

Objectif : Codages one hot et moore des FSM. Application aux séquenceurs de moteur pas à pas.

I. Introduction

Les circuits séquentiels sont des circuits dont les sorties dépendent des entrées mais également d'un état interne. La particularité des systèmes synchrones est de synchroniser l'évolution des éléments mémorisant l'état courant sur une source d'impulsions appelée horloge. La synthèse des systèmes séquentiels consiste à déterminer le bloc diagramme d'un circuit à partir d'une description comportementale telle qu'une machine à état finie. Le codage des états en vue de leur implémentation est une étape influant de manière sensible sur la solution obtenue. Dans cette manipulation, on se propose d'illustrer les techniques de conception des systèmes synchrones pour le contrôle d'un moteur.

II. L'application

Un **moteur pas à pas** permet de transformer une impulsion électrique en un mouvement angulaire. Ce type de moteur est très courant dans tous les dispositifs où l'on souhaite faire du contrôle de vitesse ou de position en boucle ouverte, typiquement dans des systèmes tel que le positionnement de la tête d'impression dans les imprimantes matricielles, à marguerite et à jet d'encre ou encore la rotation du rouleau porte-papier (qui concerne également les imprimantes laser). Un moteur pas à pas fonctionne suivant un principe identique au moteur synchrone.

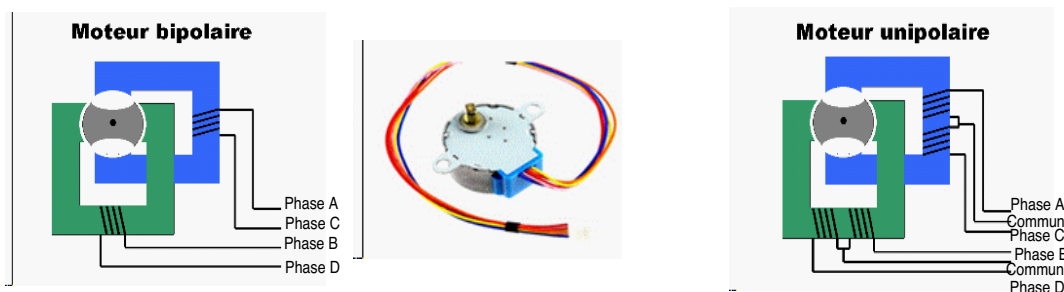


Figure 1: Vue externe et interne d'un moteur pas à pas unipolaire ou bipolaire

Ces moteurs sont composés d'un barreau de fer doux et d'un certain nombre de bobines. Lorsqu'on alimente une bobine, elle se comporte comme un électro-aimant et le barreau de fer cherche naturellement à s'orienter suivant le champ électrique. Plongé dans un champ magnétique créé par le stator (bobine), le rotor tend alors à s'aligner sur ce champ (un écart crée un couple de rappel). Le rotor peut être constitué par un aimant permanent, ce qui lui permet de rester sur sa dernière position, même en l'absence d'excitation. Les moteurs à aimant continu sont sensiblement similaires aux moteurs à reluctance variable exceptés qu'ils sont dotés des pôles Nord et Sud (d'où le terme aimanté). Il existe également des moteurs hybrides combinant reluctance variable et aimant permanent. La rotation du moteur s'obtient alors en alimentant successivement une ou deux bobines du moteur (tension de 3,3 à 48V selon le type de moteur). Selon la catégorie du moteur (bipolaire ou unipolaire avec présence d'un point milieu), les modes nécessitent 2 ou 4 phases avec différents modes de fonctionnement possibles : monophasé, biphasé, $\frac{1}{2}$ pas. Pour la suite, on s'intéresse au fonctionnement d'un moteur unipolaire. On notera que la consommation relativement importante du moteur pas à pas (de 0,2 à 1,5 A) nécessite une interface de puissance intercalée entre le circuit de commande que nous allons réaliser et le moteur.

III. Préparation au LAB

LES REPONSES AUX QUESTIONS DE CETTE PARTIE SERONT VERIFIEES ET EVALUEES EN DEBUT DE SEANCE.

1^{ème} partie : codage one-hot d'une FSM

Pour illustrer le codage one-hot d'une FSM, on se propose d'implémenter une commande du moteur de type monophasée. Dans ce type de fonctionnement, on alimente une des quatre bobines en séquence pour obtenir les champs électriques successifs dans l'ordre pas1 → pas4 (sens horaire) ou pas4 → pas1 (sens anti-horaire) selon le sens de rotation souhaité (cf figure 2).

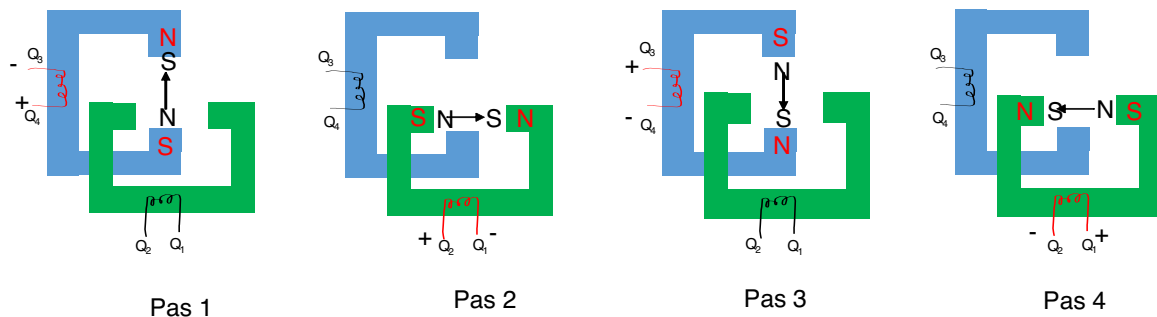


Figure 2: Séquence d'alimentation des bobines (alimentation alternée d'une seule bobine)

La table ci-dessous montre la commande à appliquer aux bobines selon le sens de rotation souhaité.

Sens horaire (S=1)				Sens anti-horaire (S=0)			
Q _A	Q _B	Q _C	Q _D	Q _A	Q _B	Q _C	Q _D
1	0	0	0	0	0	0	1
0	0	1	0	0	1	0	0
0	1	0	0	0	0	1	0
0	0	0	1	1	0	0	0

Tab1e 1 : Commandes à appliquer aux bobines du moteur

Comme le montre la Figure 3, un moteur pas à pas est constitué de plusieurs rotors et stators avec une alternance de pôles Nord et Sud. La figure ci-dessous montre l'application des 3 premières phases sur un moteur « déplié ». Pour obtenir une rotation d'un tour pour un moteur à 48 pas, il suffit d'appliquer 12 fois la séquence d'alimentation souhaitée.

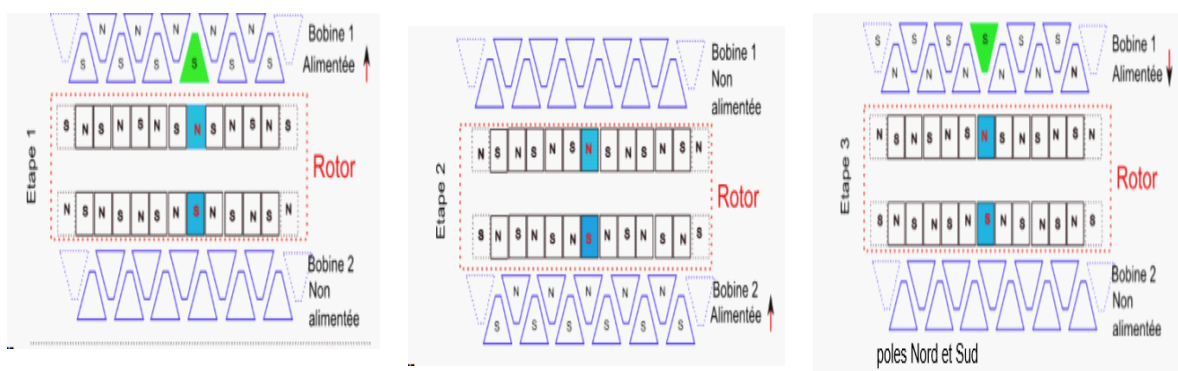


Figure 3: Répétition de la séquence pour une rotation complète

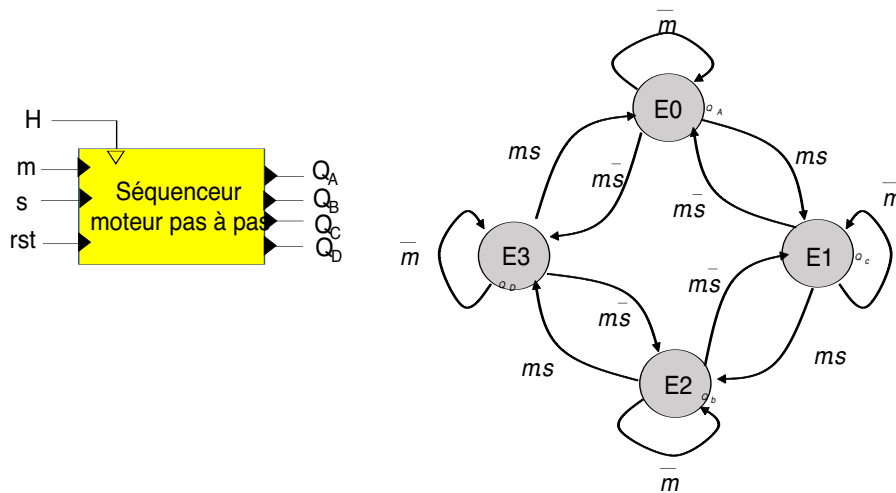


Figure 4: Diagramme des états du mode pas à pas

La figure ci-dessus montre le symbole du circuit à concevoir avec les signaux en entrée/sortie. Le signal **H** définit la fréquence de fonctionnement du moteur. On notera que la fréquence d'horloge est limitée à quelques centaines de Hertz. Le signal **M** permet de stopper (0) ou autoriser la rotation (1) du moteur. **S** définit le sens (1: sens horaire, 0 sens anti-horaire). En sortie, les quatre signaux **QA**, **QB**, **QC** et **QD** désignent les signaux d'alimentation du moteur (4 phases).

La figure ci-dessous montre une machine à état caractérisant les combinaisons de sorties et d'état suivant attendues pour les différentes combinaisons d'entrée et d'état courant. Sur cette figure, on ne représente que la phase mise à 1. Par défaut, les phases non mentionnées sont considérées à 0.

Questions : A Proposez un code one-hot encore appelé codage une bascule par état de la machine à états

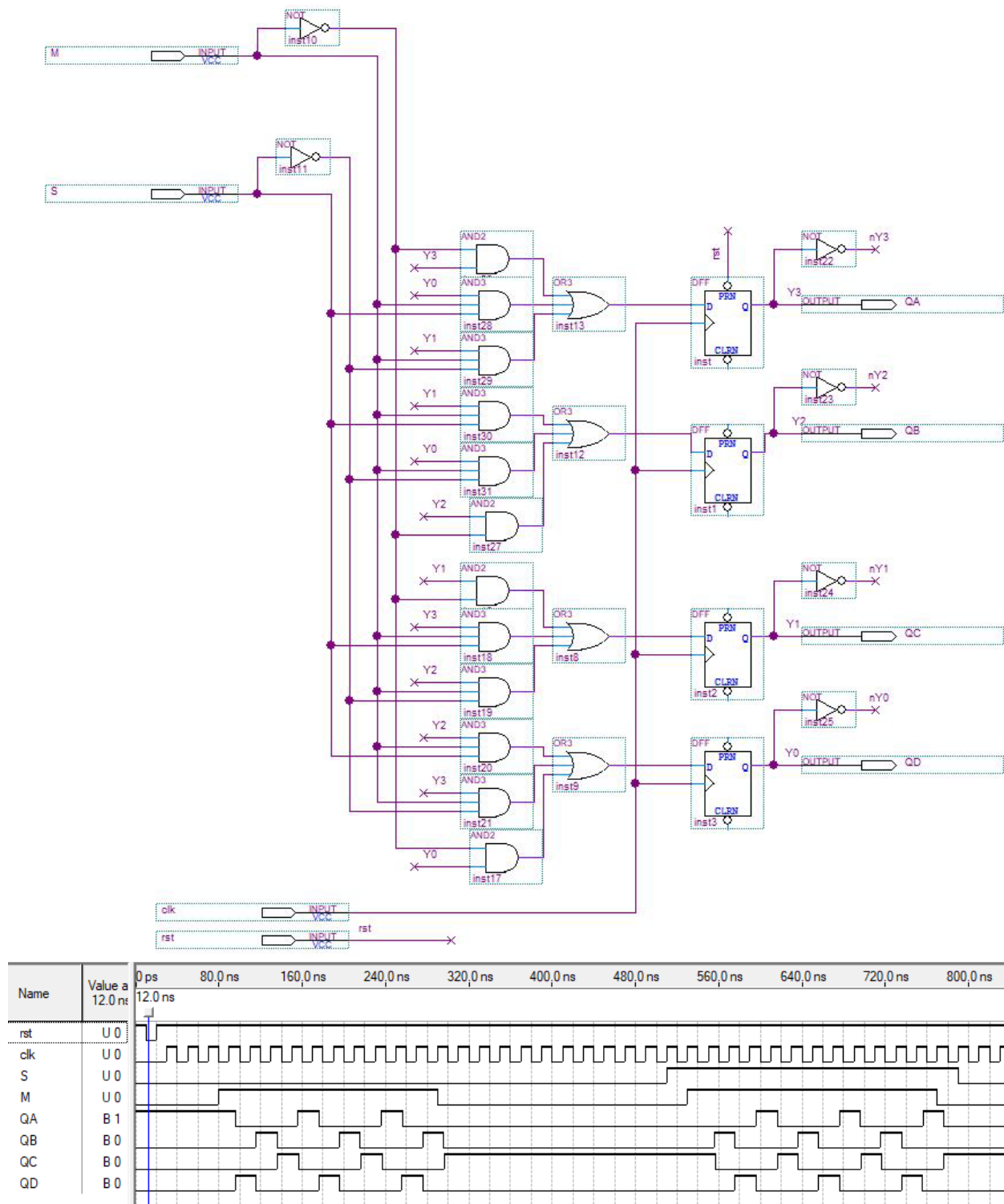
E0	1000
E1	0010
E2	0100
E3	0001

On a $QA = Y3$ $QC = Y1$ $QB = Y2$ $QD = Y0$

B. A partir de ce codage et du diagramme des transitions associé, déduisez directement les bloc diagramme du circuit en utilisant 4 FFD et des portes logiques. Un signal *rst* sera ajouté à l'entrée *prn* de la FF de l'état initial et sur les entrées *clr* pour les autres.

$Y3+ Y2+ Y1+ Y0+$ $Y3 Y2 Y1 Y0 \setminus M S$	00	01	11	10
1000	1000	1000	0010	0001
0010	0010	0010	0100	1000
0100	0100	0100	0001	0010
0001	0001	0001	1000	0100

$$\begin{aligned}
 Y3+ &= Y3./M + Y0.M.S + Y1.M./S & Y2+ &= Y2./M + Y1.M.S + Y0.M./S \\
 Y1+ &= Y1./M + Y3.M.S + Y2.M./S & Y0+ &= Y0./M + Y2.M.S + Y3.M./S
 \end{aligned}$$

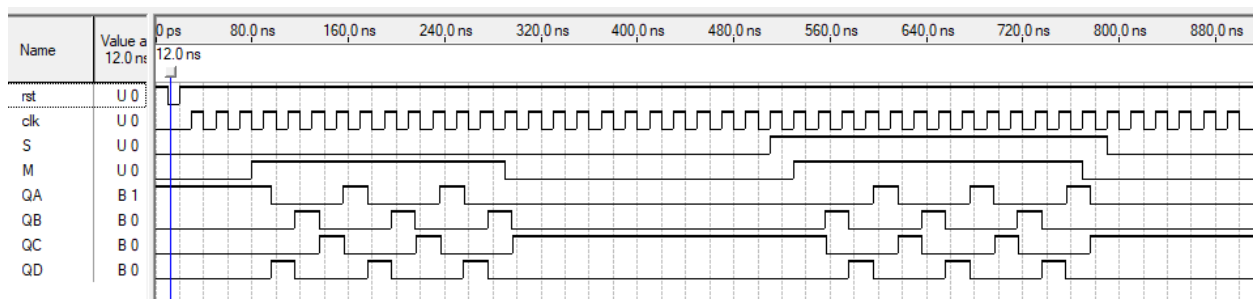
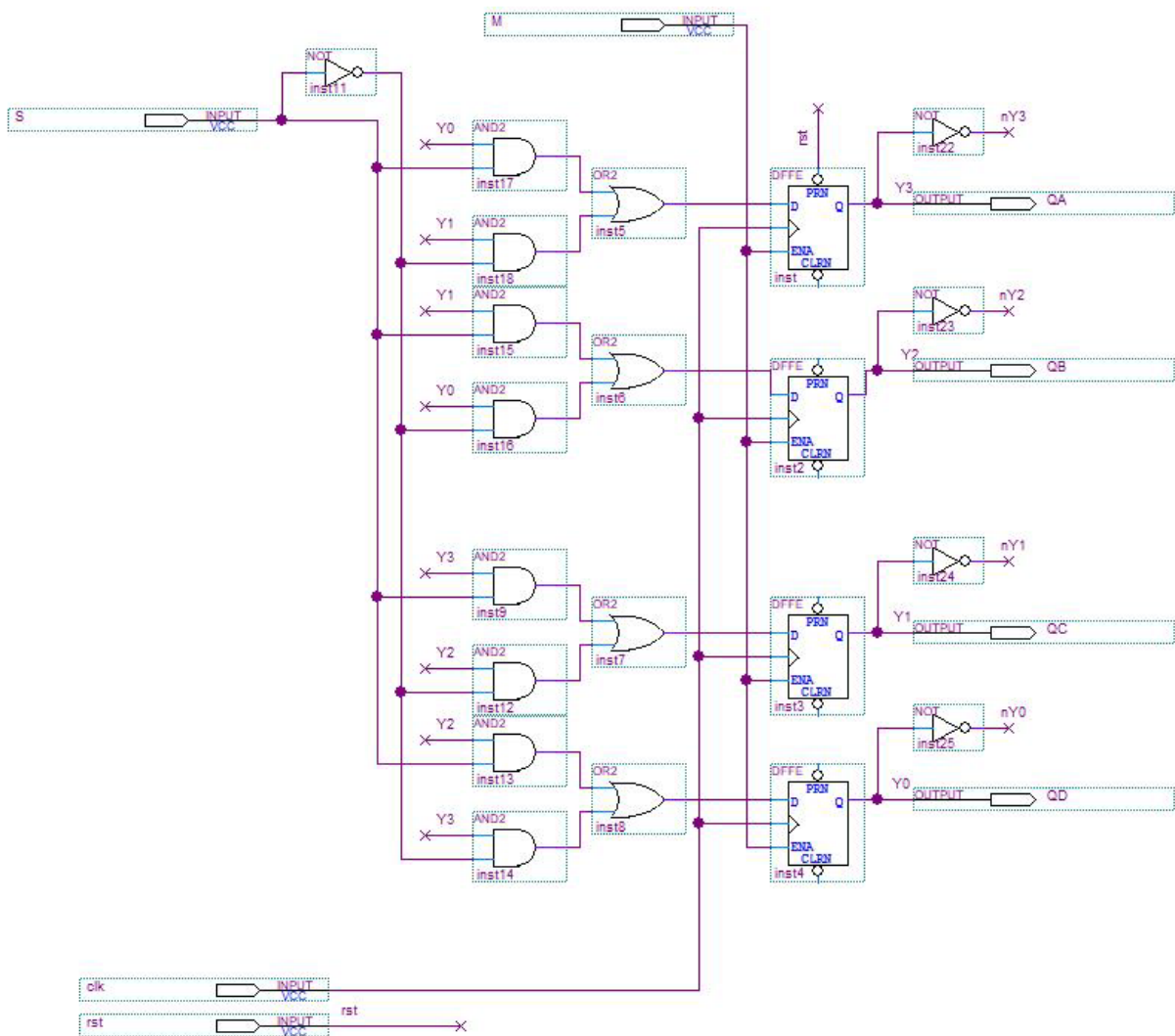


C. Modifiez le schéma en utilisant des DFFE (FF avec entrée enable). L'entrée *ena* sera raccordée au signal M.

Les nouvelles équations sont désormais :

$$Y3+ = Y0.S + Y1./S \quad Y2+ = Y1.S + Y0./S$$

$$Y1+ = Y3.S + Y2./S \quad Y0+ = Y2.S + Y3./S$$



2^{ème} partie : codage de Moore d'une FSM

On s'intéresse à présent au codage de Moore (encore appelé codage par les sorties) que l'on souhaite appliquer pour la commande biphasée du moteur. Dans ce mode, un meilleur couple (de 5 à 64 N.cm en général) peut être obtenu par l'alimentation simultanée de deux bobines à chaque pas (cf. figure 3). Le rotor se cale entre les deux stators.

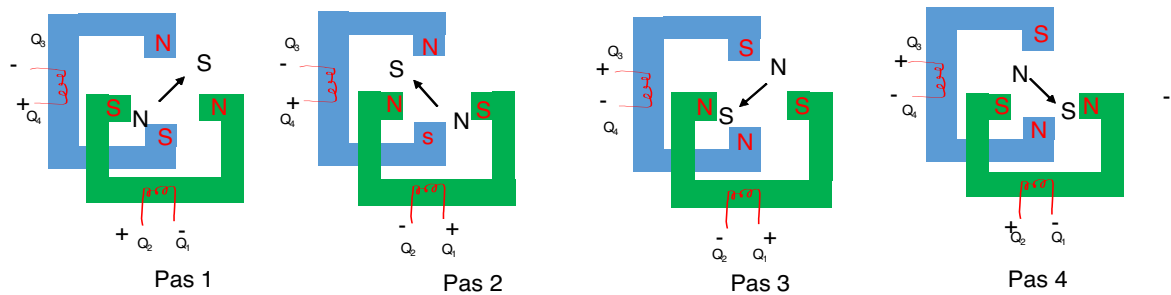


Figure 3: Séquence d'alimentation des bobines en mode couple maximal

Ce mode de fonctionnement suppose une modification des commandes à appliquer (cf. table 2).

Sens horaire (S=1)				Sens anti-horaire (S=0)			
Q _A	Q _B	Q _C	Q _D	Q _A	Q _B	Q _C	Q _D
1	0	1	0	0	1	1	0
1	0	0	1	0	1	0	1
0	1	0	1	1	0	0	1
0	1	1	0	1	0	1	0

Table 2 : Commandes à appliquer aux bobines du moteur

Pour implémenter la commande d'un moteur pas à pas en mode biphasé, on se propose de définir une solution d'implémentation à partir d'un codage de Moore. La figure ci-dessous donne un diagramme à états finis permettant d'assurer la rotation du moteur en mode pas et couple moteur renforcé :

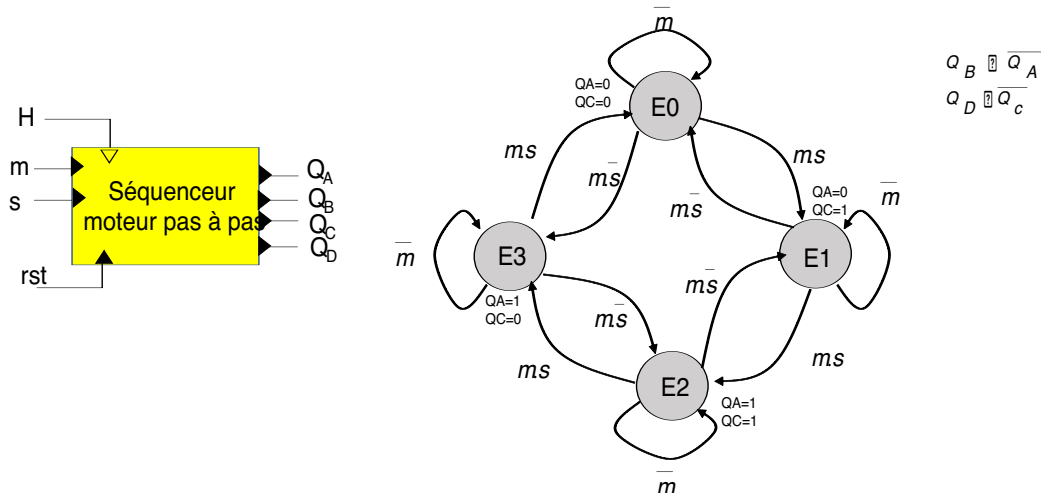


Figure 5: Spécification du comportement du séquenceur

Dans ce mode de rotation, le signal Qb est le complément de Qa à chaque phase. Même chose pour Qc et Qd. On peut donc spécifier le comportement du circuit sur le diagramme à états finis pour 2 sorties uniquement et définir les 2 autres sorties à partir d'opérateurs complément.

Questions : A. Définissez un codage des états à partir des sorties (codage de Moore). En déduire les équations des sorties induites par ce codage.

E : Y1 Y0 E0 = 11 E1 = 10 E2 = 00 E3 = 01
 QA = Y1 QB = /Y1 QC = Y0 QD = /Y0

B. Dressez la table de transition du circuit correspondante. L'entrée m ne sera pas considérée.

Etat+ Etat \ S	0	1
11	01	10
10	11	00
00	10	01
01	00	11

C. Exprimez la table d'excitation du circuit en utilisant une technologie de bascule JK

J1K1 J0K0	0	1	0	1
Etat \ S				
11	01	10	X1 X0	X0 X1
10	11	00	X0 1X	X1 0X
00	10	01	1X 0X	0X 1X
01	00	11	0X X1	1X X0

Transition	J	K
0 → 0	0	X
1 → 1	X	0
1 → 0	X	1
0 → 1	1	X

D. Donnez les équations d'entrées des 2 bascules JK.

On met dans l'ordre d'une table de Karnaugh

J1K1 J0K0	0	1	0	1
Etat \ S				
00	10	01	1X 0X	0X 1X
01	00	11	0X X1	1X X0
11	01	10	X1 X0	X0 X1
10	11	00	X0 1X	X1 0X

J1	0	1
Etat \ S		
00	1	0
01	0	1
11	X	X
10	X	X

$$J1 = /Y0./S + Y0.S$$

J0	0	1
Etat \ S		
00	0	1
01	X	X
11	X	X
10	1	0

$$J0 = /Y1.S + Y1./S$$

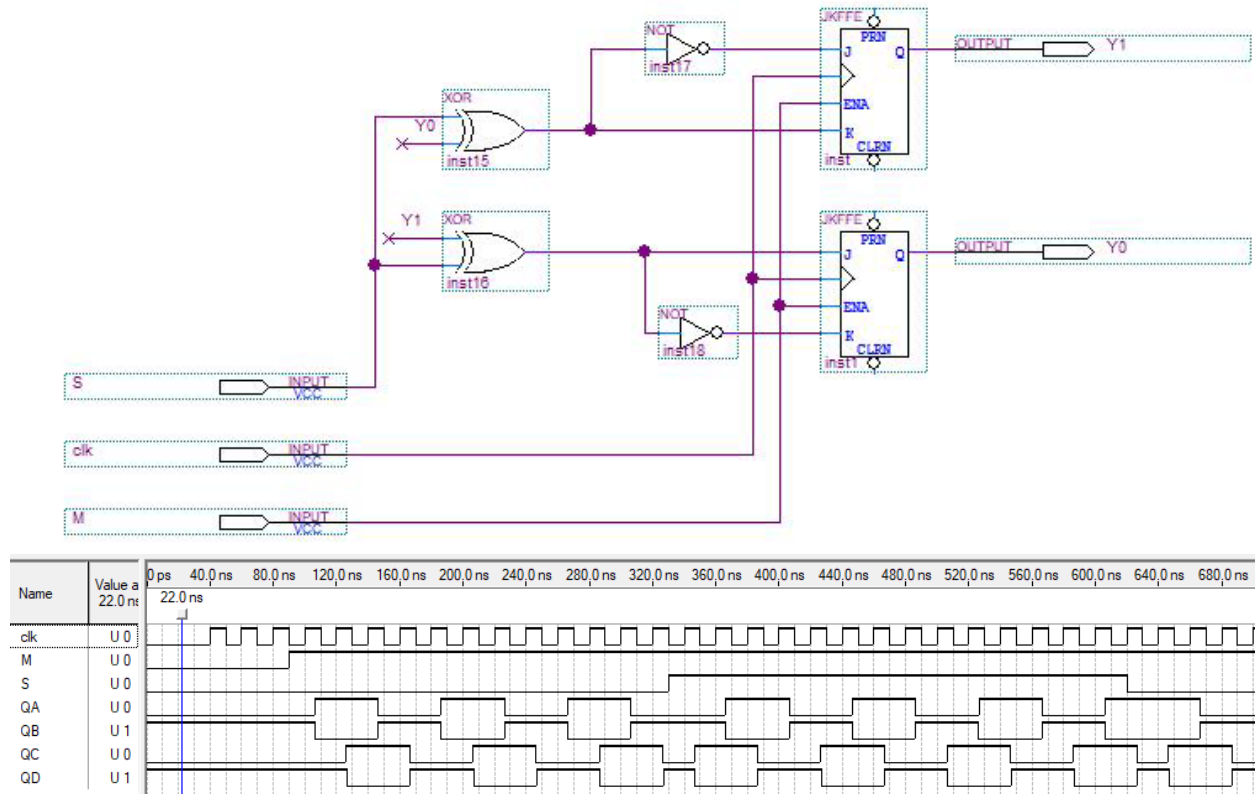
K1	0	1
Etat \ S		
00	X	X
01	X	X
11	1	0
10	0	1

$$K1 = Y0./S + /Y0.S$$

K0	0	1
Etat \ S		
00	X	X
01	1	0
11	0	1
10	X	X

$$K0 = /Y1./S + Y1.S$$

E. Dressez le bloc diagramme avec une implémentation de la logique combinatoire à partir de portes logiques et de flip flop JK avec enable. Ajoutez l'entrée *m* sur l'entrée *ena*.



IV Travail en séance

CHAQUE PARTIE DOIT ETRE VALIDEE PAR L'ENSEIGNANT AU COURS DE LA SEANCE.

1^{ère} partie : Validation fonctionnelle de la solution monophasée

- A. Créez un projet sous Quartus II que vous intitulerez *TP1*. Pour cela, cliquez sur *File* ➤ *New Project Wizard*. Dans la fenêtre qui apparaît, saisissez le chemin, le nom de projet et le nom du design (les noms doivent être de préférence identiques). Cliquez ensuite sur Next.

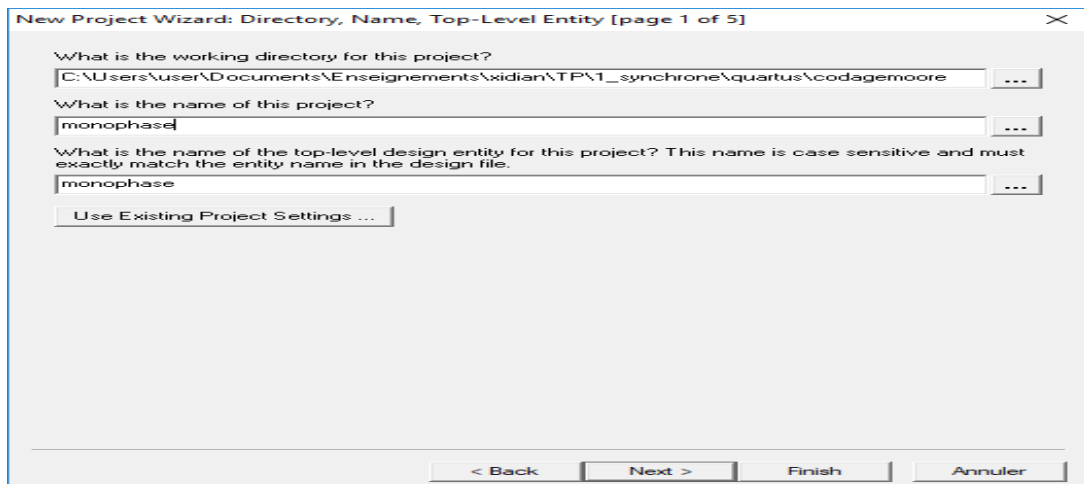


Figure 6: Fenêtre de création du projet

Une nouvelle fenêtre vous demande si vous souhaitez inclure des fichiers de conception dans le projet. Cliquez directement sur *next*. La fenêtre suivante vous propose de sélectionner la famille et le type de composant.

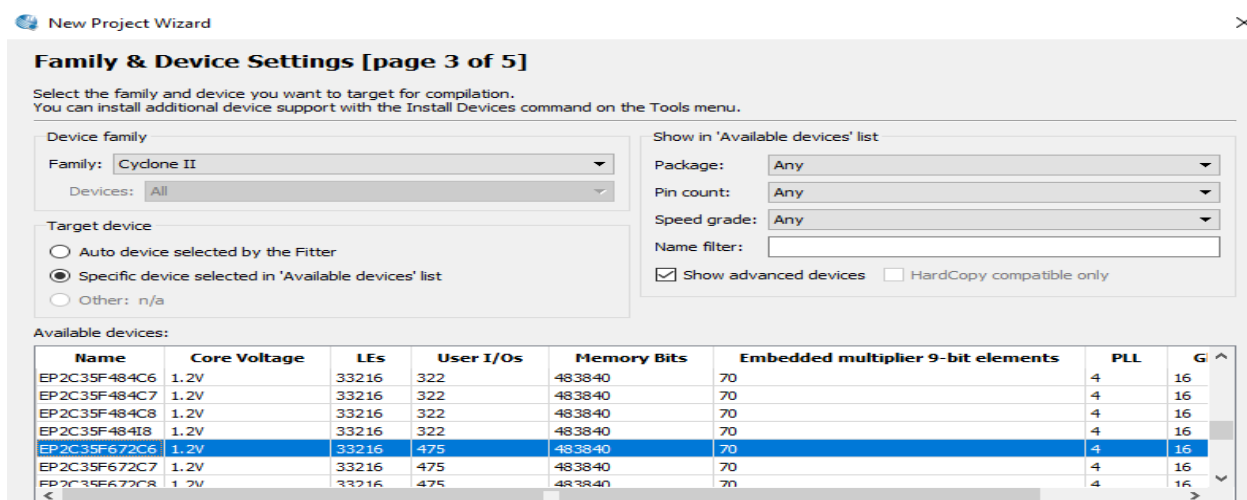


Figure 7: Saisie de la famille et du type de composant

Sélectionnez la famille *CYCLONEII* ainsi que le composant (cf. Figure 7). Cliquez ensuite sur *next* par deux fois puis sur *finish* dans la fenêtre de récapitulation des informations du projet, si tout est correct.

- B. Ouvrez ensuite un fichier de saisie schématique en cliquant sur *file* ➤ *new* puis *bloc diagram*

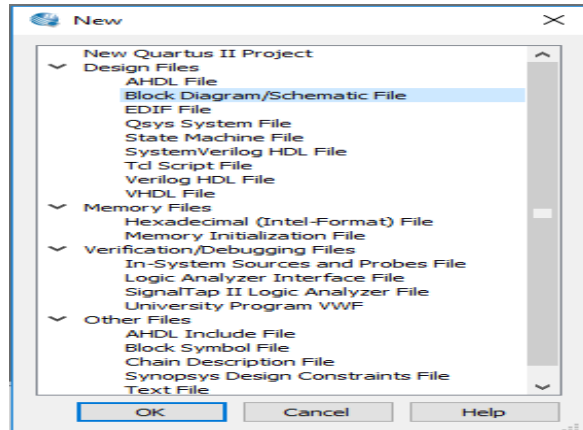


Figure 8: création d'un fichier de saisie du schéma

Procédez à la saisie du schéma du bloc diagramme issu du codage de one-hot sur l'outil quartus II. Pour cela, double cliquez sur le fond de la fenêtre. La bibliothèque de composants disponibles sous Quartus s'ouvre à vous. Sélectionnez dans les *primitives* les différentes portes (sous-répertoire *logic*) et FF (sous-répertoire *storage*) qui vous intéressent. Sauvegardez votre saisie sous l'appellation *monophase.bdf*. puis demandez l'inclusion du fichier dans le projet (menu *project* → *add current file to project*)

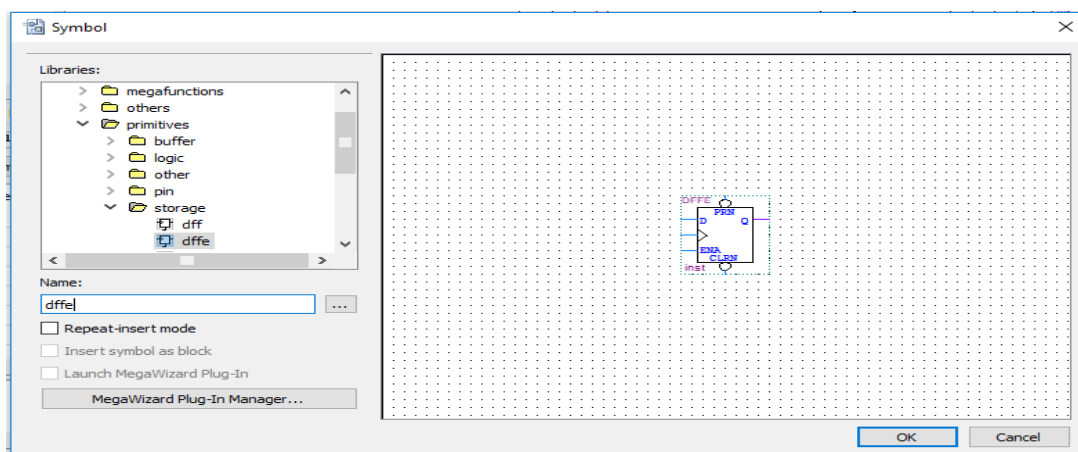


Figure 9: Sélection d'un composant dans la bibliothèque des primitives.

- C. Définissez le fichier en top-level (menu *project* → *set as top-level entity*). Compilez votre projet en cliquant *tools* → *start compilation* (ou l'icône de la barre de menu au choix).

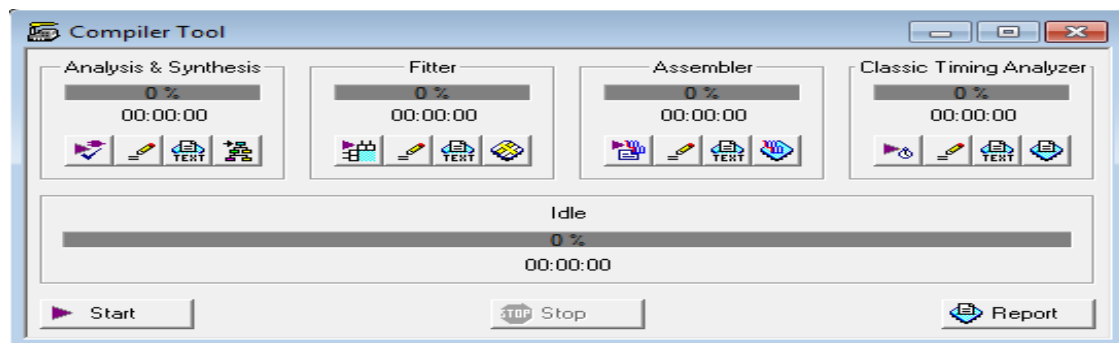


Figure 10: Compilation du projet

- D. Ouvrez un nouveau fichier de type *waveform* et procédez à la saisie du scénario de test sur le modèle ci-dessous. Sauvegardez sous *TP1.vwf*. Dans le menu *Assignments* → *settings*, définissez le

fichier comme fichier à utiliser pour la simulation.

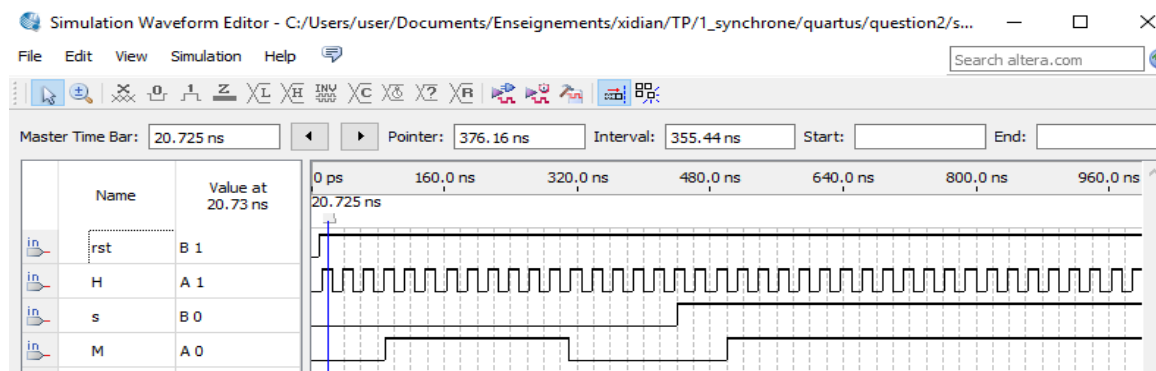


Figure 11: Création d'un scénario de simulation

E. Validez votre circuit (*processing* *start simulation*) en comparant avec le résultat ci-dessous :

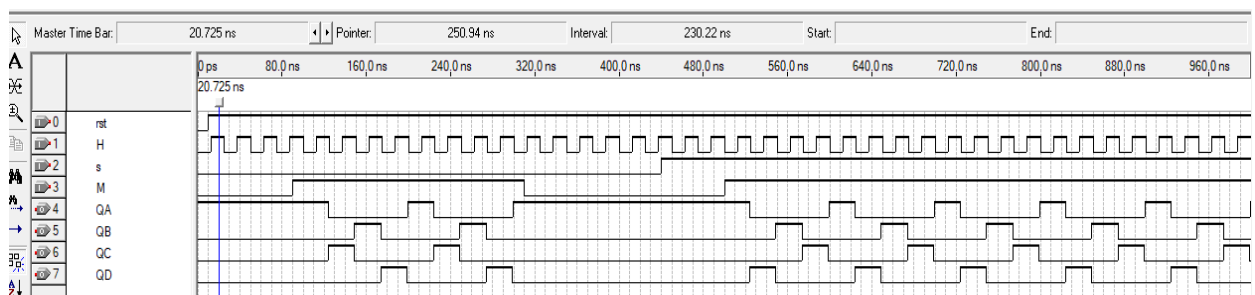


Figure 12: chronogramme du fonctionnement mode monophasé

2^{ème} partie : Validation fonctionnelle de la solution biphasée

- Dans un nouveau projet Quartus, effectuez la saisie du bloc diagramme issu du codage de Moore, schéma que vous nommerez *biphase* Compilez puis incluez (*file* *add/remove files*) le fichier de test créé lors de la 1^{ère} partie.
- Validez fonctionnellement votre solution.

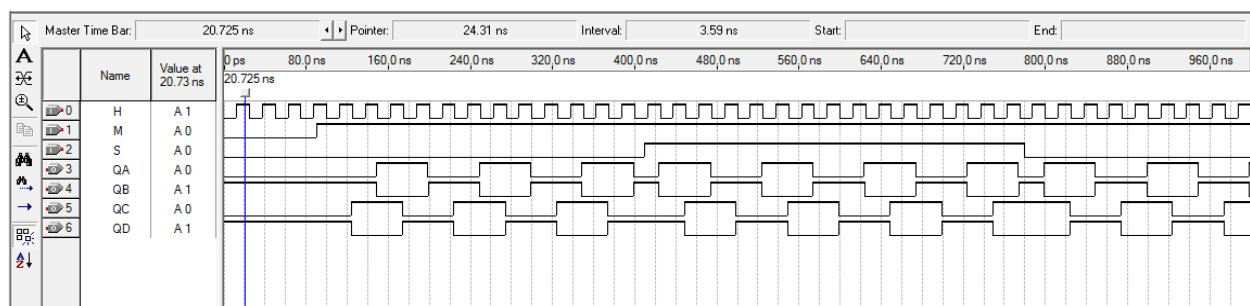


Figure 13; Chronogrammes pour la solution mode biphasé

3^{ème} partie : Test du moteur pas à pas

- Dans le projet courant, ajoutez un composant *compteur* en entrée de l'horloge. Pour cela, cliquez sur la bibliothèque des *mégafonctions*. Dans la fenêtre de paramétrage, définissez la taille à 25 bits. Connectez l'entrée d'horloge sur *H* et la sortie de la sortie 24 du compteur sur l'entrée d'horloge des FF.

- B. Procédez ensuite à l'affectation des pattes du circuit menu (*assignment* → *pin planner*). Cliquez sur la colonne *location* et entrez le numéro de la patte en respectant les conventions ci-dessous :

	Node Name	Direction	Location	I/O Bank	VREF Group	Reserved	Group	PCB layer	I/O Standard	Current Strength
1	H	Input	PIN_N2	2	B2_N1				3.3-V LVTTTL (default)	24mA (default)
2	M	Input	PIN_N25	5	B5_N1				3.3-V LVTTTL (default)	24mA (default)
3	Q1	Output	PIN_O25	5	B5_N0				3.3-V LVTTTL (default)	24mA (default)
4	Q2	Output	PIN_J22	5	B5_N0				3.3-V LVTTTL (default)	24mA (default)
5	Q3	Output	PIN_E26	5	B5_N0				3.3-V LVTTTL (default)	24mA (default)
6	Q4	Output	PIN_E25	5	B5_N0				3.3-V LVTTTL (default)	24mA (default)
7	S	Input	PIN_N26	5	B5_N1				3.3-V LVTTTL (default)	24mA (default)
8	<new node>>									

Figure 14: Affectation des pattes du composant

Cette table indique par exemple de relier l'horloge à l'oscillateur 50 Mhz présent sur la carte et disponible sur la patte N2. Les sorties sont reliées à des pattes du bornier de la carte. Recompilez ensuite votre projet.

- C. Branchez la carte *FPGA DE2* à base de *CycloneII* à l'aide de fiches bananes et vous en s'aidant du schéma ci-dessous.

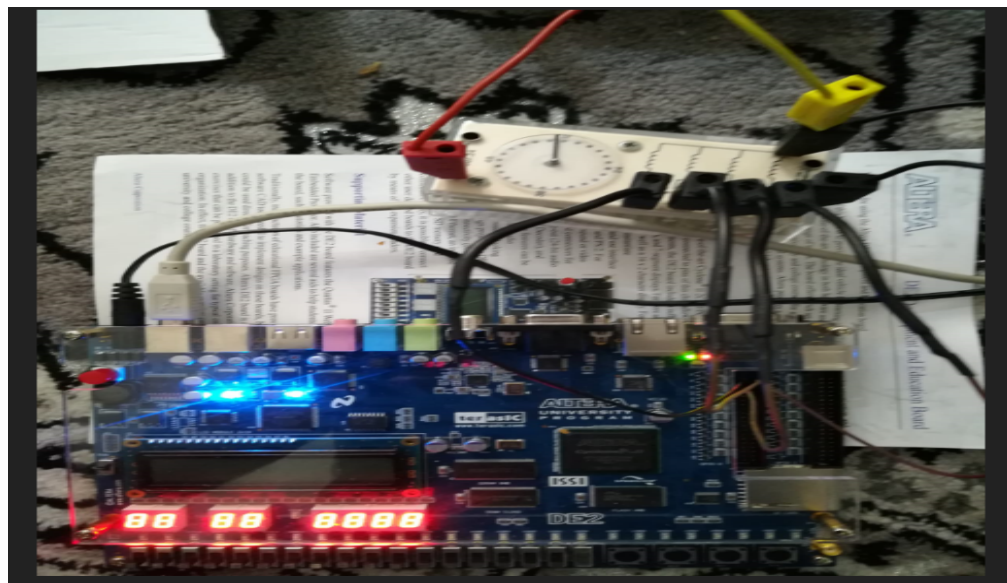


Figure 15: montage pour le test du mode biphasé

- D. Procédez à la programmation du FPGA avec la commande *tools* → *programmer*.
- E. Positionnez le bouton *M* (SW0) à marche et observez le fonctionnement du moteur en faisant varier le sens *S* (SW1).