

LABORATOIRE
REGISTRES À DÉCALAGE

Johanna Melly & Sven Rouvinez

Professeur
Carlos Andrés Pena

Assistant
Gaëtan Matthey

6 décembre 2016

1 Registre à décalage 4 bits

Sur la base du registre à décalage vu dans le cours, concevoir et implémenter un registre à décalage de 4 bits qui peut décaler à gauche (SHL), décaler à droite (SHR), charger un nibble (LOAD) ou garder son contenu (HOLD). Le registre à décalage doit être réalisé à l'aide de flip-flops D et de multiplexeurs.

Registre à décalage 4 bits- Max 1 page Insérez une capture d'écran pour présenter votre bloc Registre à décalage 4 bits (Structure interne). Accompagnez-le de commentaires et d'explications nécessaires à sa compréhension. Remplacez le texte ci-dessus par vos réponses (à l'intérieur du cadre rouge)

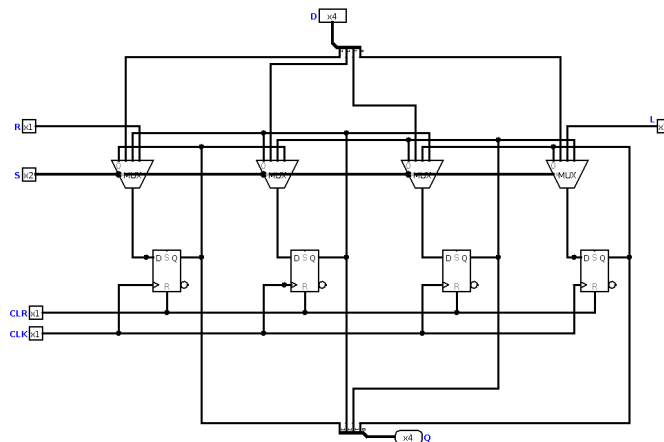


FIGURE 1 – Bloc SHIFT REGISTER 4 bits

Le registre à décalage 4 bits permet d'effectuer un décalage de tous les bits sur la droite ou sur la gauche. L'entrée **D** représente les bits qu'on veut pouvoir décaler. L'entrée **S** permet de choisir l'action à effectuer :

- 00 : HOLD, la valeur reste inchangée
- 01 : LOAD, la valeur est placée dans l'état voulu
- 10 : shift left, les bits sont tous décalés d'un cran vers la gauche
- 11 : shift right, les bits sont tous décalés d'un cran vers la droite

L'entrée CLK correspond à notre horloge, l'utilisation d'une entrée simple se justifie de la sorte que le bloc sera utilisé dans un autre circuit qui lui aura une vraie horloge.

Lorsque l'horloge passe à 1 (flanc montant), la valeur des bits dans les balances est changée, et lorsqu'elle passe à 0, la valeur des bits est conservée. Ainsi, pour effectuer un décalage, il sera tout d'abord nécessaire de mettre l'entrée **S** à 1 et d'avoir un flanc montant pour que les bits des balances prennent la valeur de l'entrée **D**. Ensuite, on peut changer la valeur de l'entrée **S** dans le but d'avoir un décalage sur la droite ou sur la gauche. Dès lors, à chaque fois que l'horloge passera de 0 à 1, les bits vont tous se décaler d'un cran.

2 Registre à décalage 8 bits

Concevoir un registre à décalage de 8 bits en utilisant deux registres à décalage de 4 bits

Registre à décalage 8 bits - Max 2 pages

Insérez une capture d'écran pour présenter votre bloc Registre à décalage 8 bits (Structure interne). Accompagnez-le de commentaires et d'explications nécessaires à sa compréhension. Insérez une capture d'écran du chronogramme qui montre le chargement (Load) d'une valeur (exemple : 0x80), puis le décalage à droite de la valeur chargée (8 fois consécutif). Accompagnez le chronogramme d'explications. Remplacez le texte ci-dessus par vos réponses (à l'intérieur du cadre rouge)

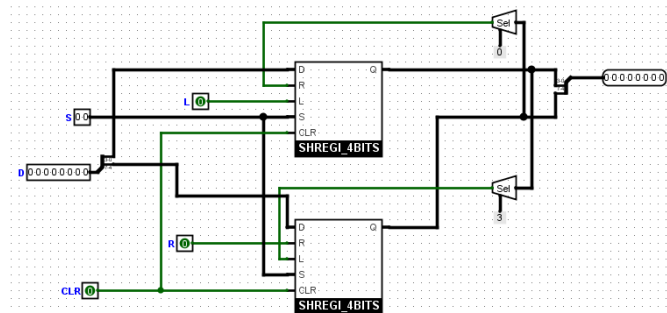


FIGURE 2 – Bloc SHIFT REGISTER

Le shift register 8 bits est composé de 2 shift register 4 bits. Ils ont comme entrée en commun l'horloge CK, S qui détermine l'action à effectuer, ainsi que le reset. L'octet est séparé en deux nibbles qui sont chacun en entrée D des register 4 bits. Lorsqu'un décalage est fait, le dernier bit d'un register doit être récupéré par le suivant, afin de ne pas perdre les bits décalés. Ainsi, pour le décalage à gauche, le bit [0] du shift register prenant en entrée D les bits [7:4] de l'octet (le SHREGI_2 de l'illustration), doit être passé en entrée R de l'autre shift register (SHREGI_1). Ainsi, au moment du décalage à gauche, le bit [0] du SHREGI_2 va bien être passé au SHREGI_1 et il y aura une continuité dans le décalage. Le même principe est appliqué pour le décalage à droite, mais cette fois c'est le bit [3] du SHREGI_1 qui va être donné en entrée L du SHREGI_2.

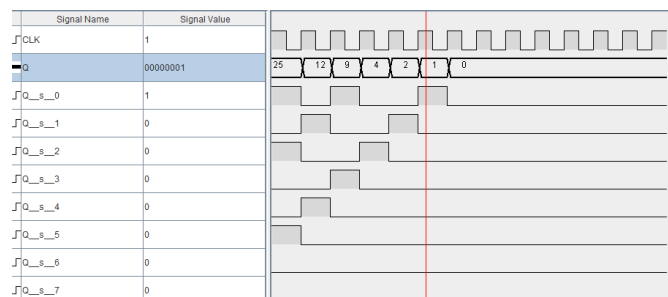


FIGURE 3 – Chronogramme shift register

Le chronogramme ci-dessus capture un shiftright dans le shift register 8 bits. En observant l'évolution de la sortie Q, on peut relever les valeurs suivantes (qui sont, dans le chornogramme, en hexadécimal) à chaque

Tick	Q
1	0b000100101
2	0b000010010
3	0b000001001
4	0b000000100
5	0b000000010
6	0b000000001
7	0b000000000

flanc montant :

3 Générateur de nombres aléatoires

Concevoir un générateur de nombres aléatoires à l'aide d'un registre LFSR (Linear Feedback Shift Register) 8-bits. Dans un registre LFSR $Q_{n+1} = Q_n$, $n > 1$ et Q_1 est la fonction XOR ou XNOR de plusieurs bits du registre. Notez que les bits du registre LFSR sont numérotés de 1 à n. Utilisez la documentation technique de Xilinx pour identifier les bits nécessaires pour calculer le bit Q_1 et simuler le registre LFSR 8-bits à l'aide de Logisim.

LFSR 8 bits - Max 2 pages

Insérez une capture d'écran pour présenter votre bloc LFSR 8 bits (Structure interne). Accompagnez-le de commentaires et d'explications nécessaires à sa compréhension. Insérez une capture d'écran du chronogramme qui comporte les 8 premières valeurs de la séquence (à partir de la valeur 0x00). Accompagnez le chronogramme d'explications. Remplacez le texte ci-dessus par vos réponses (à l'intérieur du cadre rouge)

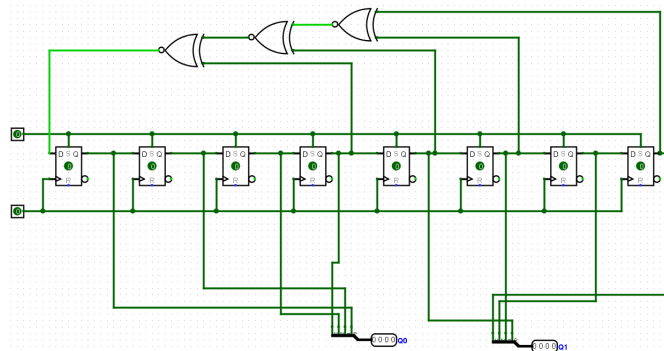


FIGURE 4 - Bloc LSFR 8 bits

Le bloc LSFR permet de générer une suite de nombre pseudo-aléatoire. En entrée, la clock qui va permettre de changer l'état des bascules bistables présentes dans le circuit. À chaque flanc montant, tous les bits vont être décalés d'un cran vers la gauche, et le bit $Q-1$ va être généré à partir de portes XNOR placées à différents Q . L'emplacement des portes XNOR a été déterminé par la table de Wayne Rahnke donnée en annexe pour ce laboratoire. Il y est indiqué que pour 8 bits, on doit placer les portes XNOR au bits 8, 6, 5 et 4. Lorsque toutes les bascules sont à 0, comme on a un premier XNOR qui prend en entrée les bits 8 et 6, qui sont forcément à 0 lorsque les bascules sont initialisées à 0, la sortie sera à 1. Le deuxième XNOR prend en entrée la sortie à 1 du premier XNOR ainsi que le bit 5, sa sortie est donc à 0. Le troisième XNOR prend en entrée la sortie à 0 du deuxième XNOR et le bit 4 qui est à 0. Sa sortie, reliée à la première bascule, est donc à 1. Cela va permettre d'entrer un premier 1, qui va se décaler à chaque tick de l'horloge. C'est grâce à ces portes XNOR qu'on peut ensuite obtenir une suite de bits pseudo-aléatoire.

4 Modification du registre LFSR

Modifier le registre LFSR afin d'initialiser les flip-flops à '1' (utilisez l'entrée preset) et tester le fonctionnement du registre LFSR.

Ajout preset LFSR - Max 1 page

Insérez une capture d'écran pour présenter votre bloc LFSR 8 bits (Structure interne). Accompagnez-le de commentaires et d'explications nécessaires à sa compréhension. Insérez une capture d'écran du chronogramme qui comporte les 8 premières valeurs de la séquence (à partir de la valeur 0x00). Accompagnez le chronogramme d'explications. Remplacez le texte ci-dessus par vos réponses (à l'intérieur du cadre rouge)

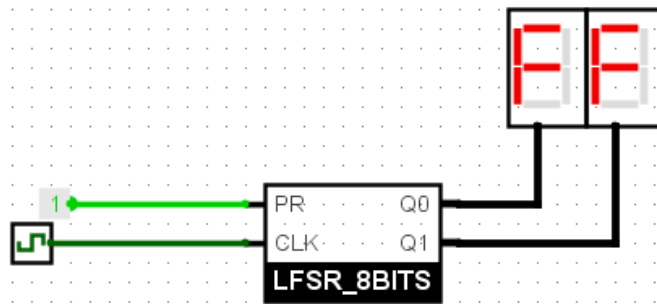


FIGURE 5 – Bloc LFSR avec PRESET à 1

Lorsqu'on ajoute un Preset à '1', on obtient 0xFF, ce qui correspond à une erreur. Ci-dessous, le détail du LFSR 8 bit lorsque le preset est à 1 permet de mieux comprendre ce phénomène :

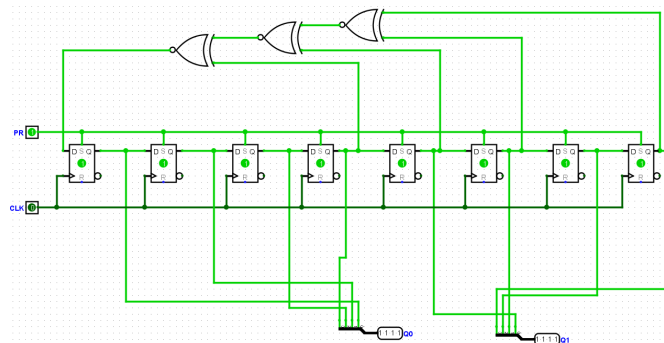


FIGURE 6 – Détail bloc LFSR avec PRESET à 1

Comme on utilise des portes XNOR, si toutes les bascules D sont à 1, toutes les sorties des XNOR seront à 1 aussi.

5 Conclusion

Modifier le registre LFSR afin d'initialiser les flip-flops à '1' (utilisez l'entrée preset) et tester le fonctionnement du registre LFSR.

Conclusion - Max 1/2 page

R emplacez le texte ci-dessus par vos réponses (à l'intérieur du cadre rouge)