



ELECINF102

Processeurs et Architectures Numériques

Contrôle de connaissances

Vendredi 13 juin 2014 à 8h30

Document autorisé : une feuille recto-verso

Durée: 1h30 minutes

Ce contrôle comporte trois parties **indépendantes** :

1. Logique CMOS et performances
2. Compteur de Gray
3. Récepteur de données série

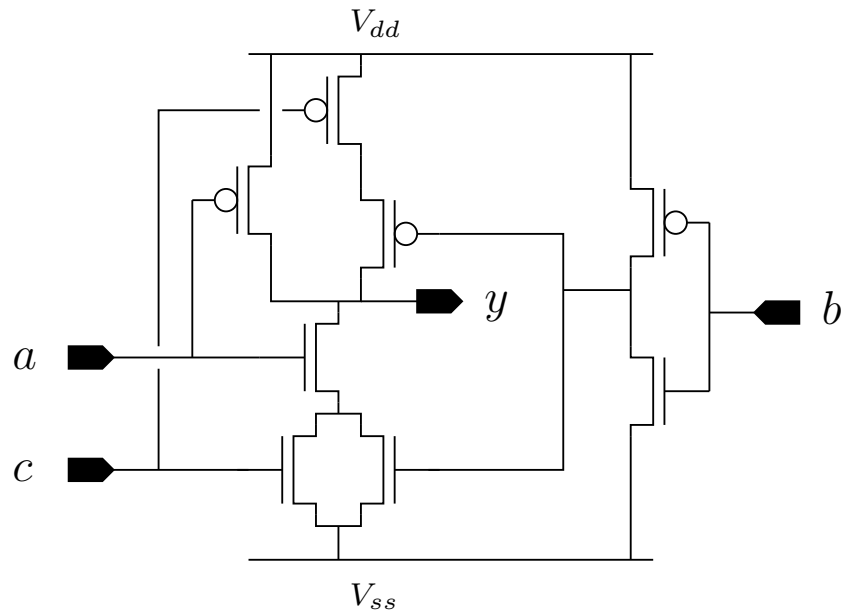
Consignes importantes : Si des **schémas** sont demandés dans les différents exercices, ils doivent être impérativement clairs, lisibles et sans ambiguïté. Les dimensions des bus doivent être indiquées. Si nécessaire le sens des signaux doit être précisé. Pour la logique synchrone, les signaux d'horloge et d'initialisation asynchrone (`reset_n`) ne seront pas représentés dans ces schémas.

N'oubliez pas d'inscrire nom, prénom, et numéro de casier sur votre copie.

Bon courage!

1 Logique CMOS

Soit le circuit suivant à base de transistors MOS. Les entrées correspondent aux signaux a , b et c , la sortie correspond au signal y .



Question 1 : Les règles de construction de la logique CMOS sont-elles respectées ? (*Justifiez votre réponse.*)

Question 2 : Donner l'équation logique de la fonction réalisée.

Question 3 : Donner une représentation schématique de cette fonction en utilisant exclusivement les portes logiques suivantes (pas de schéma à transistors) :

- ET (AND) à deux entrées,
- OU (OR) à deux entrées,
- INVERSEUR (NOT)

Pour la suite de l'exercice, nous utiliserons la représentation schématique trouvée à la question 3. Les conditions d'utilisation sont les suivantes :

- Les entrées a , b et c proviennent de sorties de bascules D,
- La sortie y est connectée à l'entrée d'une bascule D,
- Le temps de propagation de la port AND est $t_p^{AND} = 1$ ns,
- Le temps de propagation de la porte OR est $t_p^{OR} = 2$ ns,
- Le temps de propagation de la porte NOT est $t_p^{NOT} = 1$ ns,
- Le temps de propagation de la bascule D est $t_{co}^{DFF} = 1$ ns,
- Les temps de pré-positionnement t_{su} et de maintien t_h des bascules D sont nuls.

Question 4 : Déterminer la fréquence maximale de fonctionnement de ce circuit.

Question 5 : Proposer une modification de l'architecture pour augmenter cette dernière (à technologie et tension d'alimentation constantes).

Question 6 : Déterminer la nouvelle fréquence de fonctionnement.

2 Compteur de Gray

Question 1 : Codez en SystemVerilog un compteur binaire sur 2 bits avec remise à zéro (reset) asynchrone.

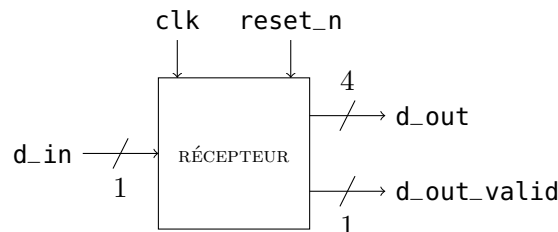
Question 2 : On désire changer la séquence binaire $0 \Rightarrow 1 \Rightarrow 2 \Rightarrow 3$ en séquence de Gray $0 \Rightarrow 1 \Rightarrow 3 \Rightarrow 2$. Ajoutez (toujours en SystemVerilog) un bloc de logique combinatoire aux sorties du compteur binaire de façon à ce que les nouvelles sorties suivent cette séquence.

Question 3 : On veut cette fois-ci réaliser la séquence de Gray avec un minimum de logique. Si l'on nomme C_1^c, C_0^c les valeurs courantes des 2 bits du compteur de gray, et si l'on nomme C_1^f, C_0^f les valeurs futures, calculez les équations booléennes de C_1^f et C_0^f en fonction de C_1^c et C_0^c . En déduire le code SystemVerilog du compteur de gray optimisé.

3 Réception de données en série asynchrone

Nous voulons réaliser un récepteur pour des **mots** de 4 bits arrivant en série (1 bit après l'autre). Chaque bit arrivant en série est appelé dans la suite **symbole** et tous les **symboles** ont la même durée. L'intérêt de ce type de transmission est de n'utiliser qu'un seul fil.

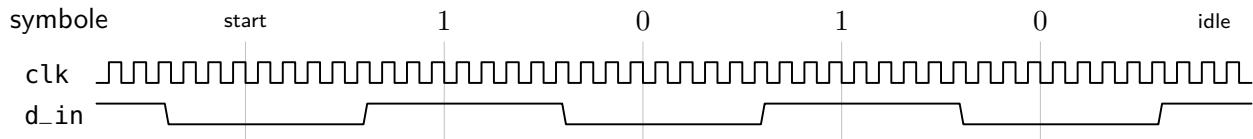
Le récepteur aura l'interface suivante :



Avec :

- **clk** une horloge dont la période est huit fois plus courte ($\div 8$) que la durée d'un **symbole**,
- **reset_n** un signal de remise à zéro asynchrone global,
- **d_in** l'entrée pour les données en série,
- **d_out** (4 bits) une sortie pour le **mot** reçu,
- **d_out_valid** qui indique en passant à 1 durant 1 cycle d'horloge que la valeur présente sur **d_out** est valide.

La figure suivante montre un exemple de transmission du **mot** 1010 arrivant sur l'entrée **d_in** du récepteur :



Comme la transmission n'utilise qu'un seul fil, le protocole suivant est utilisé

- Au repos l'entrée **d_in** est maintenue à 1. C'est l'état «*idle*».
- Pour indiquer le début d'un **mot**, on passe **d_in** à 0. Ce **symbole** ne fait pas partie du mot à transmettre et est appelé «*start*».
- Ensuite les 4 **symboles** du mot sont transmis.
- Enfin **d_in** repasse à 1.

Question 1 : Détection du symbole *start*

Il va falloir tout d'abord détecter le début d'une réception (passage de 1 à 0 sur **d_in**). Donnez l'architecture (sous forme d'un schéma ou d'un code SystemVerilog) d'un module synchrone ayant les entrées-sorties suivantes (en plus de l'horloge et du reset globaux) :

- **in**, une entrée venant du monde extérieur,
- **enable**, une entrée d'activation,
- **edge**, la sortie du module.

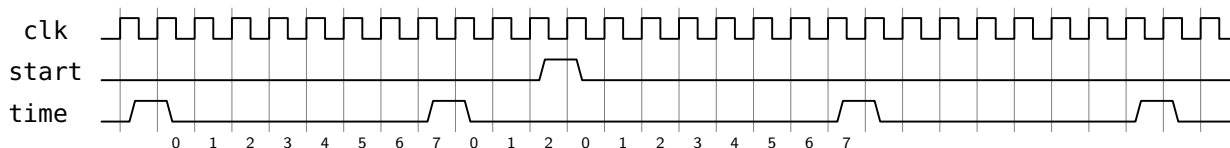
La sortie **edge** devra passer à 1 pendant 1 cycle d'horloge **clk**, lorsque l'entrée **in** passe de 1 à 0 et que l'entrée **enable** est à 1.

Question 2 : Génération d'une référence de temps

Nous devons générer une référence de temps pour échantillonner les symboles qui arrivent. Cette référence de temps doit pouvoir être resynchronisée à chaque nouvelle transmission d'un mot. Donnez l'architecture (schéma ou code SystemVerilog) d'un module synchrone ayant les entrées-sorties suivantes :

- **start**, une entrée de resynchronisation,
- **time**, la sortie du module.

Toutes les 8 périodes de l'horloge **clk**, le signal **time** passe à 1 durant une période de l'horloge **clk**. Lorsque le signal **start** vaut 1, la sortie **time** passe à 0 au cycle d'après pendant 7 cycles, avant de passer à 1 pendant un cycle, pour reprendre son rythme régulier. La figure suivante montre un exemple du comportement désiré :

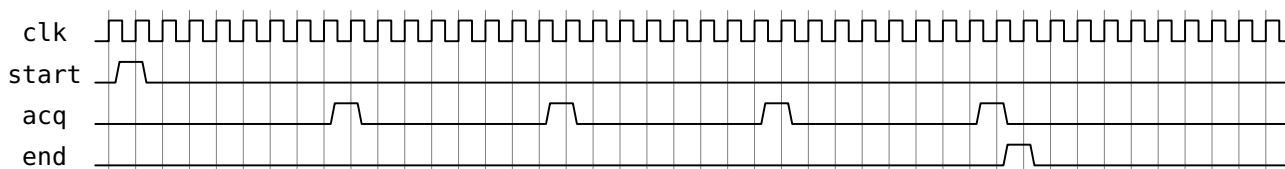


Question 3 : Comptage des symboles

Complétez le module précédent pour générer les signaux **acq** et **end** ayant le comportement suivant :

- le signal **acq** est le signal **time** mais ne passant à 1 que 4 fois après un passage à 1 du signal **start** ;
- le signal **end** passe à 1 pendant un cycle d'horloge après le 4^e passage à 1 du signal **acq**.

La figure suivante montre un exemple du comportement désiré :



Question 4 : Le récepteur complet

À partir des modules précédents, proposez une architecture (schéma ou code SystemVerilog) pour le récepteur complet.