complet. Tous les types de portes sont acceptés.

## ANNEXE

On rappelle la formule du temps de propagation pour une porte logique :

$$t_p = t_{p_0} + \alpha \sum (\text{fan-in})$$

Fonction	Fan-in	$t_{p_0}$ (ns)	$\alpha$ (ns/fan-in)
INV	1	0,09	0,05
INV4X	3,7	0,08	0,01
OR2	0,75	0,34	0,05
OR3	0,8	0,39	0,05
NOR2	0,75	0,15	0,1
NOR3	0,8	0,2	0,1
XOR2	1,6	0,3	0,1
AND2	0,85	0,34	0,05
AND3	0,9	0,39	0,05
NAND2	0,85	0,15	0,1
NAND3	0,9	0,2	0,1
NAND4	1	0,25	0,1

Fonction	Fan-in (D,J ou K)	Fan-in CLK	tp0 (ns) CLK to Q	tp0 (ns) D to Q	$\alpha$ (ns/fan-in)	t <sub>su</sub> (ns)	t <sub>H</sub> (ns)
D flip-flop	0,9	1	1,2		0,06	0,4	0,5
JK flip-flop	0,7	0,9	1,6		0,06	0,5	0,02
D latch	0,70.	0,9	0,5	0,7	0,05		