



ELECINF102

Processeurs et Architectures Numériques

Groupe M

Partie théorique

Vendredi 8 juin 2012

Sans calculatrice

Document autorisé : une feuille recto-verso

Durée: 45 minutes

Ce contrôle comporte deux parties **indépendantes** :

1. Questions de cours
2. Calcul des termes d'une suite

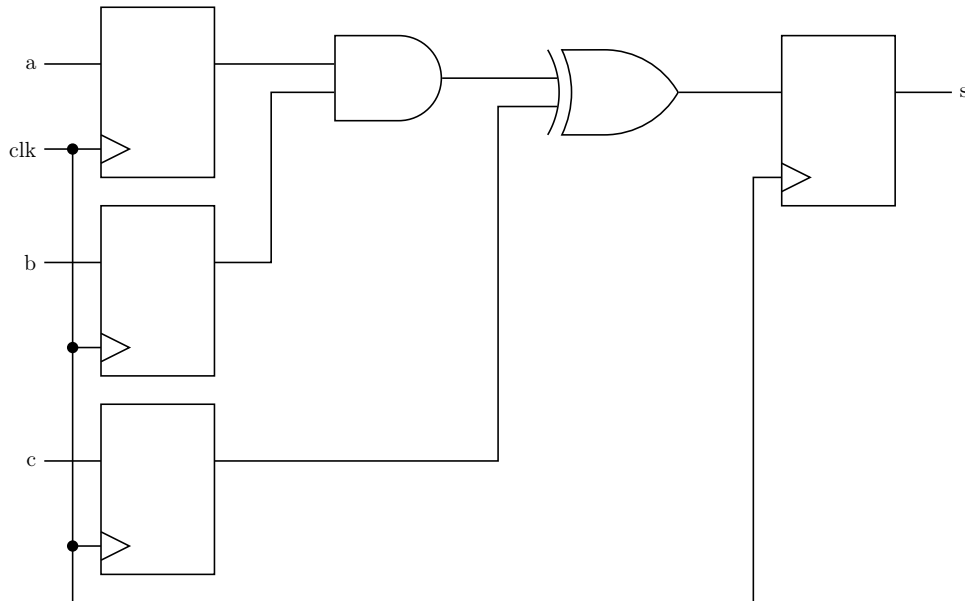
N'oubliez pas d'inscrire nom, prénom, et numéro de casier sur votre copie.

Bon courage!

1 Questions de cours

Question 1 : Indiquez la plage des valeurs représentables directement en utilisant une représentation non signée sur n bits et une représentation en complément à 2 sur n bits.

Question 2 : Soit la portion de circuit suivante :



Calculez la fréquence maximale de l'horloge (clk) pour que cette portion de circuit fonctionne correctement sachant que :

- le temps de propagation des portes (AND et XOR) est de 1 ns,
- le temps de propagation (t_{co}) des bascules est de 1 ns,
- le temps de pre-positionnement (t_{su}) des bascules est de 1 ns,
- le temps de maintien (t_h) des bascules est de 0,5 ns.

Question 3 : Décrivez, de façon concise, le comportement d'une bascule D.

Question 4 : Comment le nano-processeur (tel que vu en cours) sait que la donnée qu'il reçoit de la mémoire est une instruction, l'adresse d'un opérande ou la valeur d'un opérande ?

Question 5 : Dessinez le schéma générique de l'implémentation matérielle d'un automate fini (faites apparaître l'état courant, les entrées, les sorties, et les éventuels blocs combinatoires et séquentiels).

2 Calcul des termes d'une suite

On souhaite concevoir un module permettant de réaliser le calcul (*itératif*) d'un des termes de la suite définie par :

$$u_n = u_{n-1} + u_{n-2}$$

Les entrées du module sont :

- u_0 (8 bits non signés) : valeur du terme u_0 de la suite
- u_1 (8 bits non signés) : valeur du terme u_1 de la suite
- n (8 bits non signés) : l'indice n du terme de la suite à calculer (on supposera dans la suite que $n \geq 2$)

- `inputs_valid` (1 bit) : signal à 1 pendant un cycle d'horloge permettant d'indiquer que les données présentées sur `u_0`, `u_1` et `n` sont valides et que le calcul peut commencer. Il est prioritaire, c'est-à-dire que dès qu'il est à 1, un nouveau calcul commence, que le précédent soit terminé ou pas.
- `clk` (1 bit) : le signal d'horloge
- `reset_n` (1 bit) : signal de remise à zéro asynchrone actif à l'état bas

Les sorties du module sont :

- `u_n` (8 bits non signés) : valeur du terme u_n de la suite
- `overflow` (1 bit) : indique que le terme u_n n'est pas représentable sur 8 bits (1 : non représentable, 0 : représentable)
- `outputs_valid` (1 bit) : indique que les valeurs présentées sur les sorties `u_n` et `overflow` sont correctes (c'est-à-dire que le calcul est terminé). Il reste à 1 tant qu'un nouveau calcul n'est pas commencé.

Dans la suite cet exercice, vous aurez le droit aux éléments suivants :

- portes et bascules élémentaires vues en cours,
- blocs additionneur ou soustracteur ayant pour entrées deux mots sur 8 bits et une retenue entrante et pour sorties un mot sur 8 bit et une retenue sortante,
- opérateur d'égalité ayant pour entrée deux mots sur 8 bits et pour sortie un signal indiquant si les deux mots sont égaux.

Les schémas demandés doivent être *propres*, *lisibles* et *non ambigus*.

2.1 Question 1 — Partie traitement

Dans cette première question, on s'intéresse uniquement à la partie du circuit réalisant les calculs et pas à la partie contrôle. On suppose également que toutes les opérations sont effectuées modulo 2^8 (c'est-à-dire que l'on ne se préoccupe pas des dépassements lors des additions).

Dessinez le schéma d'un circuit calculant à chaque cycle d'horloge un terme de la suite, avec la possibilité de charger les deux premiers termes à l'aide des signaux d'entrées `u_0`, `u_1` et `inputs_valid`, et la possibilité d'arrêter le calcul à l'aide d'un signal `stop`.

2.2 Question 2 — Partie contrôle sans dépassement

On suppose que u_0 , u_1 et n sont toujours choisis de telle façon que u_n soit représentable sur 8 bits et donc qu'il n'y ait aucun dépassement ni sur le résultat final ni sur les résultats intermédiaires. Vous n'avez donc pas à vous préoccuper du signal `overflow`.

En utilisant la partie traitement que vous avez conçue lors de la première question, dessinez le schéma de la partie contrôle permettant de respecter le comportement du circuit décrit dans l'introduction.

2.3 Question 3 — Gestion du dépassement

Dans cette partie, nous ne faisons plus d'hypothèses sur u_0 , u_1 et n (excepté que $n \geq 2$) et donc il peut se produire un débordement au cours de l'une des additions.

Dessinez le schéma de ce module modifié afin de respecter le comportement décrit dans l'introduction.