**INF1500 – logique des systèmes numériques**

**Laboratoire 4:**

**Simulation et implémentation d’un circuit en logique combinatoire en VHDL**

**Groupe : 02**

**Présenté par :**

**Nathan Ramsay-Vejlens (1989944)**

**Louis Dutheil (1994257)**

**Novembre 2019**

**Département de génie informatique et de génie logiciel**

**École Polytechnique de Montréal**

**Introduction**

Dans le cadre de ce TP on nous demande de créer un circuit composé de différents modules : un multiplicateur par 2, un modulo 5, un multiplexeur, un convertisseur de binaire vers décimal et un driver fourni. La particularité de ce TP est que l’on a du coder ce circuit à partir du langage VHDL.

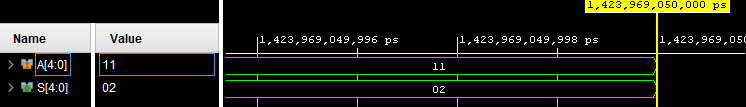
**Modulo 5**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Valeurs décimal | Valeur entrée A | | | | | A%5 | | | | | Valeurs décimale |
| A | A0 | A1 | A2 | A3 | A4 | S0 | S1 | S2 | S3 | S4 | S |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 3 |
| 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| 5 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 7 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 2 |
| 8 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| 9 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 4 |

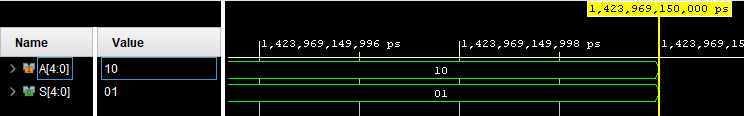
A partir de ce tableau on peut déterminer le code VHDL en s’inspirant de la slide Prime4\_arch du cours. On a donc une liste de cas (5 exactement) selon la valeur entrée A, pour chaque cas une valeur S sort, cette valeur correspond au résultat de A modulo 5.

**Simulation :**

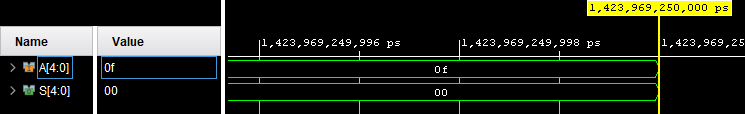
Pour la simulation on a rentré des valeurs A en code héxadécimal



17%5=2



16%5=1



15%5=0

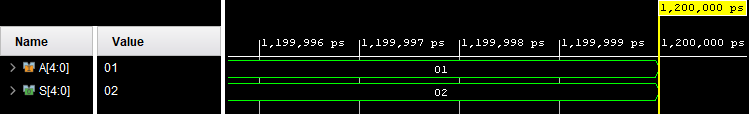
**Multiplicateur par 2**

**Description Générale :**

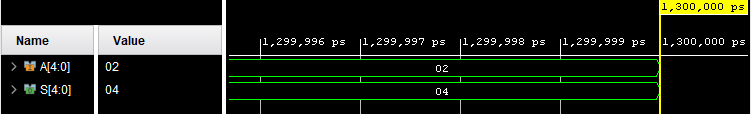
Pour le multiplicateur, on sait que lorsque l’on multiplie une valeur en binaire par 2, on décale en fait chaque bit d’un rang vers la gauche. A partir de ce principe on a alors séparé la vecteur valeur entré en 5 bits A0 A1 A2 A3 A4, A4 a pris la valeur de A3, A3 a pris la valeur de A2 , A2 a pris la valeur de A1, A1 a pris la valeur de A0 et A0 a pris la valeur 0.

**Simulation :**

Pour la simulation on a rentré des valeurs A en code hexadécimal.



1x2=2



2x2=4

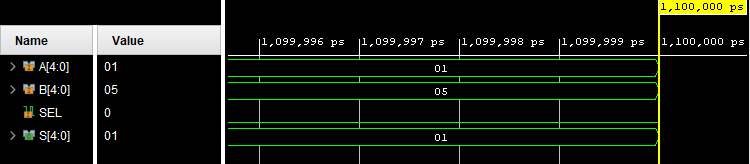
**Multiplexeur**

**Description Générale :**

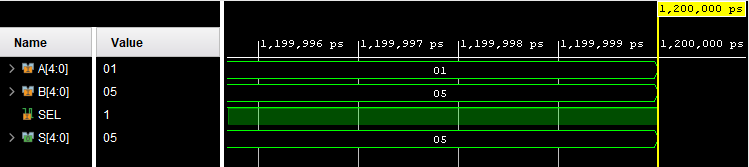
Pour le multiplexeur on a utilisé le même principe que pour le modulo, seulement au lieu de choisir parmi des valeurs en sortie on choisit parmi les variables correspondant au résultats des opérations modulo 5 et multiplication par 2, et on a implémenté une variable de sélection SEL.

**Simulation :**

­­­Pour la simulation on a juste attribué des valeurs arbitraires à A et B et vérifié si ces valeurs sortaient comme on le souhaitait.



SEL=0 donc A sort



SEL=1 donc B sort

**Convertisseur binaire vers décimal**

**Description Générale :**

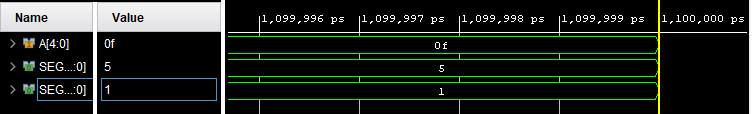
Le module BIN\_ 2\_DEC nous a posé plus de problèmes pour les conversions entre vecteur et décimal.

Pour isoler le chiffre des unités (SEG 0) on a utilisé la fonction mod de VHDL et fait le modulo 10 de la valeur entrée, le résultat de l’opération était ainsi le chiffre des unités.

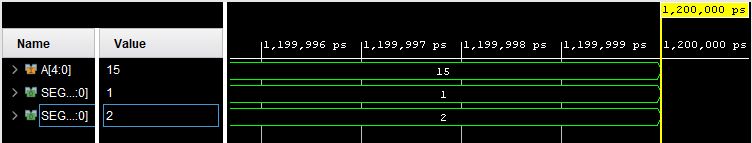
Pour le chiffre des dizaines (SEG 1) on s’est servi du chiffre des unités (SEG 0) précédemment trouvé, on a soustrait SEG 0 à la valeur entrée. On obtenait ainsi un multiple de 10 qu’on a juste divisé par 10, donnant SEG 1.

**Simulation :**

­­­Pour la simulation on a juste attribué des valeurs arbitraires à A et B et vérifié si ces valeurs sortaient comme on le souhaitait.



A=15 donc SEG0=5 et SEG1=1



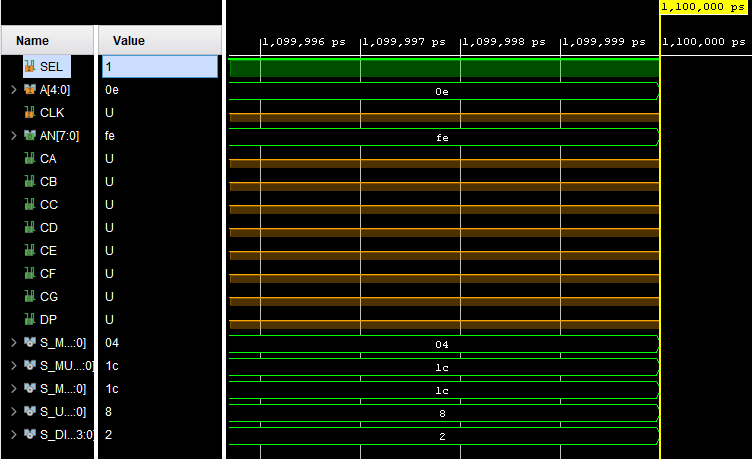
A=21 donc SEG0=1 et SEG1=2

**Design Final**

**Description générale :**

Une fois tous nos modules créés on n’avait plus qu’à relier les ports grâce à des signaux.

**Simulation :**



**Conclusion**

Pour conclure on a donc codé le circuit composé d’un multiplicateur, d’un modulo, d’un multiplexeur, d’un convertisseur binaire vers décimal et d’un driver fourni. Les codes VHDL sont en fichier .docx dans le même dossier.