

1. Avertissement

Pour être valide un résultat doit comporter une démonstration s'appuyant sur un raisonnement scientifique.

2. QUESTION SUR LE PROJET [8]

1) Décrire les fonctions principales du microcontrôleur 68HC11 [2]

Le μ processeur 68HC11 de Motorola (descendant du 6800 datant de 1975) produit par Freescale Semiconductor, est un contrôleur CISC (Complex Instruction Set Computer : nombreuses instructions + modes d'adressages complexes). Il exécute des instructions contenues dans un programme écrit en langage Assembleur et compilé en code machine.

Ces instructions sont essentiellement : lecture/écriture/traitement (opérations logiques ou arithmétiques simples) d'une donnée dans un registre ou en mémoire.

Il possède 2 accumulateurs (A, B) et 1 registre de condition (C) de 8b, 2 registres d'index (X, Y), un pointeur de pile (S : contient l'@ où est mémorisée la pile qui est un ensemble d'octets mémorisant le contenu des registre lors d'interruption, d'appel à sous programmes...) et un compteur programme (P : contient l'@ de la prochaine instruction à exécuter) de 16b.

2) A quoi sert un « Port » d'un microcontrôleur ? Donner un exemple d'utilisation d'un « Port » en écriture et d'un port en lecture. Comment programmer un « Port » bidirectionnel en lecture ou en écriture ? [2]

5 ports d'E/S de 8b servant à recevoir ou transmettre un signal avec l'extérieur. PortA : E/S, PortB : S, PortC : E/S, PortD : E/S, PortE : E. Pour lire ou écrire sur un port il suffit d'écrire ou de lire à l'@MEM réservée à ce port (PortA : @ \$1000).

Écrire « 1 » sur PA4 = OC4 : `Idaa #$10 ou Idaa #%0001 0000 staa $1000`

Lire l'état de PortA : `Idaa $1000`

Pour programmer l'E/S il faut écrire dans le registre DDRA (Data Direction Register for PortA @Mem \$0001) à l'emplacement correspondant dans PortA : « 0 » pour E et « 1 » pour S.

Le bit DDR7 du PACTL (Pule Accumulator ConTroL) permet de programmer en E ou S les bit PAI (PA7) du PortA

3) Expliquer la méthode de polling (ou scrutation) [2]

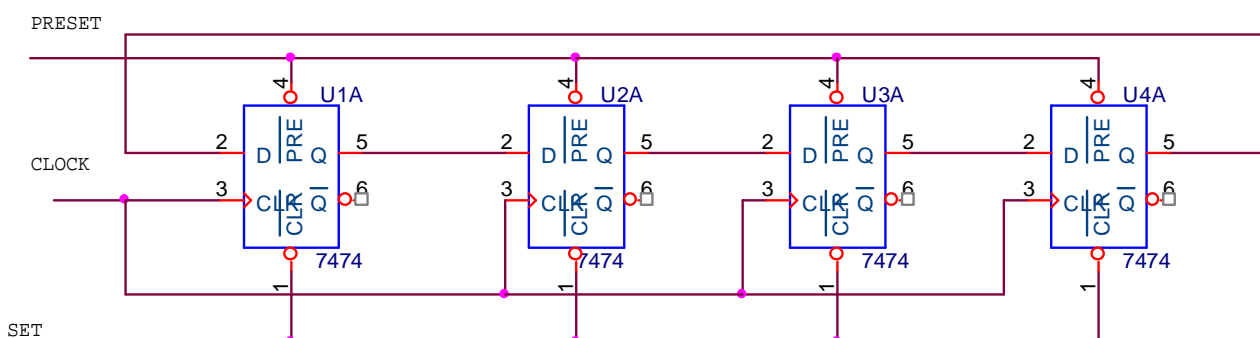
Utilisation d'une boucle conditionnelle. Dans cette boucle on lit dans le registre TFLG1 (@MEM \$1023) l'état du flag correspondant au bit d'entrée du PortA qui nous intéresse (IC3F pour PA0, IC2F pour PA1 ou IC1F pour PA2), en chargeant l'état de TFLG1 dans l'accumulateur et en y faisant une RAZ de tous les bits sauf celui qui nous intéresse : on reboucle si RegA est à « 0 » sinon on sort de la boucle. Dans TCLT2 (@MEM \$1021) on aura pris soin de programmer sur quel bit d'entrée (PA0, PA1 ou PA2) de PortA et sur quel front le flag va réagir.

4) Expliquer la programmation par interruption [2]

Utilisation d'une boucle inconditionnelle (infinie). On ne peut en sortir que par une interruption. Il faut programmer TCLT2 comme pour la scrutation et TMSK1 (@MEM \$1022) pour autoriser une interruption sur événement sur 1 bit d'entrée du PortA (IC3I pour PA0, IC2I pour PA1 ou IC1I pour PA2).


3. Exercice 1 [6]

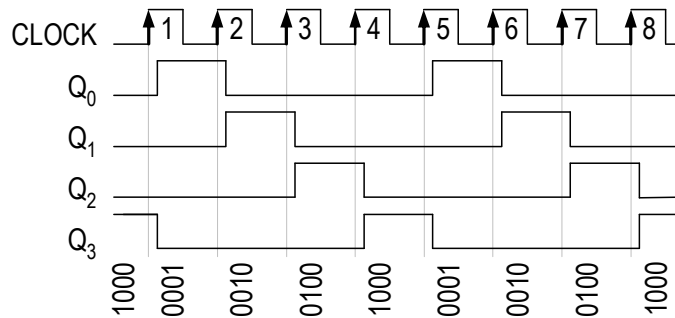
Soit le circuit de la figure suivante. Les entrées et sorties des bascules sont numérotées de 0 à 3 en partant de la gauche. Le registre est initialisé à la valeur $Q_3Q_2Q_1Q_0 = 1000_2 = 8_{10}$.



1) *Élaborer le chronogramme des sorties des bascules pour 8 impulsions d'horloge (CLOCK) [2]*

Supposons : PRESET=SET=1 \Rightarrow entrées de forçage asynchrone inactives.

CLK₀ = CLK₁ = CLK₂ = CLK₃ = CLOCK \Rightarrow circuit synchrone + toutes les bascules D actives sur CLOCK = 
 D₀ = Q₃, D₁ = Q₀, D₂ = Q₁, D₃ = Q₂ : registre synchrone bouclé ou registre en anneau ou compteur de Johnson.



2) *Donner en binaire et en décimal, le code cyclique obtenu [1]*

3) *Quelle valeur faut-il donner à PRESET (ou à SET) pour activer cette entrée [1]*

L'initialisation est maintenant obtenue par l'activation de l'entrée PRESET

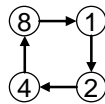
4) *Quel est le code cyclique obtenu ? [1]*

L'initialisation est maintenant obtenue par l'activation de l'entrée SET.

5) *Quel est le code cyclique obtenu ? [1]*

Cl : Q₃Q₂Q₁Q₀ = 1000

CLK	Q ₀	Q ₁	Q ₂	Q ₃	[Q ₃ Q ₂ Q ₁ Q ₀] ₁₀
0	0	0	0	1	8
1	1	0	0	0	1
2	0	1	0	0	2
3	0	0	1	0	4
4	0	0	0	1	8
5	1	0	0	0	1
6	0	1	0	0	2
7	0	0	1	0	4
8	0	0	0	1	8



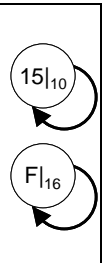
Cl : SET=0: Q₃Q₂Q₁Q₀ = 0000

CLK	Q ₀	Q ₁	Q ₂	Q ₃	[Q ₃ Q ₂ Q ₁ Q ₀] ₁₀
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0



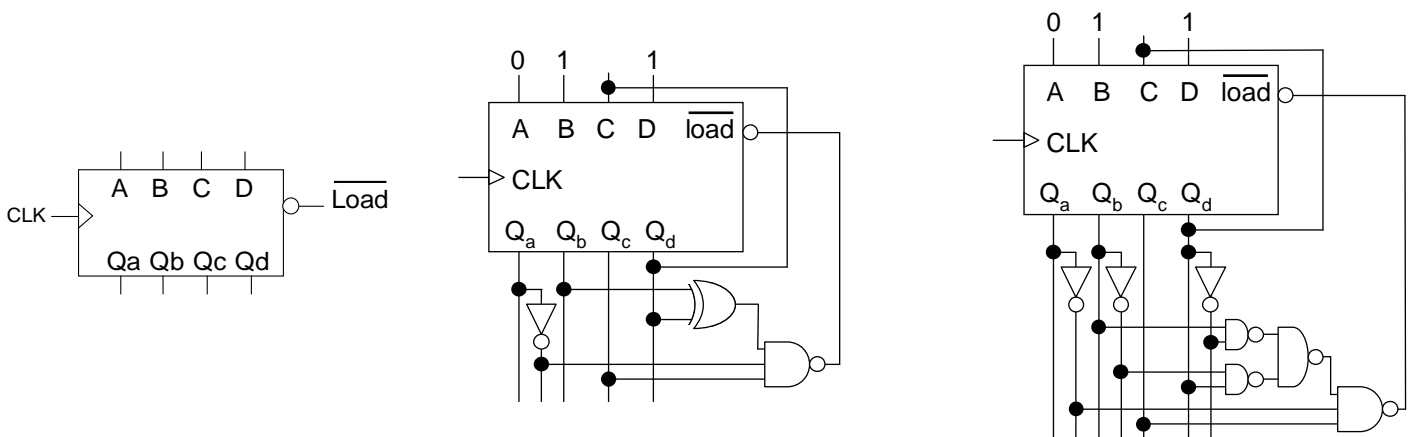
Cl : PRESET=0: Q₃Q₂Q₁Q₀ = 1111

CLK	Q ₀	Q ₁	Q ₂	Q ₃	[Q ₃ Q ₂ Q ₁ Q ₀] ₁₀
0	1	1	1	1	15
1	1	1	1	1	15
2	1	1	1	1	15
3	1	1	1	1	15
4	1	1	1	1	15
5	1	1	1	1	15
6	1	1	1	1	15
7	1	1	1	1	15
8	1	1	1	1	15



4. **Exercice 2 [5]**

Un compteur synchrone modulo 16 possède 4 entrées de chargement parallèle (A, B, C, D), 4 sorties (Q_A, Q_B, Q_C, Q_D) et une commande de chargement parallèle (Load).



1) En utilisant la commande de chargement, réaliser le cycle : 0,1,2,3,4,5,6,10,11,12,14,15

Détection des états 6 et 12 commandant le chargement parallèle :

$$6|_{10} = 0110|_2 = \overline{Q_d} Q_c Q_b \overline{Q_a} \text{ et } 12|_{10} = 1100|_2 = Q_d Q_c \overline{Q_b} \overline{Q_a} .$$

D'où :

$$\overline{load} = \overline{Q_d Q_c Q_b \overline{Q_a} + Q_d Q_c \overline{Q_b} \overline{Q_a}} = \overline{Q_c \overline{Q_a} (Q_d Q_b + Q_d \overline{Q_b})} = \overline{Q_c \overline{Q_a} (Q_d Q_b + Q_d \overline{Q_b})}$$

$$\overline{load} = \overline{Q_c \overline{Q_a} Q_d Q_b Q_d \overline{Q_b}} \text{ ou } \overline{load} = \overline{Q_c \overline{Q_a} (Q_b \oplus Q_d)}$$

La séquence sera après chaque front montant de l'horloge (\uparrow) :

$$[Q_d Q_c Q_b Q_a] = 5|_{10} \uparrow [Q_d Q_c Q_b Q_a] = 6|_{10} \Rightarrow \overline{load} = 0 \Rightarrow [DCBA] = [10]_{10} \uparrow [Q_d Q_c Q_b Q_a] = 10|_{10}$$

$$[Q_d Q_c Q_b Q_a] = 11|_{10} \uparrow [Q_d Q_c Q_b Q_a] = 12|_{10} \Rightarrow \overline{load} = 0 \Rightarrow [DCBA] = [14]_{10} \uparrow [Q_d Q_c Q_b Q_a] = 14|_{10}$$

Ainsi quand $[Q_d Q_c Q_b Q_a] = 6|_{10} = 0110|_2$ nous devons charger : $[DCBA] = [10]_{10} = 1010|_2$.

Et quand $[Q_d Q_c Q_b Q_a] = 12|_{10} = 1100|_2$ nous devons charger : $[DCBA] = [14]_{10} = 1110|_2$.

Observons que dans les 2 cas $D = 1, B = 1$ et $A = 0$, et :

$[Q_d Q_c Q_b Q_a] = 0110|_2 = 6|_{10}$ où $Q_d = 0$ et $[Q_d Q_c Q_b Q_a] = 1100|_2 = 12|_{10}$ où $Q_d = 1$ d'où : $C = Q_d$

Voir le schéma plus haut

5. Exercice 3 [6]

On souhaite fabriquer un plan mémoire à partir de circuits mémoire organisés en octet et contenant 2 Kilo octets par boîtier.

1) Indiquer combien de boîtiers sont nécessaires et réaliser le schéma de connexion de ces boîtiers (vous pouvez utiliser, si cela vous semble utile, des circuits logiques supplémentaires) pour obtenir une mémoire de 2 Kilo mots de 16 bits chacun. [1]

2K mots $\Rightarrow 2 \times 2^{10}$ mots = 2^{11} mots \Rightarrow 11 bits d'adresse

2K mots de 16 bits \Rightarrow 2 boîtiers 2K x 8bits.

Les 2 bus de données sont concaténés pour faire 2×8 bits = 16 bits.

2) Indiquer combien de boîtiers sont nécessaires et réaliser le schéma de connexion de ces boîtiers (vous pouvez utiliser, si cela vous semble utile, des circuits logiques supplémentaires) pour obtenir une mémoire de 8 Kilo octets. [2]

8K octets $\Rightarrow 4$ boîtiers x 2Ko $\Rightarrow 2 \times 2^{10} \Rightarrow$ 11 bits d'adresse / circuit.

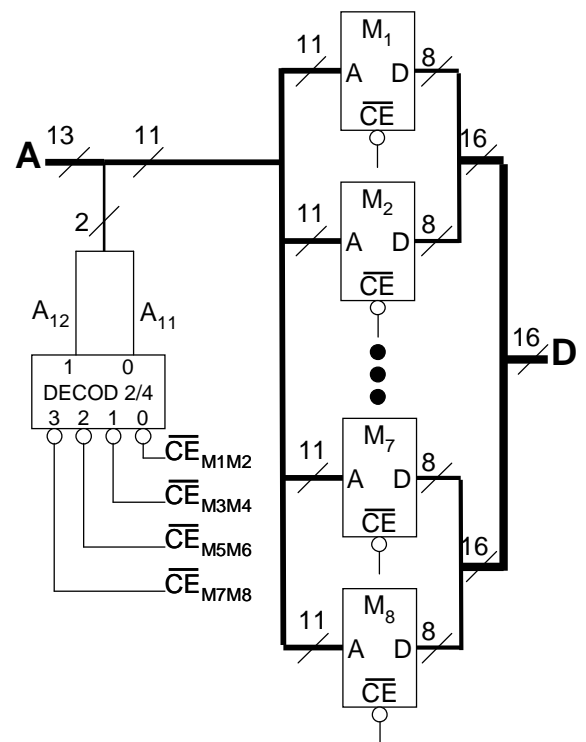
8K = $8 \times 2^{10} \Rightarrow$ 13 bits d'adresse :

$[A_{12} \rightarrow A_{11}]$ (2 bits d'@) servent à sélectionner un des 4 boîtiers (décodeur)

$[A_0 \rightarrow A_{10}]$ (11 bits d'@) sont connectées à tous les boîtiers.

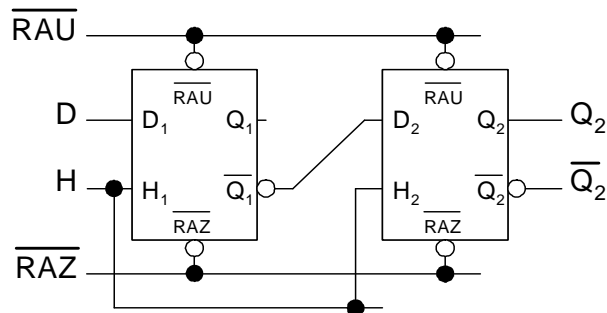
3) Indiquer combien de boîtiers sont nécessaires et réaliser le schéma de connexion de ces boîtiers (vous pouvez utiliser, si cela vous semble utile, des circuits logiques supplémentaires) pour obtenir une mémoire de 8 Kilo mots de 16 bits chacun. [3]

8K mots de 16 bits $\Rightarrow 4 \times 2$ boîtiers, (mélange des questions 1 et 2)



6. Exercice 4 [5]

Soit le circuit de la figure suivante.



1) *Quelle est la fonction de chacune des 2 entrées asynchrones ? Sur quel niveau logique sont elles actives ? [1]*

RAU : mise à «1», RAZ : mise à «0», actives mais jamais ensemble, sur niveau bas («0») :

RAU = 0 \Rightarrow $Q_1 = Q_2 = 1$

RAZ = 0 \Rightarrow $Q_1 = Q_2 = 0$

Les bascules D sont synchronisées sur niveau haut.

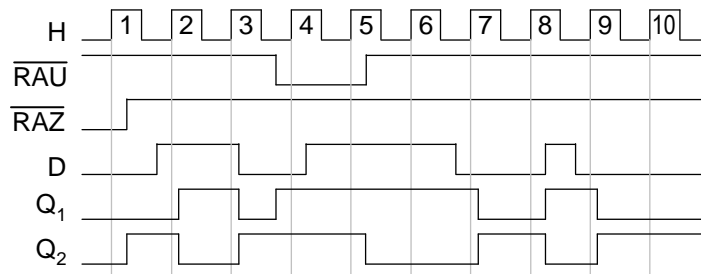
2) *Compléter le chronogramme suivant [2]*

Q_1 et Q_2 déclenchées sur le même signal H : circuit synchrone. $D_2 = \overline{Q_1}$.

Quand une entrée de forçage asynchrone devient inactive il faut attendre la prochaine impulsion d'horloge pour estimer le prochain état (mémorisation).

Tout changement de la donnée D pendant le niveau actif d'horloge (ici «1») est répercuté sur la sortie Q.

Mémorisation sur H = 0.



Les bascules D sont maintenant synchronisées sur front montant.

3) *Compléter le chronogramme suivant [2]*

Q_1 et Q_2 déclenchées sur le même front montant de H : circuit synchrone. $D_2 = \overline{Q_1}$.

Quand une entrée de forçage asynchrone devient inactive il faut attendre le prochain front d'horloge pour estimer le prochain état (mémorisation).

