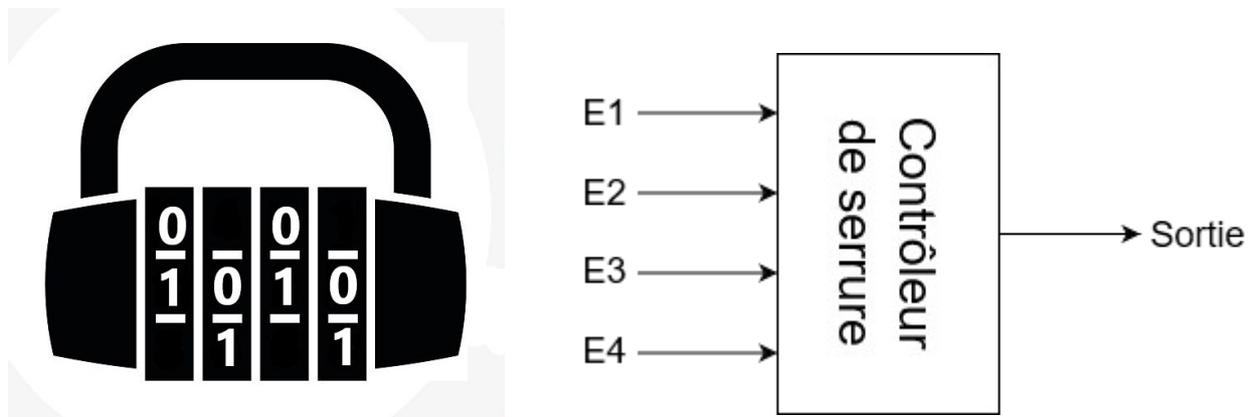


Série 1 (Circuits Combinatoires)

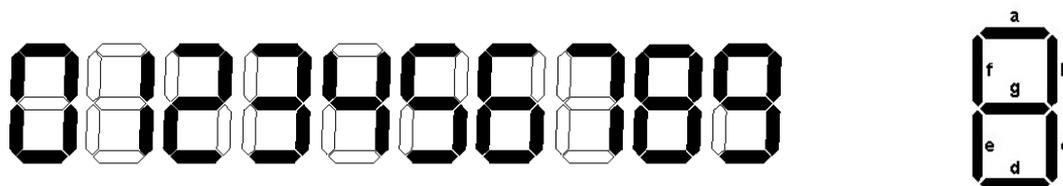
Exercice 01 :

Construire en utilisant la méthode à 5 étapes le circuit combinatoire qui gère l'ouverture et la fermeture d'une serrure (cadenas) électronique. Le clavier de la serrure dispose de 4 entrées binaires, comme sur la figure. La serrure s'ouvre mécaniquement si le circuit fait sortir la valeur 1, sinon ça reste fermée. Le code d'ouverture est 1101.



Exercice 02 :

Le circuit combinatoire d'un afficheur 7 segments est un circuit qui commande un afficheur 7 segments, il permet l'affichage d'un seul chiffre sur 7 segments comme illustré sur le schéma, il reçoit en entrée un nombre encodé sur 4 bits appartenant à l'intervalle [0,9] et en sortie il produit la combinaison des segments qui affiche le chiffre dans le système décimal.



Construire en utilisant la méthode à 5 étapes le circuit combinatoire qui gère l'affichage sur l'afficheur 7 segments, en choisissant pour chaque segment la méthode la plus réductrice possible, entre sorties indéfinies et Forme canonique disjonctive ou forme canonique conjonctive.

Exercice 03 :

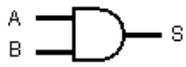
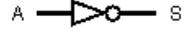
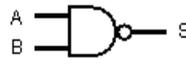
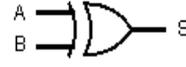
Implémentez les circuits numériques suivant en appliquant la méthode à 5 étapes :

Circuits	Caractéristiques
Multiplexeur*	2 entrées, 1 sélecteur et 1 sortie ($n=1$)
Démultiplexeur	1 entrée, 2 sélecteurs et 4 sorties ($n=2$)
Encodeur	4 entrées et 2 sorties ($n=2$)
Décodeur*	3 entrées et 8 sorties ($n=3$)
Contrôleur de priorité	4 entrées et 4 sorties ($n=4$)
Contrôleur de parité*	4 bits en entrée et 1 sortie
Additionneur*	2 entiers non signés sur 2 bits en entrée et 1 entier non signés sur 2 bits en sortie, plus la sortie de la retenue
Complément-à-1	1 entier sur 3 bits en entrée et 1 entier sur 3 bits en sortie
Shifteur* (Décaleur)	Fait le décalage à gauche, avec un entier non signé sur 2 bits en entrée plus un montant de décalage sur 2 bits. Et un entier non signé sur 2 bits en sortie
Comparateur*	2 entiers non signés sur 2 bits en entrée et 3 sorties pour le supérieur, l'inférieur et l'égale
Étendreur de signe	1 entier complément-à-2 sur 4 bits en entrée et 1 entier complément-à-2 sur 8 bits en sortie

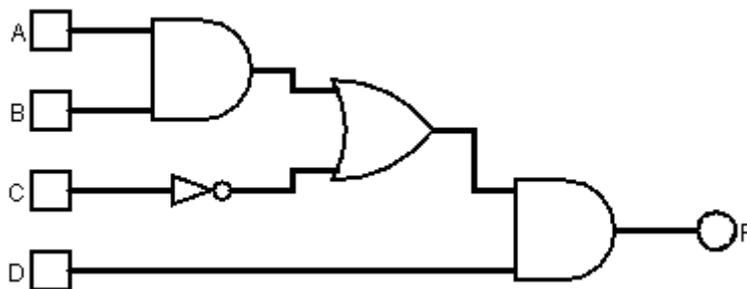
(*) Ces circuits sont obligatoire à faire en TD, par contre le reste c'est facultatif selon l'état d'avancement.

Exercice 04 :

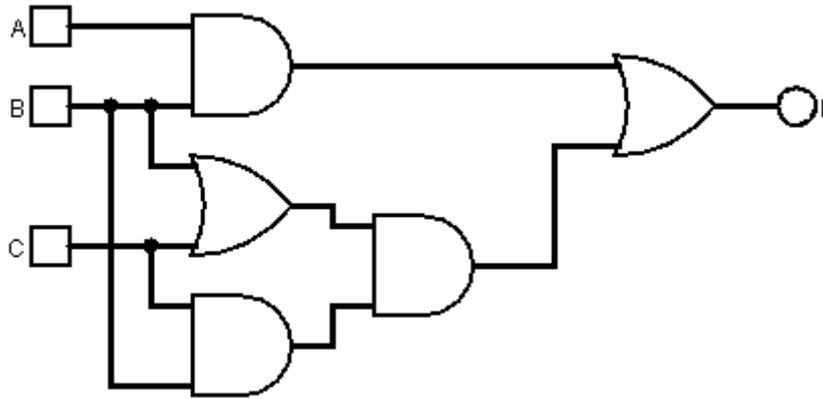
1) Trouver l'équivalent en portes universelles des portes logiques suivantes :

Porte logique	Portes NAND	Portes NOR
		
		
		
		
		
		
		

2) Transformez le circuit suivant en portes NOR en remplaçant chaque porte par son équivalent puis en utilisant la méthode du poussée de bulles.



3) Transformez le circuit suivant en portes NAND par remplacement de porte et par la méthode du poussée de bulles.



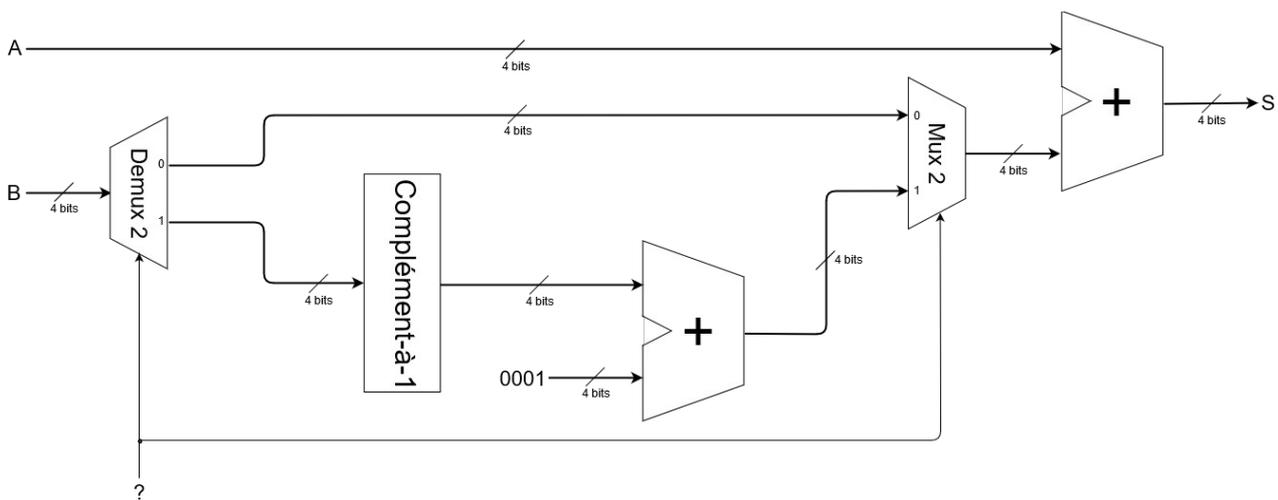
4) Donnez la conclusion de l'utilisation des deux méthodes.

Exercice 05 :

1) Utilisez les 3 méthodes suivantes pour la construction d'un Multiplexeur :

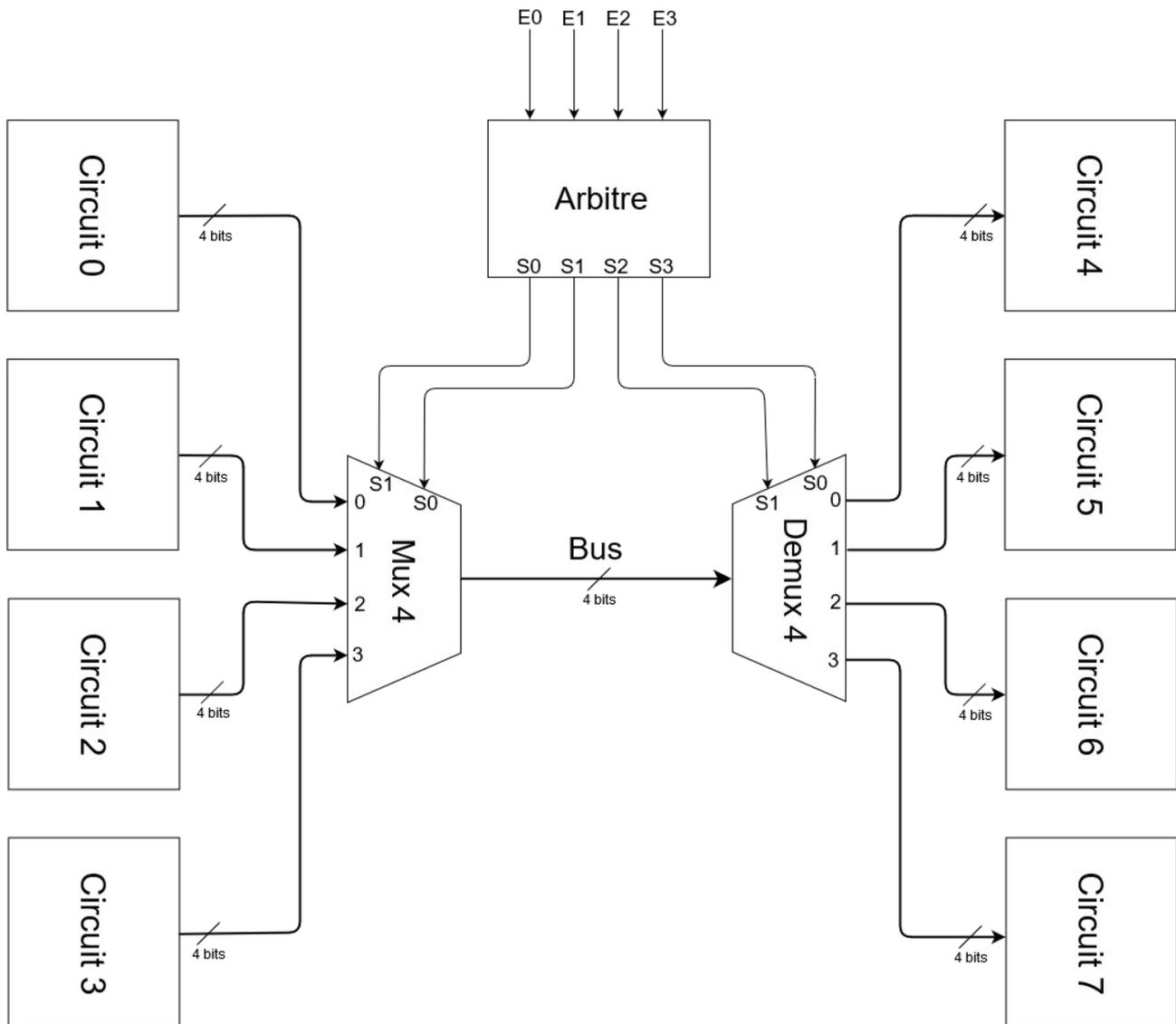
- La construction d'un Multiplexeur 2 entrées en utilisant la méthode à 5 étapes.
- La construction d'un Multiplexeur 2 entrées en utilisant les portes Tristate Buffer.
- La construction d'un Multiplexeur 8 entrées en utilisant la composition en cascade à partir de Multiplexeurs 2 entrées.

2) Décrire le fonctionnement du circuit suivant (l'Additionneur sur le schéma fait l'addition Complément-à-2/Valeur Absolue) :



- Que fait le circuit ?
- Donner un nom pour la 3-ième entrée (en ?) en bas.
- Remplacer le circuit Complément-à-1 par un circuit équivalent plus simple.
- Dessiner le schéma Global du circuit.

3) Construire le circuit combinatoire de l'Arbitre du Bus de tel-sorte que les 2 entrées E0 et E1 représentent un encodage des circuits de départ (les circuits de gauche 0-3) et les 2 entrées E2 et E3 représentent l'encodage des circuits d'arrivé (les circuits de droite 4-7)

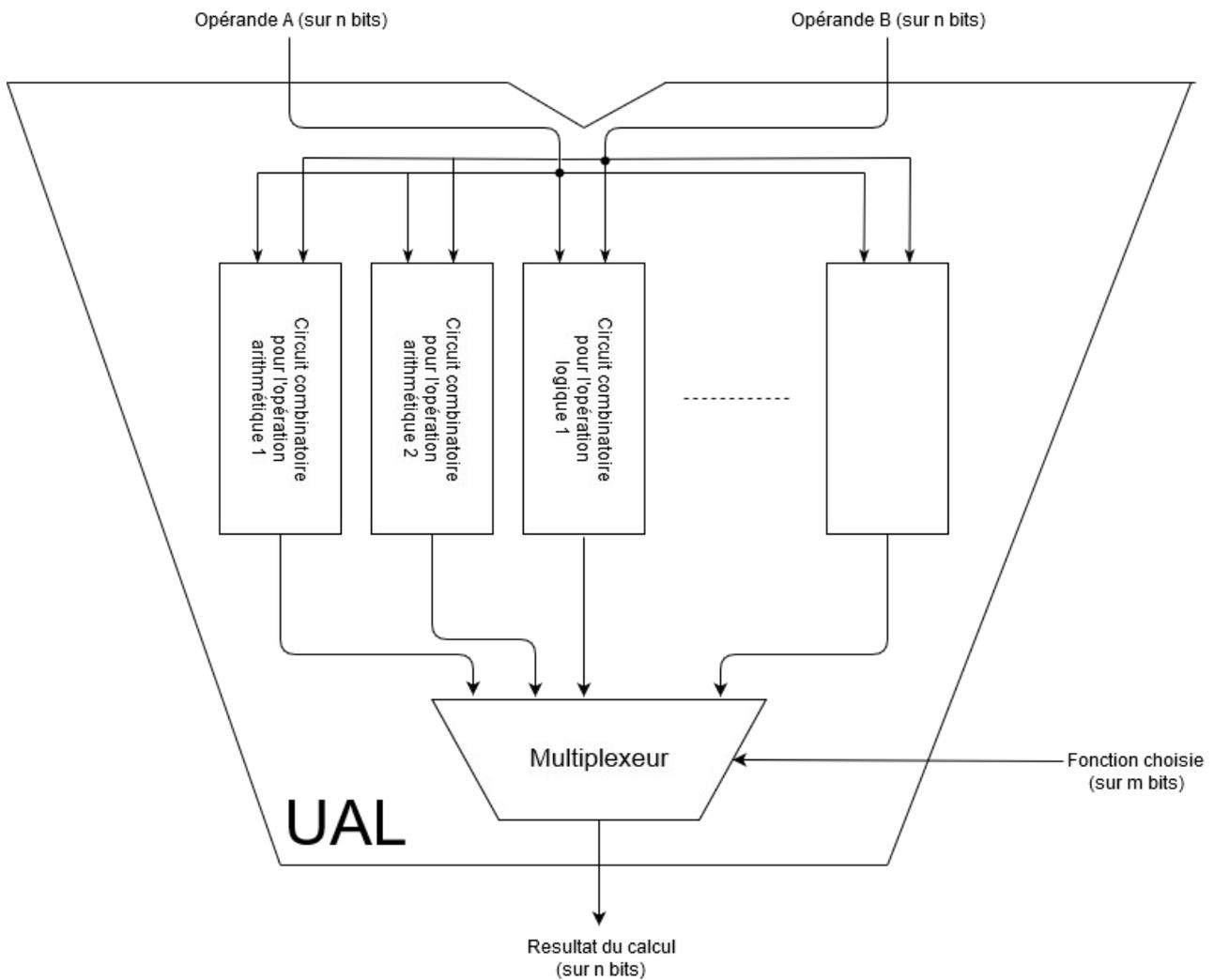


4) Construire une micro-architecture dans laquelle les transmissions de données se font en utilisant un Bus sur 8 bits à base d'état Z. La micro-architecture doit contenir 4 circuits qui peuvent tous communiquer en paires sur les 2 sens. Le Bus est régulé par un Arbitre de Bus qui contrôle les sorties des circuits par des Tristates Buffers et les entrées par le pin *enable*. Plusieurs circuits ne supportent pas en entrées la valeur Z lorsque le Bus est inactif, transformez le Bus de tel-sort à remplacer la valeur Z par un 0-faible.

Deuxièmement implémenter le circuit combinatoire de l'Arbitre qui comporte 4 entrées, pour les 3 premiers chaque combinaison représente une paire de circuit, par exemple 000 représente la paire Circuit1-Circuit2, la combinaison 001 c'est la paire Circuit1-Circuit3, et ainsi de suite jusqu'à couvrir tous les paires possibles. La dernière entrées correspond au sens de la communication, par exemple 0 pour une communication du Circuit1 vers Circuit2, et 1 inversement du Circuit2 vers le Circuit1.

Exercice 06 :

L'UAL est l'un des circuits combinatoires les plus connus dans le domaine de la micro-architecture. Théoriquement la construction d'une UAL s'appuie sur le modèle référencé sur la figure d'en bas, dans lequel l'UAL se présente sous forme de plusieurs circuits combinatoires réalisant pour chacun une opération logique ou arithmétique unique, le nombre de ces circuits est de 2^m . Tous les circuits reçoivent les 2 opérandes A et B sur n bits et font tous leurs calculs en parallèle et retournent leurs résultats au Multiplexeur, un seul résultat est choisi par le Multiplexeur parmi les 2^m proposés, le choix du Multiplexeur pour laisser passer le résultat d'un circuit précis est pris en rapport avec le nombre encodé en binaire sur m bits fourni sur l'entrée *Fonctions* du Multiplexeur. Une table est souvent fournie avec l'UAL pour spécifier le code en m bits pour chaque fonction.



1) Réaliser l'UAL qui effectue les opérations en complément-à-2 décrites dans la table en bas, tel que $n = 8$ bits et $m = 3$, donc 8 fonctions.

Fonction	F2	F1	F0	Sortie de l'UAL
0	0	0	0	A
1	0	0	1	B
2	0	1	0	B+1
3	0	1	1	A+B
4	1	0	0	A-B
5	1	0	1	not A
6	1	1	0	A and B
7	1	1	1	A or B

Remarque 1: Cette UAL est réel, elle a servie à la construction du processeur académique Mic-2.

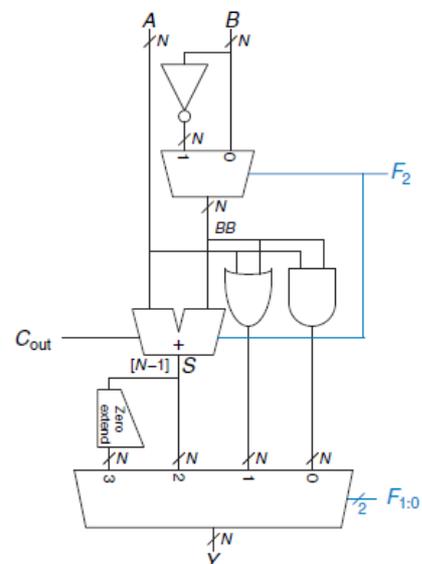
Remarque 2: Les opérations logiques de l'UAL not, and et or sont des opérations bit pas bit (*bitwise* en anglais), et pas mot par mot (un mot est sur 8 bits).

2) Pour rendre l'implémentation de l'UAL plus efficace effectuer de l'optimisation sur les opérations B+1, A+B et A-B à fin de minimiser le nombre de portes dans le circuit.

3) L'UAL représentée en bas a été directement prise du livre Digital Design and Computer Architecture (page 249), elle va servir dans le chapitre 7 du livre à la réalisation d'un pseudo-processeur MIPS. Expliquer le fonctionnement de l'UAL.

Table 5.1 ALU operations

$F_{2:0}$	Function
000	A AND B
001	A OR B
010	A + B
011	not used
100	A AND \bar{B}
101	A OR \bar{B}
110	A - B
111	SLT



4)** Construire le circuit combinatoire qui fait la Multiplication et la Division sur 4 bits en encodage Non Signé. Vous pouvez vous référencer du livre Digital Design and Computer Architecture à la page 252.

** Question difficile, à faire si il vous reste suffisamment de temps ou à faire à la maison.