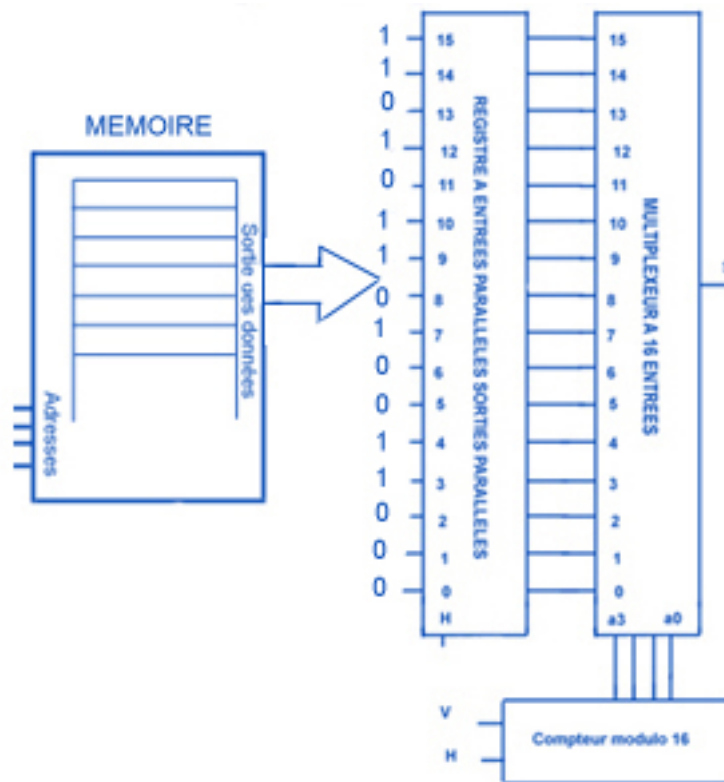


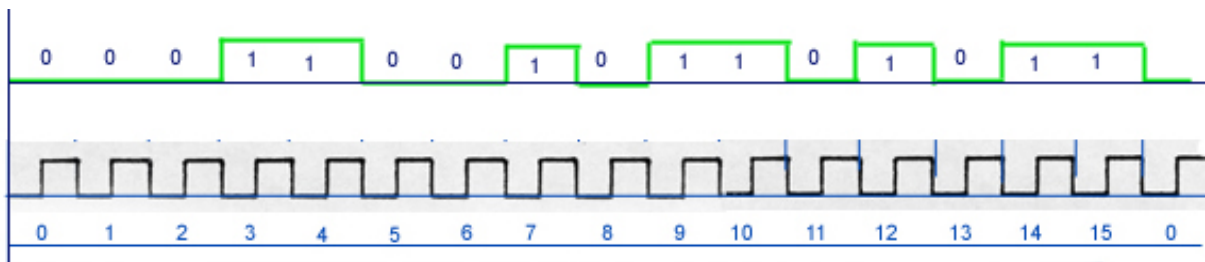
LES SEQUENCEURS PROGRAMMABLES

Un séquenceur programmable va pouvoir émettre une séquence d'impulsions dont la chronologie va être modifiable à l'aide d'un nombre binaire. Cette séquence peut être une suite de 0 et de 1 sur une seule ligne ou des impulsions qui vont servir à contrôler le fonctionnement de différents circuits à différents moments de la séquence.

Plusieurs techniques sont possibles nous en évoquerons deux qui permettent de mettre en application les composants étudiés dans les leçons précédentes.

1^{er} séquenceur

Dans la mémoire sont stockés des nombres binaires de 16 bits, un nombre correspondant à la séquence désirée est appliqué sur les entrées d'adresse de la mémoire. La mémoire place sur ses sorties le nombre de 16 bits correspondant. Une impulsion d'horloge sur l'entrée H du registre qui mémorise ce nombre. Le nombre de 16 bits est, de ce fait est présent sur les entrées du multiplexeur. Lorsque le compteur est validé, l'horloge incrémente le compteur qui va successivement balayer toutes les adresses de 0 à F envoyées au multiplexeur. Sur la sortie se succèdent les bits présents sur les entrées du multiplexeur.

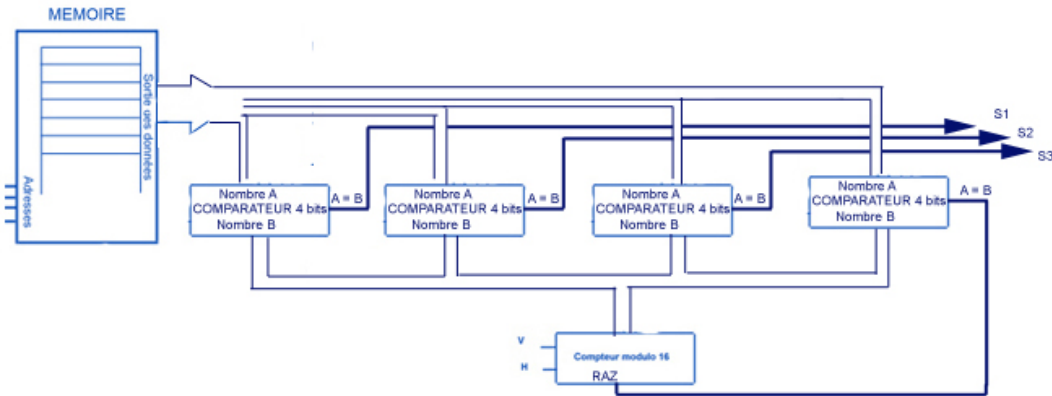


Si le compteur n'est pas inhibé lors de son recyclage la même séquence va se reproduire.

Nous venons de voir comment réaliser une succession d'impulsion sur une seule ligne, dont la disposition va dépendre du nombre binaire fourni.

Un autre problème est de réaliser une séquence d'impulsions sur différentes lignes, par exemple des registres peuvent avoir à stocker des informations qui ne se succèdent pas toujours de la même façon.

2^{ème} séquenceur



Dans le montage ci-dessus une mémoire contient des mots de 16 bits composés de 4 bytes de 4bits contenant chacun le N° d'ordre de l'impulsion à générer dans une séquence de 16 périodes d'horloge.

Supposons que le mot mémoire sélectionné par l'adresse soit 1100 1001 0101 0010 , le compteur étant à 0, la validation est effectuée

Le 1^{er} comparateur reçoit 0010, l'impulsion en S1 est émise lorsque le compteur atteint la valeur 2 et retombe au 3^{ème} pulse

Le 2^{ème} comparateur reçoit 0101 , l'impulsion en S2 est émise lorsque le compteur atteint la valeur 5

Le 3^{ème} comparateur reçoit 1001 , l'impulsion en S3 est émise lorsque le compteur atteint la valeur 9

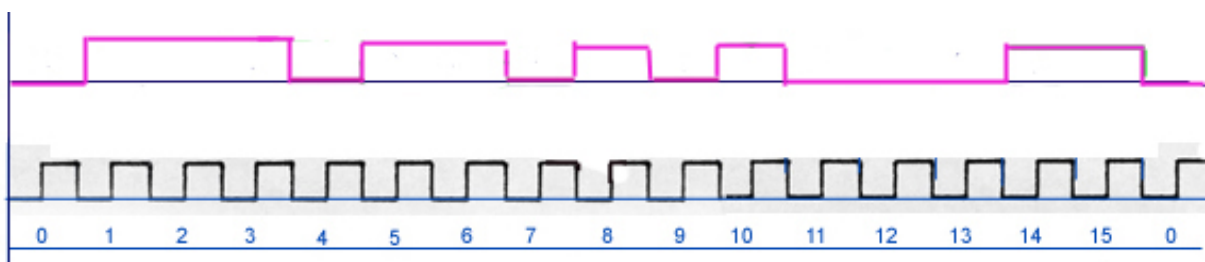
Le 4^{ème} comparateur reçoit 1100 , l'impulsion en S4 est émise lorsque le compteur atteint la valeur 12 qui reset le compteur mettant fin à la séquence.

Dans un tel système, la durée de la séquence est au maximum de 16 impulsions d'horloge mais peut s'adapter à la complexité de la tâche. Il en sera de même dans les microprocesseurs, Nous verrons que le nombre d'impulsions d'horloge nécessaires pour exécuter un tâche sera variable, la longueur de la séquence devra donc être adaptable, le contraire alignerait la séquence sur la plus longue rendant le microprocesseur particulièrement lent.

Exercices

1^{er} exercice : Dans le premier séquenceur, quel est , en hexadécimal le nombre stocké en mémoire et sélectionné par l'adressage dans l'exemple représenté

2^{ème} exercice Dans le premier séquenceur quel nombre binaire et hexadécimal doit être stocké en mémoire pour obtenir la séquence ci dessous



3^{ème} exercice Dans le premier séquenceur que peut-on dire de la séquence générée si le nombre stocké en mémoire est en hexadécimal 5555 (on peut noter l'hexadécimal comme 0x5555, notation employée dans certains assembleurs destinés à la programmation des microprocesseurs) et 0x3333

4^{ème} exercice Dans le deuxième séquenceur on veut obtenir une impulsion au temps 4 puis 5 puis 8 quel nombre devra être sélectionné par la mémoire

5^{ème} exercice Quelles modifications devraient être effectuées dans le deuxième séquenceur pour répondre au problème suivant : Un compteur génère l'adresse envoyée sur la mémoire, on veut que avant le reset du compteur du séquenceur, le compteur d'adresse pointe sur la séquence d'adresse suivante