

Applications des multiplexeurs et des démultiplexeurs

 Domaine d'application :
Transmission de l'information

 Type de document :
Travaux Pratiques

 Classe :
Terminale

Date :

☞ Mise en situation et objectifs du TP ☞

L'objectif de ce TP est de découvrir de nouveaux circuits logiques : **les multiplexeurs** et **les démultiplexeurs**. L'ensemble de ces nouveaux composants, ajouté aux composants logiques déjà connus [portes logiques, transcodeurs, bascules et compteurs], vous permettra de simuler désormais le fonctionnement d'un montage logique complexe. Après une découverte des multiplexeurs [**MUX** en abrégé] et des démultiplexeurs [**DMUX** en abrégé] vous devrez appliquer vos acquis dans différents problèmes. Ce TP est structuré en 4 grandes parties :

- * **Listes des nouveaux composants à connaître à l'issue de ce TP**
- * **Découverte du multiplexeur**
- * **Découverte du démultiplexeur**
- * **Applications à réaliser en mettant en œuvre tous vos savoir-faire**

☞ Travail demandé ☞

I - Liste des nouveaux composants à connaître à l'issue de ce TP

Vous trouverez dans le tableau suivant les composants principaux utilisés dans ce TP. Cette liste de composants complète la liste des composants déjà vus dans les TP précédents. Les composants déjà connus et utilisés fréquemment ne sont pas rappelés ici. De plus, il vous appartient aussi de découvrir par vous-même de nouveaux composants dans une catégorie ou une famille donnée [les TP ne vous indiquant **qu'un exemple** de chaque type] :

<i>Nom réel du composant</i>	<i>Nom exact du composant à saisir dans Mots clés</i>	<i>Catégorie</i>	<i>Bibliothèque</i>
Une roue codeuse à 16 positions	THUMBSWITCH-HEX	Switches & Relays	ACTIVE
Un interrupteur rotatif à 6 positions	SW-ROT-6	Switches & Relays	ACTIVE
Un bouton poussoir	BUTTON	Switches & Relays	ACTIVE
Une barre de 10 LED vertes	LED-BARGRAPH-GRN	Optoelectronics	DISPLAY
Un afficheur 7 segments à cathode commune	7SEG-COM-CATHODE	Optoelectronics	DISPLAY
Un compteur binaire 12 bits	4040	CMOS 4000 series	CMOS
Un compteur/décompteur 4 bits	4510	CMOS 4000 series	CMOS
Un décodeur BCD vers 7 segments	4511	CMOS 4000 series	CMOS
Un multiplexeur 8 vers 1	4512	CMOS 4000 series	CMOS
Un démultiplexeur 1 parmi 16	4514	CMOS 4000 series	CMOS
Un décodeur 1 parmi 10	4028	CMOS 4000 series	CMOS

Comment trouver rapidement un nouveau circuit logique dans Proteus ? Proteus propose des centaines de circuits logiques rangés dans la catégorie **CMOS 4000 series** mais aussi dans les catégories **TTL 74 xx**. Pour afficher un seul type de circuit logique il faut utiliser une « **Sous-catégorie** » dans la boîte de dialogue « **Pick devices** ». Voici le contenu des 8 sous-catégories principales des catégories **CMOS 4000 series** et **TTL 74 xx** :

<i>Catégories</i>	<i>Sous-catégorie</i>	<i>Cette sous-catégorie contient</i>
Circuits logiques dans CMOS 4000 series et dans les catégories TTL 74 xxx series	Gates & Inversors	les portes logiques
	Flip-Flop & Latches	les bascules
	Counters	les compteurs
	Multiplexers	les multiplexeurs
	Decoders	les décodeurs et les démultiplexeurs
	Registers	les registres
	Comparators	les comparateurs numériques
	Adders	les additionneurs numériques

Remarque à surligner en fluo, à retenir et à appliquer pour toutes vos futures utilisations de Proteus :

Pour filtrer l'affichage et accélérer la recherche d'un nouveau circuit logique il faut cliquer sur la sous-catégorie du composant recherché

Ouvrez le logiciel ISIS Proteus et ajoutez rapidement à votre sélecteur tous les composants énumérés ci-dessus.

Qu'est-ce qu'une roue codeuse ? Une roue codeuse est constituée de 4 contacts électriques actionnés mécaniquement en tournant la roue. Suivant la valeur affichée sur la roue, les contacts se ferment et s'ouvrent pour constituer sur les 4 sorties de la roue **un nombre binaire sur quatre bits**. Dans ce TP la roue codeuse nous permet de générer rapidement un nombre binaire sur 4 bits et remplace simplement 4 générateurs **LOGICSTATE** [ce qui évite de devoir faire les conversions en binaire de tête]. Par exemple pour générer le chiffre 5 en binaire il suffit de positionner la roue codeuse sur 5 : les sorties logiques de la roue codeuse génèrent alors **0101** en binaire, ce qui correspond bien au chiffre 5 codé en binaire. Placez une roue codeuse seule sur votre feuille de travail, lancez la simulation temps réel, actionnez la roue codeuse en modifiant sa position, observez l'état de ses 4 sorties logiques puis répondez aux questions suivantes **en retenant votre résultat pour la suite** :

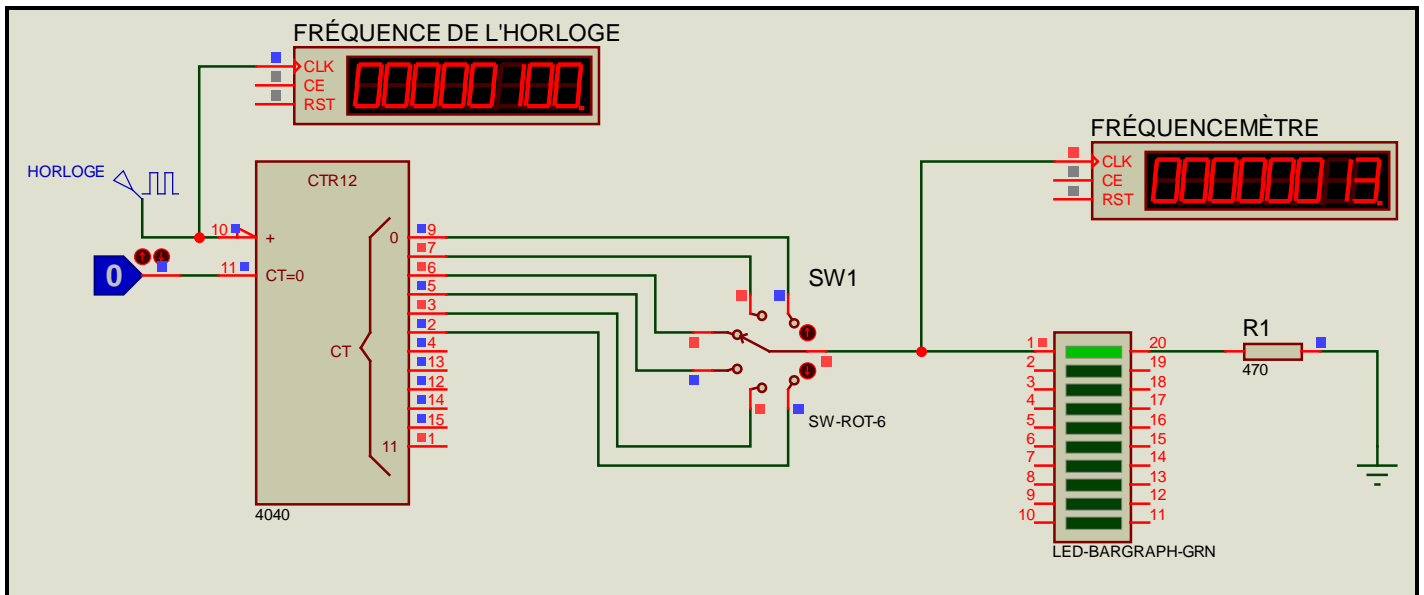
- * Où se situe le bit de **poids faible** sur la roue codeuse ? c'est la borne du **haut** c'est la borne du **bas**
- * Où se situe le bit de **poids fort** sur la roue codeuse ? c'est la borne du **haut** c'est la borne du **bas**

II - Découverte du multiplexeur

II - 1 - Réalisez dans Proteus le *Montage 1* utilisant un compteur 12 bits, un interrupteur rotatif à 6 positions, une barre de 10 LED vertes, une résistance de 470 ohms, un générateur **DCLOCK** générant un signal carré de fréquence 100 Hertz, un générateur de niveau logique **LOGICSTATE** et deux fréquemètres **COUNTER TIME**.

Remarque à surligner en fluo, à retenir et à appliquer pour toutes vos futures utilisations de Proteus :

Pour mesurer la fréquence d'un signal il faut utiliser l'appareil COUNTER TIMER en mode FREQUENCY (disponible dans le Mode Instruments Virtuels)



Montage 1

Dans ce *Montage 1* :

- * On appelle **Q0** à **Q11** les 12 sorties du compteur 4040 [Q0 est la borne n°9 et Q11 est la borne n°1]
- * On appelle **L0** à **L9** les 10 LED vertes [L0 est la LED du haut et L9 est la LED du bas]
- * On appelle **P0** à **P5** les 6 positions de l'interrupteur rotatif [en position Pn la LED L0 est reliée à Qn]
- * On appelle **H** le signal carré présent sur l'entrée d'horloge du compteur [borne n°10]

Lancez la simulation du *Montage 1* puis complétez le tableau suivant :

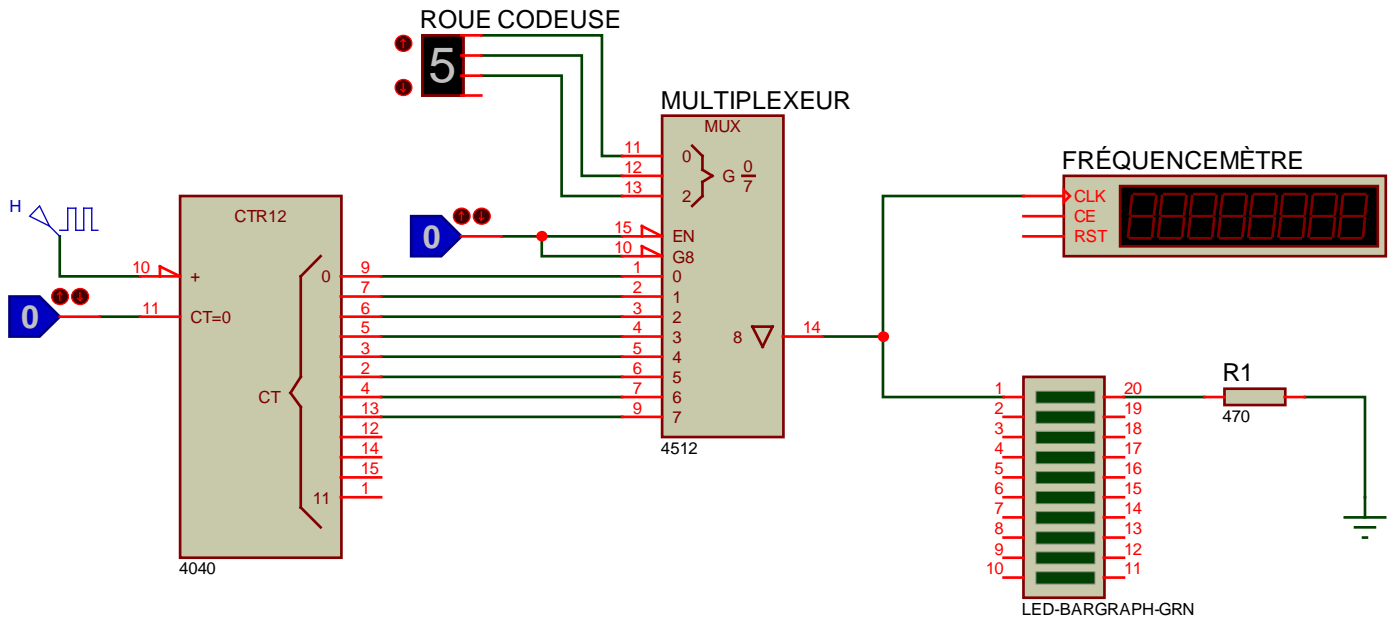
Position de l'interrupteur rotatif	Quelle sortie du compteur est reliée à L0 ?	Fréquence d'alimentation de L0
P0		
P1		
P2		
P3		
P4		
P5		

Quelle est la fréquence du signal **H** ?

II - 2 - Remplacez l'interrupteur rotatif à 6 positions par un multiplexeur 8 vers 1 et une roue codeuse connectés comme sur le *Montage 2*.

Pour le circuit 4512 [multiplexeur 8 vers 1] :

- * Les bornes 11, 12 et 13 sont appelées **les entrées d'adresse**. On les notera **A0**, **A1** et **A2**
- * Les bornes 1 à 7 et 9 sont appelées **les entrées de donnée**. On les notera **E0** à **E7**
- * La borne 14 est **la sortie de donnée** du multiplexeur. On la notera **S**.
- * Les bornes 15 (**EN**) et 10 (**G8**) sont des entrées de validation : comme le symbole l'indique par le petit triangle de complémentarité il faut que ces 2 entrées **soient à 0 pour que le circuit soit validé**



Montage 2

Lancez la simulation du *Montage 2* puis complétez le tableau suivant :

Position de la roue codeuse	A2	A1	A0	Quelle entrée du MUX arrive sur S ?	Quelle sortie du compteur est reliée à LO ?	Fréquence d'alimentation de LO
0						
1						
2						
3						
4						
5						
6						
7						

Conclusion : un multiplexeur 8 vers 1 est assimilable à un interrupteur rotatif à 8 positions : on retrouve sur la sortie du MUX l'entrée dont le numéro est programmé sur les entrées d'adresse.

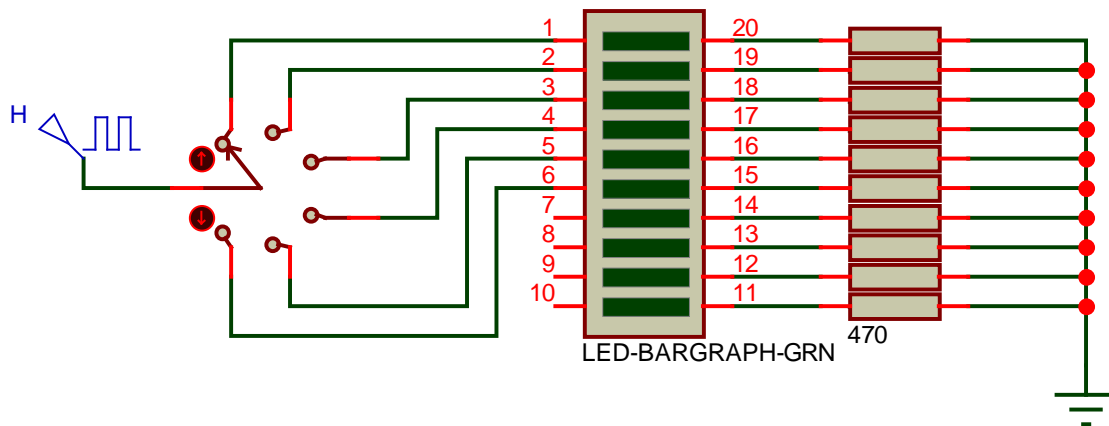
III - Découverte du démultiplexeur

III - 1 - Réalisez maintenant le *Montage 3* dans lequel :

- * Le signal H a une fréquence de 2 Hz
- * Les 10 résistances ont une valeur de 470 Ω
- * La LED LO est la LED du haut
- * Lorsque l'interrupteur rotatif est en position Pn alors H est relié à la LED Ln [avec $0 \leq n \leq 5$]

Lancez la simulation du *Montage 3* puis complétez le tableau suivant :

Position de l'interrupteur rotatif	Le signal H est relié à quelle LED ?
P0	
P1	
P2	
P3	
P4	
P5	

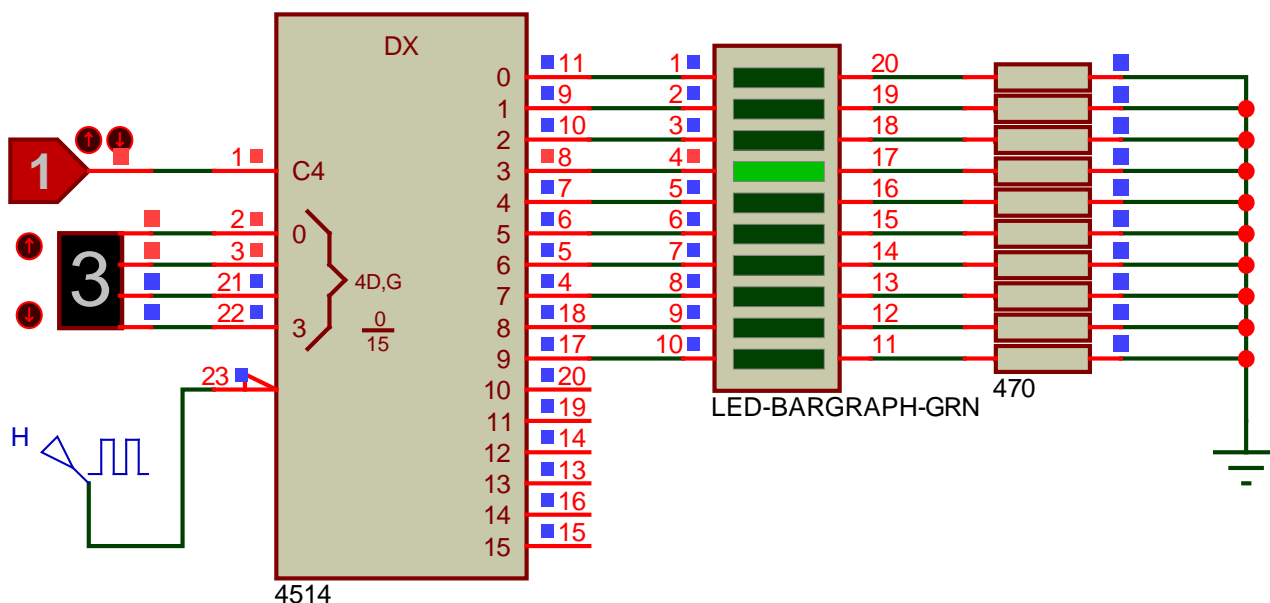


Montage 3

III - 2 - Remplacez l'interrupteur rotatif à 6 positions par un démultiplexeur 1 parmi 16 et une roue codeuse connectés comme sur le *Montage 4* :

Pour le circuit 4514 [démultiplexeur 1 parmi 16] :

- * Les bornes 2, 3, 21 et 22 sont **les entrées d'adresse**. On les notera **A0** à **A3**
- * Le borne 23 est **l'entrée de donnée**. On la notera **E**
- * La borne 1 [notée **C4**] est une entrée de validation active au niveau haut : **il faut qu'elle soit à 1 pour que le circuit soit validé**
- * Les 16 autres bornes sont **les sorties de donnée** du démultiplexeur. On les notera **S0** à **S15**.



Montage 4

Lancez la simulation du *Montage 4* puis complétez le tableau suivant [suite du tableau page 5] :

Position de la roue codeuse	A3	A2	A1	A0	L'entrée E du DMUX arrive sur quelle sortie ?	Quelle LED est reliée au signal carré H ?
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						

A						
B						
C						
D						
E						
F						

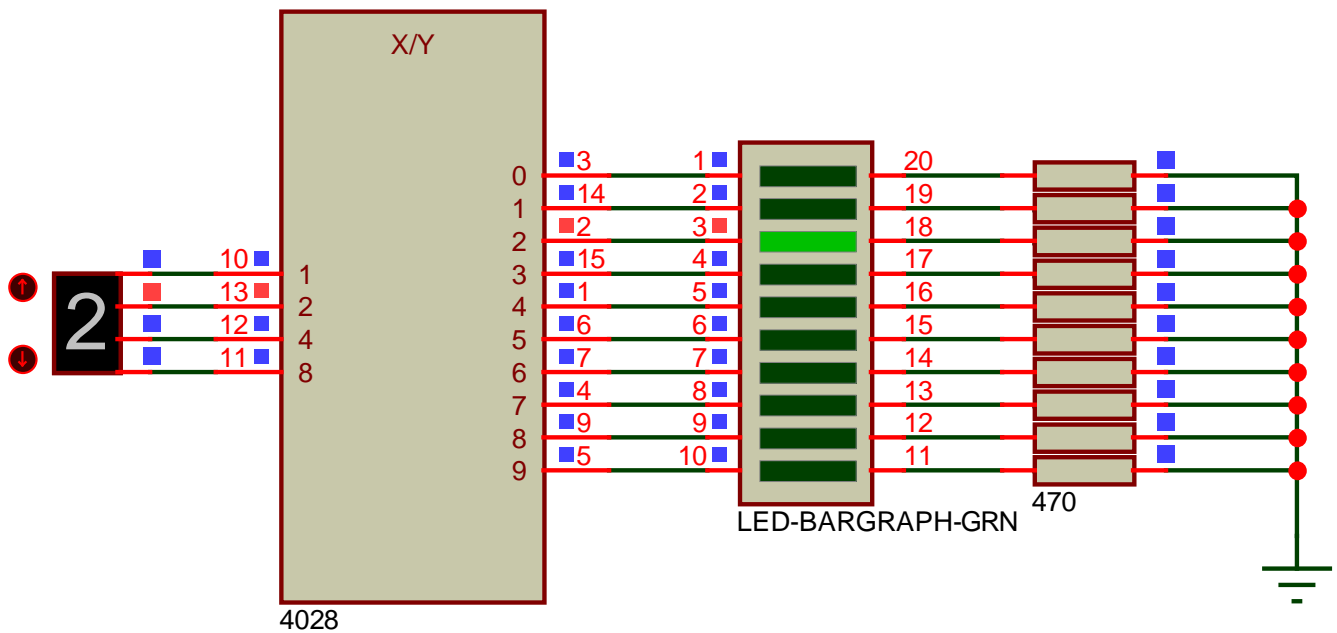
Expérimentez le rôle de l'entrée de validation **C4** du circuit 4514 puis expliquez ce qu'il se passe lorsque **C4 = 0** :

.....

.....

Conclusion : un démultiplexeur 1 parmi 16 est assimilable à un interrupteur rotatif à 16 positions : l'entrée de donnée du DMUX est reliée à la sortie dont le numéro est sur les entrées d'adresse.

III - 3 - Remplacez le DMUX 1 parmi 16 par un décodeur 1 parmi 10 connecté comme dans le *Montage 5* puis complétez le tableau ci-dessous [les 4 entrées d'adresse sont **A0** à **A3** et les 10 sorties sont **S0** à **S9**] :



Montage 5

Position de la roue codeuse	A3	A2	A1	A0	Sortie active du décodeur	LED allumée
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
A						
B						
C						
D						
E						
F						

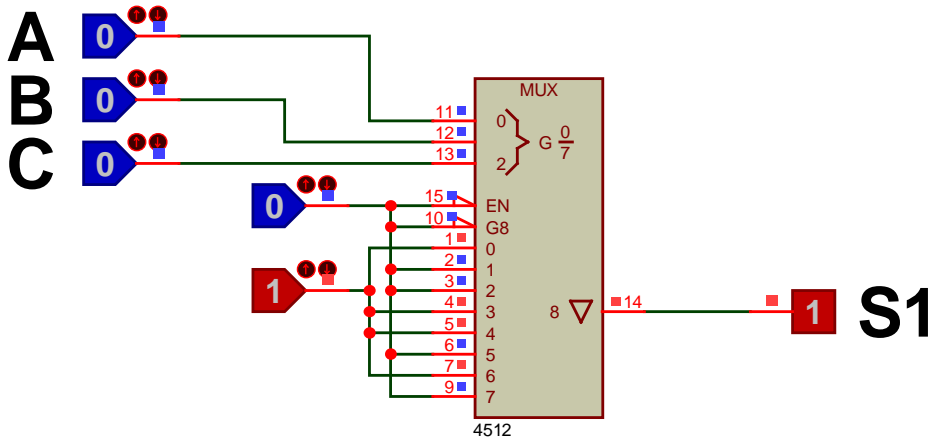
Conclusion : sur un décodeur 1 parmi 10 la seule sortie active est celle dont le numéro est sur les entrées d'adresse. La différence entre un démultiplexeur et un décodeur est que le démultiplexeur possède une entrée de donnée, alors que le décodeur ne possède pas d'entrée de donnée.

IV – Applications à réaliser en mettant en œuvre tous vos savoir-faire

IV – 1 – Application 1 : réalisation d'une équation logique avec un multiplexeur

IV – 1 – 1 – En utilisant un MUX 8 vers 1 on réalise le montage suivant dans lequel A, B et C sont 3 entrées et S1 la sortie d'une **fonction logique combinatoire**. Saisissez ce schéma dans Proteus, lancez la simulation, complétez la table de vérité de S1 puis donnez l'équation non simplifiée de S1 en fonction de A, de B et de C :

S1 =



Entrées			Sorties			
C	B	A	S1	S2	S3	S4
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

IV – 1 – 2 – On veut maintenant réaliser l'équation **S2 = A./B./C + /A./B.C + A.B.C** [dans laquelle /A se lie « A barre » et représente le complément de A]. Complétez ci-dessus la table de vérité de S2 puis proposez une solution pour réaliser la fonction logique S2 **en utilisant un MUX 4512**. Validez votre solution sous Proteus en testant bien chacune des 8 lignes de la table de vérité.

IV – 1 – 3 – On veut maintenant réaliser l'équation **S3 = A ⊕ B ⊕ C**. Complétez ci-dessus la table de vérité de S3 puis proposez une solution pour réaliser la fonction logique S3 **en utilisant un MUX 4512**. Validez votre solution sous Proteus en testant les 8 lignes de la table de vérité.

IV – 1 – 4 – On veut maintenant réaliser l'équation **S4 = A./(B + C)**. Complétez ci-dessus la table de vérité de S4 puis proposez une solution pour réaliser la fonction logique S4 **en utilisant un MUX 4512**. Validez votre solution sous Proteus en testant les 8 lignes de la table de vérité.

IV – 1 – 5 – En utilisant le même principe validez dans Proteus une solution pour réaliser l'équation logique à 4 entrées **S = A.B.C.D + /A.B./C./D + A./B.C + B./C.D** après avoir rempli sa table de vérité :

A	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
C	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
D	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
S																

Conclusion de l'application 1 : bien qu'il n'ait pas été inventé dans ce but, un multiplexeur peut parfaitement permettre de réaliser une équation logique en remplaçant tout un logigramme complexe et sans même avoir besoin de simplifier l'équation à réaliser ! Libre à vous d'utiliser ou pas cette nouvelle technique à partir d'aujourd'hui pour réaliser vos futures équations logiques dans Proteus.

IV – 2 – Application 2 : réalisation d'un chenillard bidirectionnel

IV – 2 – 1 – Proposez dans Proteus une solution pour réaliser un chenillard à partir de la barre de 10 LED vertes, c'est-à-dire l'allumage successif de chacune des LED L0 à L9 pendant une durée de 500 ms chacune. Après L9 c'est L0 qui doit se rallumer et le cycle repart, une seule LED étant allumée à la fois.

IV – 2 – 2 – Améliorez votre montage afin que le chenillard devienne bidirectionnel, c'est-à-dire que :

- * Les LED s'allument de L0 à L9 [une seule LED à la fois pendant une durée de 500 ms chacune]
- * Puis en sens inverse de L9 à L0, puis dans le sens L0 à L9, puis de L9 à L0, puis de L0 à L9, etc.

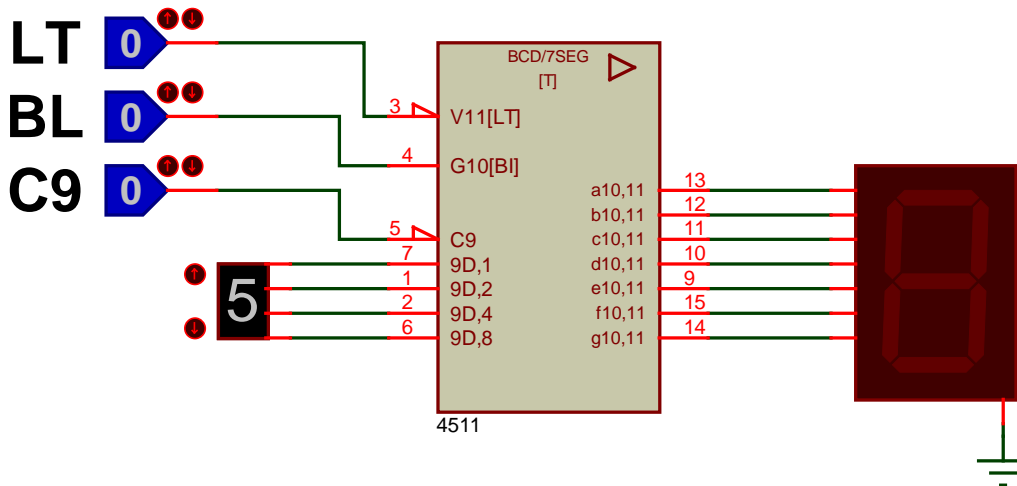
IV - 3 - Application 3 : scrutation d'un clavier et affichage de la touche enfoncée

On dispose d'un clavier à 8 touches composé de 8 boutons poussoirs B0 à B7 ainsi que d'un afficheur 7 segments. L'objectif de cette application est de proposer une solution répondant au **cahier des charges** suivant :

- * Si aucune touche du clavier n'est enfoncée alors l'afficheur doit être éteint
- * Si la touche **Bn** est enfoncée alors le chiffre **n** doit s'afficher sur l'afficheur [avec $0 \leq n \leq 7$]

IV - 3 - 1 - Principe du circuit 4511

Le circuit 4511 est un décodeur BCD / 7 segments. Son rôle est d'alimenter chacun des 7 segments d'un afficheur afin d'afficher le chiffre [entre 0 et 9] dont la valeur est entrée en BCD sur les bornes 7, 1, 2 et 6 du circuit. En plus de ces 4 entrées de données [sur lesquelles on a connecté une roue codeuse dans le schéma ci-dessous] le circuit 4511 possède 3 entrées particulières nommées LT, BL et C9 et appelées entrées de contrôle :



Voici le rôle des 3 **entrées de contrôle** du circuit 4511 :

- * Borne n°3 : l'entrée **LT** [pour *Lamp Test*] permet d'allumer tous les segments de l'afficheur pour les tester
- * Borne n°4 : l'entrée **BL** [pour *BLanking*] permet d'éteindre entièrement l'afficheur
- * Borne n°5 : l'entrée **C9** permet de mémoriser la valeur présente sur les entrées de données [bornes 7, 1, 2 et 6] et de « geler » ainsi un chiffre sur l'afficheur même si les entrées de données sont modifiées

En utilisant les informations données dans le Mémotech électronique et votre expérimentation dans Proteus, indiquer ci-dessous le comportement du circuit 4511 en fonction de l'état de ses entrées de contrôle. Ce tableau devra faire apparaître à la fois **l'état logique actif** des entrées ainsi que leur **priorité** relative :

LT	BL	C9	Comportement du 4511

Quel doit être l'état logique des entrées de contrôle LT, BL et C9 afin que le décodeur 4511 affiche sur l'afficheur le chiffre présent sur les entrées de données ? LT = BL = C9 =

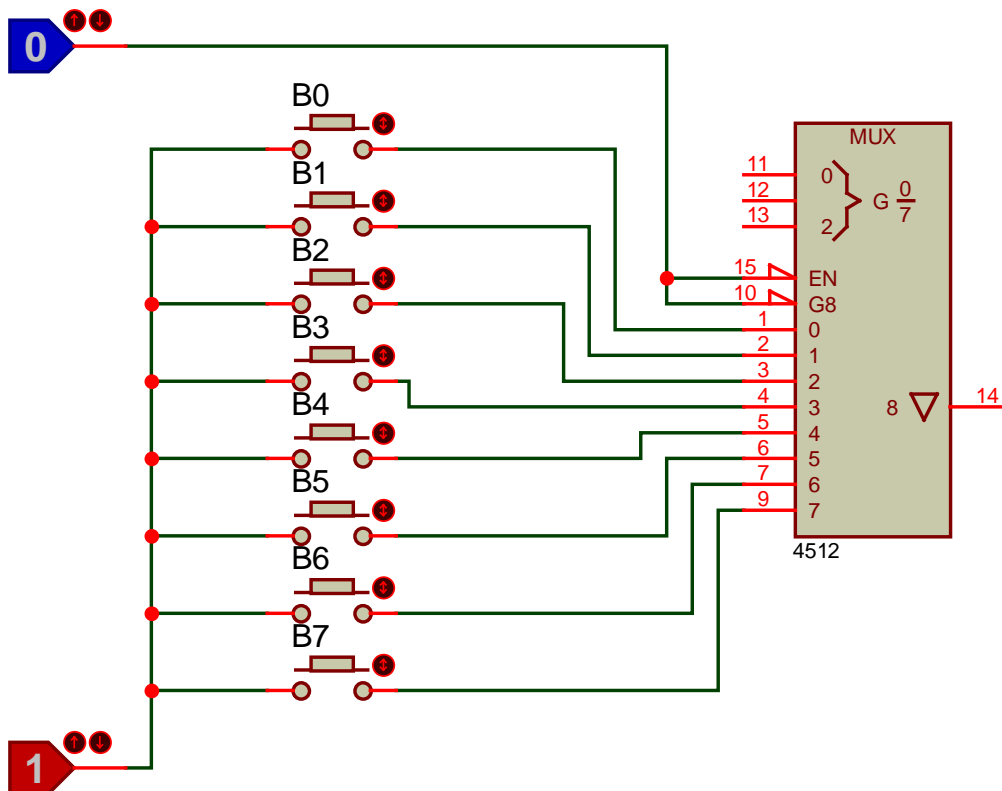
IV - 3 - 2 - Principe du clavier multiplexé et de la scrutation du clavier

Le *Montage 6* montre le principe pour multiplexer un clavier :

- * Chaque touche est reliée à l'entrée d'un multiplexeur
- * Si une touche est enfoncée elle met à 1 une entrée de donnée du multiplexeur

En mettant un compteur sur les entrées d'adresse du multiplexeur afin de « balayer » toutes les touches :

- * Si aucune touche n'est enfoncée la sortie du MUX vaut 0
- * Si une touche est enfoncée la sortie du MUX passera à 1 lorsque le compteur aura la valeur de la touche
- * **En utilisant la sortie du MUX pour bloquer le compteur, le compteur se bloquera sur la valeur de la touche enfoncée** [entre 0 et 7]
- * Une fois le compteur bloqué, il ne reste plus qu'à afficher sa valeur sur un afficheur 7 segments
- * Il suffit de relâcher la touche pour que la scrutation reparte



Montage 6

IV - 3 - 3 - Recherche d'une solution

En utilisant notamment un compteur **4510**, un multiplexeur **4512** et un décodeur BCD / 7 segments **4511** proposez une solution fonctionnelle répondant au cahier des charges. Testez, analysez puis validez votre solution sous Proteus. En cas de dysfonctionnements faites évoluer votre proposition afin de supprimer les mauvais fonctionnements. Il ne s'agit pas de trouver du premier coup une solution fonctionnelle parfaite, mais le travail que l'on vous demande ici est d'être capable d'analyser pourquoi votre montage ne fonctionne pas, et de tendre progressivement vers la solution idéale en le faisant évoluer dans la bonne direction.

IV - 3 - 4 - Extensions du problème :

Extension 1 : clavier à 10 touches avec affichage de 0 à 9

Vous disposez d'un clavier composé de 10 boutons poussoirs notés **B0** à **B9** et d'un afficheur 7 segments. En utilisant cette fois un décodeur 1 parmi 10 [circuit **4028**] à la place du multiplexeur, proposez une solution pour la scrutation du clavier afin que :

- * l'afficheur soit éteint si aucune touche du clavier n'est enfoncée
- * le chiffre **n** s'affiche sur l'afficheur si la touche **Bn** est enfoncée [avec $0 \leq n \leq 9$]

Extension 2 : clavier à 10 touches avec affichage de 01 à 10 sur 2 chiffres

Vous disposez d'un clavier composé de 10 boutons poussoirs notés **B1** à **B10** et de 2 afficheurs 7 segments. Proposez une solution pour la scrutation du clavier afin que :

- * l'afficheur soit éteint si aucune touche du clavier n'est enfoncée
- * le nombre **n** s'affiche sur les deux afficheurs si la touche **Bn** est enfoncée [avec $01 \leq n \leq 10$]

Extension 3 : clavier à 16 touches avec affichage de 00 à 15

Vous disposez d'un clavier composé de 16 boutons poussoirs notés **B0** à **B15** et de 2 afficheurs 7 segments. Proposez une solution pour la scrutation du clavier afin d'afficher le numéro de la touche enfoncée :

- * Si aucune touche du clavier n'est enfoncée alors l'afficheur doit être éteint
- * Si la touche **Bn** est enfoncée alors le nombre **n** doit s'afficher sur les deux afficheurs [avec $00 \leq n \leq 15$]

Ce TP vous a montré comment utiliser les circuits logiques multiplexeurs et démultiplexeurs dans le logiciel Proteus. Il vous appartient désormais d'utiliser naturellement ces composants et de savoir rechercher et découvrir par vous-même tous les autres circuits logiques non énumérés ici.