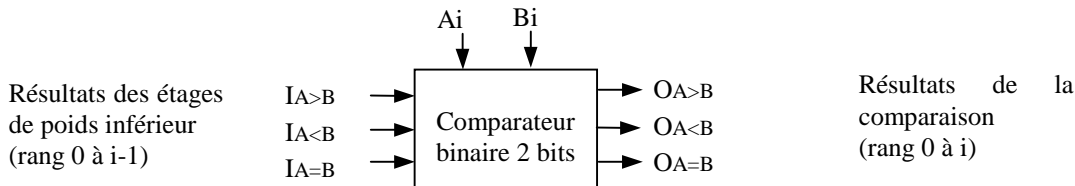


1. Etude préliminaire

On désire réaliser un système qui compare deux mots A et B de longueur n bits. Ce comparateur est constitué de n comparateurs 2 bits mis en cascade. La comparaison s'effectue bit à bit du poids faible vers le poids fort. A l'étage i , le comparateur 2 bits admet 5 entrées et 3 sorties. Les trois sorties sont le résultat de la comparaison : $A > B$, $A < B$ et $A = B$. Les entrées correspondent aux deux bits A_i et B_i à comparer et du résultat des étages de poids inférieur.



On peut ainsi établir la table de vérité du comparateur 2 bits :

	RA>B	RA<B	RA=B	IA>B	IA<B	IA=B	OA>B	OA<B	OA=B
Ai > Bi	1	0	0	X	X	X	1	0	0
Ai < Bi	0	1	0	X	X	X	0	1	0
Ai = Bi	0	0	1	1	0	0	1	0	0
Ai = Bi	0	0	1	0	1	0	0	1	0
Ai = Bi	0	0	1	0	0	1	0	0	1

- R correspond à un résultat intermédiaire provenant de la comparaison des bits A_i et B_i sans tenir compte du résultat des rangs inférieurs ($IA > B$, $IA < B$, $IA = B$).
- Les X correspondent à des états indifférents ("0" ou "1");

1. Donnez le logigramme de la fonction intermédiaire R ($RA > B$, $RA < B$, $RA = B$).
2. Donnez le logigramme du comparateur 2 bits complet à partir de R et I ($IA > B$, $IA < B$, $IA = B$).

2. Réalisation à l'aide de portes logiques élémentaires

A l'aide des circuits intégrés à votre disposition, réaliser sur les maquettes de montage le comparateur 2 bits complet. Pour vérifier le bon fonctionnement de votre montage vous utiliserez les interrupteurs et afficheurs à votre disposition.

3. Réalisation à l'aide d'un outil d'aide à la conception

L'objectif de cette partie est de se familiariser avec les outils de développement pour circuits logiques programmables. Avant de commencer le TP créez-vous un répertoire dans le dossier F:\MI41\ où vous stockerez vos fichiers.

3.1. Saisie du schéma logique.

Dans cette partie vous allez saisir le logigramme du comparateur 2 bits dans le logiciel de développement Max Plus 2.

- Lancer le logiciel MaxPlus2
- Ouvrir un nouveau fichier : File → New → Graphic Editor file
- Une fenêtre de saisie graphique s'ouvre, dans cette fenêtre le logigramme va être saisi en implantant les différents composants utilisés et en les reliant entre eux. Avant toute chose il faut définir ce fichier comme étant votre projet. Sauvegardez votre fichier dans votre répertoire et définissez le comme étant votre projet : File → project → Set Project to Current File.
- Choisissez un composant cible pour l'implantation. Exemple : Assign → Device → EPM7032SLC44-10 (famille MAX 7000s, décochez la case « show only fastest devices »)
- Saisie d'un composant : les différents composants disponibles sont appelés "symboles" et sont accessibles dans différentes librairies. Les composants élémentaires sont dans la librairie "prim" (portes ET, OU, ...). Double-cliquez dans la fenêtre là où vous souhaitez placer votre nouveau composant. Choisissez votre librairie et votre composant. Cliquez sur OK.
- Votre nouveau composant est maintenant dessiné dans votre page graphique. Procédez de même pour les autres composants dont vous avez besoin (vous pouvez également utiliser le copier/coller).
- Pour relier vos composants entre eux, pointer avec la souris sur un nœud d'entrée ou de sortie (le pointeur devient une petite croix) et cliquer en maintenant le bouton enfoncé pour dessiner une liaison.
- Pour saisir les entrées et sorties de votre comparateur vous devez choisir les symboles "input" et "output" dans la librairie "prim".

Lorsque tout votre schéma est saisi, compilez-le : Max plus2→compiler. Vérifiez qu'il n'y a pas d'erreur. Créez un nouveau composant correspondant à votre schéma : File→Create Default Symbol.

3.2. Simulation.

Afin de vérifier le bon fonctionnement de votre schéma vous allez effectuer une simulation. Pour cela il vous faut créer des stimuli (signaux à appliquer sur vos entrées) et visualiser la réponse de votre circuit à ceux-ci.

- Ouverture d'un fichier de simulation : File→new→Waveform Editor file
- Une nouvelle fenêtre apparaît. Dans cette fenêtre vous allez tracer les signaux d'entrées en fonction du temps et visualiser la réponse de votre circuit. Pour cela il faut commencer par définir la durée de simulation : File→end Time que vous fixerez à 600ns et le pas de la grille de visualisation Option→Grid Size que vous fixerez à 50ns.
- Mise en place des signaux à visualiser : Nod→Enter Node from SNF, cliquez sur "List" les entrées/sorties de votre schéma apparaissent dans la fenêtre de gauche, transférez les dans celle de droite et "OK".
- Saisie des stimuli : sélectionnez un signal complet ou un tronçon de signal et donnez lui une valeur Edit→Overwrite→...:
 - horloge (0,1 successifs)
 - high ('1')
 - low ('0')
- Effectuez la simulation : Max Plus 2→Simulator→start

3.3. Réalisation d'un comparateur 4 bits

3.3.1. Saisie

Ouvrez un nouveau fichier de saisie graphique afin de créer un comparateur 4 bits à partir du comparateur 1 bit réalisé précédemment. Désignez ce nouveau fichier comme étant votre projet: File→project→Set Project to Current File. Pour récupérer votre composant procéder comme précédemment en allant le chercher dans la librairie correspondant à votre répertoire.

En entrée vous utiliserez deux bus : saisissez une entrée comme fait précédemment en la nommant *nom*[3..0] et en nommant chaque connexion entrantes ou sortantes par *nom*[*i..j*] ou *nom*0 (poids faible), *nom*1, *nom*2 et *nom*3 (poids fort) (sélectionnez le fil de connexion et tapez directement au clavier son nom) l'association entre votre entrée "bus" et les fils de connexion s'effectue par nom et non de manière graphique.

Note : le type de dessin associé à un fil de bus doit être une ligne large.

3.3.2. Simulation

Simuler votre nouveau schéma (Choisissez des stimuli adaptés de manière à vérifier le bon fonctionnement de votre schéma)

3.4. Implantation

Cette dernière partie est consacrée à l'implantation du comparateur 4 bits dans le circuit programmable MAX7032S d'Altera. Ce composant est disponible sur vos plaquettes d'essai.

Lors de la dernière compilation les broches d'entrées/sorties du composant ont été affectées automatiquement. Pour connaître cette affectation :

- Ouvrez l'éditeur graphique d'affectation des noeuds : maxplus2→Floorplan Editor.
- Affichez le composant vu de l'extérieur : Layout→Device View.
- Affichez les nœuds à assigner : Layout→Last Compilation Floorplan.

Si vous souhaitez changer cette affectation procédez comme suit :

- Affichez les nœuds à assigner : Layout→Current Assignments Floorplan. Dans la fenêtre en haut à droite se trouve alors la liste des nœuds du composant (entrées/sorties et connexions internes).
- Assignez les entrées/sorties du composant en faisant glisser l'entrée ou la sortie présente dans la fenêtre en haut à droite sur la patte du composant de votre choix.
- Compilez à nouveau le projet afin que votre assignation soit prise en compte.

Pour programmer le composant :

- Programmer le composant : MaxPlus2→Programmer et "Program".
- Vérifiez le bon fonctionnement de votre implantation à l'aide des interrupteurs et afficheurs à votre disposition

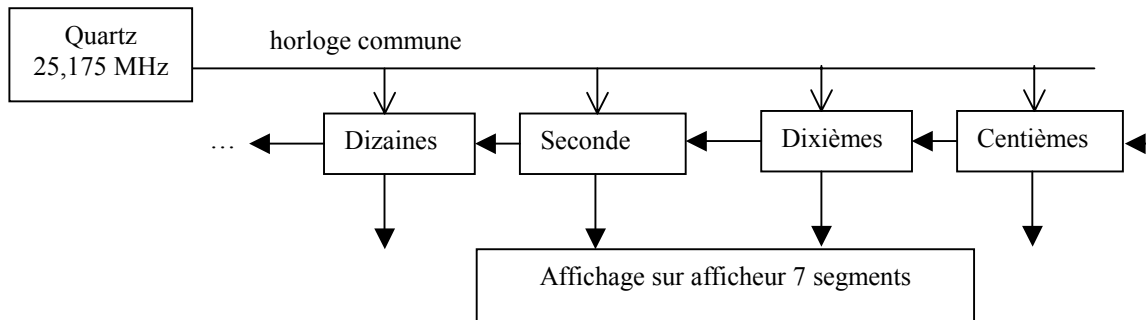
1. Présentation

Il s'agit dans ce TP de réaliser un compteur en vue de réaliser un chronomètre. Les maquette d'expérimentation ne comportant que 2 afficheur, on se limitera ici à l'affichage de deux digits (secondes et dixièmes de seconde).

Ce compteur est commandé par un bouton séquentiel permettant :

- un appui bref : (inférieur à 1 seconde) : démarrage du compteur si ce dernier était arrêté, arrêt du compteur si celui-ci était en phase de comptage.
- un appui long supérieure à 1 seconde : remise à zéro du compteur et arrêt.

Synoptique non complet :



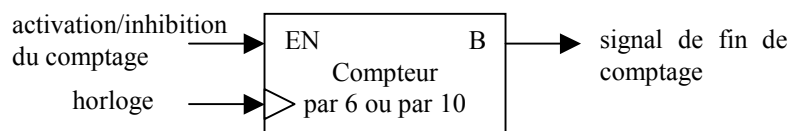
2. Réalisation

2.1. Comptage synchrone

Dans un premier temps on ne s'occupe pas de la fréquence d'horloge.

Pour le comptage, on souhaite disposer de compteurs par 10 et par 6 synchrones cascables. En effet, tous les compteurs étant reliés à la même horloge, il est nécessaire de pouvoir :

- inhiber le comptage
- envoyer un signal en fin de comptage pour les compteurs de poids supérieurs



1. Réaliser le compteur par 10 et vérifier le bon fonctionnement par simulation
2. Réaliser le compteur par 6 et vérifier le bon fonctionnement par simulation
3. Réaliser un compteur par 60 à l'aide des deux précédents et vérifier le bon fonctionnement par simulation
4. Visualiser le fonctionnement du compteur par 60 en implantant votre système sur les maquette d'expérimentation (Max 7032S).
5. Pour la suite on se contentera d'un compteur par 1000 (comptage au centième de seconde et limité à 10 secondes)

2.2. Affichage (VHDL)

On souhaite intégrer le système complet sur la maquette de TP comportant un PLD MAX7128S relié à 2 afficheurs 7 segments. Le système doit donc intégrer un transcodeur BCD 7 segment afin de pouvoir visualiser les valeurs du compteur.

➔ Saisir la description VHDL du transcodeur BCD/ 7 segments

2.3. Division d'horloge (VHDL)

La maquette utilisée utilise un quartz à 25 175 000 Hz. Cette horloge sera reliée directement à votre compteur. Bien entendu, la fréquence est trop élevée pour un comptage au centième de seconde.

➔ Ajouter les éléments nécessaires au système pour avoir un fonctionnement correct du compteur au centième de seconde.

2.4. Gestion des interrupteurs (VHDL)

➔ Prendre en charge la gestion de l'interrupteur

Note : l'interrupteur a de forte chance de produire des oscillations d'une durée inférieure à la milliseconde (on pourra se contenter d'une précision de notre chronomètre au dixième de seconde).

1. Séquenceur de moteur pas à pas

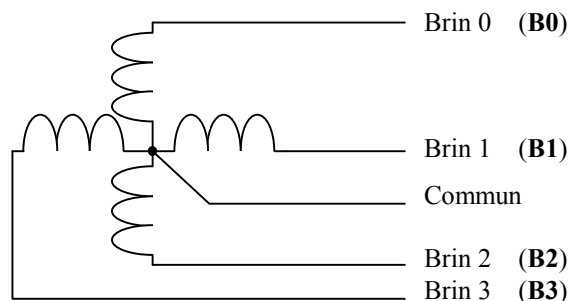
Il s'agit dans ce TP de réaliser la commande d'un moteur pas à pas de manière à ce que sa vitesse soit proportionnelle à une fréquence d'horloge.

1.1. Structure du moteur

Le principe d'un moteur pas à pas consiste à alimenter à tour de rôle les bobines du moteur.

2 types de commandes simples existent :

- par pas entier où on alimente 2 bobines adjacentes à tour de rôle (4 commandes différentes pour un cycle)
- par demi-pas où on alimente alternativement une bobine seule, 2 bobines adjacentes, une bobine seule,



1.2. Cahier des charges du séquenceur

On souhaite une rotation dans les deux sens du moteur par demi pas. Pour tourner dans un sens, la séquence de commande doit être la suivante :

B3	B2	B1	B0
1	0	0	0
1	1	0	0
0	1	0	0
0	1	1	0
0	0	1	0
0	0	1	1
0	0	0	1
1	0	0	1

Ordre d'envoi des commandes

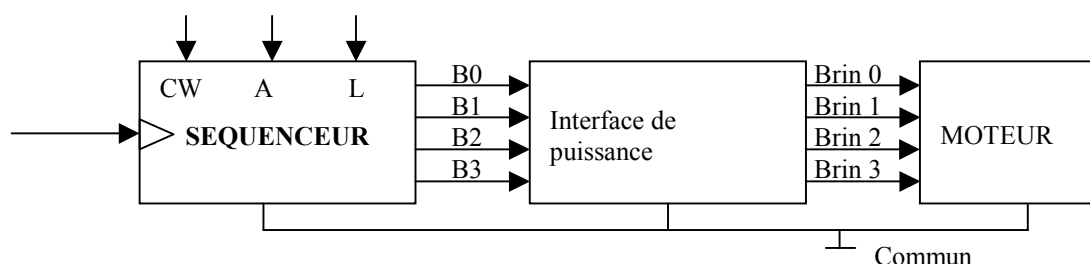
- Un 1 logique correspond à l'alimentation de la bobine associée.
- Pour une rotation en sens inverse il faut remonter le tableau.
- Lorsqu'aucune bobine est alimentée B3...B0 = 0000 le moteur est libre (pas de couple).
- Lorsque la commande est maintenue le moteur est à l'arrêt avec un couple de maintien.

On souhaite 4 fonctionnements possibles en fonction de 3 variables d'entrée :

- **CW** : Lorsque CW = 1 on a une rotation dans le sens des aiguilles d'une montre. Lorsque CW est à 0 on a une rotation dans le sens inverse des aiguilles d'une montre.
- **A** : Lorsque A = 1 le moteur est à l'arrêt avec couple.
- **L** : lorsque L = 1 le moteur est libre (pas de commande et donc pas de couple).

Lorsque L et A sont à 0 on est en rotation, lorsque L et A sont tous les deux à 1 le moteur est libre (L prioritaire sur A).

1.3. Schéma de la commande du moteur



2. Réalisation du graphe d'état

Pour chacune des commandes possibles on associe un état.

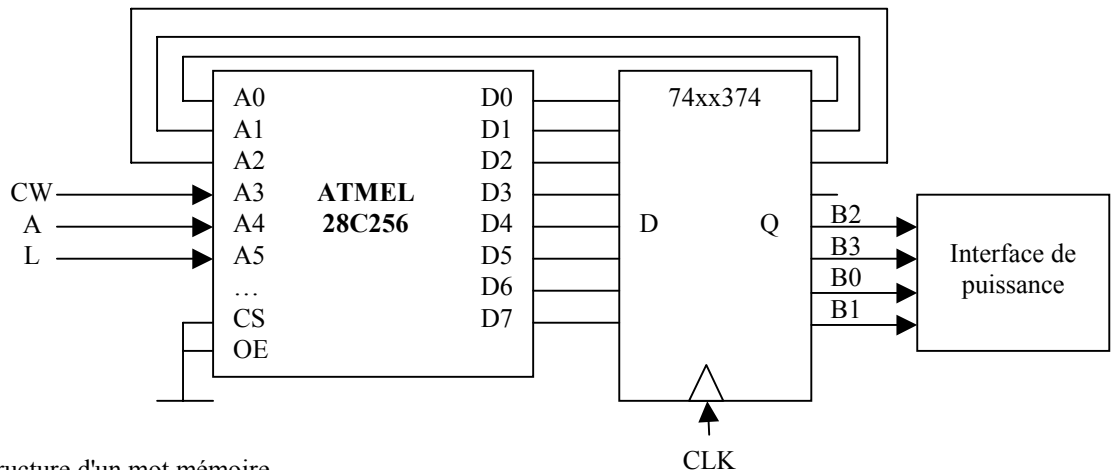
- Représenter le graphe à états qui décrit le fonctionnement du séquenceur de commande du moteur.

3. Implémentation en logique câblée

Réaliser **sur papier** ce séquenceur à l'aide de bascules D actives sur front montant et de portes logiques élémentaires.

4. Implémentation à l'aide d'une mémoire

On utilise pour réaliser le séquenceur une mémoire EEPROM 28C256 (32Kx8) et 8 bascules D (74LS374).
Le choix du câblage est le suivant :



- Donnez la structure d'un mot mémoire
- Etablissez le contenu de 64 premiers octets de la mémoire (donner les valeurs en hexadécimal). Attention à l'ordre des brins
- Saisissez les dans le logiciel ChipWin (DataIO/ChipWriter)
 - Sélectionner le "device" ATMELE/EEPROM/28C256
 - Remplissez les 64 premiers octets (valeur en hexadécimal).
 - Sauver votre fichier dans votre répertoire
- Avec l'aide de l'intervenant de TP, programmer la mémoire sur le poste programmation.
- Testez

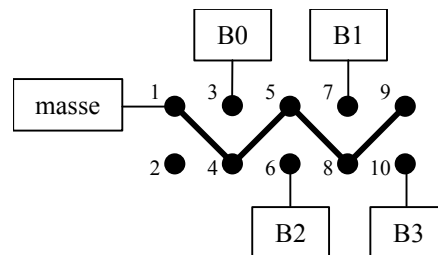
Remarque : la réalisation doit correspondre exactement à votre graphe d'état...

5. Implémentation à l'aide d'un circuit programmable

On utilise cette fois ci le PLD Max7032s d'Altera pour implémenter le séquenceur.

- Ecrivez le programme VHDL correspondant au graphe d'état sous MaxPlusII.
- Vérifiez le bon fonctionnement par simulation.
- Réaffectez les broches d'entrées/sorties si vous le souhaitez
- Programmez le composant.
- Réalisez les connexions avec l'interface de puissance

Brochage du connecteur HE10 :



6. Conclusion

- Commenter les différentes réalisations possibles
 - ➔ facilité d'implantation, vitesses maximales des commandes, souplesse face aux modifications (par exemple si on souhaite mettre une commande par pas entier), coût ...
- On souhaiterait pouvoir contrôler approximativement la vitesse du moteur entre 32 et 500 demi-pas par seconde.
 - Expliquez clairement comment pourrait-on effectuer ce contrôle de la vitesse à partir d'une horloge à 1khz.
 - Est-ce que ce contrôle de vitesse peut-être implanté dans le circuit programmable ? (On ne demande pas de réaliser ce contrôle de vitesse).

7. Annexes : contenu mémoire

Adresses (Hexa)	Contenu case (binaire)	Contenu (Hexa)
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
0A		
0B		
0C		
0D		
0E		
0F		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
1A		
1B		
1C		
1D		
1E		
1F		

Adresses (Hexa)	Contenu case (binaire)	Contenu (Hexa)
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
2A		
2B		
2C		
2D		
2E		
2F		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
3A		
3B		
3C		
3D		
3E		
3F		