



ELECINF102

Processeurs et Architectures Numériques

Contrôle de connaissances

Juillet 2015 Rattrapage

Document autorisé : une feuille recto-verso

Durée: 1h30 minutes

Ce contrôle comporte 3 parties **indépendantes** :

1. Reverse-engineering
2. Comptage de moutons
3. Techniques basse consommation

Bon courage!

1 Où le lecteur découvre la Rétroconception (reverse-engineering)

La rétroconception n'est pas la conception à la mode antique, mais l'analyse de la conception d'autres ingénieurs ou scientifiques de manière à comprendre, copier et éventuellement améliorer un dispositif existant.

Vous disposez du code SystemVerilog de l'opérateur **F1**, qui est manifestement un opérateur synchrone :

```
module F1(
    input  logic clk,debut,valide,
    input  logic [3:0] A,B,
    output logic [7:0] S
);

always @(posedge clk)
begin
    if(debut)
        S <= 8'b0;
    else if(valide)
        S <= A * B + S;
end

endmodule
```

Question 1 : Expliquez le traitement effectué par l'opérateur **F1**.

Vous disposez maintenant d'une variante **F2** du code SystemVerilog de l'opérateur.

```
module F2(
    input  logic clk, debut, valide,
    input  logic [3:0] A,B,
    output logic [7:0] S
);

logic [7:0] P;
always @(posedge clk)
begin
    if(debut)
        S <= 8'b0;
    else if(valide)
        begin
            P <= A * B;
            S <= P + S;
        end
end

endmodule
```

Question 2 : L'opérateur **F2** réalise-t-il le même calcul que l'opérateur **F1** ? Justifiez votre réponse qu'elle soit positive ou négative.

Vous disposez d'une troisième variante **F3** du code SystemVerilog de l'opérateur.

```
module F3(  
    input logic clk, debut, valide,  
    input logic [3:0] A,B,  
    output logic [7:0] S  
);  
logic [7:0] P,X;  
  
always @( * )  
    P <= A * B;  
  
always @( * )  
begin  
    if(debut)      X <= 8'b0;  
    else if(valide) X <= P + S;  
    else          X <= S;  
end  
  
always @(posedge clk)  
    S <= X;  
endmodule
```

Question 3 : L'opérateur **F3** réalise-t-il le même calcul que l'un des opérateurs **F1** ou **F2** des questions précédentes ? Justifiez votre réponse.

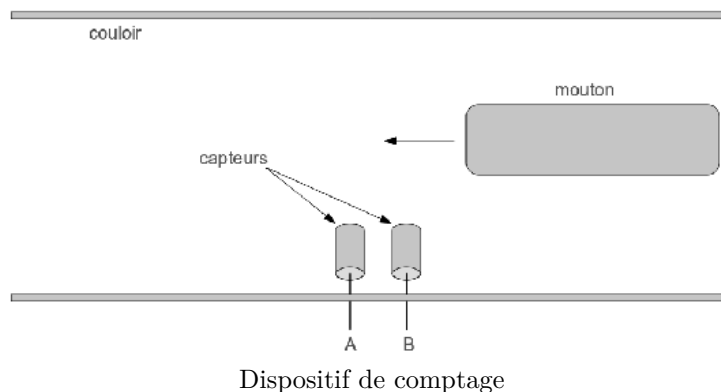
Question 4 (BONUS) : Nous désirons choisir, parmi **F1**, **F2** ou **F3**, l'opérateur pouvant fonctionner à la fréquence d'horloge la plus élevée (**indépendamment la fonction réalisée**). Quel opérateur choisissez vous ? Justifiez votre réponse.

2 Comptage de moutons

Notes :

- Toutes les bascules de vos circuits seront synchrones sur une horloge **clk** et **uniquement sur celle-ci**. Elle seront de plus initialisées par un signal **reset_n** asynchrone actif à l'état bas (**0**).
- Tous les signaux utilisés devront être explicitement déclarés.
- La taille (nombre de bits) de tous les signaux utilisés devra être explicitement déclarée.
- Il sera inutile de déclarer les entêtes (**module xxx(input...)**), et fin (**endmodule**) des modules.

On cherche à compter les moutons d'un troupeau (comportant moins de 2^{32} moutons). Pour cela, les moutons passent l'un après l'autre dans un couloir où sont placés deux détecteurs. Chaque détecteur renvoie un signal sur 1 bit, valant **1** lorsqu'un mouton passe en face de lui et **0** sinon.



Les deux détecteurs, *A* et *B*, sont éloignés d'environ 10cm : lors du passage d'un mouton, il y a donc un moment durant lequel les deux détecteurs voient en même temps le mouton : $A = B = 1$.

On supposera que les sorties des détecteurs sont synchrones sur l'horloge **clk** de votre circuit, et que l'horloge va suffisamment vite pour qu'il s'écoule plusieurs cycles entre deux événements successifs sur le bus de sortie des capteurs.

Question 1

On suppose pour l'instant que les moutons ne font qu'avancer de la droite vers la gauche.

En vous aidant éventuellement d'un chronogramme (réalisé par vos soins) des sorties des capteurs lors du passage d'un mouton, donnez le code SystemVerilog d'un système comptant le nombre de moutons passés dans le couloir.

Question 2

Les moutons étant des animaux capricieux, une fois arrivés tout au bout du couloir ils peuvent faire demi-tour et revenir dans l'enclos de départ. Il faut dans ce cas les dé-compter !

Donnez une **nouvelle** version de votre code de façon à pouvoir maintenir le compte correct des moutons arrivés à destination.

Question 3 (BONUS)

L'imagination des moutons étant sans bornes, ils peuvent avoir envie de faire demi-tour au moment précis où ils passent devant les détecteurs.

Si nécessaire, donnez une nouvelle version de votre code, de façon à prendre en compte le fait que les moutons peuvent faire demi-tour n'importe quand.

3 Techniques basse consommation

L'objectif de cet exercice est d'explorer des techniques architecturales permettant de réduire la consommation des circuits intégrés. Nous disposons d'une fonction de calcul combinatoire \mathbf{F} pour laquelle nous connaissons :

- Le temps de calcul T_{calc} en fonction de la tension d'alimentation V_{dd} (voir **figure 3**)
- L'énergie consommée par calcul E_{calc} en fonction de la tension d'alimentation V_{dd} (voir **figure 4**)

Nous rappelons que la puissance consommée par un opérateur de calcul est égale au produit de sa fréquence de fonctionnement F_{calc} par l'énergie consommée à chaque calcul :

- $P_{calc} = F_{calc} * E_{calc}$

3.1 Analyse d'une structure synchrone utilisant la fonction (F)

La **figure 1** présente une utilisation de la fonction (F) dans un environnement synchrone. La structure reçoit une suite de valeurs A_0, A_1, \dots et doit générer une suite de valeurs $\mathbf{F}(A_0), \mathbf{F}(A_1), \dots$

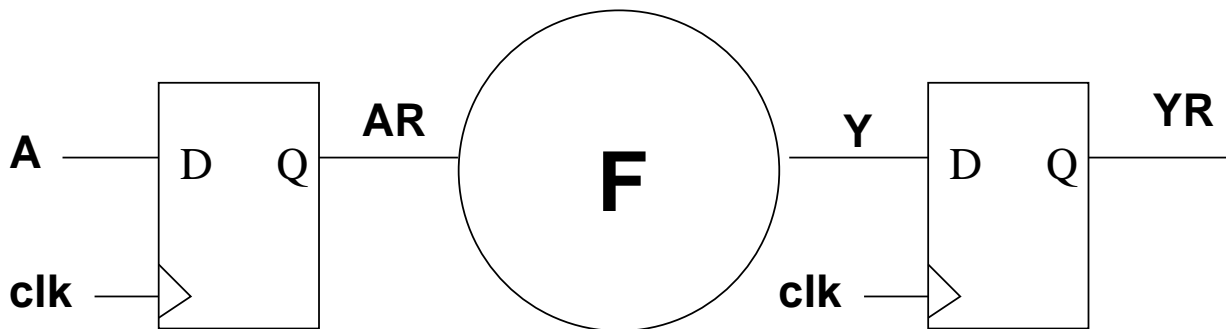


Figure 1 – Architecture synchrone

Question 3.1.1 Complétez le chronogramme de la **figure 5**.

Question 3.1.2 En négligeant le temps de propagation des bascules, déterminez la tension d'alimentation minimale V_{ref} permettant de faire fonctionner cette structure avec une horloge **clk** de fréquence $F_{ref} = 1GHz$.

Question 3.1.3 En négligeant la consommation des bascules, déterminez la puissance consommée par cette structure pour la tension V_{ref} et la fréquence F_{ref} . N'oubliez pas de préciser les unités...

3.2 Analyse d'une structure synchrone parallélisée

La **figure 2** présente une version parallélisée de l'architecture précédente, dans laquelle nous dédoublons la fonction **F**.

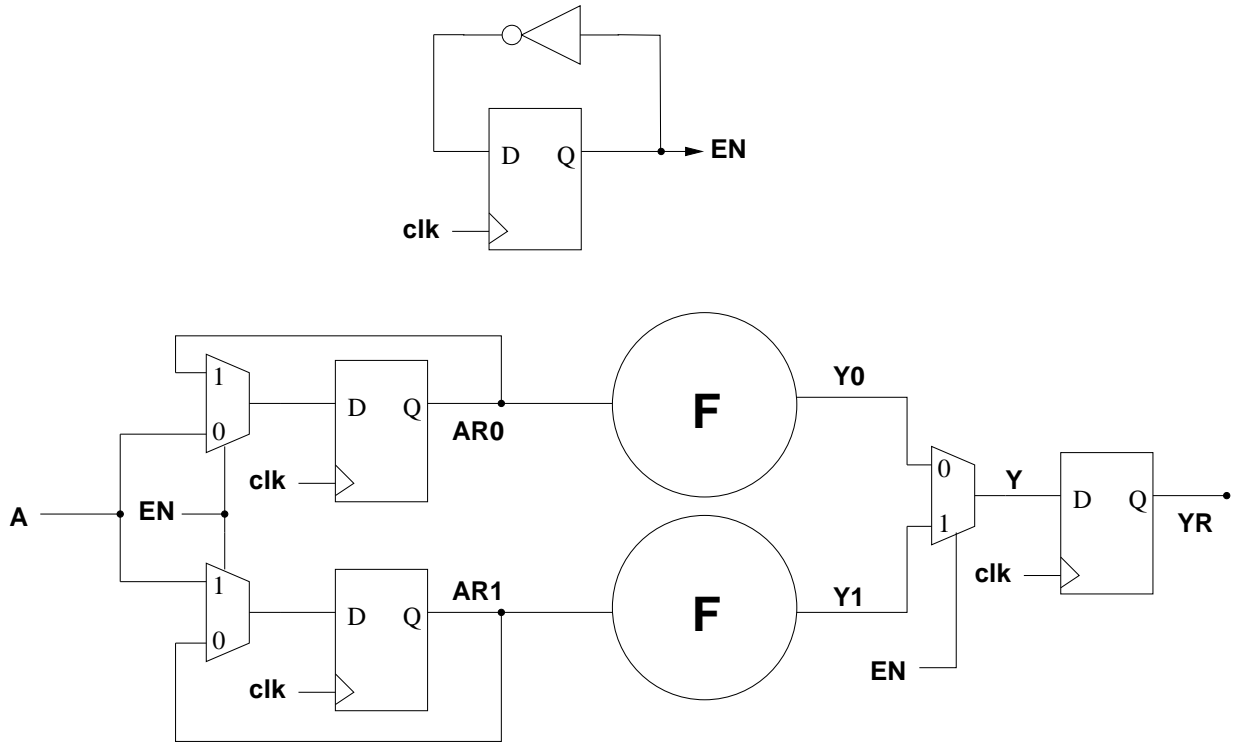


Figure 2 – Architecture synchrone parallèle

Question 3.2.1 Complétez le chronogramme de la **figure 6**. N'oubliez pas la génération du signal **EN**.

Question 3.2.2 Expliquez le fonctionnement de cette structure.

Question 3.2.3 Pendant combien de périodes de l'horloge **clk** chacune des données A_i est elle maintenue en entrée de l'une ou l'autre des fonctions **F**? De combien de temps dispose-t-on pour le calcul de chaque $F(A_i)$ sachant que l'horloge a une fréquence $F_{ref} = 1GHz$? (on négligera le temps de propagation dans les multiplexeurs).

Question 3.2.4 Compte tenu du nouveau temps de calcul disponible, montrez qu'il est possible de diminuer la tension d'alimentation du montage tout en fixant la fréquence F_{ref} à $1GHz$. Déterminez la tension d'alimentation minimale V_{min} permettant de conserver cet fréquence de $1GHz$.

Question 3.2.5 En déduire l'énergie E_{min} consommée par la fonction **F** à la tension V_{min} . Puis la puissance totale consommée par la structure parallèle. N'oubliez pas la encore de préciser les unités.

Question 3.2.6 (BONUS) Nous avons utilisé une technique de parallélisme pour diminuer la consommation d'une structure de calcul synchrone. Pensez vous qu'une technique de pipeline (pipeline de la fonction **F** en deux sous-fonctions **F1** et **F2** de temps de calcul $T_{calc}/2$) permettrait d'obtenir un résultat similaire? Expliquez.

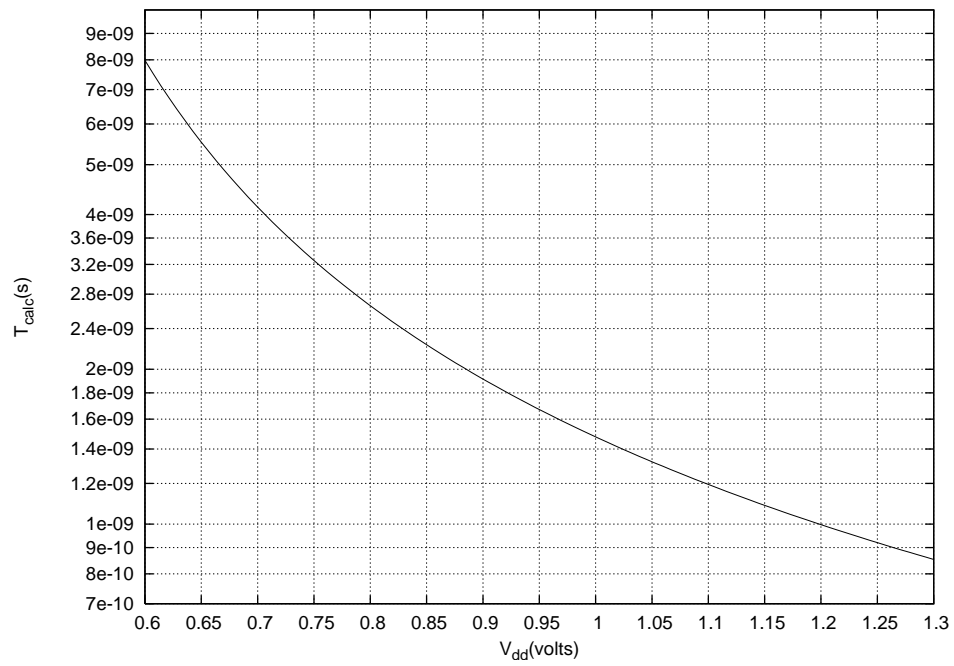


Figure 3 – Temps de calcul de la fonction **F**

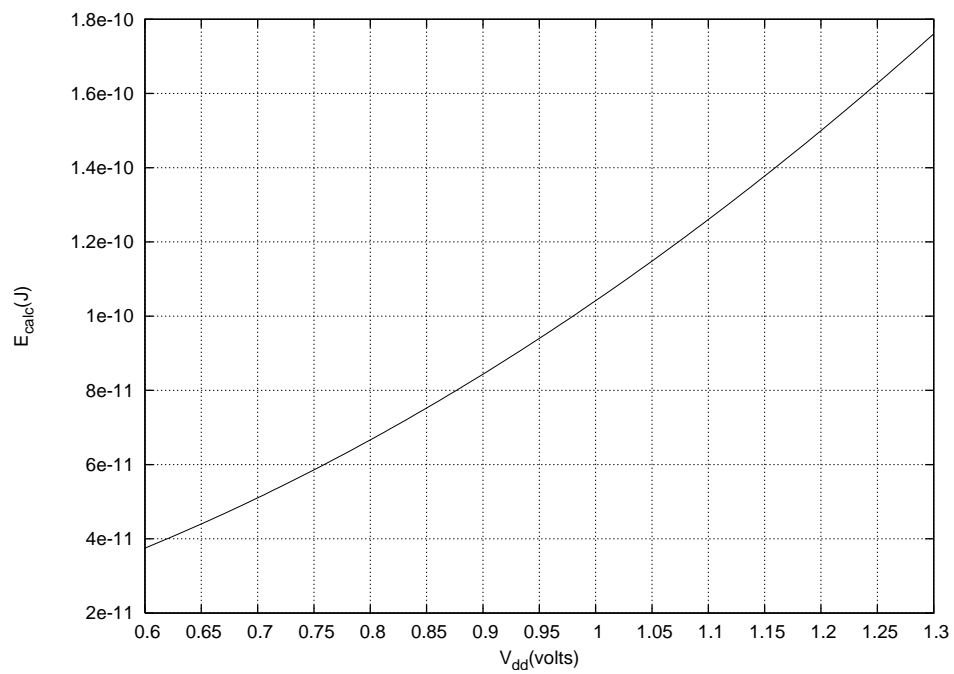


Figure 4 – Energie consommée par calcul de la fonction **F**

Nom / Prénom :
Groupe :
Numéro de casier :

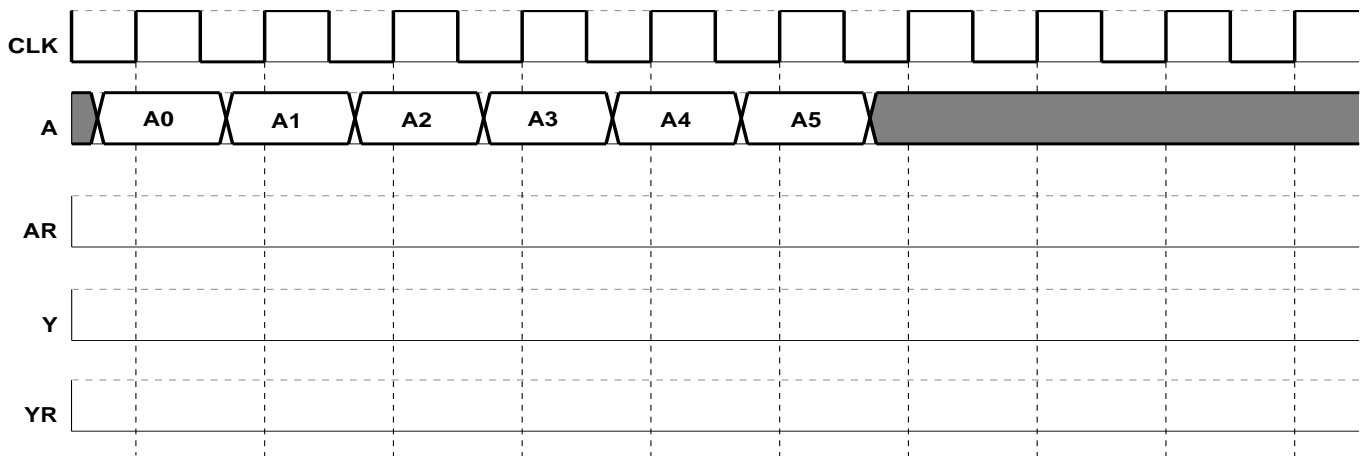


Figure 5 – Chronogramme de fonctionnement de la structure synchrone (Question 3.1.1)

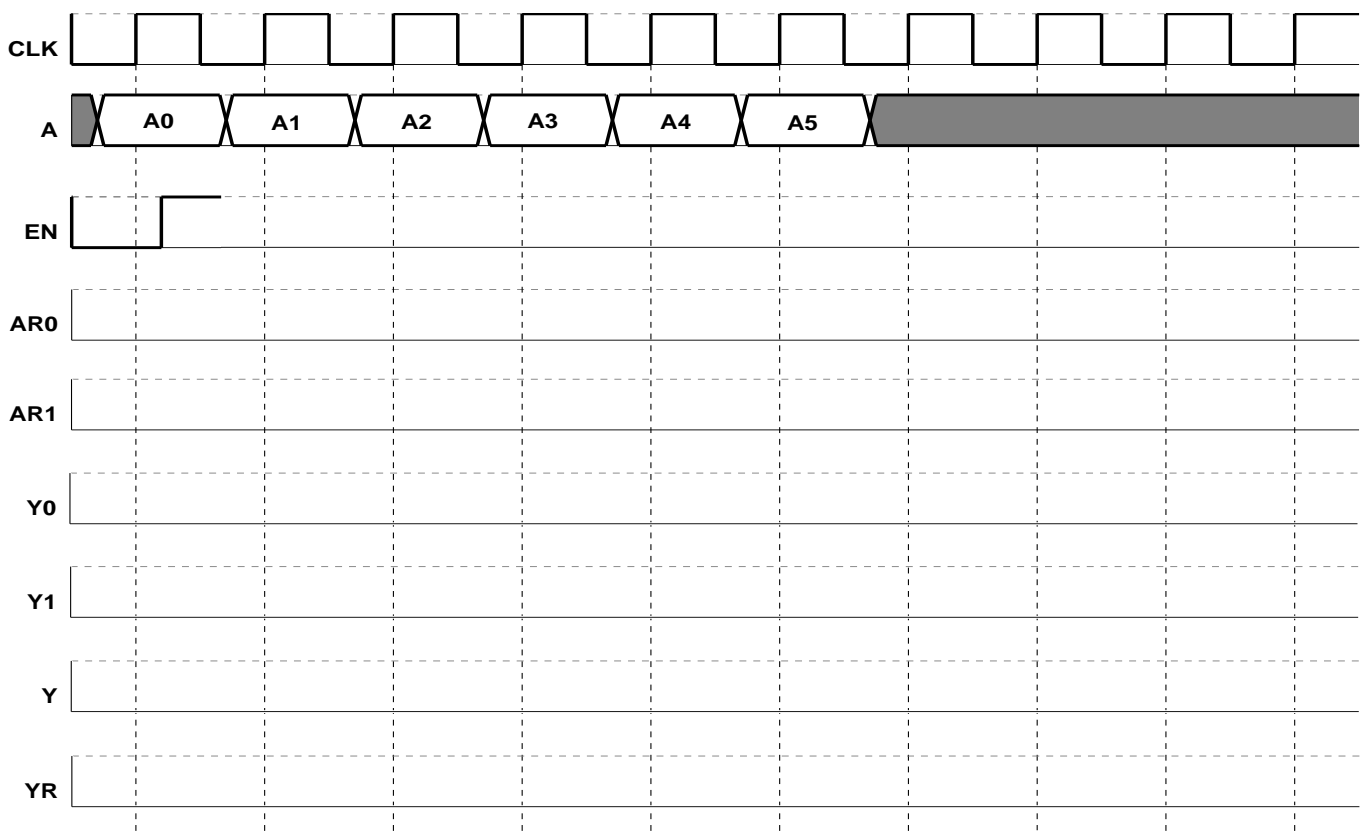


Figure 6 – Chronogramme de fonctionnement de la structure parallèle synchrone (Question 3.2.1)