TD - Architecture des Circuits Numérique

Victor Lezaud

19 novembre 2017

Sommaire

Séance de Travaux Dirigés I

Représentation des nombres et arithmétique entière

1 Entiers représentables et ordres de grandeur

1.1 Valeur maximale codable

Quels sont les plus grands nombres codables en base 2 sur les tailles suivantes. On demande une valeur exprimée en décimal.

— 1 bit : 1
— 4 bits : 15
— 8 bits : 255
— 1 byte : 255
— 10 bits : 1048 $\sim 10^3$
— 2 bytes : 65 536
— 4 bytes : 4×10^9
— 8 bytes : 16×10^{18}

1.2 Ordres de grandeur

Pour chacune des paires suivantes quel est le plus grand nombre des deux
— 10^{33} et 2^{80} : $2^{80} \sim 10^{24}$ donc $10^{33} > 2^{80}$
— 10^{10} et 2^{35} : $2^{35} \sim 32 \times 10^9$ donc $2^{35} > 10^{10}$

2 Nombres entiers relatifs, codage en complément à 2

2.1 Conversion sur 3 bits

Écrire la table des correspondances binaire $\leftrightarrow$ décimal pour tous les nombres entiers signés codés sur trois bits en complément à deux

Binaire	Décimal
000	0
001	1
010	2
011	3
100	-4
101	-3
110	-2
111	-1

2.2 Signe et valeur en complément à 2

Comment pouvez-vous savoir si l'entier, codé en complément-à-deux sur 16 bits, 1010 1110 1010 1111 est positif ou négatif ? Quelle est sa valeur ? Donnez

le code binaire de son opposé.
Le bit de poids fort est à 1, ce nombre est donc négatif. Le code binaire de son opposé est 0101 0001 0101 0001. Sa valeur est donc $-1-16-64-4096-16484 = -20061$.

2.3 Nombre négatif en complément à 2

Décimal	Binaire
-11	1111 0101
-22	1110 1010
-44	1101 0100
-47	1101 0001
-125	1111 1101

2.4 Conversion binaire → décimal

Binaire	Décimal
1111 1111	-1
0100 1111	79
1001 1111	-97
1111 1110	-2

2.5 Conversion en décimal

En utilisant la technique décrite dans le poly, convertissez $(11011)_2$ puis $(56)_9$ en décimal
$(11011)_2 = 1 + 2 + 8 + 16 = 27$
$(56)_9 = 6 + 5 \times 9 = 52$

2.6 Conversion à partir de la base décimal

Utilisez la méthode à base de divisions euclidiennes décrite dans le poly pour convertir $(414)_{10}$ en binaire, puis $(3452)_{10}$ en base 8

$414 = 2 \times 207 + 0$
$207 = 2 \times 103 + 1$
$103 = 2 \times 51 + 1$
$51 = 2 \times 25 + 1$
$25 = 2 \times 12 + 1$
$12 = 2 \times 6 + 0$
$6 = 2 \times 3 + 0$
$3 = 2 \times 1 + 1$
$1 = 2 \times 0 + 1$
$(414)_{10} = (000110011110)_2$

$3452 = 8 \times 431 + 4$
$431 = 8 \times 53 + 7$
$53 = 8 \times 6 + 5$
$6 = 8 \times 0 + 6$
$(3452)_{10} = (6574)_8$

2.7 Conversion en base 2,8,16

Entre les bases 2, 8 et 16, des méthodes plus directes peuvent être utilisées : par exemple, tout chiffre octal est représenté par un entier sur trois bits, et tout entier sur trois bits est représenté par un chiffre octal. Justifiez cette méthode

de conversion entre les bases 2 et 8. Convertissez $(34521)_8$ en base 2 puis 16.

Réponse : Si on prend un nombre en binaire et qu'on regroupe par groupe de trois on obtient des chiffres en base 8, de plus le n^{me} chiffre est multiplié par $(2^3)^{n-1}$ soit 8^{n-1}. Avec le tableau ci-dessous on trouve donc aisément $(34521)_8 = (011100101010001)_2$ et avec une méthode équivalente $(0011100101010001)_2 = (3951)_{16}$ donc $(34521)_8 = (3951)_{16}$.

Binaire	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

3 Opérations en complément à deux

3.1 Addition et calcul de l'opposé

3.1.1 Addition sur 8 bits

	$(10001010)_{\bar{2}}$			$(10001010)_{\bar{2}}$			$(01001010)_{\bar{2}}$	
$+$	$(00001011)_{\bar{2}}$		$+$	$(10001011)_{\bar{2}}$		$+$	$(11001010)_{\bar{2}}$	
$=$	$(10010101)_{\bar{2}}$		$=$	$(00010101)_{\bar{2}}$		$=$	$(00010100)_{\bar{2}}$	

3.1.2 Calcul de l'opposé et vérification

$-(10001010)_{\bar{2}} = (01110101)_{\bar{2}} + (00000001)_{\bar{2}}$
$-(10001010)_{\bar{2}} = (01110110)_{\bar{2}}$
$(10001010)_{\bar{2}} + (01110110)_{\bar{2}} = (00000000)_{\bar{2}}$

3.1.3 Dépassement de capacité

En complément à 2 sur p bits, quel est le seul cas cas produisant un dépassement de capacité ? Quel que soit p seul 0 donne un dépassement de capacité. Le calcul de -0 nous donne la somme de 1 et du nombre codé par p 1. Le résultat est donc 1 suivi de p 0. En gardant que les p derniers bits on retrouve bien $-0 = 0$.

3.1.4 Soustraction

Calculer en binaire la soustraction $1101_2 - 0110_2$.
$1101_{\bar{2}} - 0110_{\bar{2}} = 1101_{\bar{2}} + 1001_{\bar{2}} + 0001_{\bar{2}} = 10111_{\bar{2}}$
$\Leftrightarrow -3_{10} - 6_{10} = -9_{10}$

3.2 Conversion en complément à 2

3.2.1 Conversion

Comment sont représenté $(34)_{10}$ et $(-42)_{10}$ en complément à 2 sur 8 bits ?
$(34)_{10} = (0010\,0010)_{\bar{2}}$
$(-42)_{10} = -(0010\,1010)_{\bar{2}} = (1101\,0110)_{\bar{2}}$

3.2.2 Extension de signe

Comment sont représenté $(34)_{10}$ et $(-42)_{10}$ en complément à 2 sur 8 bits ?
$(34)_{10} = (0000\,0010\,0010)_{\bar{2}}$
$(-42)_{10} = (1111\,1101\,0110)_{\bar{2}}$

3.3 Multiplication

3.3.1 Calcul du produit

Effectuer en binaire les opérations suivantes :
$1111\,1111_2 \times 1_2 = 1111\,1111_2$
$1111\,1111_2 \times 10_2 = 1\,1111\,1110_2$
$1111\,1111_2 \times 100_2 = 11\,1111\,1100_2$
$1101\,0011_2 \times 1001_2 = 110\,1001\,1000_2 + 1101\,0011_2 = 111\,0110\,1011_2$
Dans une multiplication on a : $taille_{rsultat} = taille_{oprande1} + taille_{oprande2} - 1$

4 Nombre binaire fractionnaire

4.1 Conversion binaire->décimale

Quelle est la valeur décimale du nombre binaire positif 11,001001 ?
$11,001001_2 = 2 + 1 + \frac{1}{4} + \frac{1}{32}$
$11,001001_2 = 3 + \frac{9}{32}$

4.2 Écriture finie en binaire

Parmi les nombres fractionnaires suivants, lesquels ont une écriture finie en binaire ?

- $0,1 \Rightarrow$ NON
- $0,2 \Rightarrow$ NON
- $0,3 \Rightarrow$ NON
- $0,4 \Rightarrow$ NON
- $0,5 \Rightarrow 0,1_2$

Séance de Travaux Dirigés II

Calcul booléen

5 Calcul booléen

5.1 Règle de De Morgan

Rappeler la règle de De Morgan sous ses deux formes.
$\overline{a+b} = \bar{a}\bar{b}$
$\overline{ab} = \bar{a} + \bar{b}$

5.2 Equivalences

Prouver les équivalences suivantes :

5.2.1 $ab + \bar{a}b = b$

$ab + \bar{a}b = b(a + \bar{a})$ Par distributivité
$ab + \bar{a}b = b$ Par tautologie

5.2.2 $a + ab = a$

$a + ab = a(1 + b)$ Par distributivité
$a + ab = a$ Par absorption

5.2.3 $a + ab = a$

$a + ab = a(1 + b)$ Par distributivité
$a + ab = a$ Par absorption

5.2.4 $a + \bar{a}b = a + b$

a	b	$a + \bar{a}b$	$a + b$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

Selon la table de vérité ci dessus les expressions sont bien équivalentes

5.2.5 $ab + \bar{a}c = ab + \bar{a}c + bc$

a	b	c	$ab + \bar{a}c$	$ab + \bar{a}c + bc$
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

Selon la table de vérité ci dessus les expressions sont bien équivalentes

5.3 Simplification

Simplifier les expressions suivantes

5.3.1 $s_1 = \overline{\overline{(x+y)}(x+z)}(y+z)$

$s_1 = ((x+y) + \overline{(x+z)})(y+z)$
$s_1 = ((x+y) + \bar{x}\bar{z})(y+z)$
$s_1 = xy + yz + \bar{x}\bar{z}y + \bar{x}\bar{z}z$
$s_1 = y(x + z + \bar{x}\bar{z})$
$s_1 = y((x+z) + \overline{(x+z)}$
$s_1 = y$

5.3.2 $s_2 = \overline{\overline{x\bar{y}} + yz} + \overline{(x+z)\overline{(y+\bar{x}z)}}$

$s_2 = ((xy)\overline{(yz)}) + \overline{(x+z)} + (y + \bar{x}z)$
$s_2 = (xy(\bar{y}+\bar{z})) + \bar{x}\bar{z} + y + \bar{x}z$
$s_2 = xy\bar{y} + xy\bar{z} + y + \bar{x}(\bar{z}+z)$
$s_2 = y + \bar{x}$

5.3.3 $s_3 = \overline{y(\bar{x}+z) + \overline{x(\bar{y}+z)}} + \overline{(y+z)}(xy + \overline{x(\bar{y}+\bar{z})})$

$s_3 = \overline{(y(\bar{x}+z))}(x(\bar{y}+z)) + \bar{y}\bar{z}(xy + \bar{x} + \overline{(\bar{y}+\bar{z})})$
$s_3 = (\bar{y} + \overline{(\bar{x}+z)})(x(\bar{y}+z)) + \bar{y}\bar{z}(xy + \bar{x} + yz)$
$s_3 = (\bar{y} + x\bar{z})(x\bar{y} + xz) + \bar{y}\bar{z}\bar{x}$
$s_3 = x\bar{y} + \bar{y}\bar{z}\bar{x}$

6 Expression algébrique

6.1 Depuis un texte

Donner une expression booléenne de la fonction $f(x, y, z)$ qui vait 1 si et seulement si la majorité de ses trois arguments vaut 1.
$f(x, y, z) = xy + xz + yz$

6.2 Depuis une table

Donner (sans la simplifier) une expression booléenne de la fonction $g(a, b, c)$ définie par la table de vérité suivante :

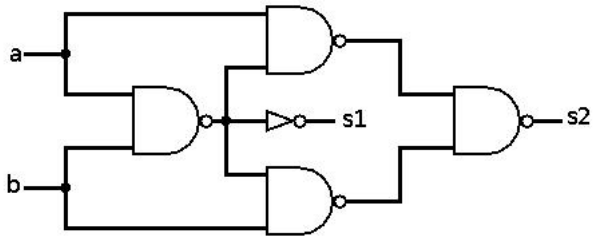
a	b	c	s
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

$g(a,b,c) = \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}bc + a\bar{b}\bar{c} + ab\bar{c} + abc$

7 Circuits logiques

7.1 Circuit logique vers algèbre

Donner une expression algébrique des sorties s_1 et s_2. Etablir la table de vérité. La fonction calculée par ce circuit vous est-elle familière ?

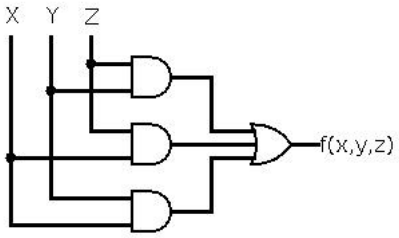


$s_1 = ab$
$s_2 = \overline{\overline{(a\overline{(ab)})} \times \overline{(b\overline{(ab)})}}$
$s_2 = a\overline{(ab)} + b\overline{(ab)}$
$s_2 = a(\bar{a} + \bar{b}) + b(\bar{a} + \bar{b})$
$s_2 = a\bar{a} + a\bar{b} + b\bar{a} + b\bar{b}$
$s_2 = a\bar{b} + b\bar{a}$
s_1 est la fonction ET. s_2 est la fonction XOR.

7.2 Algèbre vers circuit logique

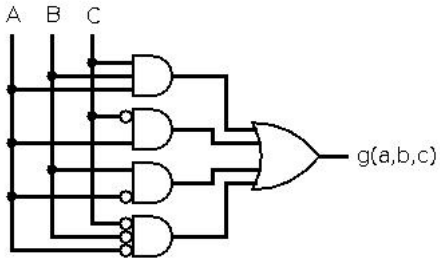
Dessiner un circuit logique pour chacune des fonctions f et g de la partie précédente en utilisant unique des portes logiques (ET, OU, NON).

7.2.1 $f(x, y, z) = xy + xz + yz$



7.2.2 $g(a, b, c) = \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}bc + a\bar{b}\bar{c} + ab\bar{c} + abc$

$g(a, b, c) = \bar{a}\bar{b}\bar{c} + \bar{a}b + a\bar{c} + abc$



8 Multiplexeur et démultiplexeur

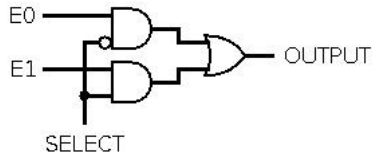
8.1 Multiplexeur 2 vers 1

8.1.1 Table de vérité

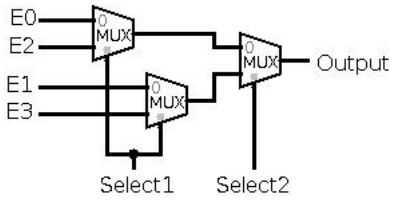
e_0	e_1	select	output
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

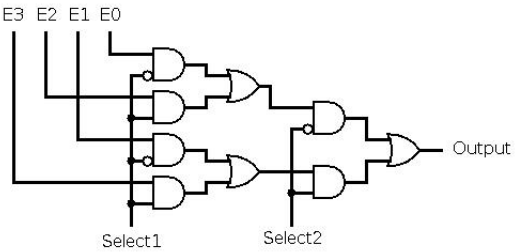
8.1.2 Expression booléenne et circuit logique

$o = e_0\bar{s} + e_1 s$

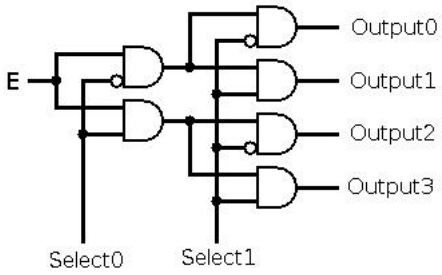


8.2 Multiplexeur 4 vers 1





8.3 Démultiplexeur 1 vers 4



Séance de Travaux Dirigés III

Circuits combinatoires

9 Additionneur 1 bit

9.1 Principe

Un additionneur 1 bit est un circuit combinatoire à 3 entrées et deux sorties qui permet de réaliser l'addition de deux opérandes à 1 bit avec éventuellement une retenue en entrée sur 1 bit :

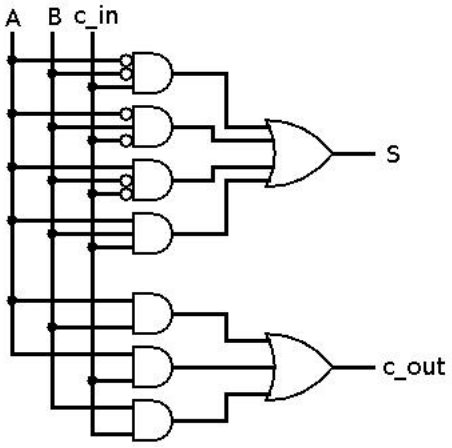
- Les entrées :
  - a : entrée à 1 bit (opérande 1 de l'addition)
  - b : entée à 1 bit (opérande 2 de l'addition)
  - c_{in} : retenue d'entrée de l'addition
- Les sorties :
  - s : sortie à 1 bit résultat de l'addition
  - c_{out} : retenue de sortie de l'addition

9.2 table de vérité

a	b	c_{in}	s	c_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$s = \bar{a}\bar{b}c_{in} + \bar{a}b\bar{c_{in}} + a\bar{b}\bar{c_{in}} + abc_{in}$ $\qquad$ $c_{out} = bc_{out} + ac_{out} + ab$

9.3 Circuit combinatoire



10 Additionneur 8 bits

10.1 Principe

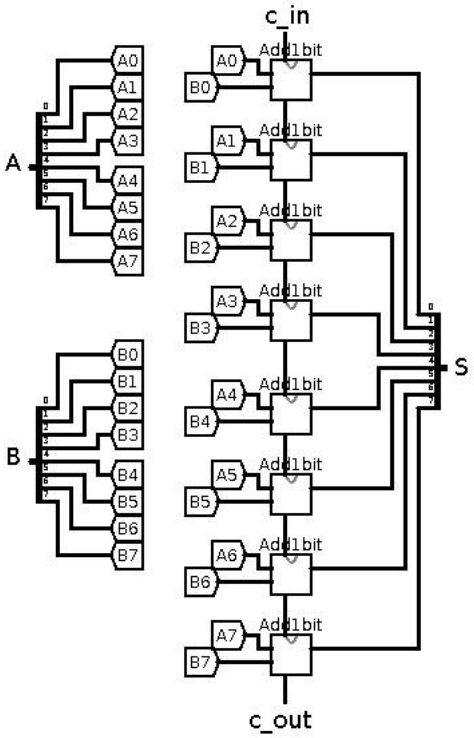
Un additionneur 8 bits est un circuit combinatoire à 3 entrées et deux sorties qui permet de réaliser l'addition de deux opérandes à 8 bits avec éventuellement une retenue en entrée sur 1 bit :

- Les entrées :
  - a : entrée à 8 bit (opérande 1 de l'addition)
  - b : entée à 8 bit (opérande 2 de l'addition)
  - c_{in} : retenue d'entrée de l'addition sur 1 bit
- Les sorties :
  - s : sortie à 8 bits, résultat de l'addition
  - c_{out} : retenue de sortie de l'addition sur 1 bit

10.2 Réutilisation

L'additionneur 8 bits revient à additionner chaque bit séparément avec le transfert des retenus d'un bit au suivant. On utilise donc l'additionneur 1 bit déjà réalisé pour créer celui-ci.

10.2.1 Circuit combinatoire



11 Additionneur/Soustracteur 8 bits

11.1 Principe

La soustraction est en réalité extrêmement proche de l'addition. Elle revient à additionner l'opposé du nombre que l'on souhaite soustraire. Or en complément à 2 l'opposé est trouvé en prenant le complément du nombre (inverser tous les bits) et en ajoutant 1. On fait donc la différence entre un ordre d'addition et de soustraction par la valeur de la retenue d'entrée($c_{in} = 0$ pour l'addition et $c_{in} = 1$ pour la soustraction). Ensuite il suffit de prendre le complément de la 2^{me} opérande pour la soustraction et on réutilise l'additionneur 8 bits.

11.2 Inverseur

11.2.1 Principe

L'inverseur est la partie du circuit qui doit inverser les bits de la deuxième opérande si l'on fait une soustraction et ne rien faire pour une addition

- Les entrées :
  - b : entée à 8 bits (opérande 2 de l'addition/soustraction)
  - c_{in} : retenue d'entrée de l'addition/soustraction
- Les sorties :
  - b' : sortie à 8 bit, opérande prête pour l'additionneur

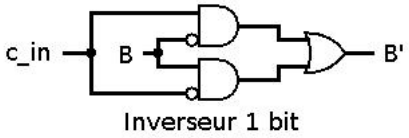
11.2.2 table de vérité

Chaque bit est indépendant donc on en représente qu'un seul.

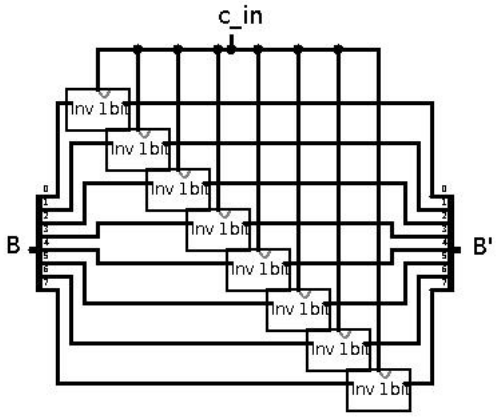
b	c_{in}	b'
0	0	0
0	1	1
1	0	1
0	0	1

$$b' = \bar{b}c_{in} + b\overline{c_{in}}$$

11.2.3 Circuit combinatoire

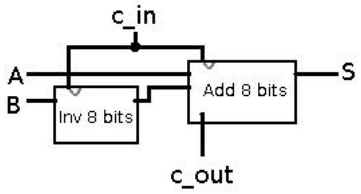


Inverseur 1 bit



Inverseur 8 bits

11.3 Implémentation Add/Sous 8 bits



Séance de Travaux Dirigés IV

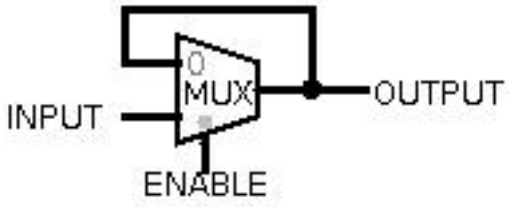
Registres et mémoires

12 Registres

12.1 Latch et FlipFlop

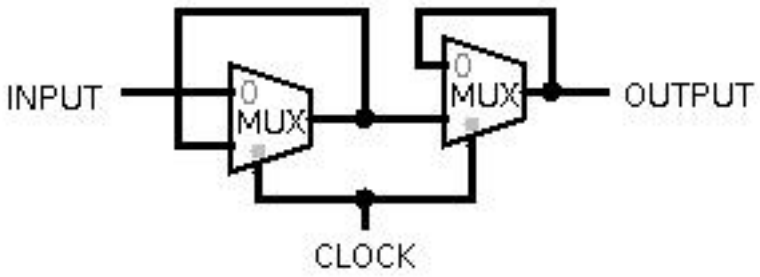
12.1.1 Lactch ou verrou

Le verrou permet de stocker une valeur quand l'entrée enable est à 0 et modifier cette velaur stockée quand enable est à 1. C'est un circuit séquentiel crée comme ceci :



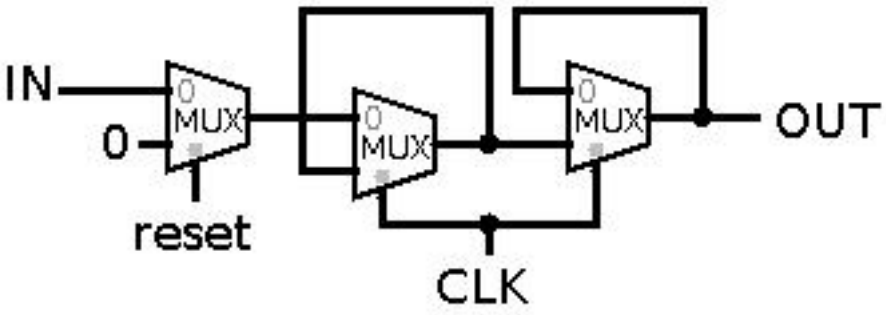
12.1.2 Flip-Flop

Le flip-flop permet de réaliser une bascule synchrone, c'est à dire que la sortie a toujours la valeur qu'avait l'entrée au dernier top d'horloge. Il est réalisé comme ci-dessous :



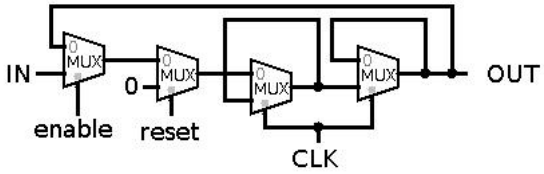
12.2 Flip-Flop avec reset

On complète le flip-flop avec un reset qui permet de remettre la sortie à 0 quelque soit l'entrée. Nous allons un réaliser un reset synchrone, cela signifie que la remise à 0 ne sera effective qu'à un top d'horloge. On doit donc avoir une entrée reset qui, quand elle est à 1, force l'entrée à 0.



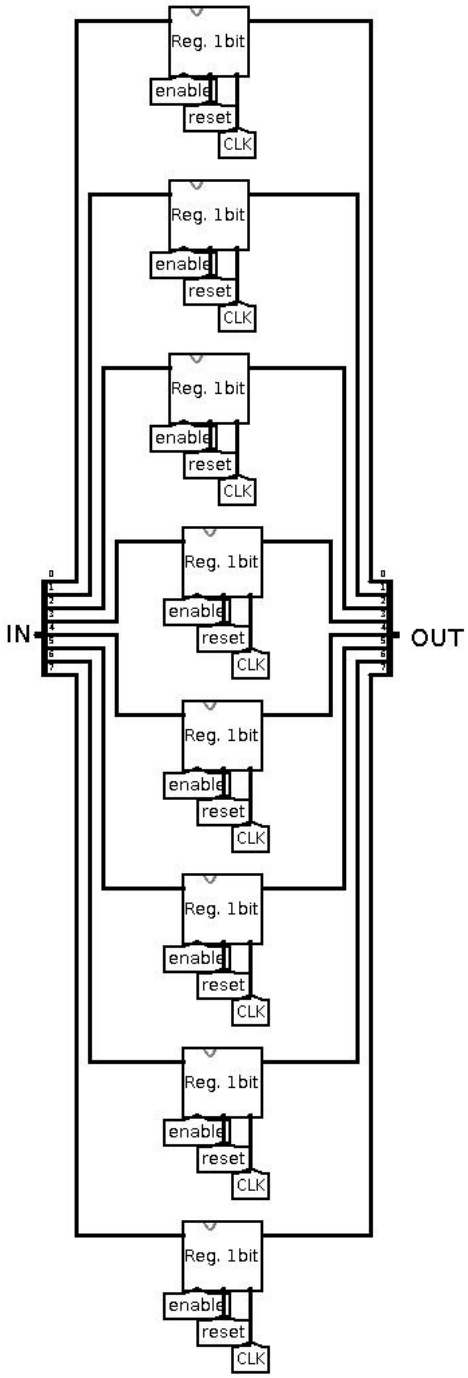
12.3 Registre à commande de chargement

La commande de chargement (communément appelé enable) permet de choisir quand on modifie ou non la valeur stockée. Pour ajouter cette commande au circuit précédent il suffit de "l'entourer" par le verrou vu juste au dessus :



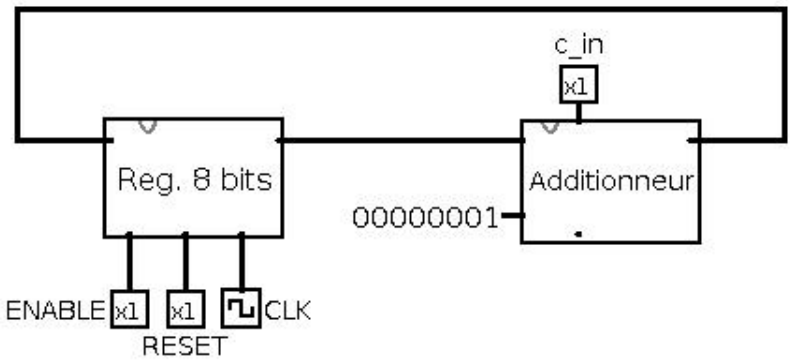
12.4 Registre à n bits

Nous avons réussi à créer un registre complet à 1 bit. Pour la suite nous allons vouloir stocké beaucoup plus de valeur. Mais le travail est quasiment déjà fait puisqu'il nous suffit de mettre côte à côte autant de registre que l'on veut stocker de bits. Par exemple un registre 8 bits est fait ainsi :



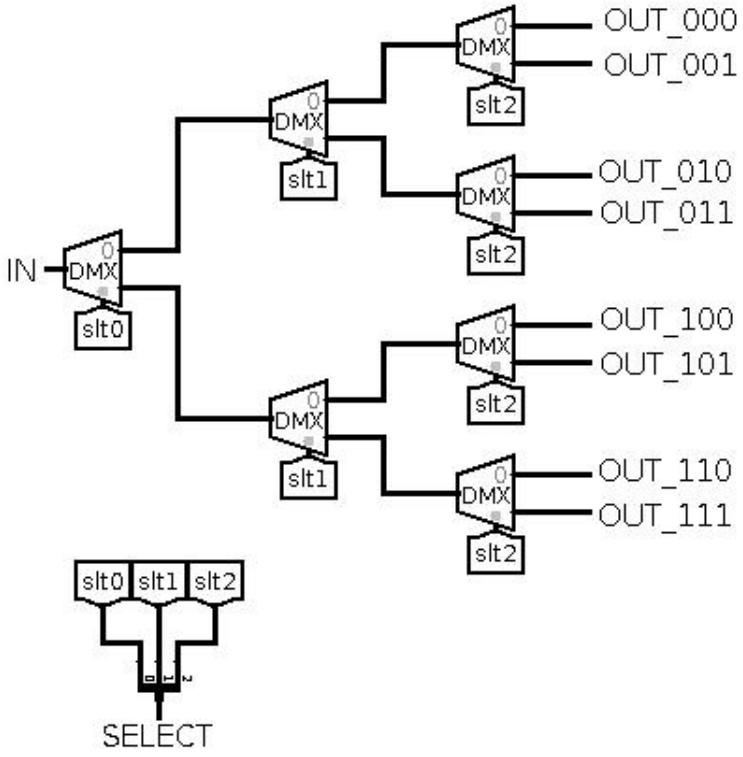
12.5 Un petit compteur

Nous allons réaliser un circuit qui va compter les tops d'horloge dans un registre 8 bits. Le compteur peut aller de 0 à $2^8 - 1$

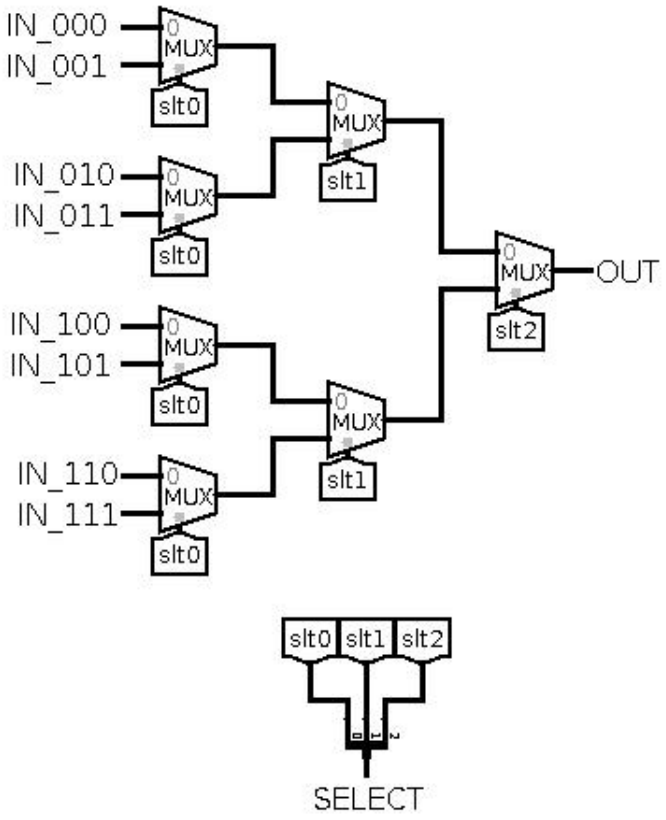


13 Mémoires adressables

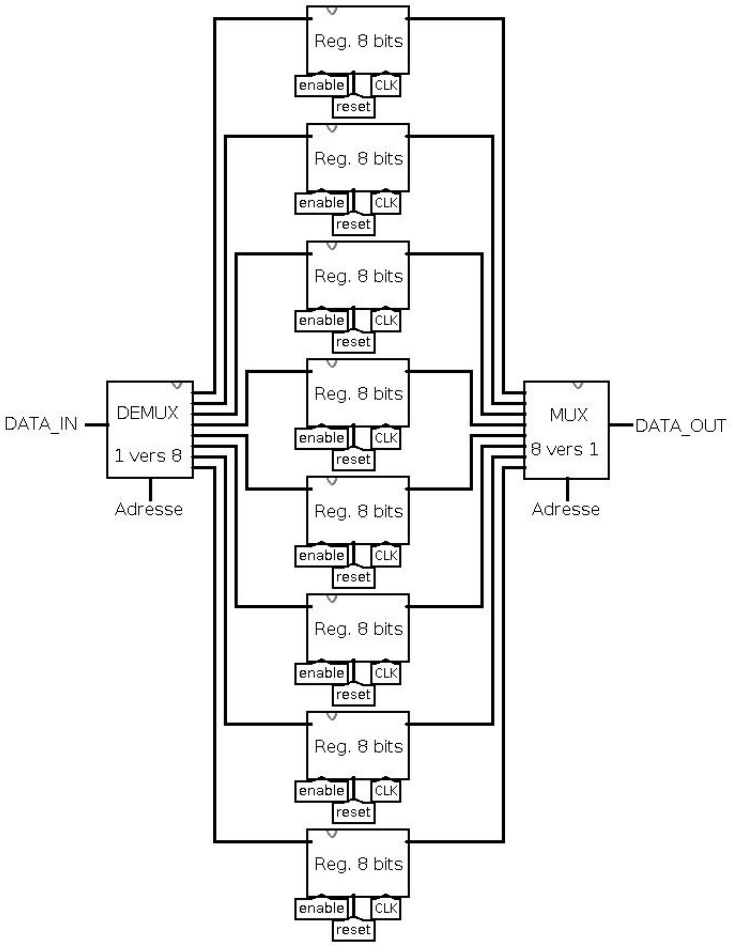
13.1 Démultiplexeur de 1 vers 8



13.2 Multiplexeur de 8 vers 1



13.3 Mémoire 8 mots de 8 bits



Séance de Travaux Dirigés V

Automates