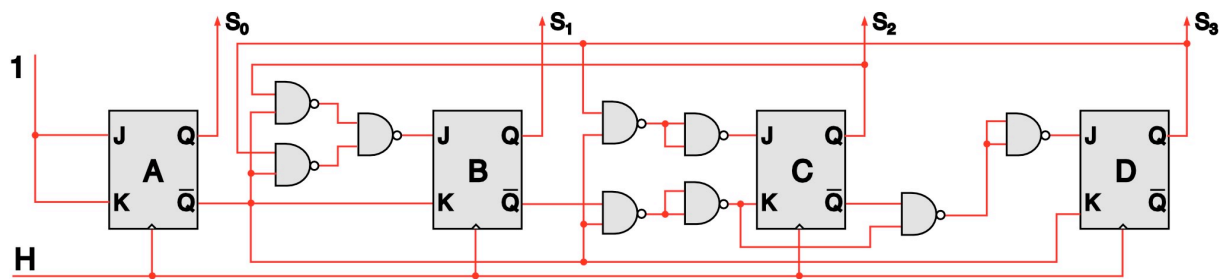


Structure des ordinateurs

Durée de l'épreuve : 3 heures. Documents autorisés.

Exercice 1 : Etude d'un circuit séquentiel synchrone.

Le circuit de la figure ci-dessous comporte 4 bascules JK, notées A, B, C, D et par souci d'homogénéité, seulement des portes NAND (\mathcal{G}). Ce circuit possède une seule entrée notée H, l'horloge. Les 4 sorties de ce circuit doivent représenter le code d'un chiffre en DCB (décimal codé binaire), où S_0 est le bit poids faible et S_3 est le bit poids fort.



Questions :

1. Analyser ce circuit (équations de commande des bascules, matrice et graphe des phases) et indiquer son fonctionnement normal.
2. On suppose un démarrage aléatoire du circuit (son état interne est quelconque au démarrage). Indiquer le nombre maximum de cycles d'horloge qu'il faut attendre pour que le circuit ait atteint son fonctionnement normal.
3. Ajoutez à ce circuit une entrée d'initialisation qui le force à prendre son fonctionnement normal lorsqu'elle est activée.

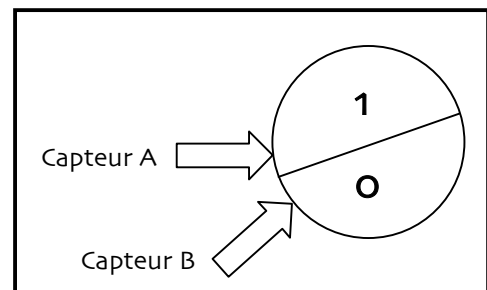
Exercice 2 : Synthèse d'un circuit séquentiel asynchrone

On désire fabriquer un circuit séquentiel asynchrone qui va déterminer le sens de rotation d'une roue.

Pour le réaliser, on dispose du montage ci contre. La sortie de chacun des capteurs est égale à la valeur indiquée dans le secteur avec lequel il est en contact.

Vous devez donc réaliser un circuit à deux entrées, A, B et une sortie S. La valeur de la sortie sera :

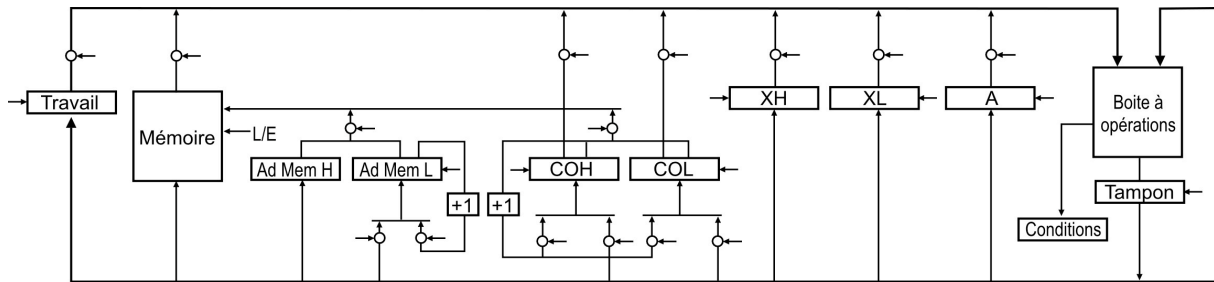
$S = 1$ le sens de rotation est celui des aiguilles d'une montre.
 $S = 0$ le sens de rotation est l'inverse de celui des aiguilles d'une montre.



Questions :

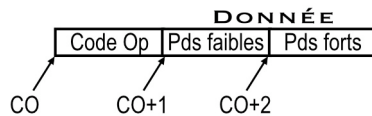
1. Quel est le sous graphe représentatif du sens de rotation horaire.
2. Quel est le sous graphe représentatif du sens de rotation anti-horaire.
3. Réunissez les deux graphes afin d'explicitier le fonctionnement global du circuit.
4. Tracez la matrice des phases correspondant au graphe ainsi obtenu.
5. Affectez les variables auxiliaires afin d'éviter toute course critique et déterminer les équations.

Exercice 3 : Langage de transfert



Voici la structure d'une machine comportant un bus de données (« celui qui fait le tour ») et un bus d'adresses recevant soit le compteur ordinal (COH/COL), soit une adresse quelconque stockée dans les registres AdMemH/AdMemL. Les seuls transferts d'information réalisables sont ceux rendus possibles par la structure de cette machine.

1. On constate que le seul moyen de passer de la partie haute du bus des données à sa partie basse s'opère au travers de la boîte à opérations.
2. Tous les composants de stockage (registres : Travail, COH/COL, XH/XL, A et la mémoire) délivrent leur contenu sur le bus des données au travers d'une barrière figurée par $\rightarrow\bigcirc$. Par contre, pour la plupart d'entre eux, ils sont chargés sur déclenchement d'un signal d'écriture représenté par \rightarrow .
3. Le compteur ordinal dispose d'un circuit combinatoire d'incrémentation (COL) avec gestion de report éventuel vers les poids forts (COH).
4. Les instructions machines sont codées sur un nombre d'octets variant de 1 à 3¹, le premier octet contenant le code de l'opération à effectuer. Dans le cas d'une instruction nécessitant une donnée codée sur deux octets, le second octet de l'instruction contiendra l'octet poids faible de la donnée et le troisième octet de l'instruction contiendra l'octet poids fort de la donnée.



5. Au moment d'exécuter une instruction, le compteur ordinal (CO) est supposé être positionné sur le premier octet de l'instruction à réaliser (qui contient le code opération). Un tronçonnage commun à toutes les instructions est exécuté :
 - $RI^2 \leftarrow (Mem(CO))$ Le registre RI reçoit le code
 - $CO^3 \leftarrow (CO)+1$ Le compteur ordinal est incrémenté (pour la suite)
 - Le code est ensuite analysé pour accéder à la portion décrivant, en terme de « langage de transfert », les actions à effectuer pour réaliser l'instruction.
6. Dans les écritures, les parenthèses ont un rôle important. Lorsqu'un composant est entre parenthèse, on fait référence à son contenu et non au composant lui-même. Par exemple en écrivant $CO \leftarrow (CO)+1$, on signifie que l'on ajoute 1 au contenu de CO, le résultant étant ensuite affecté au composant CO.
7. Chaque accès mémoire retourne un octet, mais les adresses sont codées sur deux octets !

Questions :

1. (Compréhension du cours) : en reprenant l'explication 2, indiquez l'utilité des barrières $\rightarrow\bigcirc$.
2. (Compréhension du problème) : dans l'explication 3, on fait état d'un montage sur CO permettant l'incrémentation avec report éventuel vers les poids forts. On constate que sur AdMem, on a aussi un circuit incrémenteur, mais seulement sur les poids faibles et sans gestion de report. Pouvez-vous dire pourquoi ce montage est ainsi fait et à quoi il sert ?
3. Décrire l'exécution du saut indexé indirect codé *JMIX Ad*, ayant pour fonction de réaliser $CO \leftarrow ((Mem(Ad))+(X))$.

¹ Afin de respecter les accès mémoire par adresses paires, dans le cas où une instruction contient un nombre impair d'octets, un octet complémentaire (inutilisé) sera prévu après l'instruction.

² Le registre RI n'apparaît pas dans le schéma de la machine par souci de simplification (il ne vous est d'aucune utilité).

³ Lorsqu'on parle d'un composant dans sa globalité, on assimile ses octets poids forts et poids faibles. Par exemple, on parlera de AdMem (regroupant AdMemH et AdMemL) ou de X (regroupant XH et XL)

CORRECTION

Exercice 1 :

1) L'évaluation des fonctions de commande ne pose pas de problèmes particuliers. Il suffit de "suivre les fils" et de connaître la loi de de Morgan...

$$J_A = 1 \text{ et } K_A = 1$$

$$J_B = \overline{\overline{Q_A} \cdot Q_C} \cdot \overline{\overline{Q_A} \cdot Q_D} = \overline{Q_A} \cdot Q_C + \overline{Q_A} \cdot Q_D$$

$$K_B = \overline{Q_A}$$

$$J_C = \overline{\overline{Q_A} \cdot Q_D} = \overline{Q_A} \cdot Q_D$$

$$K_C = \overline{\overline{Q_A} \cdot Q_B} = \overline{Q_A} \cdot \overline{Q_B}$$

$$J_D = \overline{\overline{Q_C} \cdot (\overline{Q_A} \cdot \overline{Q_B})} = \overline{Q_A} \cdot \overline{Q_B} \cdot \overline{Q_C}$$

$$K_D = \overline{Q_A}$$

Afin de dresser la matrice des états, j'ai choisi, pour une meilleure compréhension, de faire apparaître les valeurs des fonctions de commandes en regard de l'état de la bascule correspondante.

| | Etat Préc. | | | | Etat Suivant | | | | | | | | | | | | |
|----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----|
| | Q _D | Q _C | Q _B | Q _A | K _D | J _D | Q _D | K _C | J _C | Q _C | K _B | J _B | Q _B | K _A | J _A | Q _A | |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 9 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 2 |
| 4 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| 5 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 4 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 5 |
| 7 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 6 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7 |
| 9 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 8 |
| 10 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 5 |
| 11 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 10 |
| 12 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| 13 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 12 |
| 14 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 5 |
| 15 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 14 |

Pour déterminer l'état suivant, il faut se souvenir du fonctionnement de la bascule JK que l'on a vu en cours...

Si à temps précédent, $Q=0$ alors, Q prends la valeur 1 si J (et éventuellement K) =1. Il reste à 0 dans les autres cas.

Réciproquement, si $Q=1$, alors, Q prends la valeur 0 si K (et éventuellement J) =1. Il reste à 1 dans les autres cas.

2) J'ai nommé les divers états par leur valeur décimale. Le cycle le plus long traverse les états suivants :

$$9 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0 \rightarrow 9 \dots \text{ soit 10 étapes.}$$

Tous les états ne sont pas représentés dans ce cycles !

On a aussi des états d'entrée qui nécessitent des étapes supplémentaires avant de rejoindre le cycle:

$$10 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \dots, 14 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \dots, 12 \rightarrow 3 \rightarrow 2 \dots \text{ (1 étape),}$$

$$11 \rightarrow 10 \rightarrow 5 \rightarrow 4 \rightarrow 3 \dots, 15 \rightarrow 14 \rightarrow 5 \rightarrow 4 \rightarrow 3 \dots, 13 \rightarrow 12 \rightarrow 3 \rightarrow 2 \dots \text{ (2 étapes)}$$

Le but de la question 3 est de forcer à rejoindre le cycle dès qu'une entrée supplémentaire I est activée.

3) Tout d'abord, choisissons un état du cycle vers lequel, quel que soit l'état dans lequel on est, le cycle va être amorcé, par exemple, l'état 0.

Cet état correspond à une configuration de $Q_A Q_B Q_C Q_D = 0000$.

D'après les rappels précédents sur le fonctionnement de la bascule JK, il faut donc que $J_X = 0$ et $K_X = 1$, $\forall X \in \{A, B, C, D\}$. Par ailleurs, il faut que lorsque I retombera à 0, les fonctions de commande fonctionnent normalement.

Par exemple pour $I = 0$, J_A doit être égal à 1 (valeur prévue dans le montage initial) et pour $I = 1$, J_A doit être égal à 0 (conclusion précédente). Donc finalement, $J_A = \bar{I}$. K_A , lui, doit être égal à 1 dans les deux cas.

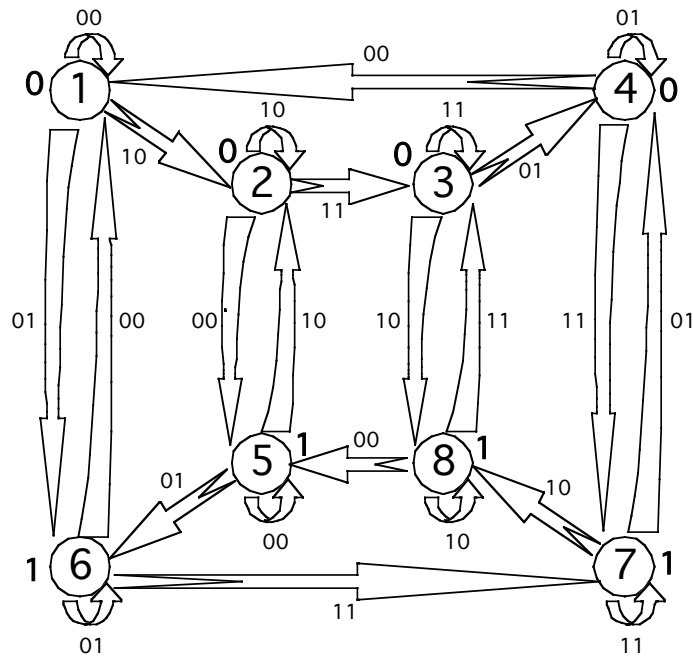
Autre exemple, K_B doit être égal à 1 si $I = 1$, et à \bar{Q}_A si $I = 0$. Donc $K_B = \bar{Q}_A \bar{I} + I$.

En définitive, on obtient :

| | | | |
|------------------------------|--|--|--|
| $J_A = \bar{I}$ et $K_A = 1$ | $J_B = (\bar{Q}_A \cdot Q_C + \bar{Q}_A \cdot Q_D) \cdot \bar{I}$ $K_B = \bar{Q}_A \bar{I} + I$ | $J_C = (\bar{Q}_A \cdot Q_D) \cdot \bar{I}$ $K_C = (\bar{Q}_A \cdot \bar{Q}_B) + I$ | $J_D = (\bar{Q}_A \cdot \bar{Q}_B \cdot \bar{Q}_C) \cdot \bar{I}$ $K_D = \bar{Q}_A \bar{I} + I$ |
|------------------------------|--|--|--|

Je vous conseille de chercher une autre solution en supposant, par exemple, que l'entrée dans le cycle s'opère non plus par 0, mais, par exemple, par 5.

Exercice 2 :



Affectation des variables auxiliaires :

| | |
|---|-----|
| 1 | 000 |
| 2 | 001 |
| 3 | 011 |
| 4 | 010 |
| 5 | 101 |
| 6 | 100 |
| 7 | 110 |
| 8 | 111 |

Exercice 3 :

Tampon \leftarrow (Mem(CO))

Tampon \leftarrow (Tampon) + (XL)

Travail \leftarrow (Tampon)

CO \leftarrow (CO) + 1

Tampon \leftarrow (Mem(CO))

Tampon \leftarrow (Tampon) + (XH) + r

AdMemH \leftarrow (Tampon)

Tampon \leftarrow (Travail)

AdMemL \leftarrow (Tampon)

Tampon \leftarrow (Mem(AdMem))

COL \leftarrow Tampon

AdMemL \leftarrow (AdMemL) + 1

Tampon \leftarrow (Mem(AdMem))

COH \leftarrow (Tampon)