

Travail complémentaire pour les A3

Sommaire

I.	Description du cahier des charges	2
II.	Mode de fonctionnement	2
III.	Les vues externe et interne du système	3
IV.	Codage :	6
1.	Codage du compteur (horloge) et décompteur (minuteur)	6
2.	Codage du Décodeur 7 segments	7
3.	Codage du Top level.....	7
V.	Simulation.....	8
VI.	Reference et mots clé.....	10



DBIBIH Oussama
4eme année S3E, Promo 1 Groupe 1
Module : Electronique numérique séquentielle et VHDL

I. Description du cahier des charges

Nous souhaitons réaliser un système qui permette suivant les besoins de donner l'heure (Format **heures:minutes:secondes**) ou de servir de minuteur. Un sélecteur "mode" permet de passer dans l'un ou l'autre des modes de fonctionnement.

Mode = 1 => fonctionnement en horloge

Mode = 0 => fonctionnement en minuteur

Le système est cadencé par un signal d'horloge de 1Hz que nous appellerons "**clk**" et les afficheurs sont des afficheurs 7 segments avec les segments allumés sur un niveau logique 1.



Figure1 : Exemple minuteur

II. Mode de fonctionnement

Description du mode 1 :

L'appui sur le bouton start charge les valeurs **dh** (dizaine d'heure), **uh** (unité d'heure), **dm** (dizaine de minute) et **um** (unité minute) et remet à 0 les secondes.

Lorsque le bouton start est relâché l'affichage s'incrémente toutes les secondes au format

HH:MM:SS

Description du mode 0 :

L'appui sur le bouton start charge les valeurs **dh** (dizaine d'heure), **uh** (unité d'heure), **dm** (dizaine de minute) et **um** (unité minute) et remet à 0 les secondes.

Lorsque le bouton start est relâché l'affichage décrémente toutes les secondes au format **HH:MM:SS**.

Lorsque le compteur arrive à **00:00:00** la sortie alarme est activée pendant **3 secondes**.

III. Les vues externe et interne du système

Voici les vues externe et interne de l'**Afficheur_7_seg** :

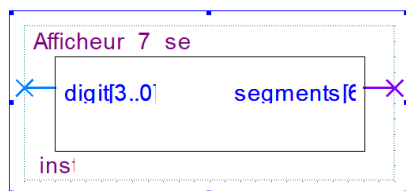


Figure2 : Vue externe

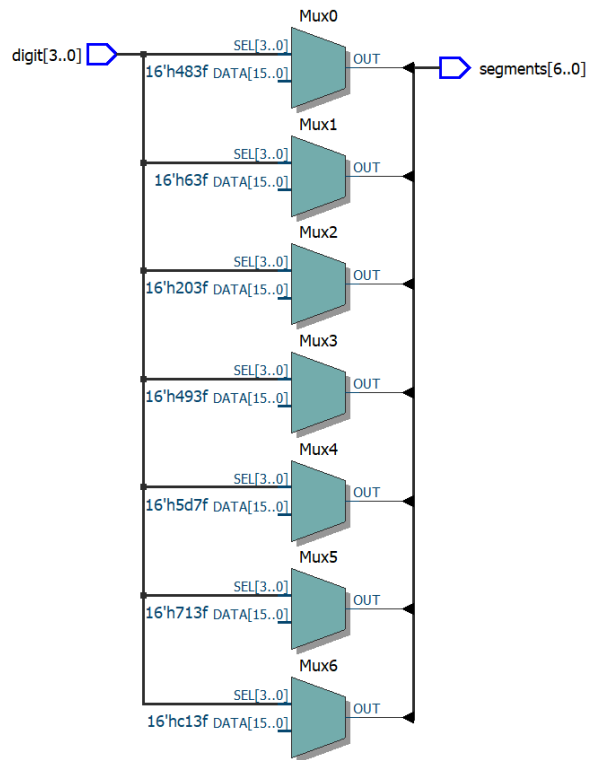


Figure3 : Vue interne

Voici les vues externe et interne du **cpt_dcp** :

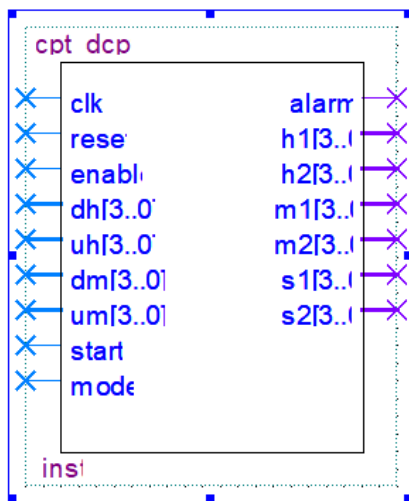


Figure4 : Vue externe

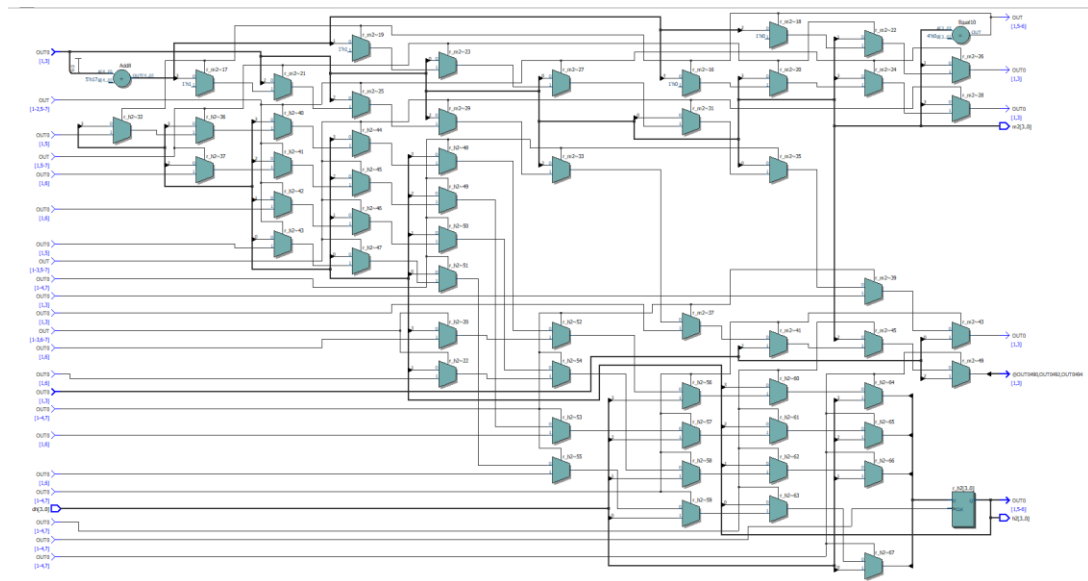


Figure5 : Vue externe

Voici les vues externe et interne du **Top_level** :

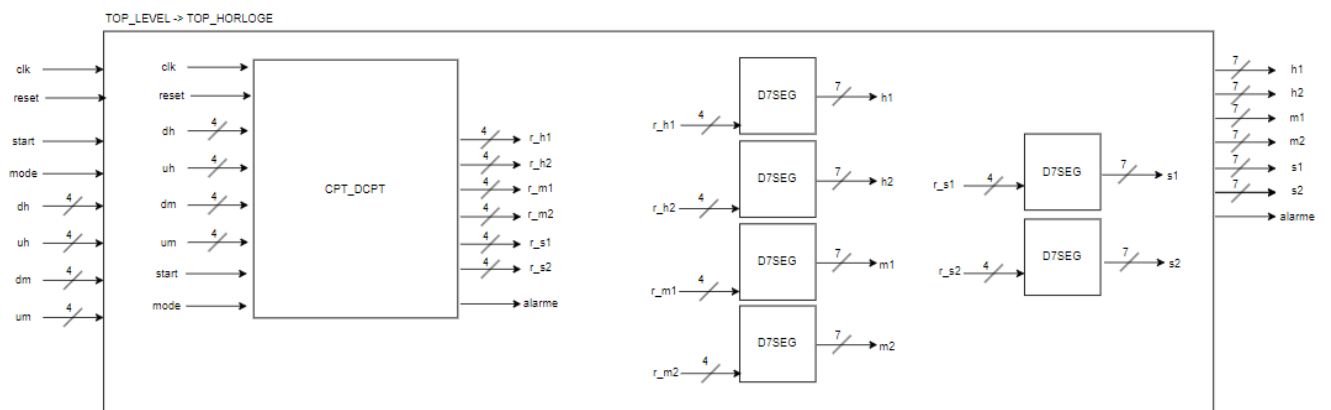


Figure6 : Vue externe

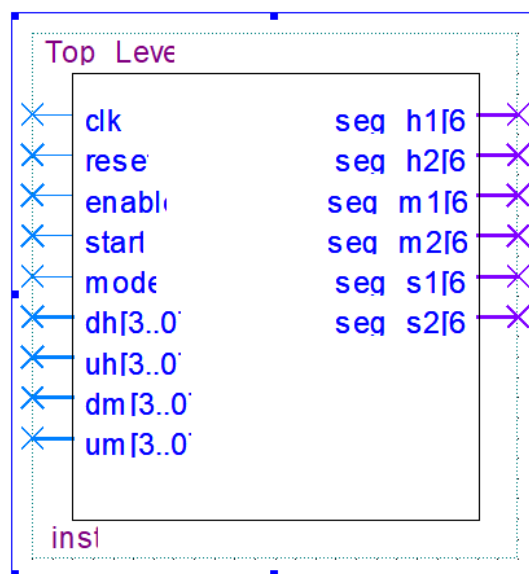


Figure7 : Vue externe

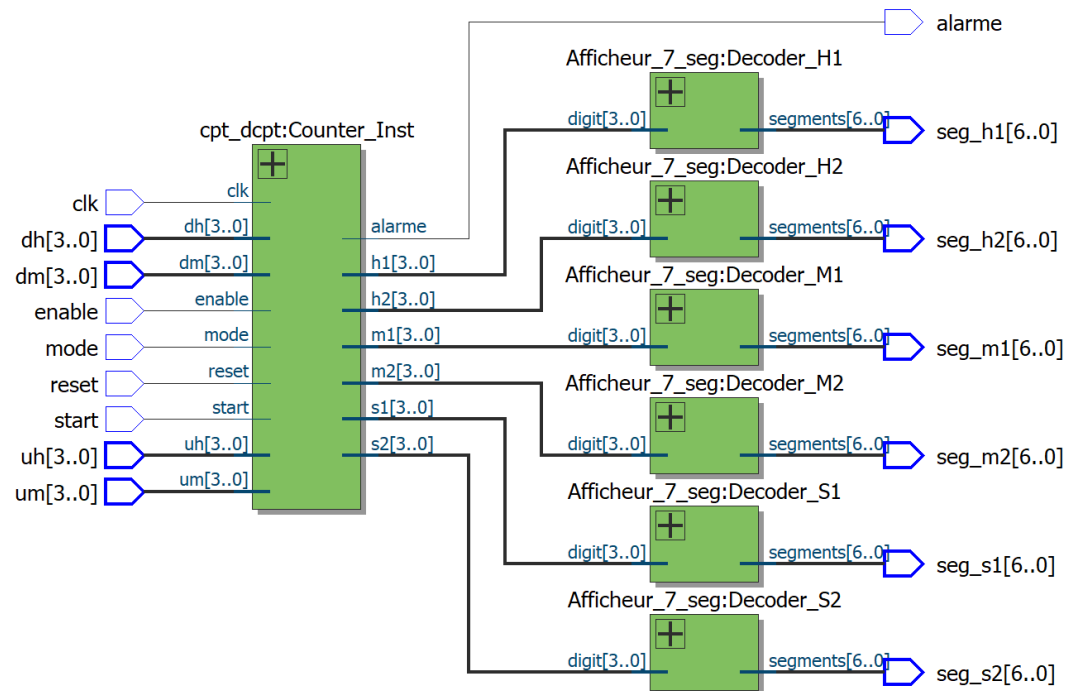


Figure8 : Vue interne

IV. Codage :

1. Codage du compteur (horloge) et décompteur (minuteur)

Le script ***cpt_dcpt.vhd*** implémente un compteur pour le mode horloge et un décompteur pour le mode minuteur. Lors de son codage, j'ai pris soin de respecter la logique des transitions d'une horloge réelle.

Fonctionnalités principales :

- ✓ Mode horloge :
 - Les secondes s'incrémentent jusqu'à atteindre 59, moment où elles se réinitialisent à 00 et ajoutent une minute.
 - Les minutes suivent une logique similaire : lorsqu'elles atteignent 59, elles se réinitialisent à 00 et ajoutent une heure.
 - Les heures vont de **00:00:00 à 23:59:59**, après quoi elles se réinitialisent à **00:00:00**.
- ✓ Mode minuteur :
 - Les secondes décrémentent jusqu'à atteindre 00, moment où elles passent à 59 et soustraient une minute.
 - Les minutes décrémentent de la même manière : lorsqu'elles atteignent 00, elles passent à 59 et soustraient une heure.
 - Lorsque le minuteur atteint **00:00:00**, un signal d'alarme est activé pendant trois secondes.
- ✓ Logique des transitions temporelles :
 - L'implémentation utilise une approche hiérarchique :
 - Les secondes gèrent les transitions vers les minutes.
 - Les minutes gèrent les transitions vers les heures.
 - Les heures respectent les limites de 23 pour garantir le format **24 heures** en mode horloge.

Le script s'inspire directement du fonctionnement d'une horloge réelle.

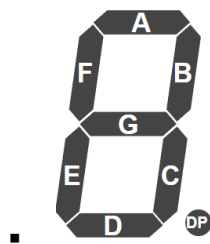
2. Codage du Décodeur 7 segments

Le script **Afficheur_7_seg.vhd** est basé sur le fonctionnement d'un décodeur **7segments** standard. Il transforme une valeur binaire de 4 bits en un signal de 7 bits correspondant aux segments d'un afficheur.

Fonctionnalités principales :

✓ Entrées et sorties:

- L'entrée est un vecteur binaire de 4 bits représentant un chiffre décimal.
- La sortie est un vecteur de 7 bits activant les segments correspondants (a, b, c, d, e, f, g).



✓ Logique de décodage:

- Le décodeur utilise une instruction **case** pour associer chaque chiffre binaire à la combinaison appropriée de segments.
- Une gestion par défaut (others) désactive tous les segments pour les valeurs non valides.

3. Codage du Top level

Le script **Top_Level.vhd** rassemble le compteur/décompteur **cpt_dcpt** et les six instances de **l'Afficheur_7_seg** pour construire le système global, répondant ainsi au cahier des charges.

Fonctionnalités principales :

✓ Connexion des composants :

- Les sorties du compteur (heures, minutes, secondes) sont directement connectées aux entrées des décodeurs 7 segments.
- Les sorties des décodeurs pilotent les afficheurs 7 segments.

✓ Gestion des modes :

- Les signaux de contrôle (mode, start, reset, enable) permettent de passer entre les modes horloge et minuteur.
- En mode minuteur, le signal d'alarme est activé lorsque le temps atteint **00:00:00**.

✓ Structure modulaire :

- Le design top-level est conçu pour intégrer facilement les modules de base, tout en assurant leur interconnexion logique.

V. Simulation

Simulation *Afficheur_7_seg*:

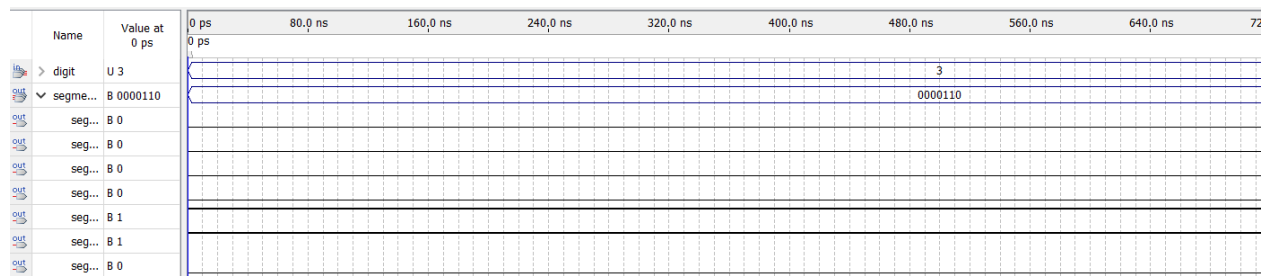


Figure9 : Simulation Waveform dec7.vwf

On voit que si digit prend '0011' segments prend '0000110' qui représente 3.

Simulation *cpt_dcpt*:

Mode = 1 :

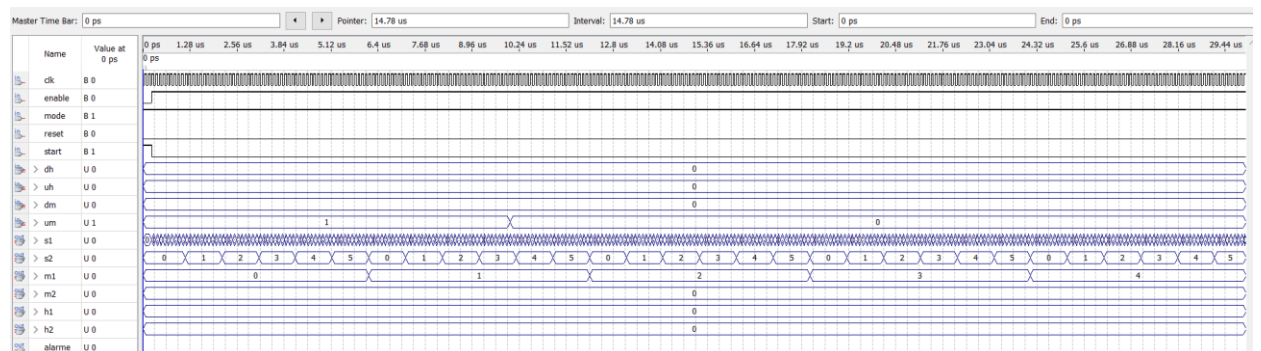


Figure10 : Simulation Waveform cpt dcpt 1.vwf

On voit que les secondes passent à 59 avant d'ajouter une minute

Mode = 0 :

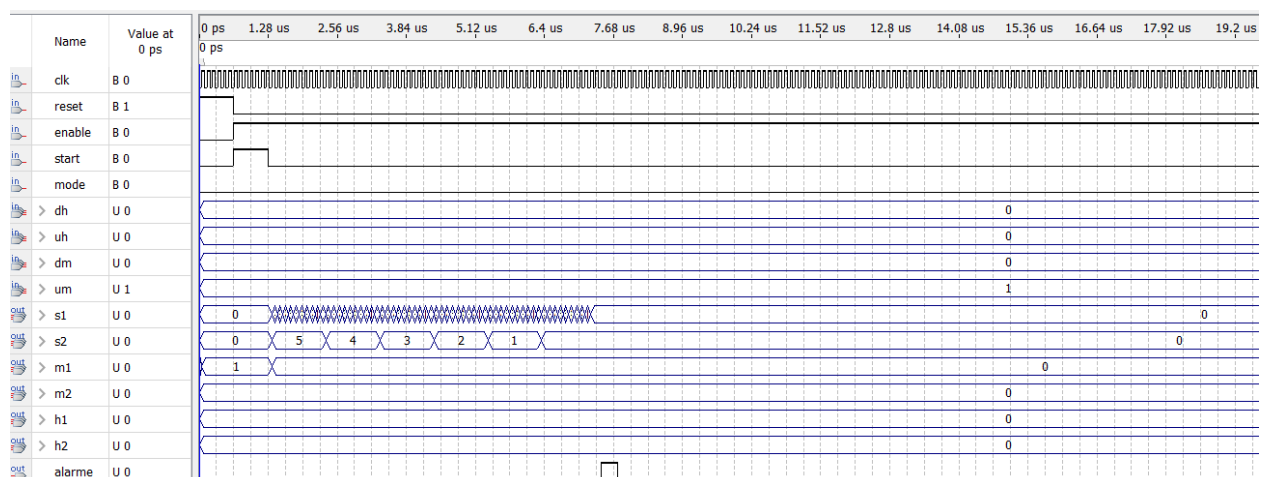


Figure11 : Simulation Waveform cpt dcpt 0.vwf

On voit la décrémentation jusqu'à 00:00:00 et l'activation de l'alarme.

Simulation du *top_level* :

Mode = 1 :

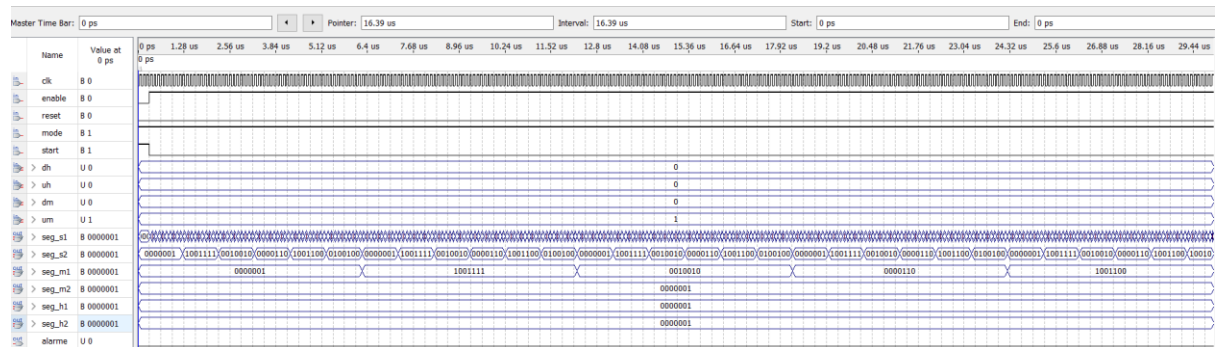


Figure12: Simulation Waveform *top_level_1.vwf*

Mode = 0:

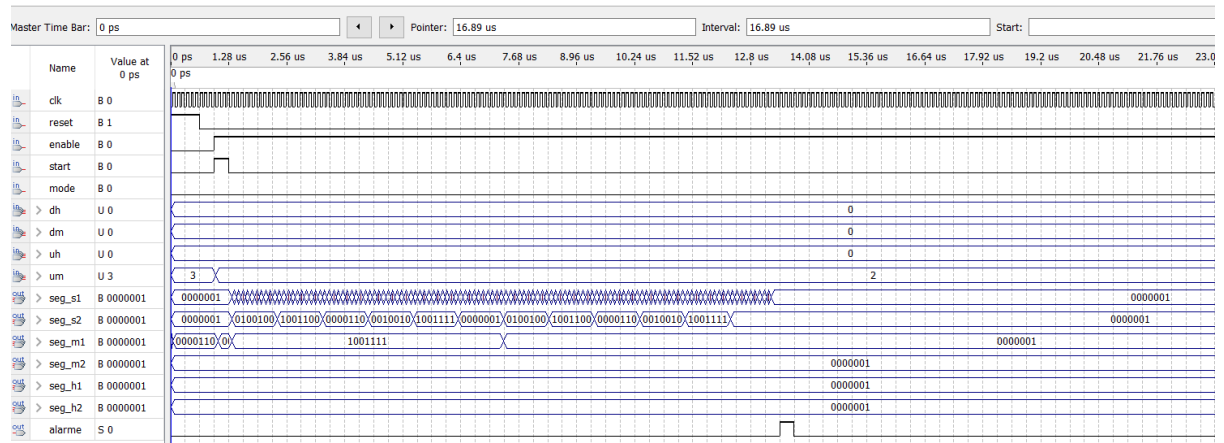


Figure13: Simulation Waveform *top_level_0.vwf*

VI. Reference et mots clé

Mots clé :

- ***Cpt_dcpt*** : compteur/décompteur.
- ***Top_level*** : le module principal qui rassemble et coordonne les différents composants du système.