



ELECINF102

Processeurs et Architectures Numériques

Contrôle de connaissances

30 juin 2017 à 15h15

Document autorisé : une feuille recto-verso

Durée: 1h30 minutes

Ce contrôle comporte 3 parties **indépendantes** :

1. Compteur modulo
2. Fibonacci ?
3. Etude d'une fonction de rendez-vous

Consignes importantes : Si des **schémas** sont réalisés, ils doivent être impérativement clairs 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) ne seront pas représentés dans ces schémas, mais l'état des bascules D à l'initialisation sera indiqué.

Si des **codes SystemVerilog** sont écrits, tous les signaux utilisés doivent être correctement déclarés, leur taille (nombre de bit) doit être définie. Les processus synchrones, ou combinatoires doivent être clairement distingués.

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

Bon courage !

1 Compteur modulo

On souhaite concevoir un module permettant de réaliser un compteur synchrone permettant au choix un comptage modulo 10 ou modulo 3.

Voici la description exhaustive des entrées et des sorties de ce module :

Signal	Direction	Nombre de bits	Description
clk	Entrée	1	L'horloge
reset	Entrée	1	Signal de remise à zéro synchrone actif à l'état haut
comptage	Entrée	1	Signal autorisant le comptage (s'il est à 1 lors d'un front d'horloge, le compteur s'incrémente, sinon, il garde sa valeur)
m3	Entrée	1	Signal indiquant si le comptage est modulo 3 (m3=1) ou modulo 10 (m3=0). Au front d'horloge, si le mode de comptage change alors, le compteur doit <ul style="list-style-type: none">• repartir à zéro si la valeur courante n'est pas compatible avec le modulo choisi,• continuer en séquence si la valeur courante est compatible avec le modulo choisi.
valeur	Sortie	4	Valeur actuelle du compteur

Écrivez le code SystemVerilog d'un tel module. Ce code doit être complet, il doit contenir la définition du nom du module, des ses entrées/sorties, ainsi que le code interne.

2 Fibonacci ?

Un concepteur prétend que le code SystemVerilog suivant génère la suite de Fibonacci en calculant sur 8 bits les nombres U_n .

On rappelle que la suite de Fibonacci est donnée par : $U_n = U_{n-1} + U_{n-2}$, avec $U_1 = 1$ et $U_0 = 0$.

```
module fibonacci(clk,
                reset_n,
                U);

input logic      clk;
input logic      reset_n;
output logic [7:0] U;

// Un-1 (= Un au cycle precedent)
logic [7:0]      U_1;

// Un-2 (= Un-1 au cycle precedent)
logic [7:0]      U_2;

always @(posedge clk or negedge reset_n)
  if(!reset_n)
    begin
      U   <= 1;
      U_1 <= 1;
      U_2 <= 0;
    end
  else
    begin
      // Un = Un-1 + Un-2
      U   <= U_1 + U_2;
      // Un-1 = Un retarde d'un cycle
      U_1 <= U;
      // Un-2 = Un-1 retarde d'un cycle
      U_2 <= U_1;
    end
end
endmodule
```

1. Réalisez un tableau des valeurs de **U**, **U_1** et **U_2** pendant les 10 premiers cycles d'horloge (après relâchement du **reset_n**).
2. Tracez le schéma du circuit modélisé par ce code (l'horloge et le signal d'initialisation ne seront pas représentés).
3. Quelle est l'équation de la suite U_n générée ?
4. Proposez une version corrigée du code SystemVerilog de façon à générer la suite de Fibonacci.

3 Etude d'une fonction de rendez-vous

Les figures 1 et 2 présentent deux implémentations d'une fonction logique **RV** à deux entrées **A** et **B** et une sortie **Y**. Bien que vous n'ayez pas encore étudié de telles structures, vous disposez des bases nécessaires pour les analyser.

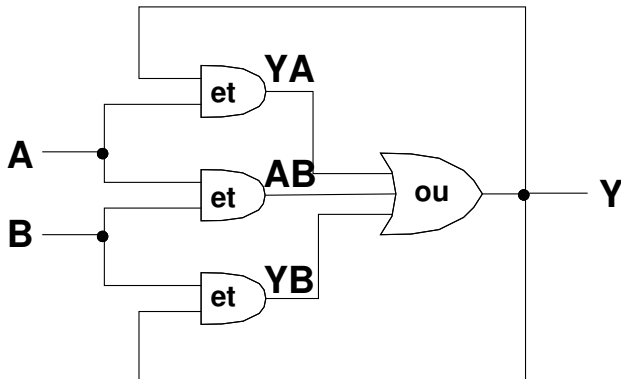


Figure 1 – Schéma de **RV** basé sur des portes logiques

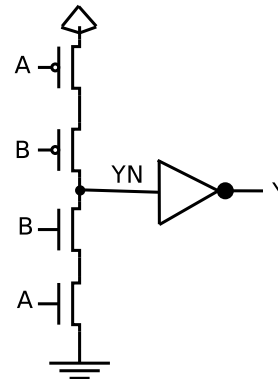


Figure 2 – Schéma de **RV** basé sur une construction à base de transistors

Analyse du schéma de la figure 1

- Les portes **ET** et **OU** ont toutes un temps de propagation de 1 dans une unité arbitraire.

Question 1 Complétez, de manière précise (avec les temps de propagation), le chronogramme de la **figure 3** page suivante.

Aide : A l'instant 0, **A** et **B** sont nuls, et donc **Y** est nul.

Question 2 Démontrez, en utilisant le chronogramme, que la fonction **RV** est une fonction séquentielle. Pourquoi appelle-t-on cette fonction, fonction de “rendez-vous” ?

Question 3 Complétez, de manière précise (avec les temps de propagation), le chronogramme de la **figure 4** page suivante. Qu'en déduisez vous sur les conditions limites d'utilisation de cette fonction ?

Analyse du schéma de la figure 2

Pour analyser le schéma on supposera

- Qu'un transistor bloqué a une résistance équivalente infinie
- Qu'un transistor passant a une résistance équivalente nulle
- Que l'entrée d'un inverseur est équivalent à une capacité connectée entre cette entrée et la masse.
- Que les temps de propagations sont considérés comme **nuls**.

Question 4 Complétez le chronogramme de la **figure 5** page suivante en indiquant l'état des réseaux de transistors **N** et **P** (**B**loqué ou **P**assant) dans chaque situation.

Question 5 Que se passe t'il lorsque le réseau **P** et le réseau **N** sont tous les deux bloqués ?

On utilise maintenant un modèle plus réaliste des transistors :

- Même dans l'état bloqué, la résistance équivalente des transistors n'est pas infinie.

Question 6 Quelles sont les conséquences sur les conditions limites d'utilisation de cette porte ?

Nom / Prénom :
 Numéro de casier :

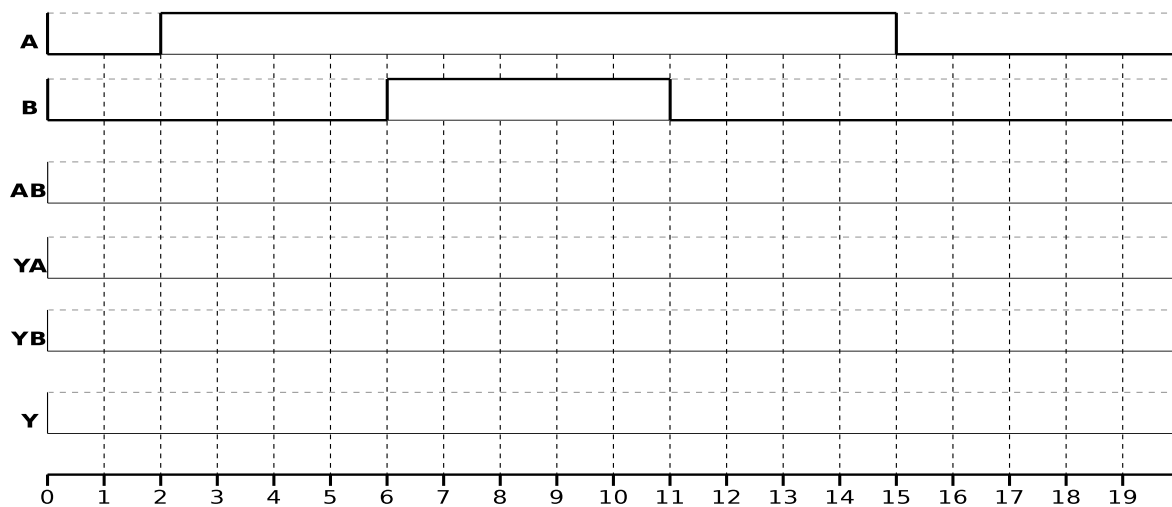


Figure 3 – Chronogramme de la question 1 (le temps est indiqué par incrément de "1")

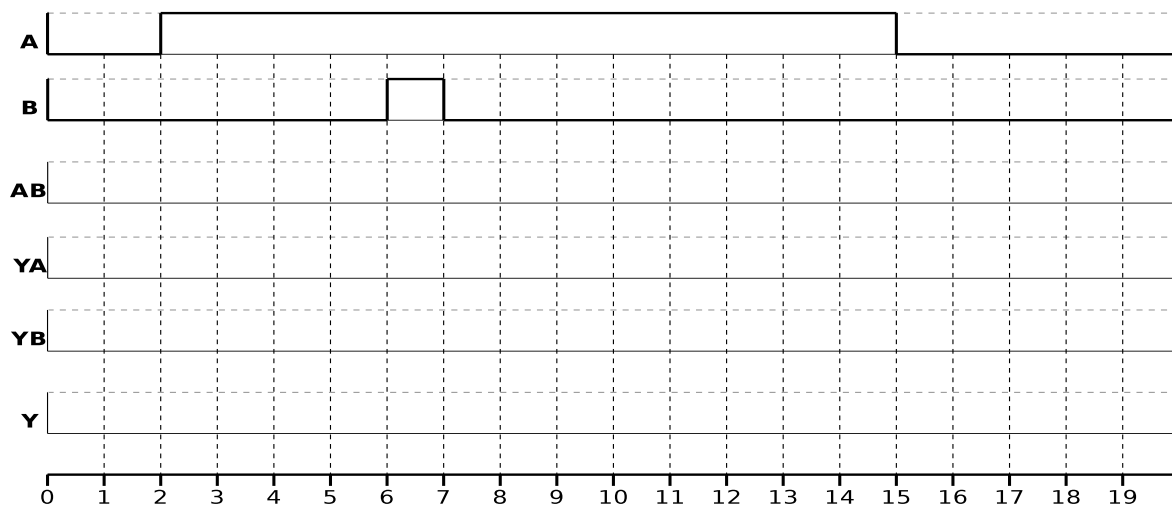


Figure 4 – Chronogramme de la question 3 (le temps est indiqué par incrément de "1")

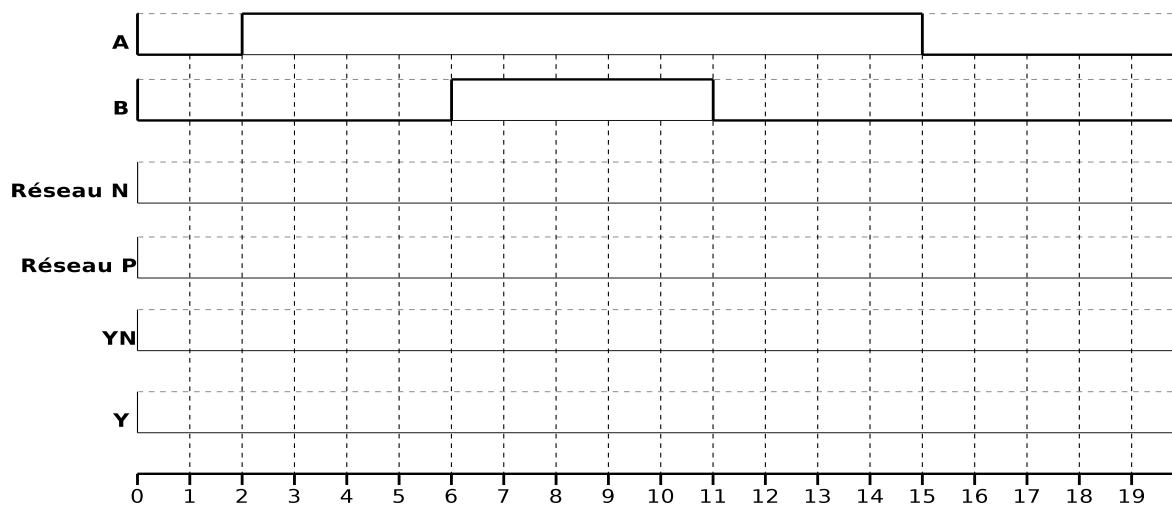


Figure 5 – Chronogramme de la question 4