

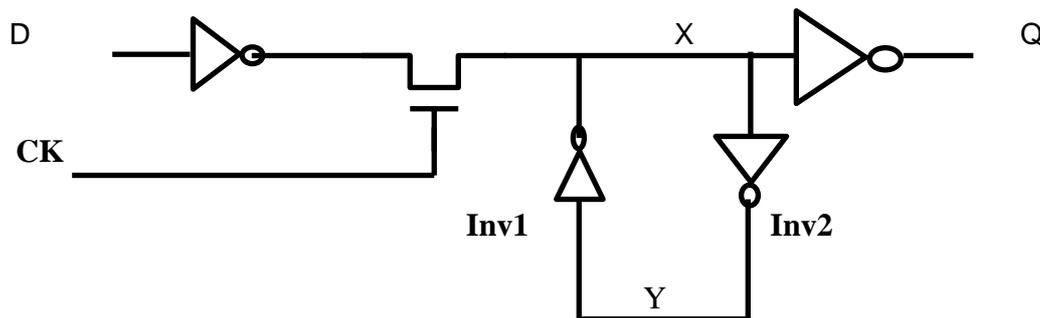


Majeure Micro-Nano-Optoelectronique
Cours : "Circuits électroniques"
PHY568
Sujet Examen

Durée 2h30
Polycopiés de cours et d'amphi autorisés
Corrigés de Petite Classe non autorisés

1^{er} Exercice : Dimensionnement d'un latch (6 points)

On considère le schéma de latch ci-dessous, appelé « latch à conflit » :



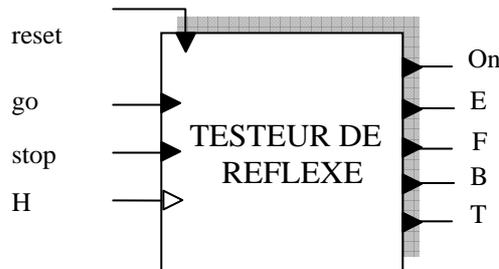
On suppose que l'inverseur d'entrée a pour dimensions $W_n = 10 \lambda$, $W_p = 20 \lambda$, $L_n = L_p = 1 \lambda$. Les dimensions de l'interrupteur N sont $W_n = 10 \lambda$, $L_n = 1 \lambda$.

- Q1) Fournir le chronogramme de fonctionnement de ce latch. Quelles sont les contraintes temporelles que doit respecter le signal D par rapport au signal CK ?
- Q2) Expliquer quel est le problème posé par l'écriture d'une nouvelle valeur dans ce latch.
- Q3) On suppose que le latch contient la valeur 1 ($X = 0$, $Y = 1$), et qu'on cherche à écrire la valeur 0 ($D = 0$, $CK = 1$). Donner une condition sur la valeur de résistance R_{on} du transistor N de l'inverseur « inv1 »
- Q4) On suppose qu'on cherche maintenant à écrire la valeur 1. Donner une condition sur la valeur de résistance R_{on} du transistor P de l'inverseur « inv1 »
- Q5) Proposez un dimensionnement pour les deux inverseurs « inv1 » et « inv2 » constituant la boucle de mémorisation, en justifiant votre réponse.



2^{ème} Exercice : Testeur de réflexes (8 points)

On souhaite réaliser un dispositif permettant de tester les réflexes, après une soirée arrosée par exemple. Ce testeur est réalisé comme un automate de Moore. Outre le signal d'horloge **H**, le circuit possède trois entrées : **go**, **stop**, **reset** et 5 sorties : **On**, **T**(richeur), **E**(xcellent), **B**(on), **F**(aible).



Le principe consiste à « mesurer » le temps écoulé entre l'activation successive des signaux **go** et **stop**. L'« unité de mesure » est le temps de cycle de l'horloge **H**. Le comportement de ce testeur peut être décrit comme suit :

- On suppose que le circuit a été initialisé grâce au signal **reset** dans un état où toutes les sorties valent 0. Le signal **reset** est synchrone, c'est à dire que sa valeur n'est prise en compte qu'au moment du front de l'horloge **H**.
- La sortie **On** passe à 1 lorsqu'une personne (le passager d'une voiture par exemple) active le signal **go**.
- La sortie **T** passe à 1 si la personne active le signal **stop** avant ou en même temps qu'elle d'active le signal **go**.
- Si le signal **stop** est effectivement activé après le signal **go**,
 - **E** passe à 1, si **stop** est activé un cycle après **go**.
 - **B** passe à 1, si **stop** est activé deux cycle après **go**.
 - **F** passe à 1, si **stop** est activé plus de deux cycle après **go**.

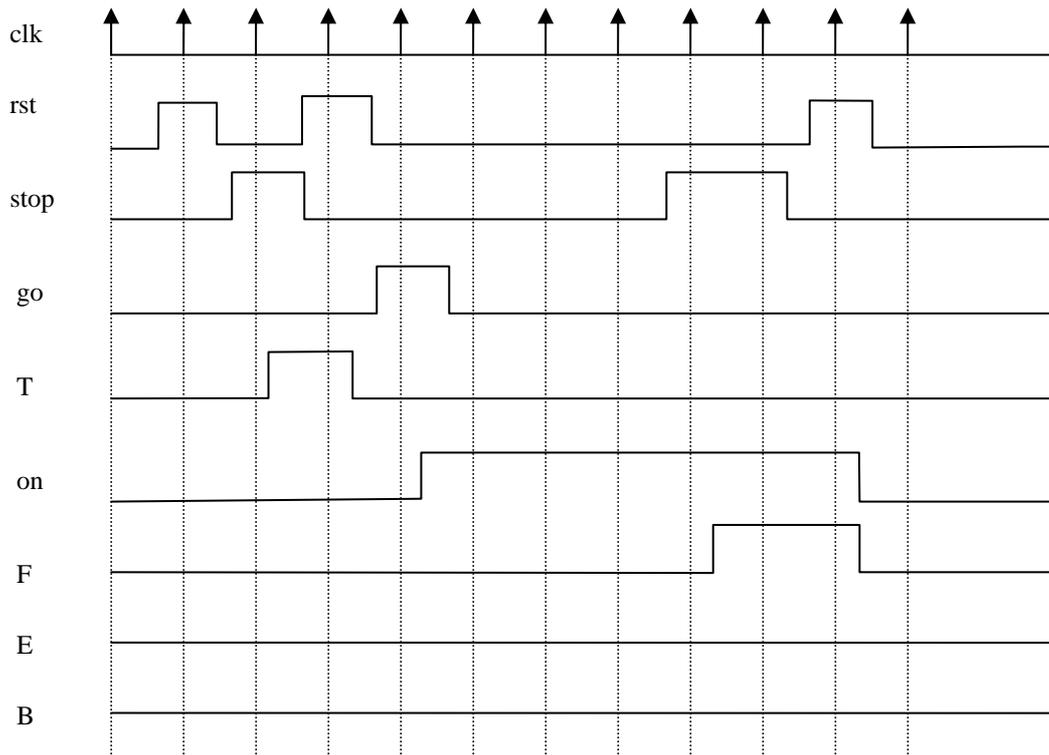
On notera que **F** ne passe à 1 qu'à partir du moment où l'utilisateur appuie sur le bouton **stop**. Il faut activer le signal **reset** pour remettre à 0 les signaux de sortie et démarrer un nouveau test, et le signal **reset** n'est pris en compte que si l'un des signaux **E,F , B ou T** a la valeur 1.

Q1) Représentez graphiquement l'automate de Moore à 8 états décrivant le comportement de ce testeur. On attachera à chaque transition l'expression Booléenne dépendant des entrées qui valide cette transition.

Q2) Construire le tableau définissant pour chaque état la valeur des 5 signaux de sortie.

Q3) En supposant un codage de type « one-hot » (une bascule par état), en déduire les expressions booléennes des 8 signaux d'entrée dans les 8 bascules représentant le registre d'état. Ces expressions Booléennes dépendent des 3 signaux d'entrée **go**, **stop**, et **reset**, et des 8 variables stockées dans le registre d'état.

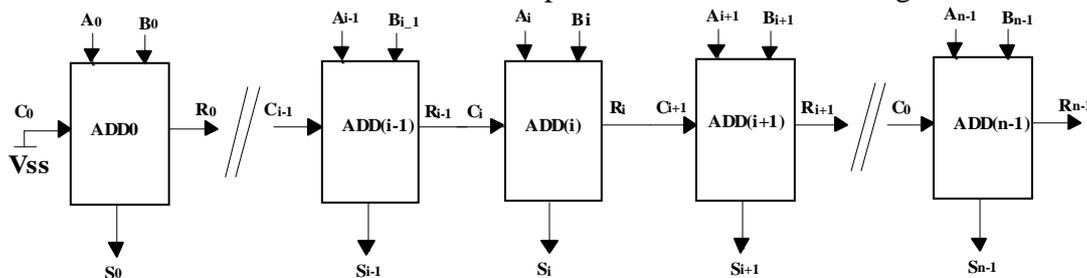
Q4) Donner les expressions booléennes des 5 signaux de sortie.



Exemple de scénario

3^{ème} Exercice : additionneur CMOS (6 points)

Un additionneur de 2 mots binaires de n bits ($A_{n-1} \dots A_i \dots A_0$ & $B_{n-1} \dots B_i \dots B_0$) peut être réalisé en utilisant n additionneur binaires complets comme décrit sur la figure suivante.



Le chemin critique au niveau temporel correspond à la propagation de la retenue. Nous chercherons donc une structure permettant d'accélérer la propagation de la retenue.

L'additionneur binaire complet permet d'additionner 3 bits de poids numérique égaux (a,b,c), et de recoder la valeur sur deux bits (r,s) où « s » est le bit de somme et « r » le bit de retenue.

Q1) Construire la table de vérité de l'additionneur binaire complet.

a	0	1	0	1	0	1	0	1
b	0	0	1	1	0	0	1	1
c	0	0	0	0	1	1	1	1
s								
r								



- Q2) Donner l'expression canonique de r et s sous forme de somme de produits, en utilisant les opérateurs OR, AND et NOT. Dans la forme canonique, tous les produits contiennent 3 termes.
- Q3) Simplifier les expressions Booléennes de r et s pour diminuer le nombre de littéraux, en utilisant les 3 mêmes opérateurs que dans la question précédente.
- Q4) Re-écrire le résultat précédent en utilisant les lois de Morgan de façon à n'utiliser que des opérateurs NAND, NOR, ou NOT et en minimisant le nombre de portes logiques sur le chemin de propagation de la retenue.
- Q5) Proposer un schéma complet pour l'additionneur binaire complet, et en déduire le nombre de portes traversées sur le chemin critique dans le cas de deux mots de 16 bits.