

Étude de laboratoire

SLEA

Binôme A11
SIMON Léo, LEVY-FALK Hugo
Supélec, promo 2020

30 janvier 2018

Table des matières

I	Objectifs de ce TL	2
II	Étude préliminaire	3
1	Saisie du schéma du montage	3
2	Affectation des entrées et sorties	3
3	Définition des vecteurs de simulation	3
4	Simulation fonctionnelle	3
5	Simulation temporelle	4
6	Analyse temporelle	4
7	Programmation	4
III	Traitement d'un bouton poussoir	6
8	Synthèse avec une machine de MOORE	6
9	Synthèse avec une machine de MEALY	8
IV	Diviseur de fréquence	10
10	Structure du diviseur	10
11	Détermination du premier rapport de division	10
12	Compteur	10
13	Diviseur par 2	11
14	Réalisation du diviseur	11
V	Partie opérative et séquenceur	13
15	Chemin de données	13
16	Séquenceur	14
VI	Réalisation du chronomètre	16
VII	Conclusion	16

Première partie

Objectifs de ce TL

L'objectif de cette étude de laboratoire est la réalisation d'un chronomètre basique. Elle se découpe entre une étude préliminaire et la réalisation à proprement parler du chronomètre.

L'étude préliminaire permet de prendre en main le logiciel qui permettra la réalisation du chronomètre.

On dispose d'un cahier des charges pour le chronomètre. Celui-ci est constitué entre autre de deux boutons poussoirs et de deux afficheurs 7 segments. On souhaite que :

- Chaque appui sur **START/STOP** permette de commuter entre le mode comptage (affichage du temps sur deux digits : secondes et dixièmes de seconde) et le mode arrêt (affichage du temps mémorisé) ;
- Un appui sur **RESET** remette l'afficheur à zéro. Cette fonction doit pouvoir être activée à tout moment et maintenir l'affichage à zéro.

Pour réaliser cela, on propose un découpage tel que montré par la figure 1.

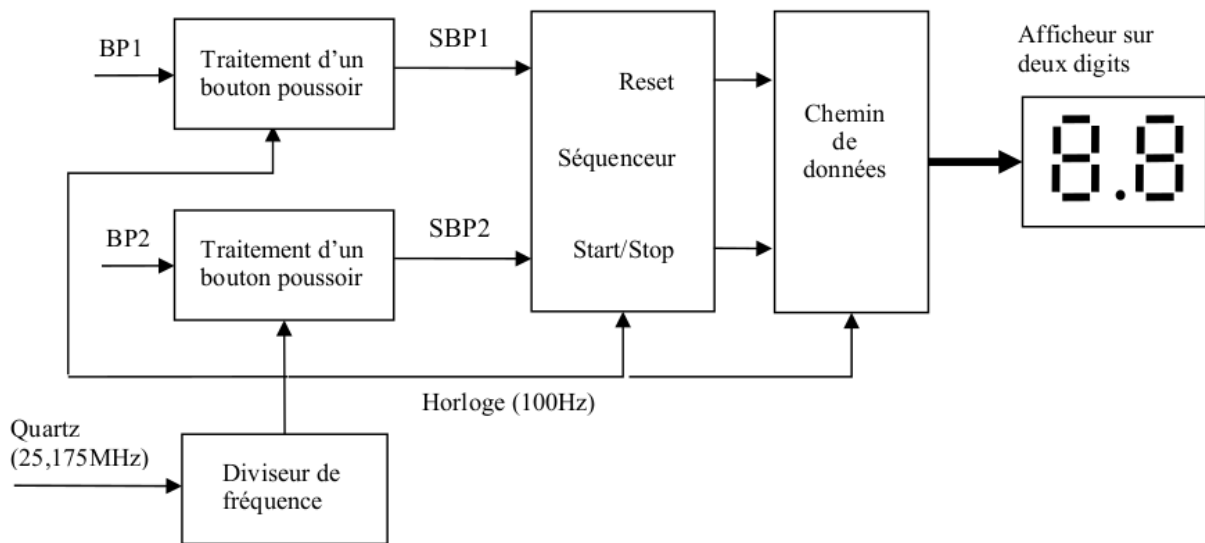


FIGURE 1 – Découpage du projet.

On consacrera donc une partie au traitement des boutons poussoirs, une au diviseur de fréquence et une au séquenceur et au chemin de donnée.

Deuxième partie

Étude préliminaire

Afin de prendre en main le logiciel de simulation Quartus, on simule un décodeur BCD vers 7 segments.

1 Saisie du schéma du montage

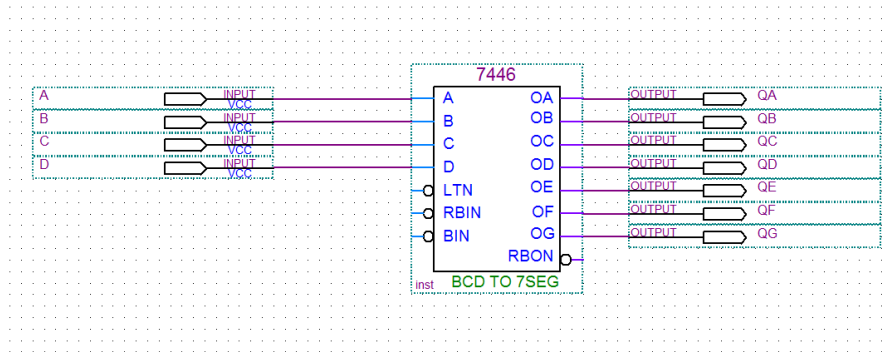


FIGURE 2 – Schéma saisi

La figure 2 montre le montage que nous simulerons. On commence par le compiler le projet une première fois, en utilisant le mode `smart compilation`.

2 Affectation des entrées et sorties

En consultant la documentation de la carte ALTERA disponible sur moodle.supelec.fr, on repère les numéros des entrées correspondant aux micro-interrupteurs à la page 12 et les numéros des sorties correspondant à l'afficheur numéro 1. On crée la table d'assignation reproduite à la figure 3.

	Node Name	Direction	Location
1	A	Input	PIN_33
2	B	Input	PIN_34
3	C	Input	PIN_35
4	D	Input	PIN_36
5	QA	Output	PIN_6
6	QB	Output	PIN_7
7	QC	Output	PIN_8
8	QD	Output	PIN_9
9	QE	Output	PIN_11
10	QF	Output	PIN_12
11	QG	Output	PIN_13
12	<<new node>>		

FIGURE 3 – Assignation des I/O

3 Définition des vecteurs de simulation

On crée un fichier `decodeur.vwf` dans lequel on spécifie les chronogrammes imposés aux entrées. Après avoir lancé la simulation on obtient les chronogrammes de la figure 4.

4 Simulation fonctionnelle

On a lancé une simulation fonctionnelle, il n'y a pas d'aléas. D'autre part, page 9 de la documentation nous informe sur l'afficheur 7 segments.

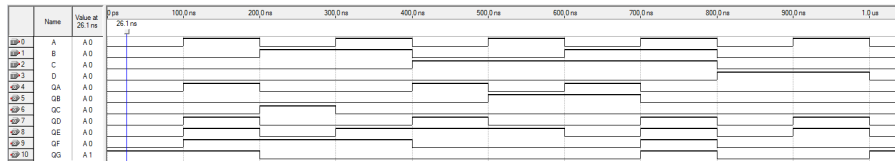


FIGURE 4 – Chronogramme obtenu par simulation fonctionnelle

Each LED segment of the display can be illuminated by driving the connected EPM7128S device I/O pin with a logic 0. Figure 4 shows the name of each segment.

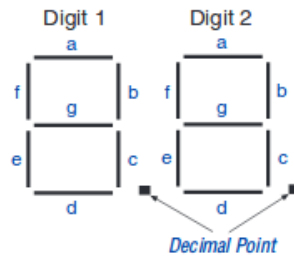


FIGURE 5 – Figure 4 de la documentation.

On vérifie ainsi que le chronogramme correspond à l’affichage des chiffres de 0 à 9.

5 Simulation temporelle

On lance ensuite une simulation temporelle. La figure 6 montre l’apparition d’aléas dûs aux temps de propagation.

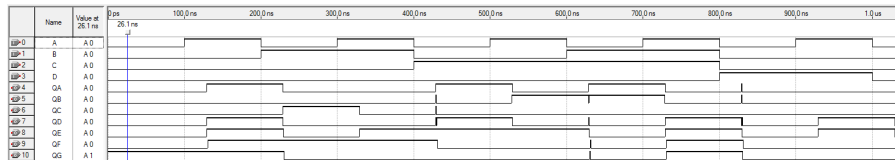


FIGURE 6 – Simulation temporelle

6 Analyse temporelle

L’analyse temporelle donne le temps de propagation entre les entrées et les sorties. La figure 7 montre que le temps de propagation est d’environ 30 ns.

7 Programmation

On programme la carte et on vérifie le fonctionnement en changeant l’état des interrupteurs, comme le montre les figures 8 et 9.

tpd					
	Slack	Required P2P Time	Actual P2P Time	From	To
1	N/A	None	31.300 ns	D	QF
2	N/A	None	31.100 ns	A	QF
3	N/A	None	31.000 ns	B	QF
4	N/A	None	30.700 ns	A	QG
5	N/A	None	30.700 ns	C	QF
6	N/A	None	30.600 ns	D	QG
7	N/A	None	30.600 ns	B	QG
8	N/A	None	30.600 ns	C	QG
9	N/A	None	29.600 ns	D	QC
10	N/A	None	29.600 ns	D	QA
11	N/A	None	29.500 ns	D	QB
12	N/A	None	29.400 ns	A	QE
13	N/A	None	29.400 ns	A	QD
14	N/A	None	29.300 ns	C	QE
15	N/A	None	29.300 ns	C	QD
16	N/A	None	29.300 ns	B	QC
17	N/A	None	29.300 ns	C	QC
18	N/A	None	29.300 ns	B	QA
19	N/A	None	29.300 ns	C	QA
20	N/A	None	29.200 ns	B	QB
21	N/A	None	29.100 ns	A	QC
22	N/A	None	29.100 ns	C	QB
23	N/A	None	29.100 ns	A	QA
24	N/A	None	29.000 ns	B	QE
25	N/A	None	29.000 ns	B	QD
26	N/A	None	28.900 ns	A	QB

FIGURE 7 – Analyse temporelle.

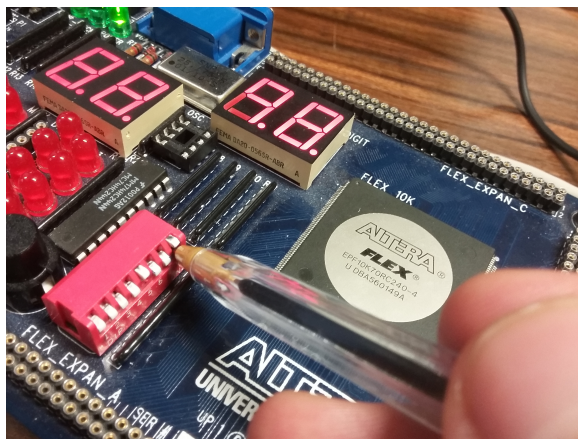


FIGURE 8 – Affichage du chiffre 9

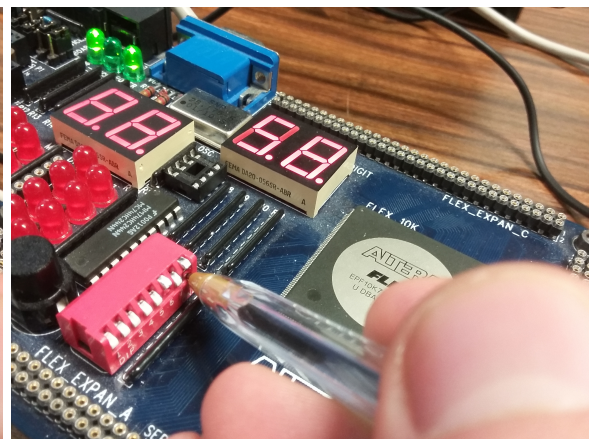


FIGURE 9 – Affichage du chiffre 5

Troisième partie

Traitement d'un bouton poussoir

Afin de détecter l'appui sur un bouton poussoir, on réalise un traitement de l'entrée associée. En effet il faut d'abord éliminer un éventuel bruit créé par les rebonds créés lors du changement de l'état du bouton et synchroniser le signal avec celui de l'horloge afin d'obtenir le signal BPS. Cette partie est réalisée avec une bascule D (on suppose donc que le bruit dure moins d'une période d'horloge). On souhaite ensuite passer la sortie à l'état haut pendant une période d'horloge sur front montant de BPS. On se propose de synthétiser cette partie avec une machine de MOORE puis avec une machine de MEALY.

Note : On remarquera que du fait du câblage des boutons (figure 10), un appui sur un bouton se traduira par le passage à l'état bas de l'entrée correspondante. Par soucis de simplicité, les calculs de cette partie se font en supposant le fonctionnement inversé. Ce n'est pas un problème puisqu'il suffit d'ajouter une porte inverseuse entre le bouton et l'entrée de notre système pour corriger cela. Les schémas présentés en fin de section auront donc une porte inverseuse en entrée.

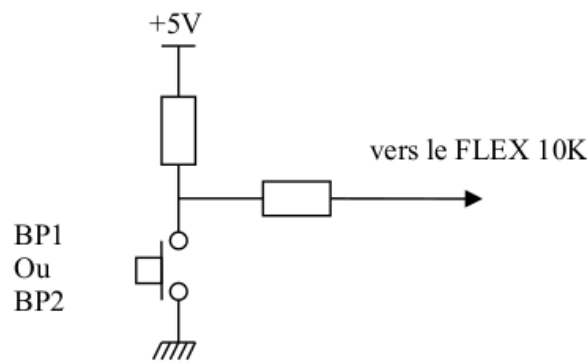


FIGURE 10 – Câblage d'un bouton poussoir

8 Synthèse avec une machine de MOORE

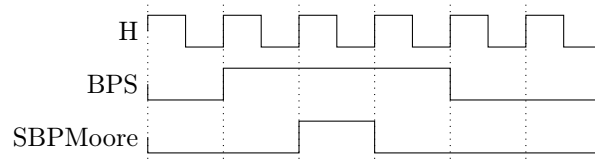


FIGURE 11 – Chronogramme recherché

On commence par résumer à l'aide d'un chronogramme le comportement recherché pour le système, représenté à la figure 11.

On souhaite réaliser une machine de MOORE, c'est à dire $S = g(E_p)$ et $E_S = f(E, E_p)$, avec S , E_p et E tels que représentés à la figure 12.

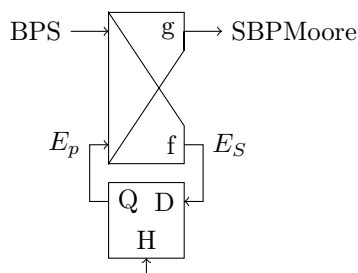


FIGURE 12 – Schéma général d'un système séquentiel synchrone

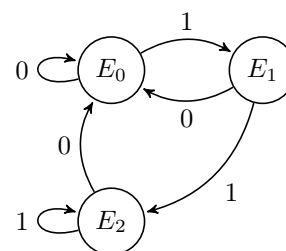


FIGURE 13 – Automate des états de la machine de MOORE

On synthétise ensuite le fonctionnement de la machine en dessinant le graphe des états, représenté à la figure 13. On obtient un graphe à trois états. Un premier état de repos, et deux autres permettant de passer la sortie au niveau 1 pendant un cycle d'horloge sur front montant de l'entrée.

On peut alors écrire le tableau des états (table 2), en utilisant le codage des états reproduit dans la table 1 (on pose $E_p = (y_1, y_0)$).

	y_1	y_0
E_0	0	0
E_1	0	1
E_2	1	0

TABLE 1 – Codage des états de la machine de MOORE

	BPS		
E_p	0	1	S
E_0	E_0	E_1	0
0 0	0 0	0 1	
E_1	E_0	E_2	1
0 1	0 0	1 0	
E_2	E_0	E_2	0
1 0	0 0	1 0	

TABLE 2 – Tableau des états de la machine de MOORE

On peut ensuite synthétiser l'état suivant E_S et la sortie S en fonction de BPS , y_0 et y_1 . On réalise pour cela trois tableaux de KARNAUGH (tables 3, 4 et 5).

	y_1y_0			
BPS	00	01	11	10
0	0	0	-	0
1	1	0	-	0

TABLE 3 – Tableau de KARNAUGH pour la variable Y_0

	y_1y_0			
BPS	00	01	11	10
0	0	0	-	0
1	0	1	-	1

TABLE 4 – Tableau de KARNAUGH pour la variable Y_1

	y_1y_0			
BPS	00	01	11	10
0	0	1	-	0
1	0	1	-	0

TABLE 5 – Tableau de KARNAUGH pour la variable S

On peut alors obtenir l'équation de chaque variables.

$$Y_0 = BPS \cdot \bar{y}_0 \cdot \bar{y}_1 \tag{1}$$

$$Y_1 = BPS \cdot (y_0 + y_1) \tag{2}$$

$$S = y_0 \tag{3}$$

Enfin, on vérifie que le fonctionnement est celui attendu par simulation, ce que montrent les figures 14 (présenté avec la porte inverseuse en entrée pour tenir compte du câblage du bouton) et 15.

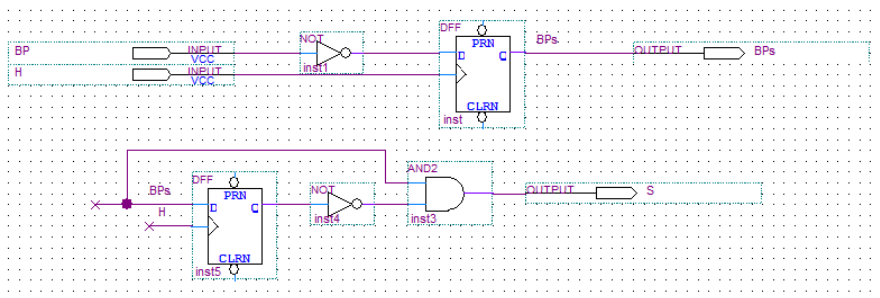


FIGURE 14 – Schéma du montage

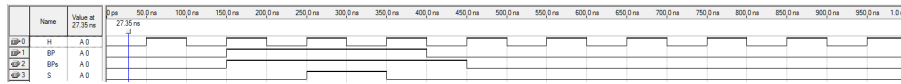


FIGURE 15 – Chronogramme de la simulation

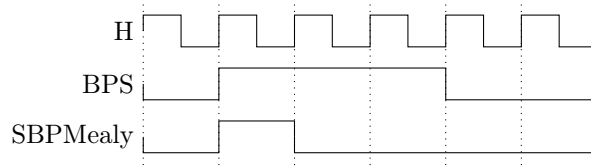


FIGURE 16 – Chronogramme recherché

9 Synthèse avec une machine de MEALY

On commence par résumer à l'aide d'un chronogramme le comportement recherché pour le système, représenté à la figure 16.

On souhaite réaliser une machine de MEALY, c'est à dire $S = g(E, E_p)$ et $E_S = f(E, E_p)$, avec S , E_p et E tels que représentés à la figure 17.

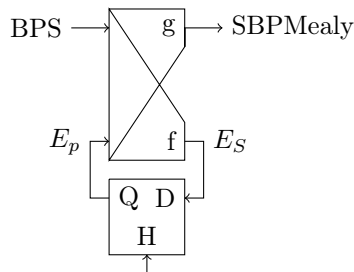


FIGURE 17 – Schéma général d'un système séquentiel synchrone

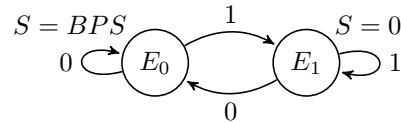


FIGURE 18 – Automate des états de la machine de MOORE

On synthétise ensuite le fonctionnement de la machine en dessinant le graphe des états, représenté à la figure 18. On obtient un graphe à deux états. Un premier état de repos qui permet de passer la sortie à l'état un dès que le bouton poussoir est enfoncé, et un autre permettant empêchant la sortie de rester au niveau haut plus d'un cycle d'horloge lors de l'appui.

On peut ensuite synthétiser l'état suivant E_S et la sortie S en fonction de BPS et E_p . On réalise pour cela deux tableaux de KARNAUGH (tables 6, 7).

		E_p	
	BPS	0	1
0	0	0	1
1	1	0	1

TABLE 6 – Tableau de KARNAUGH pour la variable E_S

		E_p	
	BPS	0	1
0	0	0	1
1	1	0	0

TABLE 7 – Tableau de KARNAUGH pour la sortie

On peut alors obtenir l'équation de chaque variables.

$$E_S = BPS \tag{4}$$

$$S = BPS \cdot \bar{E}_p \tag{5}$$

Enfin, on vérifie que le fonctionnement est celui attendu par simulation, ce que montrent les figures 19 (présenté avec la porte inverseuse en entrée pour tenir compte du câblage du bouton) et 20.

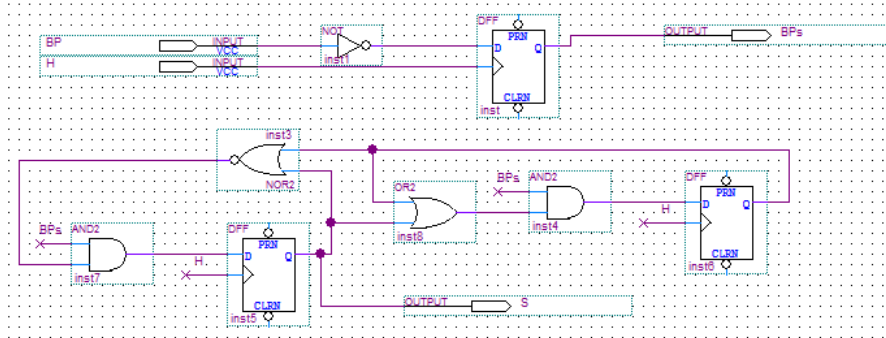


FIGURE 19 – Schéma du montage

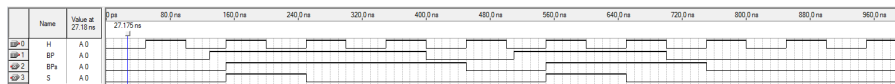


FIGURE 20 – Chronogramme de la simulation

Quatrième partie

Diviseur de fréquence

Le quartz de la carte délivre un signal d'horloge à 27.175MHz. On souhaite baisser cette fréquence à 100Hz afin de faciliter le comptage pour le chronomètre et d'offrir une expérience proche du temps réel à l'utilisateur.

10 Structure du diviseur

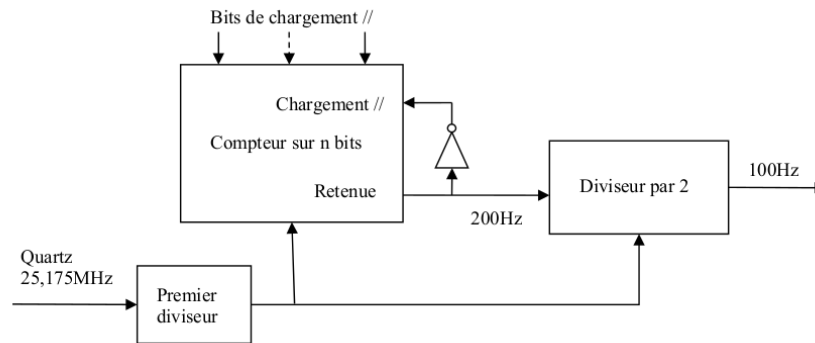


FIGURE 21 – Schéma du diviseur de fréquence.

11 Détermination du premier rapport de division

On cherche dans un premier temps à déterminer le rapport de division de fréquence que doit offrir le premier diviseur. On synthétise la première division de fréquence par la figure 22, avec α le rapport de division du premier diviseur.

$$25\,175\,000\text{ Hz} \xrightarrow{\frac{1}{\alpha}} f_1 \xrightarrow{\frac{1}{\beta}} 200\text{ Hz}$$

FIGURE 22 – Synthèse de la réduction de fréquence.

On souhaite de plus que β soit entier. Puisqu'il n'y a qu'un nombre réduit de possibilités pour le premier diviseur, on va chercher parmi les différentes valeurs possibles de α celle qui donne β entier. On a la relation $\alpha \times \beta = \frac{251\,750}{2}$. À l'aide d'un tableur (figure 23), on détermine $\alpha = 5$ et $\beta = 25\,175$.

alpha	2	4	8	16	1	2	3	4	5	6	7	8	9	10
beta	62938	31469	15734,38	7867,1875	125875	62937,5	41958,33333	31468,75	25175	20979,16667	17982,14286	15734,375	13986,11111	12587,5

FIGURE 23 – Utilisation d'un tableur pour déterminer α

12 Compteur

On va ensuite réaliser la division de fréquence d'un facteur β à l'aide d'un compteur 16 bits (réalisé par mise en cascade de deux compteurs 8 bits `8count`). On détermine la valeur x jusqu'à laquelle le compteur 16 bits doit compter :

$$x = \frac{256 \times 256}{65\,536} - \beta$$

=65 536=val. max de comptage

On va donc compter jusqu'à $40\,361_{(10)} = 1001\,1101\,1010\,1001_{(2)}$.

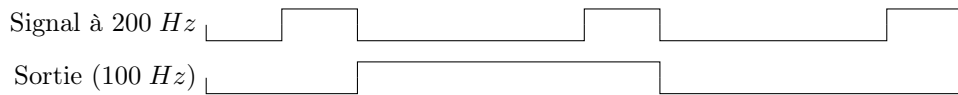


FIGURE 24 – Chronogramme du comportement recherché

13 Diviseur par 2

Afin d'avoir un rapport cyclique de $\frac{1}{2}$ on utilise un diviseur par deux en sortie. Le fonctionnement recherché est résumé par le chronogramme de la figure 24.

On synthétise alors le graphe des états correspondant (figure 25).

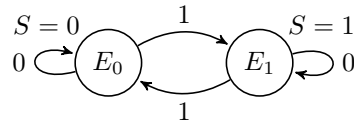


FIGURE 25 – Automate des états de la machine de MOORE

On réalise le tableau des états (table 8) et on en déduit l'équation de la sortie en fonction de l'entrée E .

E_p	Entrée		S
	0	1	
0	0	1	0
1	1	0	1

TABLE 8 – Tableau des états du diviseur par 2

On a ainsi :

$$S = \bar{E}_p \cdot E + E_p \cdot \bar{E}$$

On reconnaît l'équation d'une bascule T. On utilisera donc cette bascule pour réaliser le diviseur par deux.

14 Réalisation du diviseur

On modélise d'abord le schéma dans Quartus (figure 26).

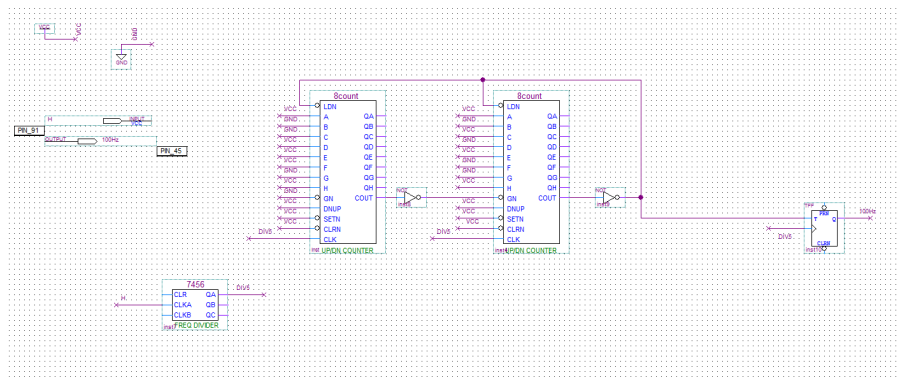


FIGURE 26 – Schéma réalisé dans Quartus

Puis on vérifie à l'oscilloscope que l'on obtient bien un signal à 100 Hz en sortie (figure 27).

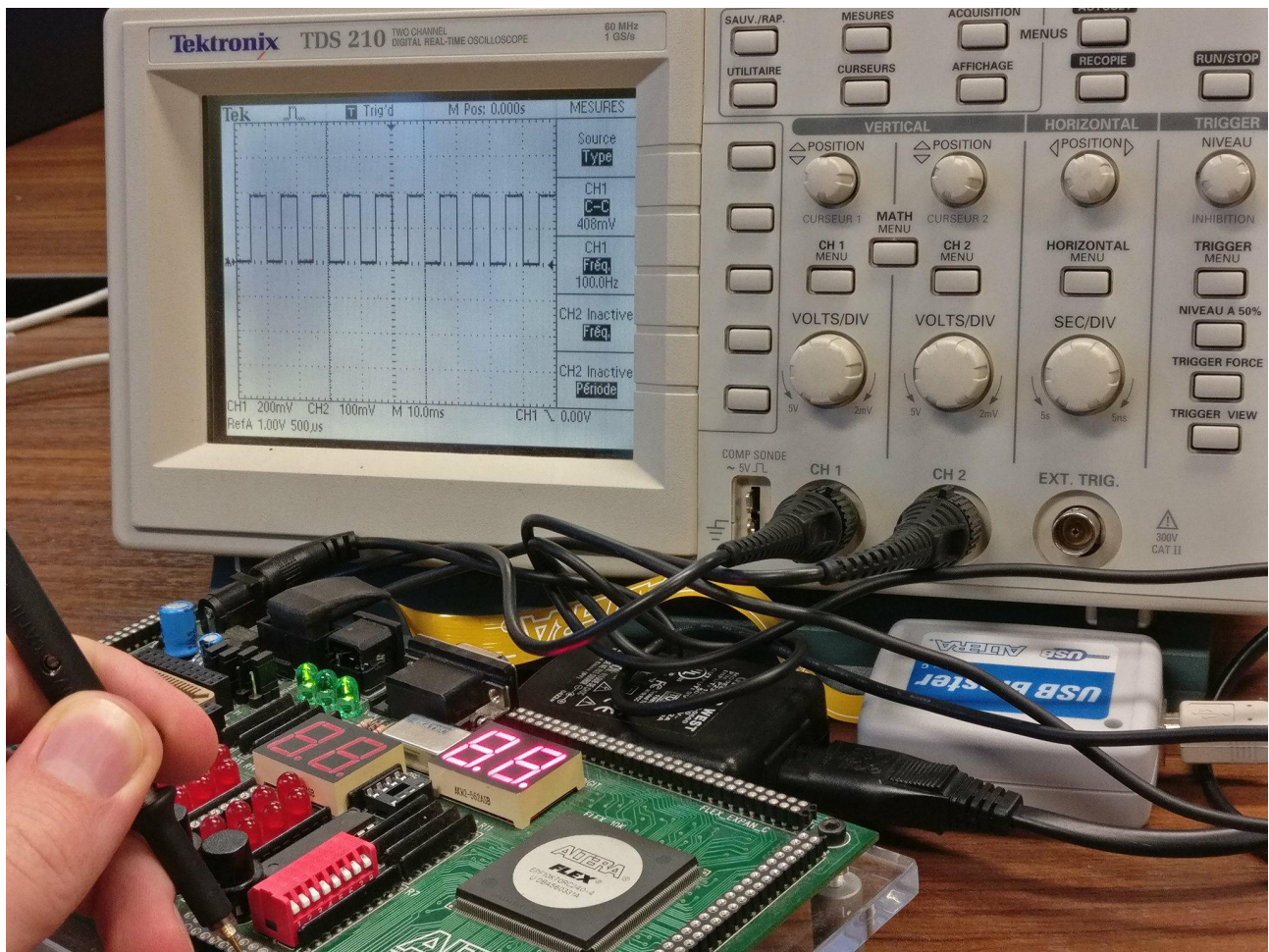


FIGURE 27 – Mesure à l'oscilloscope

Cinquième partie

Partie opérative et séquenceur

Cette partie vise à synthétiser la partie opérative et le séquenceur. La figure 28 montre l'interfaçage de ces deux parties.

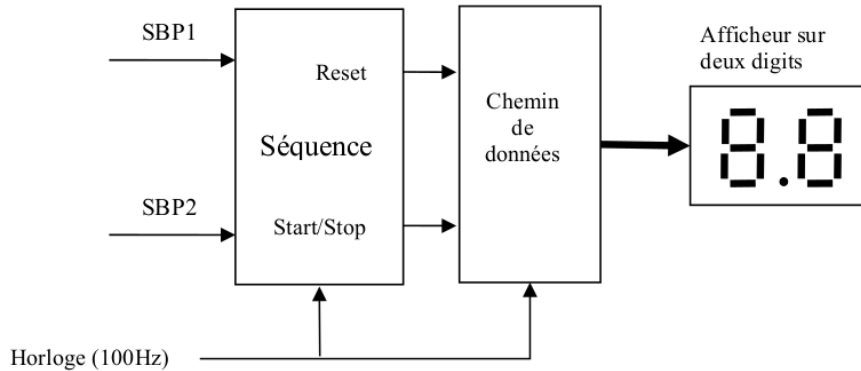


FIGURE 28 – Séquenceur et chemin de données

15 Chemin de données

On souhaite réaliser le chemin de données du chronomètre. Cette partie devra décompter le temps à partir du signal à 100Hz de l'horloge, et l'afficher sur les afficheurs 7 segments de la carte. De plus elle acceptera deux entrées : une pour réinitialiser le chronomètre et une autre pour autoriser ou stopper le décompte du temps.

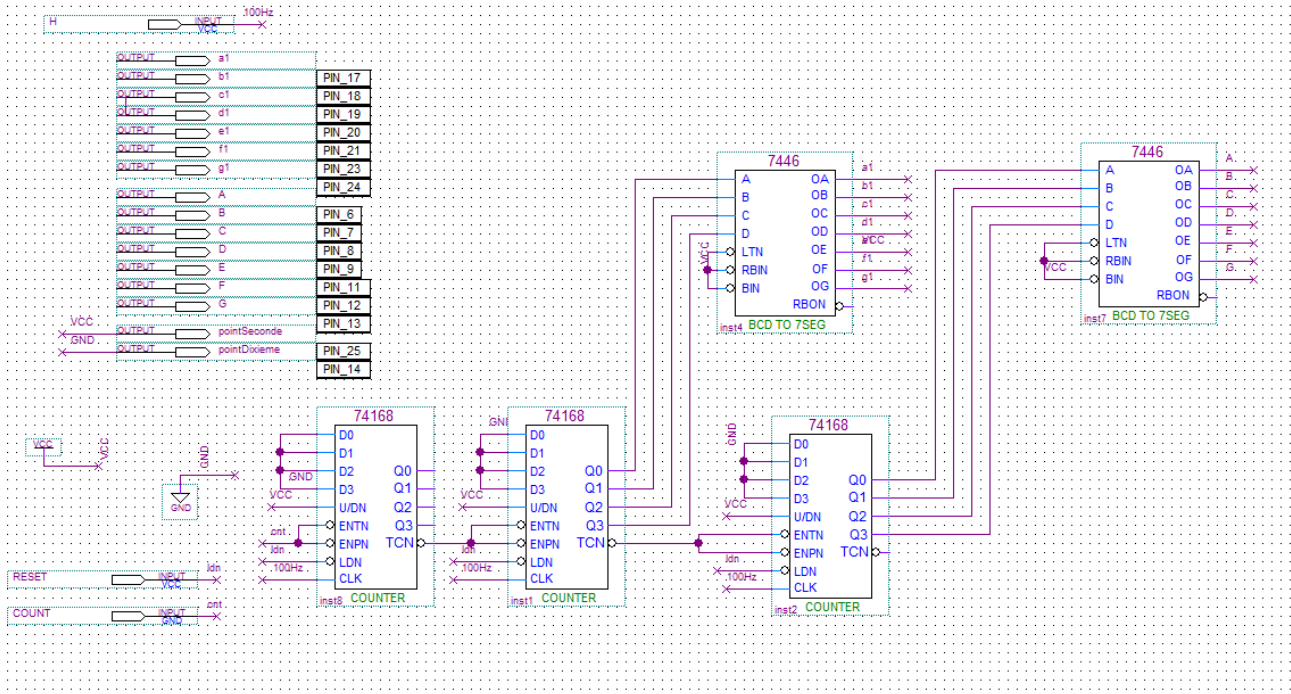


FIGURE 29 – Schéma réalisé dans Quartus

Pour réaliser le décompte du temps, on utilise une association de compteurs similaire à celle du diviseur de fréquence. Le composant utilisé est le 74168. On utilise 3 compteurs afin d'obtenir le décompte des centièmes de secondes (non affiché), des dixièmes de secondes et des secondes. On utilise des décodeurs 7446 pour piloter les afficheurs 7 segments. Le décompte est autorisé par le passage à l'état bas de l'entrée COUNT du chemin de données (reliée à ENTN et ENPN du premier compteur 74168). On peut réinitialiser la valeur du chronomètre en

passant l'entrée RESET à l'état bas (reliée au chargement parallèle des compteurs). On obtient le schéma de la figure 29.

Afin de tester cette partie du montage, on introduit dans le bloc l'horloge à 100Hz réalisée précédemment et on fixe les valeurs de COUNT et RESET. Après téléchargement sur la carte, on vérifie que l'on obtient bien un décompte des secondes et des dixièmes de secondes (figure 30).

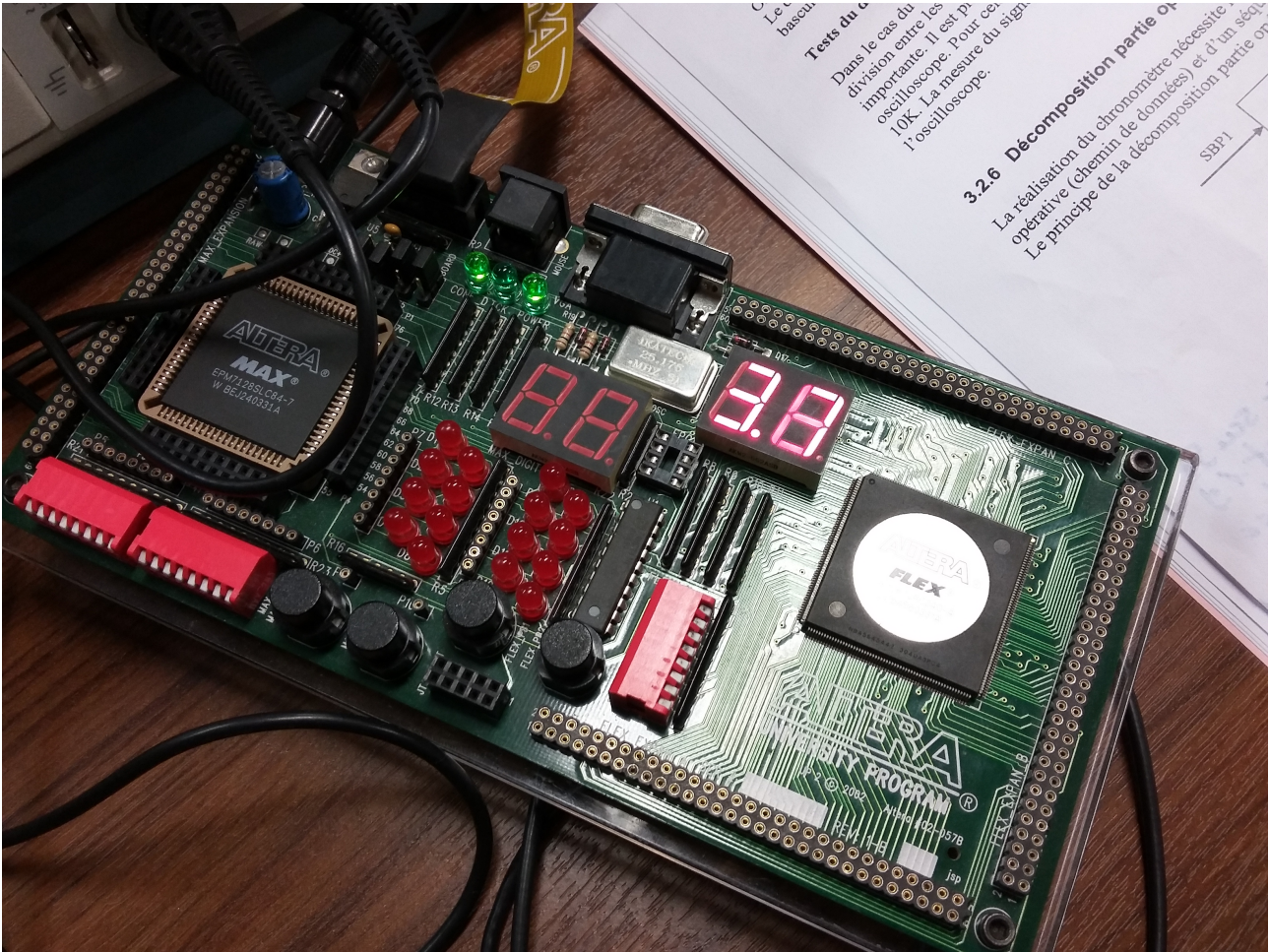


FIGURE 30 – Test du chemin de données

16 Séquenceur

On souhaite maintenant réaliser le séquenceur. Il prendra deux entrées : SPB1 et SPB2, et deux sorties : COUNT et RESET qui viendront s'interfacer avec le chemin de données. On cherche le comportement suivant :

- Si l'on appuie sur BP1 le signal RESET est actif (passage à zéro) ;
- Le bouton BP2 permet de naviguer entre les modes comptage et décomptage (passage de COUNT à 1 ou 0).

On synthétise ce comportement en une machine de MEALY comme le montre le graphe des états en figure 31.

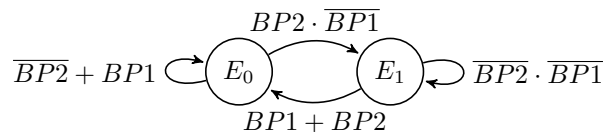


FIGURE 31 – Automate des états de la machine de MOORE

L'état E_0 correspond à un état d'attente et E_1 de comptage.

Puis on synthétise les transitions dans le tableau binaire des états représenté en table 9 (puisque'il n'y a que deux états, on se permettra de confondre E_P avec son codage sur un bit). On synthétise également l'état des

		<i>BP1 BP2</i>			
<i>E_p</i>		00	01	11	10
	0	0	1	0	0
	1	1	0	0	0

TABLE 9 – Tableau de KARNAUGH pour l'état suivant

		<i>BP1 BP2</i>			
<i>E_p</i>		00	01	11	10
	0	1	1	1	1
	1	0	0	0	0

TABLE 10 – Tableau de KARNAUGH pour *COUNT*

		<i>BP1 BP2</i>			
<i>E_p</i>		00	01	11	10
	0	1	1	0	0
	1	1	1	0	0

TABLE 11 – Tableau de KARNAUGH pour *RESET*

sorties (tables 10 et 11).

On peut alors en déduire les équations de l'état suivant et des sorties.

$$COUNT = \overline{E_P} \tag{6}$$

$$RESET = \overline{BP1} \tag{7}$$

$$E_S = \overline{BP1} \cdot (BP2 \oplus E_P) \tag{8}$$

$$\tag{9}$$

Enfin on réalise le montage sur Quartus (figure 32) et on le simule (figure 33).

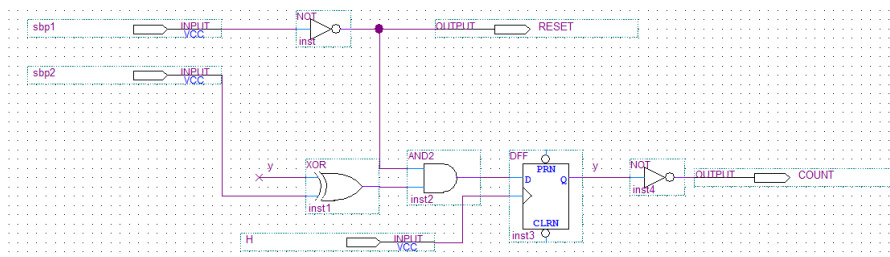


FIGURE 32 – Schéma du séquenceur

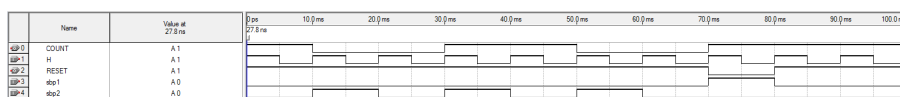


FIGURE 33 – Simulation fonctionnelle du séquenceur

Sixième partie

Réalisation du chronomètre

En assemblant les éléments synthétisés dans les parties précédentes selon ce qui est proposé dans le sujet (figure 34), on obtient le schéma de la figure 35.

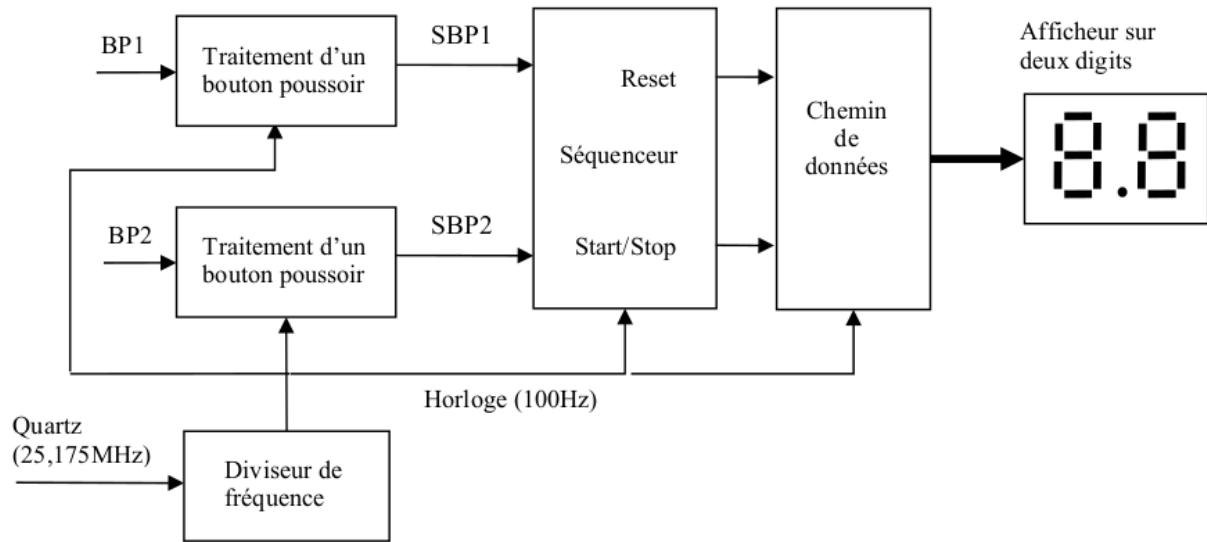


FIGURE 34 – Structure du chronomètre

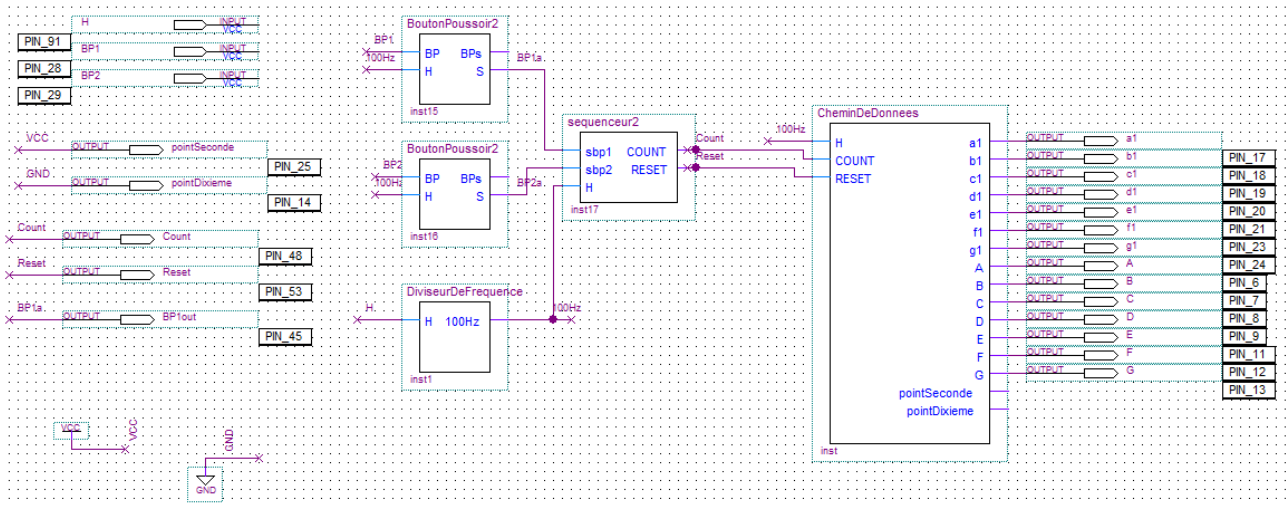


FIGURE 35 – Schéma du chronomètre

Septième partie

Conclusion

En téléversant le schéma sur la carte on vérifié que l'on obtient bien le comportement recherché :

- Un appui sur START/STOP arrête ou démarre le chronomètre ;
- Un appui sur RESET arrête le chronomètre et le remet à zéro ;
- Le chronomètre compte 10 secondes et se réinitialise, en affichant les secondes et les dixièmes de seconde ;
- Le point le point lumineux entre les secondes et les dixièmes de secondes et affiché.

On a donc répondu au cahier des charges, ce qui conclue cette étude.

Table des figures

1	Découpage du projet.	2
2	Schéma saisi	3
3	Assignation des I/O	3
4	Chronogramme obtenu par simulation fonctionnelle	4
5	Figure 4 de la documentation.	4
6	Simulation temporelle	4
7	Analyse temporelle.	5
8	Affichage du chiffre 9	5
9	Affichage du chiffre 5	5
10	Câblage d'un bouton poussoir	6
11	Chronogramme recherché	6
12	Schéma général d'un système séquentiel synchrone	6
13	Automate des états de la machine de MOORE	6
14	Schéma du montage	7
15	Chronogramme de la simulation	8
16	Chronogramme recherché	8
17	Schéma général d'un système séquentiel synchrone	8
18	Automate des états de la machine de MOORE	8
19	Schéma du montage	9
20	Chronogramme de la simulation	9
21	Schéma du diviseur de fréquence.	10
22	Synthèse de la réduction de fréquence.	10
23	Utilisation d'un tableur pour déterminer α	10
24	Chronogramme du comportement recherché	11
25	Automate des états de la machine de MOORE	11
26	Schéma réalisé dans Quartus	11
27	Mesure à l'oscilloscope	12
28	Séquenceur et chemin de données	13
29	Schéma réalisé dans Quartus	13
30	Test du chemin de données	14
31	Automate des états de la machine de MOORE	14
32	Schéma du séquenceur	15
33	Simulation fonctionnelle du séquenceur	15
34	Structure du chronomètre	16
35	Schéma du chronomètre	16