

TP VHDL 2 – DIGICODE

Objectif :

L'objectif de ce TP est de réaliser un digicode, c'est-à-dire enchaîner une séquence de quatre touches afin de pouvoir commander l'ouverture d'une porte, par exemple.

Le TP se décompose en deux parties.

La première partie vous permettra de commander l'ouverture de la porte lorsque le code correspond au code par défaut (BTND, BTNR, BTNU, BTNL).

Le code par défaut est le code stocké dans la mémoire à l'appui sur le Reset (BTNC). Lorsque le code est correct, une LED s'éclaire et l'ouverture de la porte (via un interrupteur mis à 1) est signalée par l'éclairage d'une autre LED. Deux autres LED pourront être utilisées pour indiquer la valeur de l'adresse sur 2 bits. La porte devra être refermée avant de pouvoir saisir à nouveau le code.

La deuxième partie a pour objectif le changement du code par défaut. Le code ne pourra être changé que lorsque le code actuel (séquence par défaut, par exemple) a été composé. Le mode changement de code sera matérialisé par un interrupteur.

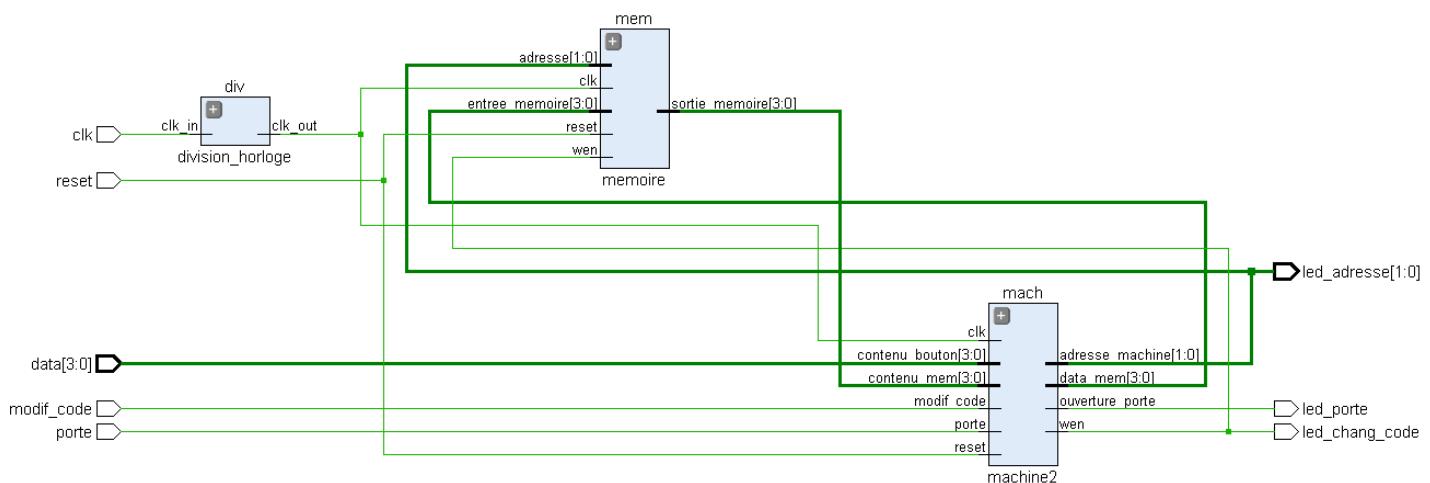
Interrupteurs :
PORTE
MODIF_CODE



Boutons poussoirs :
Code : BTND, BTNR, BTNU, BTNL
Reset : BTNC

LEDs
LED_ADRESSE (2 leds)
LED PORTE
LED CHANG_CODE

A l'issue de la 2^{ème} partie, le schéma de la page principale est le suivant :



- ✓ Une description VHDL de la partie mémoire où sera stocké le code est donnée sur Moodle. Le bloc mémoire est initialisé avec la séquence BTND, BTNR, BTNU, BTNL à la remise à zéro du circuit. L'ordre d'appui des boutons sera défini par une adresse.

Ordre	Adresse	Bouton	Codage bouton
1	00	BTN D	0001
2	01	BTN R	0010
3	10	BTN U	0100
4	11	BTN L	1000

Première partie : ouverture de la porte par le code par défaut

- ✓ Ecrire la partie séquentielle du digicode à l'aide d'une machine d'états. Un template est accessible sur Moodle, dans la section du TP2.
- ✓ Il faudra que la machine d'états prenne en compte la vérification du relâchement des boutons pousoirs avant le passage d'un état à l'autre.
- ✓ Simuler la machine d'états.
- ✓ Ecrire la page principale où vous regrouerez les composantes :
 - mémoire
 - machine d'états
 - diviseur d'horloge
- ✓ Simuler la page principale.
- ✓ Définition des entrées-sorties (I/O Floorplaner) :

Les E/S sont de type IOS STANDARD LVCMOS3.

 - CLK : E3
 - RESET ou BTNC : N17
 - DATA(0) ou BTND : P18
 - DATA(1) ou BTNR : M17
 - DATA(2) ou BTNU : M18
 - DATA(3) ou BTNL : P17
 - LED_ADRESSE(0) : au choix
 - LED_ADRESSE(1) : au choix
 - PORTE : au choix
- ✓ Programmer la carte NexysA7

Deuxième partie : changement de code

- ✓ Modifier la machine d'états pour gérer la modification de code. Une fois le code actuel saisi, utiliser un interrupteur pour passer en mode changement de code. Le nouveau code est alors saisi et enregistré dans la mémoire.
- ✓ Modifier la mémoire pour qu'elle puisse utilisée en mode Lecture (read) ou écriture (write).
- ✓ Modifier la page principale.

- ✓ Simuler la page principale
- ✓ Mettre à jour les entrées sorties. Vous rajouterez les interrupteurs et leds à votre convenance.
- ✓ Programmer la carte NexysA7