POLYTECHNIQUE
MONTRÉAL

INF1500
Logique des systèmes numériques

Laboratoire 2

Soumis par :
Boukaftane, Hamza - 2183376
Lidder, Arman - 2174916

22 février 2022

1- Description du système (2 points)

a) Description fonctionnelle

Le circuit à réaliser est une mini-UAL (unité arithmétique et logique) effectuant soit une opération mathématique soit un test de division. Les entrées de ce circuit sont A (4 bits) représentant les valeurs numériques dans un intervalle de 0 à 15 en binaire (décomposé en 4 entrées A0, A1, A2 et A3 de 1bit) et SEL (1bits) dont la valeur est soit 0 soit 1. La sortie du circuit est S (2bits). Quand SEL vaut 0, la sortie S représente la valeur résultante de l'opération mathématique effectuée sur l'entrée A et, quand SEL vaut 1, la sortie S représente la valeur résultante du test de division effectuée sur l'entrée A. Le circuit est composé de 3 modules : Op_math, Div_Test et MUX_2_1_2B. Pour ce qui est de Op_math (opération mathématique), ce module effectue la division entière de A par 2 et applique au résultat un modulo 4 (reste de la division entière). La valeur résultante de cette opération est représentée par la sortie S1 (2 bits). En ce qui concerne Div_Test (test de division), ce module divise l'entrée A par 2 et par 4. Si le reste de la division par 2 est de 0 S2[0] est égale à 1, sinon S2[0] est égale à 0. Dans la même optique, si le reste de la division par 4 est de 0 S2[1] est égale à 1, sinon S2[1] est égale à 0. Ainsi, la combinaison résultante de ce module en fonction de A est la sortie S2 (2bits) du module. Enfin, les sortie S1 et S2 des deux modules précédents converge vers le module MUX_2_1_2B (multiplexeur deux vers 1) et deviennent les entrées du multiplexeur. Ce module est composé de deux sous-modules identiques dont un traite les entrées S1[0] et S2[0], alors que l'autre traite les entrées S1[1] et S2[2]. Ce module permet de sélectionner quelle valeur entre S1 et S2 sera transmise à la sortie S en fonction de la valeur de l'entrée SEL comme expliqué précédemment.

b) Table de vérité du circuit

1- Table de vérité du module Op_Math :

Entrée					Sortie		
A					S1		
Valeur décimale	A3	A2	A1	A0	Valeur décimale	S1[1]	S1[0]
0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0
2	0	0	1	0	1	0	1
3	0	0	1	1	1	0	1
4	0	1	0	0	2	1	0
5	0	1	0	1	2	1	0
6	0	1	1	0	3	1	1
7	0	1	1	1	3	1	1
8	1	0	0	0	0	0	0
9	1	0	0	1	0	0	0
10	1	0	1	0	1	0	1
11	1	0	1	1	1	0	1
12	1	1	0	0	2	1	0
13	1	1	0	1	2	1	0
14	1	1	1	0	3	1	1
15	1	1	1	1	3	1	1

2- Table de vérité du module Div_Test :

Entrée					Sortie		
A					S2		
Valeur décimale	A3	A2	A1	A0	Valeur décimale	S2[1]	S2[0]
0	0	0	0	0	3	1	1
1	0	0	0	1	0	0	0
2	0	0	1	0	1	0	1
3	0	0	1	1	0	0	0
4	0	1	0	0	3	1	1
5	0	1	0	1	0	0	0
6	0	1	1	0	1	0	1
7	0	1	1	1	0	0	0
8	1	0	0	0	3	1	1
9	1	0	0	1	0	0	0
10	1	0	1	0	1	0	1
11	1	0	1	1	0	0	0
12	1	1	0	0	3	1	1
13	1	1	0	1	0	0	0
14	1	1	1	0	1	0	1
15	1	1	1	1	0	0	0

3- Table de vérité du module MUX_2_1_2bit :

Pour S1[0] et S2[0] :

Entrées				Sortie
Valeur décimale	SEL	S2[0]	S1[0]	S[0]
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Pour S1[1] et S2[1] :

Entrées				Sortie
Valeur décimale	SEL	S2[1]	S1[1]	S[1]
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

4- Table de vérité du circuit mini_UAL :

Entrée					Sortie	
SEL	A				S	
	A3	A2	A1	A0	S[1]	S[0]
0	0	0	0	0	0	0
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	0	1	1	0	1
0	0	1	0	0	1	0
0	0	1	0	1	1	0
0	0	1	1	0	1	1
0	0	1	1	1	1	1
0	1	0	0	0	0	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	0	1	1	0	1
0	1	1	0	0	1	0
0	1	1	0	1	1	0
0	1	1	1	0	1	1
0	1	1	1	1	1	1
1	0	0	0	0	1	1
1	0	0	0	1	0	0
1	0	0	1	0	0	1
1	0	0	1	1	0	0
1	0	1	0	0	1	1
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	0	1	1	1	0	0
1	1	0	0	0	1	1
1	1	0	0	1	0	0
1	1	0	1	0	0	1
1	1	0	1	1	0	0
1	1	1	0	0	1	1
1	1	1	0	1	0	0
1	1	1	1	0	0	1
1	1	1	1	1	0	0

c) Tables de Karnaugh des sorties :

1- Table Karnaugh du module Op_Math:

Pour S1[0]:

A1A0\A3A2	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

Pour S1[1]:

A1A0\A3A2	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	0	1	1	0

2- Table Karnaugh du module Div_Test:

Pour S2[0]:

A1A0\A3A2	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

Pour S2[1]:

A1A0\A3A2	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

3- Table Karnaugh du module MUX_2_1_2B:

Pour S[0]:

S1[0]\SELS2[0]	00	01	11	10
0	0	0	1	0
1	1	1	1	0

Pour S[1]:

S1[1]\SELS2[1]	00	01	11	10
0	0	0	1	0
1	1	1	1	0

d) Équations du circuit

1- Les équations du module de l'opération mathématique (Op_math) sont :

$S1[0] = A1$

$S1[1] = A2$

2- Les équations du module du test de division (Div_Test) sont :

$S2[0] = \overline{A0}$

$S2[1] = \overline{A1}.\overline{A0} = \overline{A1+A0}$

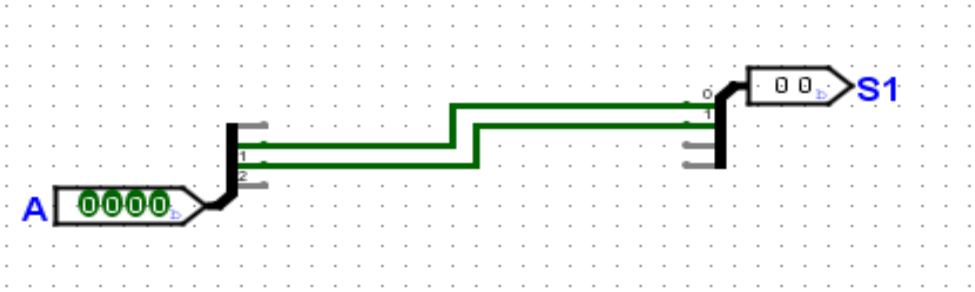
3- Les équations du multiplexeur 2 vers 1 (MUX_2_1_2B) donnant la sortie du circuit complet sont :

$S[0] = \overline{SEL}.S1[0] + SEL.S2[0]$

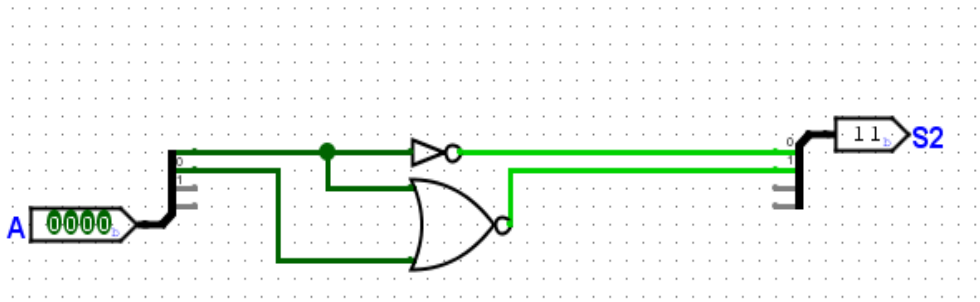
$S[1] = \overline{SEL}.S1[1] + SEL.S2[1]$

e) Schéma du circuit

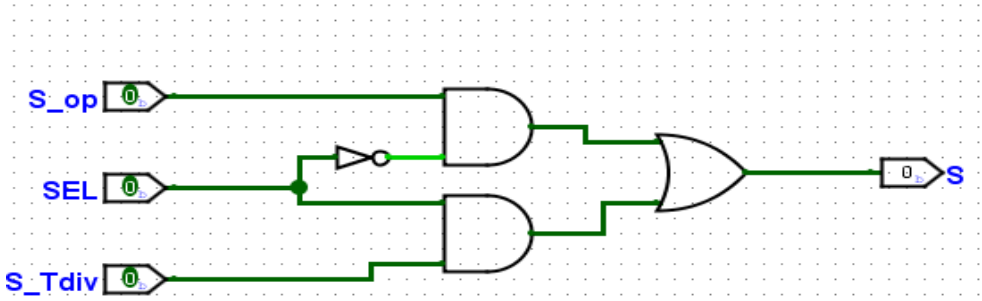
1- Module de l'opération mathématique (Op_Math) :



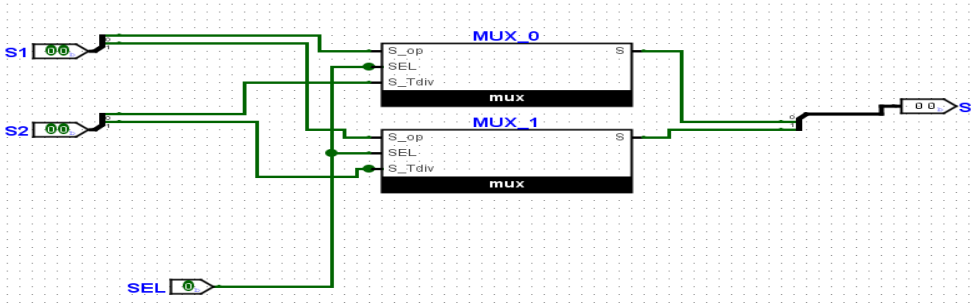
2- Module du test de la division (Div_Test) :



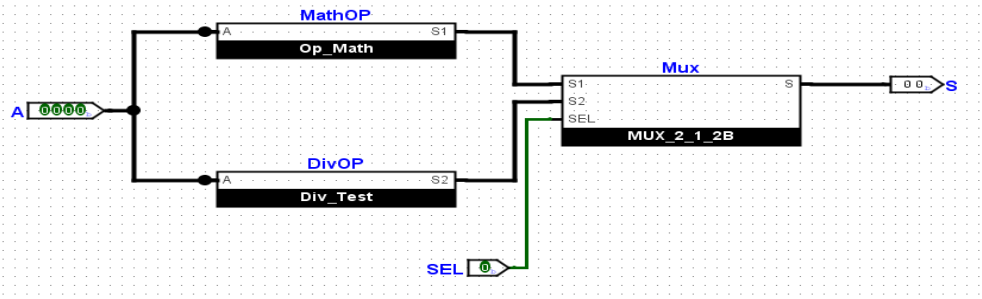
3- Module du multiplexeur 2 vers 1 de 1 bit (MUX) :



4- Module du multiplexeur 2 vers 1 de 2 bit (MUX_2_1_2B) :



5- Schéma du circuit complet incluant tous les modules précédents :



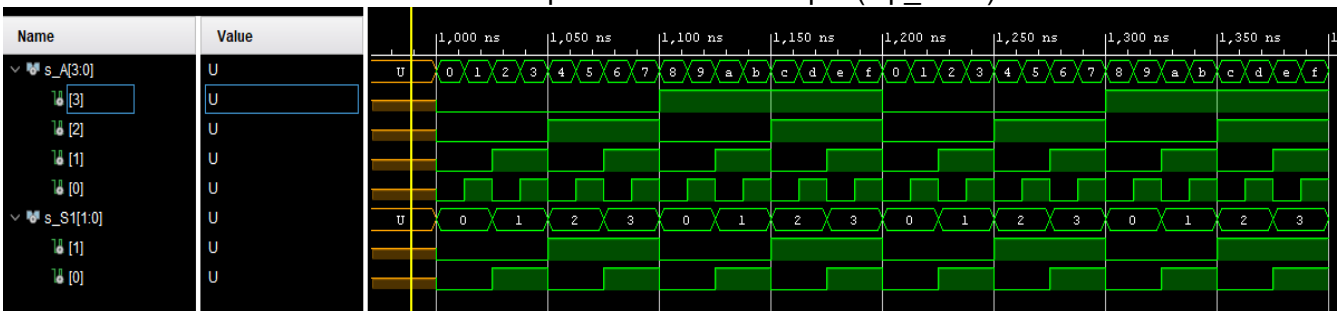
2- Vérification du système

a) Choix de validation

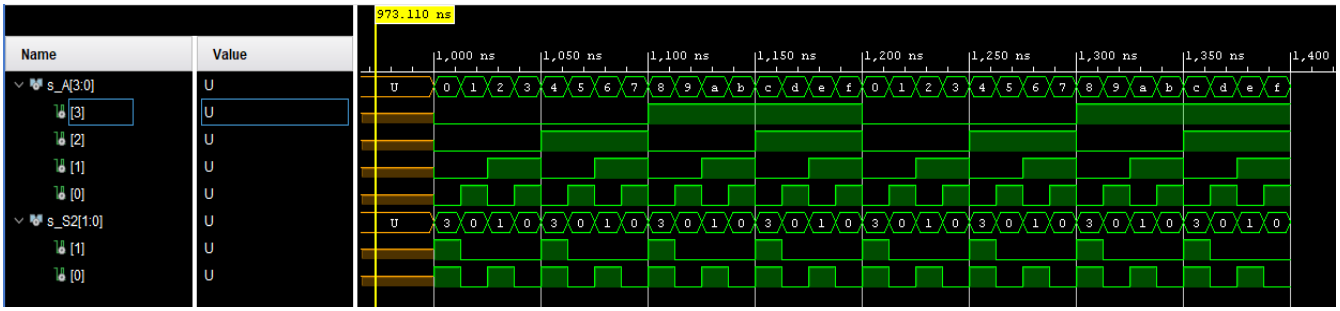
Nous avons utilisé un test exhaustif afin de valider individuellement tous les modules de notre circuit, c'est-à-dire le module de l'opération mathématique, le module du test de division, le module du multiplexeur 2 vers 1 de 1 bit et, enfin, le module du multiplexeur 2 vers 1 de 2 bits. Une fois tous les modules validés, nous avons procédé au test exhaustif du circuit complet. Le test exhaustif permet de tester toutes les combinaisons d'entrées possibles en paramétrant l'intervalle de temps dans lequel une valeur d'entrée change de 0 à 1. En accordant un temps doublement plus grand à la valeur d'entrée suivante, nous pouvons observer les valeurs de sorties correspondant à chaque combinaison d'entrée. Ainsi, nous pouvons observer le comportement du circuit et déterminer sa validité en fonction du comportement attendu.

b) Simulations

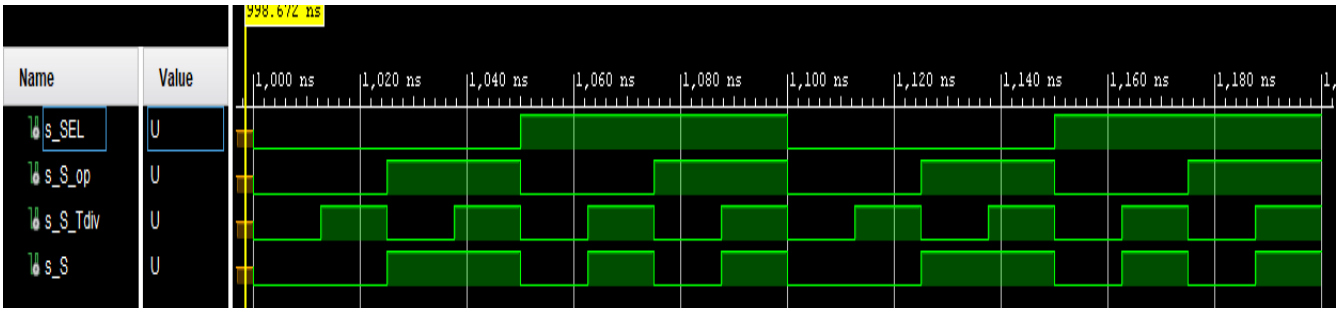
1- Simulation du module de l'opération mathématique (Op_Math) :



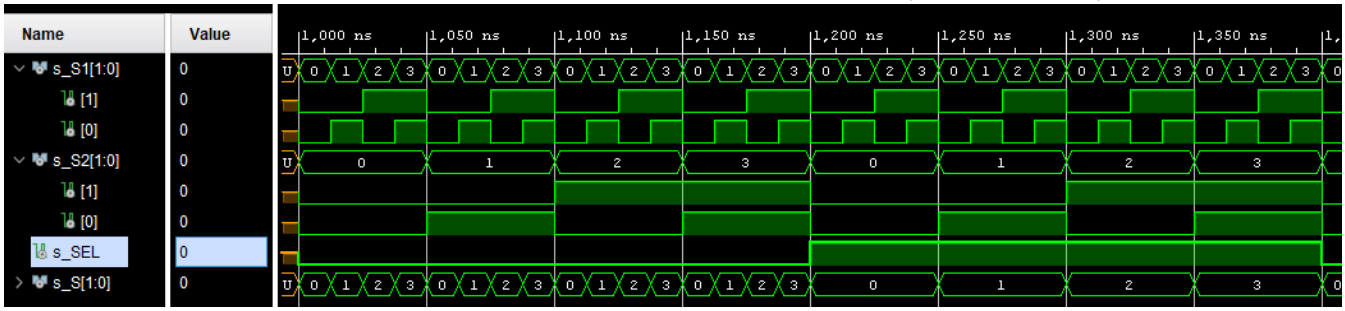
2- Simulation du module du test de la division (Div_Test) :



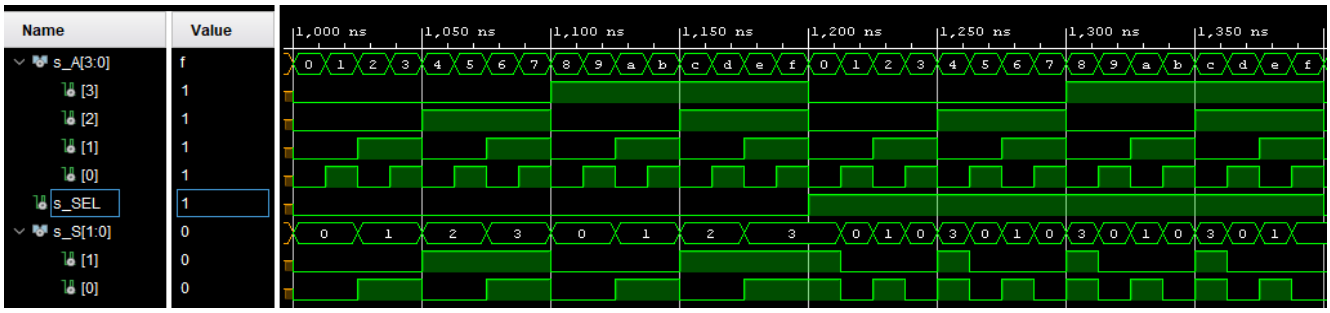
3- Simulation du module du multiplexeur 2 vers 1 de 1 bit (MUX) :



4- Simulation du module du multiplexeur 2 vers 1 de 2 bit (MUX_2_1_2B) :



5- Simulation du circuit complet :



c) Observations et validation

Après avoir effectué les simulations ci-dessus, nous pouvons constater que pour le test exhaustif (concept expliqué dans la section 2.a) du rapport) de chaque module les différentes combinaisons d'entrées et les valeurs des sorties associées à chacune des combinaisons uniques d'entrées correspondent parfaitement aux tables de vérité de chaque module (voir section 1.b). 1 à 4). Ainsi, puisque chaque module de notre sont valides, nous pouvons procéder au test exhaustif du circuit complet. Encore une fois, dans la simulation, les valeurs de sorties spécifiques à chaque combinaison d'entrée possible correspondent parfaitement à la table de vérité du circuit complet (voir section 1.b).5 du rapport). Dans cette optique, nous pouvons affirmer que notre circuit est valide, car la simulation de ce dernier respecte les équations définissant le circuit et concorde parfaitement avec la table de vérité du circuit.

3- Réponses aux questions

Question 1 : Après avoir fait la table de vérité du Test de divisibilité que remarquez-vous ?

En ce qui à trait la division par 2, nous remarquons que peu importe la valeur de A, lorsque le bit le moins significatif A0 vaut 0, alors la valeur de A est divisible par 2 et S2[0] vaut 1. Dans cette même optique, quand le bit le moins significatif A0 vaut 1, alors la valeur de A n'est pas divisible par 2 et S2[0] vaut 0. Ainsi, S2[0] correspond à la valeur binaire inverse de A0. Pour ce qui est de la division par 4, nous remarquons que lorsque A1 et A0 sont tous deux égales à 0, alors la valeur de A est divisible par 4 et S2[1] vaut 1. De ce fait, quand A1 et A0 ne valent pas tous les deux 0, alors la valeur de A n'est pas divisible par 4 et S2[1] vaut 0.

Question 2 : A quelle opération sur les bits vous fait penser la division par 4 ou par 2 ?

Pour diviser un nombre binaire par 2, il faut y enlever le dernier bit. Par exemple, soit le nombre 24 en binaire : 11000. Si on le divise par deux, on enlève le dernier bit et on aura :1100, dont la valeur décimale est de 12. Pour diviser un nombre binaire par 4, il suffit d'enlever les 2 derniers bits et pour le même exemple, on aura 110, dont la valeur décimale est de 6. Cette opération sur les bits, plus précisément, donnera le résultat de la division entière. Par exemple, pour la valeur 17, on a : 10001 et si on le divise par 4, on a: 1000, dont la valeur décimale est 4, ce qui correspond au résultat de la division entière de 17/4. Le reste de la division est la valeur décimale des bits ou du bit enlevé(s).