

ENSEEIH

RAPPORT BE CONCEPTION D'UNITÉS SÉQUENTIELLES ET COMBINATOIRE

realise par : BOUZID TAHA

Table des matières

1	Introduction	2
2	cahier de charges	2
3	principe de réalisation	2
4	Description des differents module	5
4.1	Module MA7 afficheur 7 segments	5
4.2	Description du Module MCV Compteur/Décompteur 1-6	5
4.3	description du module MD20	6
4.4	Description du Module MCC	6
4.5	Description du Module BCD/MA20	7
4.6	Description du Module MPAV	7
4.7	Description du Module MCS	8
5	creation des blocs	8
5.1	BGV	8
5.2	BGC	8
5.3	BGS	9
5.4	chef d'orchestre	10
6	Conclusion	10
7	Annexes	11
7.1	MA7	11
7.2	MCV	12
7.3	md20	14
7.4	MCC	15
7.5	MA20	17
7.6	MCS	18
7.7	MPAV	19
7.8	validation intermediaire des different modules	20

1 Introduction

L'objectif de ce BE est de concevoir un système automatisé pour la gestion d'une armoire à vidéoprojecteurs, utilisant la plaquette électronique FPGA Altera DE2 Board et le logiciel Quartus II 17.1. Nous appliquons les concepts de logiques combinatoires et séquentielles, implémentant la synthèse de fonctions, des compteurs et des machines d'états.

Guidés par un cahier des charges structuré, nous décomposons le projet en composants distincts, facilitant la validation et l'optimisation indépendantes.

Ce compte-rendu offre un aperçu des étapes franchies pour mener à bien ce projet, partageant notre démarche et les défis relevés pour aboutir à une solution fonctionnelle pour la gestion automatisée de l'armoire à vidéoprojecteurs.

2 cahier de charges

- L'utilisateur doit tout d'abord choisir le vidéoprojecteur qu'il souhaite rendre ou emprunter. Lors d'une restitution, il faudra vérifier que l'emplacement correspondant au vidéoprojecteur choisi est bien libre. Lors d'un emprunt, il faudra s'assurer que le vidéoprojecteur choisi est bien présent dans son emplacement. Le numéro du vidéoprojecteur choisi devra être visualisable sur un afficheur.
- Une fois le choix effectué et validé, l'utilisateur dispose de 20 secondes pour taper un code qui lui permettra d'ouvrir la serrure correspondant à l'emplacement choisi. Le temps restant pour taper le code devra être visualisable sur un afficheur.
- L'application comportera un reset général permettant, à tout moment de la procédure, de remettre le système dans son état initial, c'est-à-dire toutes les serrures fermées et les temporisations initialisées à 20 secondes.
- En cas d'erreur, l'utilisateur devra renouveler la procédure de choix. Si le code est correct, la serrure s'ouvre durant 20 secondes afin de lui permettre de restituer ou d'emprunter le vidéoprojecteur choisi. Au bout de 20 secondes, la serrure se referme et un nouveau cycle peut commencer. Le temps pendant lequel la serrure reste ouverte devra être visualisable sur un afficheur.
- S1 Le cahier des charges impose que le développement soit réalisé uniquement à l'aide de systèmes séquentiels synchrones.

3 principe de réalisation

pour répondre de façon optimale nous allons développer notre application en quatre blocs qui représentent les quatres étapes de son utilisation

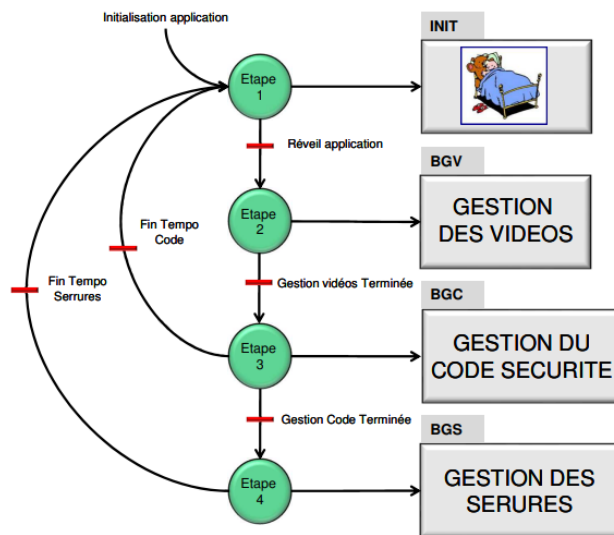


FIGURE 1 – Description de votre image.

ou chacun des blocs représentés correspond à l'une des étapes de l'utilisation de notre application et ils sont tous reliés grâce à un bloc 'chef d'orchestre' qui permet d'avoir qu'un seul bloc actif à la fois. Nous allons les expliciter ci-dessous :

Étape 1 : Initialisation

Lorsqu'elle n'est pas utilisée, l'armoire est en mode « sommeil ». Elle est réveillée par l'appui sur l'un des boutons de la carte.

Étape 2 : Choix de l'emplacement

L'utilisateur peut alors choisir, à l'aide de deux boutons (un pour l'incrément, l'autre pour choisir entre le comptage et le décomptage), le numéro de l'emplacement du vidéoprojecteur qu'il souhaite emprunter ou restituer. Le numéro de cet emplacement s'affiche sur l'un des afficheurs sept segments de la carte. En utilisant un interrupteur, il sélectionne l'emprunt ou la restitution, puis valide son choix. Un contrôle est effectué : l'emplacement choisi doit être libre pour une restitution et occupé pour un emprunt. Si l'emprunt ou la restitution n'est pas possible, le système revient au choix de l'emplacement. S'il est possible, le choix validé est affiché sur un second afficheur, et on passe à l'étape de la saisie du code.

Étape 3 : Saisie du code

L'utilisateur dispose de vingt secondes pour saisir son code. Le décompte est affiché sur la carte via deux afficheurs sept segments. Si, au bout des vingt secondes, la séquence correcte n'a pas été détectée, le système retourne à l'étape initiale. Si elle est détectée, on passe à l'étape de l'ouverture de la serrure.

Étape 4 : Ouverture de la serrure

La serrure correspondant au choix validé s'ouvre pendant vingt secondes, durant lesquelles l'utilisateur peut emprunter ou restituer le vidéoprojecteur. Comme à l'étape de saisie du code, le décompte est affiché sur la carte via deux afficheurs sept segments. À la fin du décompte, le système revient à l'étape initiale : la serrure se referme.

Chacun des blocs principaux contient quelques modules élémentaires qui permettent la gestion des relations entrées/sorties. On détaillera les entrées/sorties dont on dispose puis nous

allons réaliser les différent modules pour pouvoir les relier et obtenir notre résultat final. **entrées**

sur la carte DE1 on dispose de :

- 9 switchs.
- 4 boutons poussoir.
- et sur l'armoire (virtuelle dans le cas de notre developpement)
- 6 capteurs de presence

sorties

- 6 afficheur 7 segment
- et aussi on a une information comme sortie c'est
- la commandes d'ouverture fermeture des serrures

shema de réalisation des different blocs

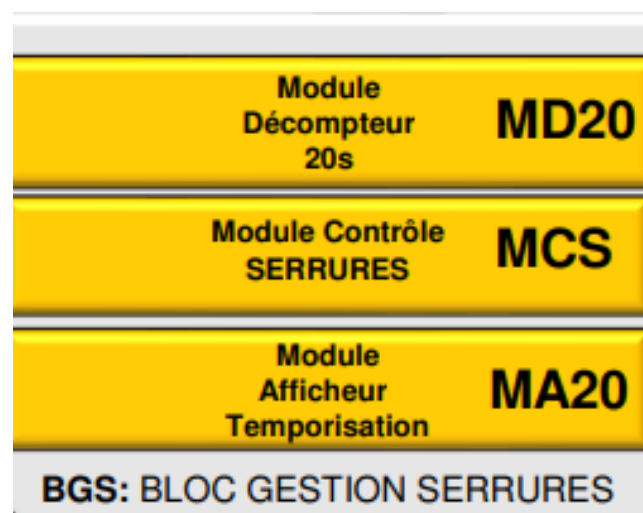
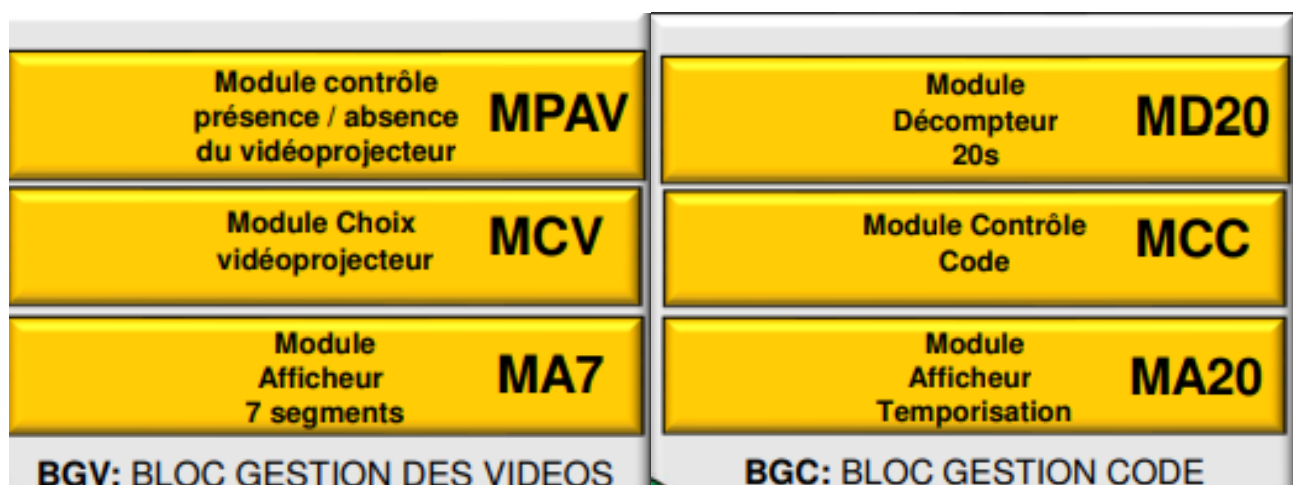


FIGURE 2 – Constituant des different blocs.

nous allons détailler les entrées et sortie des blocs entiers plus trad commençant d'abbord par réaliser chacun des modules élémentaires et le tester independemment

4 Description des différents module

4.1 Module MA7 afficheur 7 segments

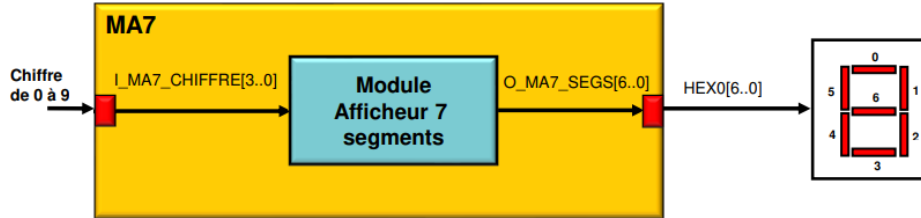


FIGURE 3 – module MA7.

ce module prend comme entrée un nombre décimal codé en binaire sur 4 bits et génère 7 signaux binaires qui commandent les segments de l'afficheur.

4.2 Description du Module MCV Compteur/Décompteur 1-6

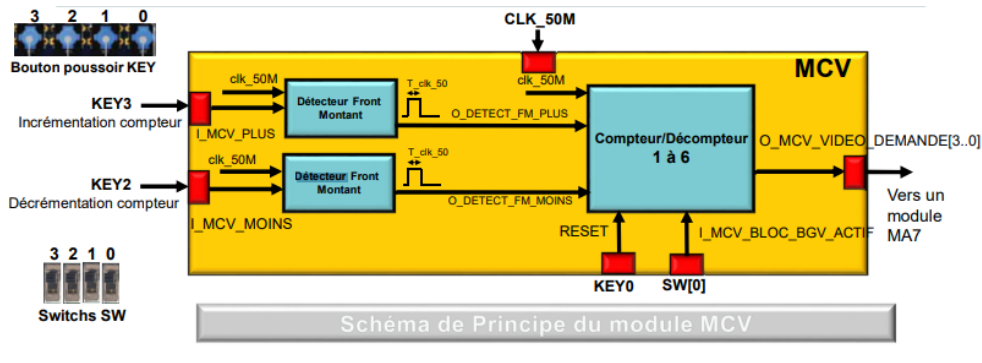


FIGURE 4 – module MCV.

Ce composant doit offrir à l'utilisateur la possibilité de sélectionner le numéro du vidéoprojecteur qu'il souhaite rendre ou emprunter, avec des numéros allant de 1 à 6. Une fois le choix effectué, ce numéro sera transmis à l'afficheur MA7, déjà programmé auparavant. Les entrées/sorties du module sont les suivantes :

En Entrée :

- I_MCV_BLOC_BGV_ACTIF : Signal d'activation du bloc 2 (BGV)
- I_MCV_PLUS (KEY3) : Incrémentation du compteur
- I_MCV_MOINS (KEY2) : Décrémentation du compteur
- I_MCV_BP_RESET (KEY0) : Reset compteur
- CLK_50M : Horloge de synchronisation

En Sortie :

- O_MCV_VIDEO_DEMANDE[3..0] : Numéro du vidéo en cours de sélection (sous 4 bits)

voir annexe page 1X pour détail de construction de ce bloc

4.3 description du module MD20



FIGURE 5 – module MD20.

Ce module permet le décomptage de 20 à 0. Il est utilisé dans les blocs BGC et BGS pour réaliser le décompte de 20 secondes. Dans le cas du BGC, si le bon code n'a pas été entré par l'utilisateur avant la fin de la temporisation, il doit signaler que le temps accordé est écoulé.

Le module se compose de trois entrées :

- un bit d'activation du module (**ACTIF**),
- un bit de réinitialisation (**RESET**),
- un bit d'horloge (**CLK_1HZ**) pour la synchronisation de l'horloge.

En sortie :

- un bit **TIME_OUT** qui enverra un signal au module concerné lorsque que le décompteur sera à 0,
- un bus de 5 bits qui correspond au temps qu'il reste pour taper le code.

4.4 Description du Module MCC

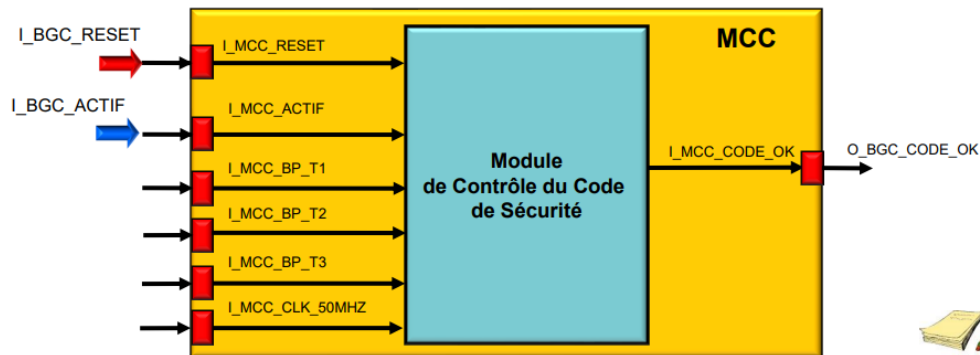


FIGURE 6 – module MCC.

le but de de module est de verifié le code secret saisi par l'utilisateur pour décider si l'ouverture des serrures doit être autorisée. Il génère une sortie signalant que le code saisi par l'utilisateur est correct.

Les entrées de ce module comprennent :

- Un bit activant le module (**ACTIF**),
- Un bit de réinitialisation (**RESET**),
- Un bit d'horloge (**CLK_1HZ**) pour synchroniser l'horloge,
- 3 bits correspondant aux 3 boutons-poussoirs utilisés pour entrer le code (1 si appuyé 0 sinon).

Si le code entré est correct avant la fin du décompte de 20 secondes, un signal de validation est transmis au bloc de gestion des serrures, permettant ainsi l'activation de celles-ci.

En cas de non-saisie du bon code avant la fin du décompte, le MCC signale cette situation par un signal, empêchant toute action jusqu'à une nouvelle validation du choix du vidéoprojecteur.

4.5 Description du Module BCD/MA20

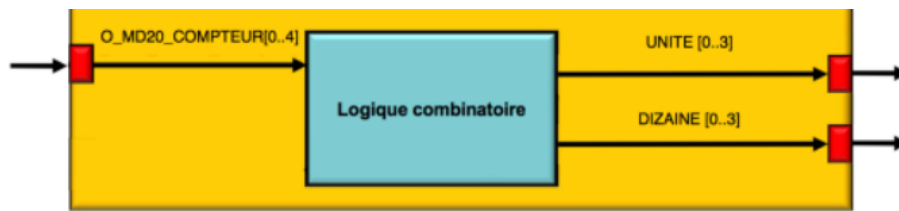


FIGURE 7 – module MA20

Ce bloc a pour fonction de convertir un nombre compris entre 0 et 20 codé en binaire provenant du bloc MD20 en deux codes (un pour les unités comprises entre 0 et 9 et un pour les dizaines comprises entre 0 et 2) pour les afficheurs sept segments. Pour y parvenir, nous avons utilisé un transcodeur convertissant un code binaire 5 bits en deux codes binaires 4 bits, chacun relié à un module MA7. En résultent donc deux signaux compatibles avec des afficheurs sept segments que nous avons envoyés à deux afficheurs adjacents sur la carte. On teste alors les modules MD20 et MA20 en associant à l'entrée initialisant le décompte à un bouton poussoir, la sortie marquant la fin du décompte à une LED et bien sûr l'affichage du décompte à deux afficheurs adjacents de la carte.

4.6 Description du Module MPAV

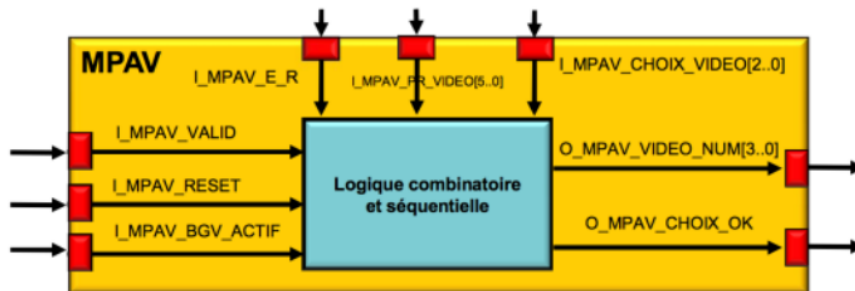


FIGURE 8 – module MPAV.

La fonction de ce bloc est de vérifier si le choix validé de l'utilisateur est cohérent (emplacement vide pour un retour de vidéoprojecteur et emplacement occupé pour un emprunt de vidéoprojecteur). Pour ce faire, ce module est composé de deux sous-modules : - le module MPAV-C : il nous permet de vérifier si le choix avant validation de l'utilisateur est cohérent, en fonction de la présence ou non de vidéoprojecteurs dans leur emplacement, ainsi que du choix de restitution ou d'emprunt de la part de l'utilisateur. Le câblage complet ainsi que les table de vérité et de Karnaugh de ce module se trouvent dans les annexes. - le module MPAV-M : il permet, à l'aide de bascules D, avec pour signal-horloge le bouton de validation, de faire passer le choix de l'utilisateur de l'état « en cours » à l'état « validé », ce-dernier état dépendant de la cohérence du choix, donnée par le module

Le module possède les entrées suivantes :

- Un bus de six bits `PR_VIDEO[5..0]` indiquant la présence du vidéoprojecteur (1) ou l'absence du vidéoprojecteur (0) dans chacun des 6 casiers.
- Un bus de trois bits `CHOIX_VIDEO[2..0]` indiquant le choix du vidéoprojecteur.
- Un bit `E_R` permettant de choisir soit d'emprunter (1) ou de restituer (0).
- Un bit de validation : `VALID`.

- Un bit d'activation **ACTIF**.
 - Un bit de **RESET**.
- Et comme sorties :
- Un bit **CHOIX_OK** indiquant si le choix de l'utilisateur est cohérent.
 - Un bus de 4 bits **VIDEO_NUM[3..0]** donnant le numéro du vidéoprojecteur choisi.

4.7 Description du Module MCS

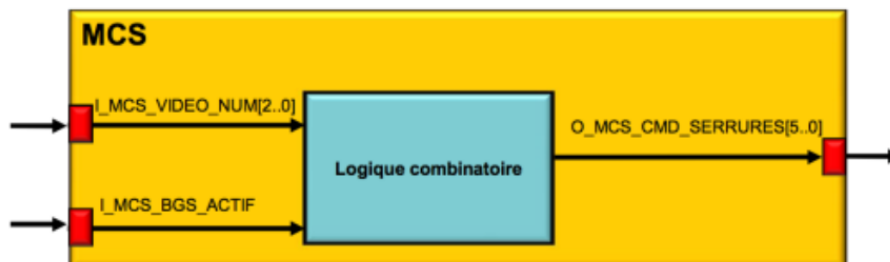


FIGURE 9 – module MCS.

Ce module a pour fonction de contrôler l'ouverture et la fermeture des serrures de l'armoire en fonction du numéro de vidéoprojecteur choisi. Les entrées du module sont les suivantes :

- Un signal d'activation **ACTIF**,
- Le numéro du choix du vidéoprojecteur validé, codé en binaire **VIDEO_NUM[2..0]**.

La sortie du module est un bus de six bits **SERRURES[5..0]**, où chaque bit correspond au numéro d'une serrure.

5 creation des blocs

apres avoir realise nos differents modules(details de conception en annexe) il est temps de les relier pour former le differents blocs de notre application

5.1 BGV

Ce bloc permet de gérer le choix des vidéoprojecteurs à rendre / emprunter. il correspon a l'étape 2 dans la conception de notre application (fig 1) il est composé des modules MCV, MPAV, et de deux MA7. Il prend en charge le processus de sélection de l'emplacement du vidéoprojecteur par l'utilisateur. Le bloc dispose de sept entrées, dont quatre sont gérées par l'utilisateur, représentant les boutons disponibles pour le choix du vidéoprojecteur, ainsi que la validation (+ le reset général). Il présente également trois sorties, comprenant deux afficheurs 7 segments et un bit confirmant le choix de l'utilisateur, marquant ainsi le passage à l'étape 3.

5.2 BGC

Ce bloc supervise la saisie correcte du code secret par l'utilisateur et la temporisation de 20 secondes, qui est affichée (correspond a l'étape 3) .Il est constitué des modules MCC, MD20, MA20, et de deux MA7. Le bloc comporte également sept entrées, mais seulement trois sont directement gérées par l'utilisateur : les trois boutons-poussoirs du code secret (+ reset général). Les sorties comprennent les deux afficheurs qui indiquent le temps restant, ainsi qu'un bit validant la saisie correcte du code et marquant le passage à l'étape suivante.

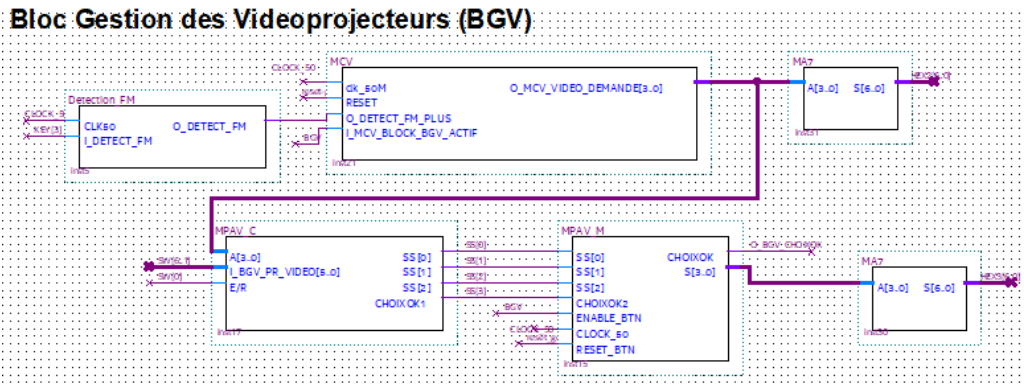


FIGURE 10 – bloc BGV.

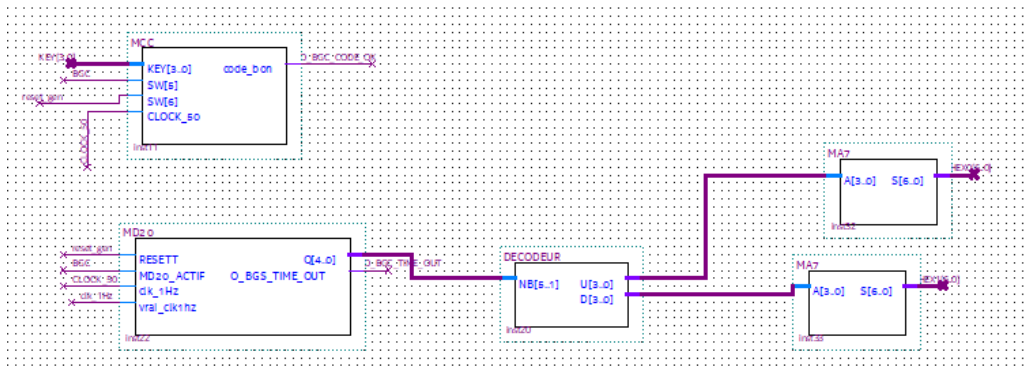


FIGURE 11 – bloc BGC.

5.3 BGS

Ce bloc supervise l'ouverture des serrures pendant 20 secondes (un décompte est affiché) (etape 4). Il est constitué des modules MD20, MA20, et MCS. Le bloc comprend quatre entrées, mais aucune n'est directement gérée par l'utilisateur. Il possède également quatre sorties : le numéro de la serrure (où tous les bits sont à 0, sauf celui correspondant à la serrure choisie), le temps restant pour emprunter ou rendre le vidéoprojecteur, ainsi qu'un time-out pour arrêter le programme.

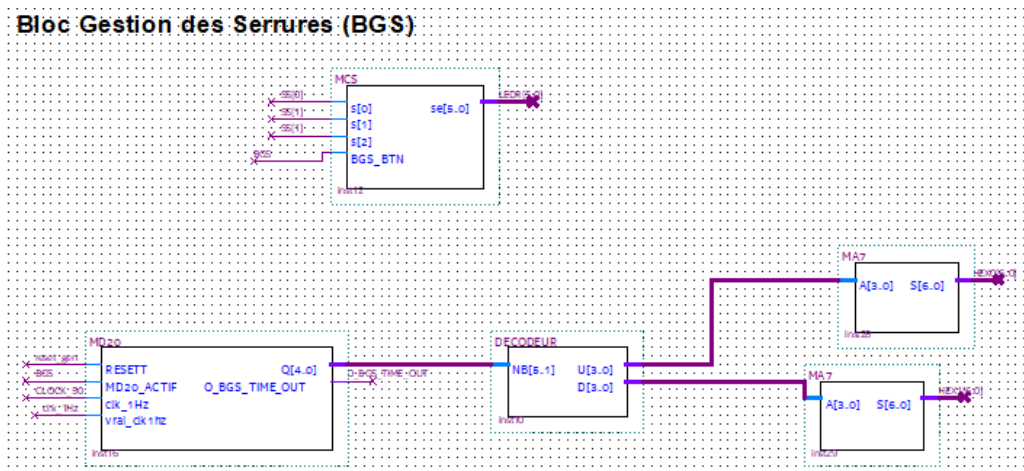


FIGURE 12 – bloc BGS.

5.4 chef d'orchestre

maintenant qu'on a tous les blocs proprement cree il ne reste que les relie dans un deroulemnt chronologique sous les 4 etapes preciser par la figure ou chque etape correspond a un bloc. c'est la ou vinet l'improtance de notre dernier bloc le chef d'orchestre c'est lui qui va donner vie a notre application en reliant les different blocs grace au element marquants le debut et fin d'activation de chaque blocs

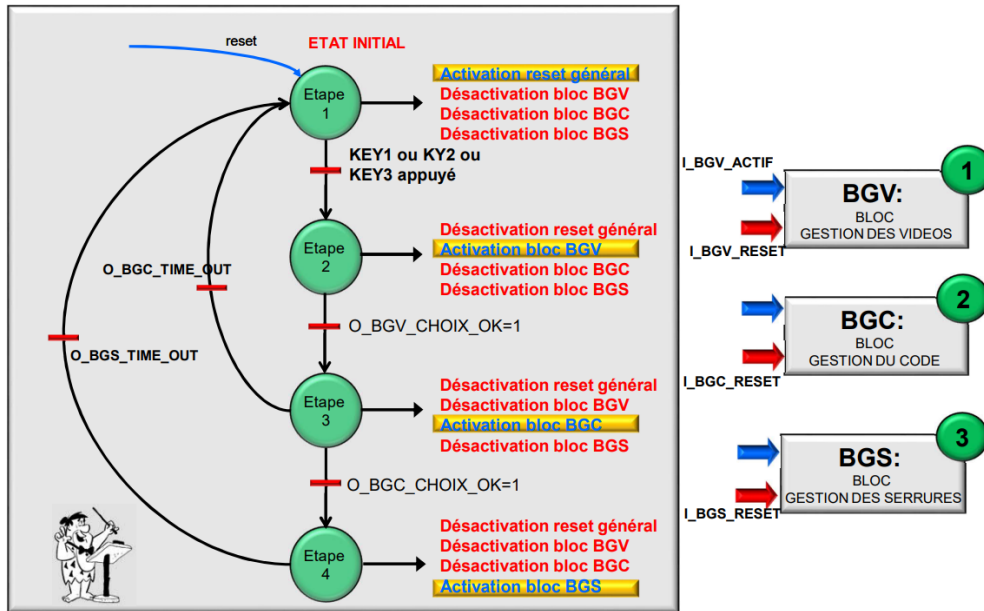


FIGURE 13 – graphe d'etats du chef d'orchestre.

grqce a cette mqchine d'etats on pourrq reqliser un blocs qui prend en entrée et genere les signaux d'activations et de resets des different blocs ainsi eremettant l'execution d'une seul tache (bloc) a la fois et d'avoir notre application finale

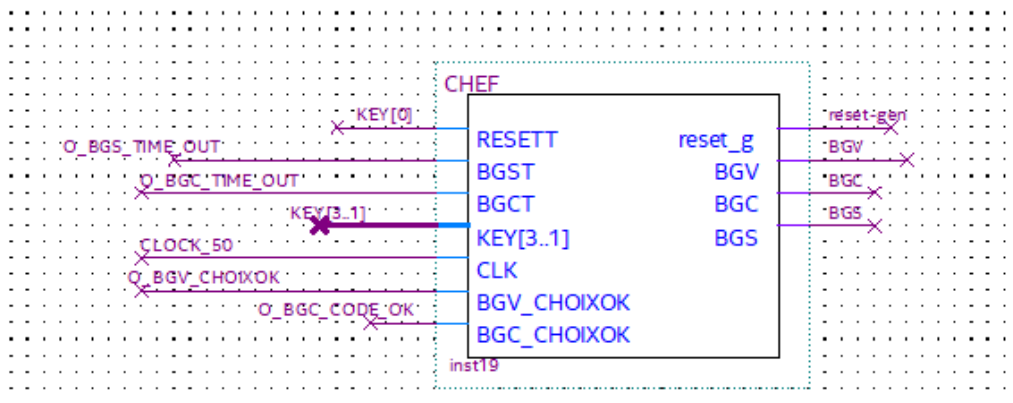


FIGURE 14 – blocs du chef d'orchestre.

6 Conclusion

Au cours de ces six séances de bureau d'étude, nous avons eu l'opportunité de mettre en œuvre les différentes approches présentées en cours, en répondant à un cahier des charges spécifique. Ce bureau d'étude nous a offert une première expérience de ce qui est attendu

d'un ingénieur, notamment la capacité à s'adapter à un logiciel et à le maîtriser, ainsi que la résolution de problèmes module par module, bloc par bloc, avec patience.

Cependant, le système que nous avons développé pourrait encore être amélioré, par exemple en permettant à l'utilisateur de créer son propre code de sécurité au lieu de l'avoir prédéterminé à 31.

Finalement nous souhaitons exprimer notre gratitude envers les enseignants et les encadrants qui ont su nous apporter leur aide tout en nous laissant suffisamment d'autonomie pour comprendre et répondre au cahier des charges.

7 Annexes

7.1 MA7

TABLE 1 – Tableau de vérité pour le pilotage d'un afficheur à segments

D_3	D_2	D_1	D_0	$s0$	$s1$	$s2$	$s3$	$s4$	$s5$	$s6$
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	1	0	0	1	1	1	1
1	0	1	1	1	0	0	1	1	1	1
1	1	0	0	1	0	0	1	1	1	1
1	1	0	1	1	0	0	1	1	1	1
1	1	1	0	1	0	0	1	1	1	1
1	1	1	1	1	0	0	1	1	1	1

alors pour réaliser ce premier module il suffit de réaliser la fonction sortie (déjà fait en TD) de chacun des segments grâce aux portes logiques sur quartus voici le schéma de réalisation du module sous quartus

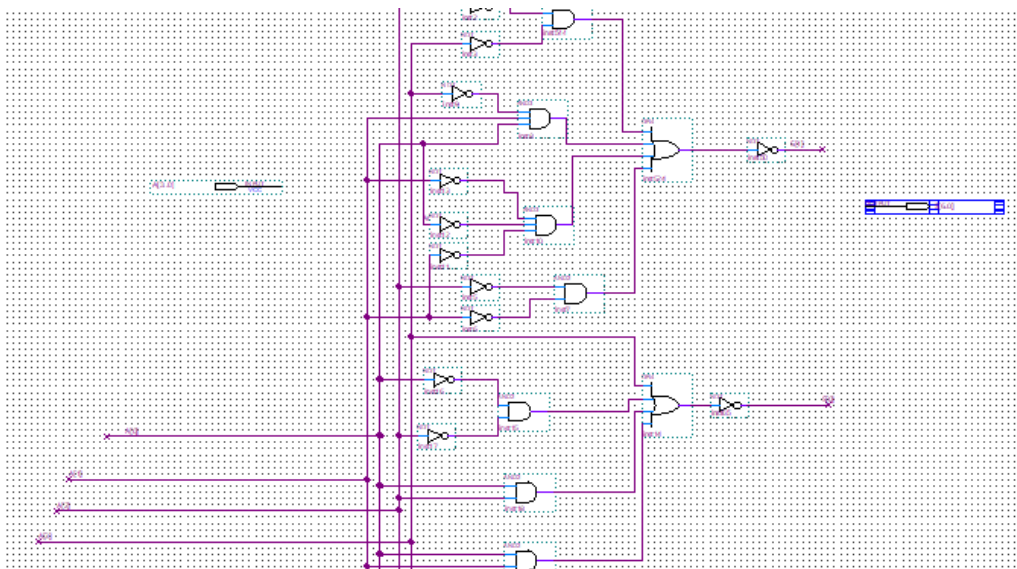


FIGURE 15 – cablage du module MA7

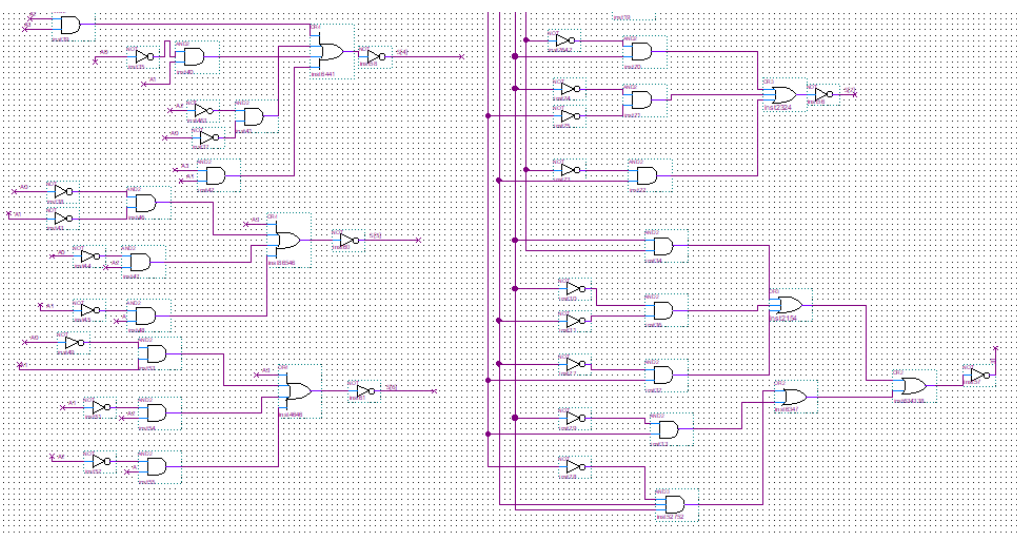


FIGURE 16 – cablage de MA7 suite

7.2 MCV

ce module comporte deux sous modules un detecteur de fronts et un compteur/decompteur 1 a 6 realisons celui ci : la table de verité traduisant notre machine d'etats du compreur/decompteur est la suivante (les etats correspond au chiffre et il sont code en binaire naturelle)

Compteur								Décompteur							
Entrées				Sorties				Entrées				Sorties			
Etat	Q3	Q2	Q1	Etat+	Q3+	Q2+	Q1+	Etat	Q3	Q2	Q1	Etat+	Q3+	Q2+	Q1+
1	0	0	1	2	0	1	0	1	0	0	1	6	1	1	0
2	0	1	0	3	0	1	1	2	0	1	0	1	0	0	1
3	0	1	1	4	1	0	0	3	0	1	1	2	0	1	0
4	1	0	0	5	1	0	1	4	1	1	0	3	0	1	1
5	1	0	1	6	1	1	0	5	1	0	1	4	1	0	0
6	1	1	0	1	0	0	1	6	1	1	0	5	1	0	1

les equation d'entrées de bascules D sont alors donnée par :

$$Q_0^+ = \overline{Q_0}$$

$$Q_1^+ = \overline{Q_2 Q_0} + Q_1 \cdot \overline{Q_0}$$

$$Q_2^+ = \overline{Q_2 \cdot Q_1} + Q_1 \cdot Q_0$$

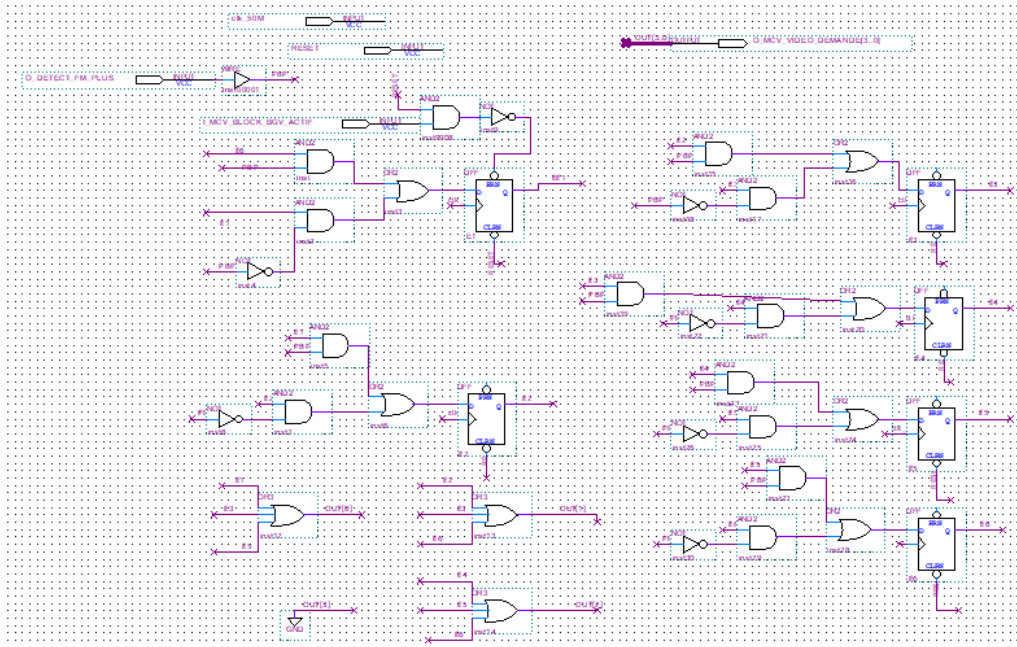
et pour le decompteur

$$Q_0^+ = \overline{Q_0}$$

$$Q_1^+ = \overline{Q_1 Q_0} + \overline{Q_2} Q_0$$

$$Q_2^+ = \overline{Q_1} Q_0 + Q_2 Q_1$$

on implemente alors ces equation grqce qu portes logique sous quartus



7.3 md20

les etats sont toujours en binaire naturelle la table de verite de ce module est

États N	Q4	Q3	Q2	Q1	Q0	États N+1	Q4+	Q3+	Q2+	Q1+	Q0+
20	1	0	1	0	0	19	1	0	0	1	1
19	1	0	0	1	1	18	1	0	0	1	0
18	1	0	0	1	0	17	1	0	0	0	1
17	1	0	0	0	1	16	1	0	0	0	0
16	1	0	0	0	0	15	0	1	1	1	1
15	0	1	1	1	1	14	0	1	1	1	0
14	0	1	1	1	0	13	0	1	1	0	1
13	0	1	1	0	1	12	0	1	0	1	1
12	0	1	1	0	0	11	0	1	0	1	0
11	0	1	0	1	1	10	0	1	0	0	1
10	0	1	0	1	0	9	0	1	0	0	0
9	0	1	0	0	1	8	0	1	0	0	0
8	0	1	0	0	0	7	0	0	1	1	1
7	0	0	1	1	1	6	0	0	1	1	0
6	0	0	1	1	0	5	0	0	1	0	1
5	0	0	1	0	1	4	0	0	1	0	0
4	0	0	1	0	0	3	0	0	0	1	1
3	0	0	0	1	1	2	0	0	0	1	0
2	0	0	0	1	0	1	1	0	0	0	0
1											
1	0	0	0	0	1	1	0	0	0	0	0
0											
0	0	0	0	0	0	0	0	0	0	0	0
0											

les equations des entrées des bascules D sont alors données par

$$Q_0^+ = \overline{Q_0} \cdot (Q_1 + Q_2 + Q_3 + Q_4)$$

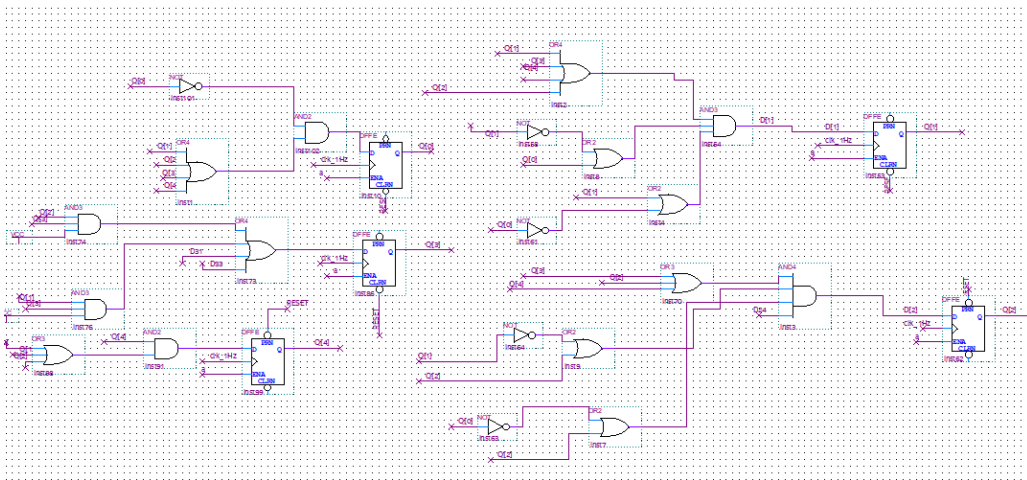
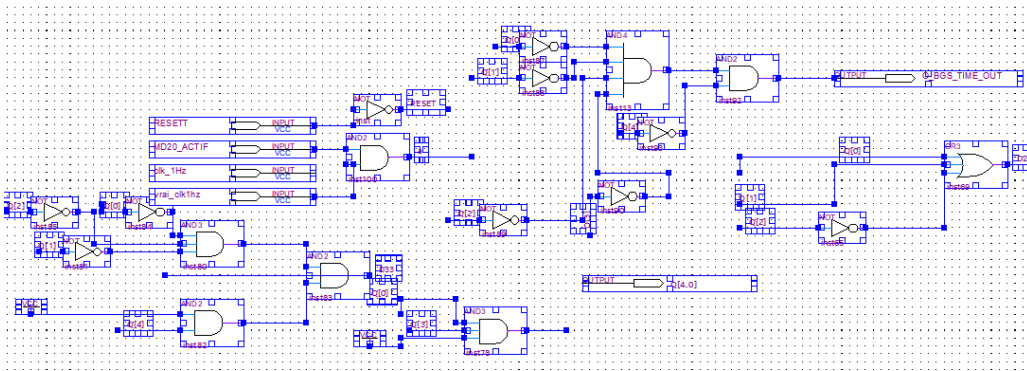
$$Q_1^+ = (\overline{Q_0} + Q_1) \cdot (Q_0 + \overline{Q_1}) \cdot (Q_1 + Q_2 + Q_3 + Q_4)$$

$$Q_2^+ = (\overline{Q_1} + Q_2) \cdot (\overline{Q_0} + Q_2) \cdot (Q_0 + Q_1 \overline{Q_2})$$

$$Q_3^+ = (Q_3 + Q_4) \cdot (\overline{Q_1} + \overline{Q_4}) \cdot (\overline{Q_2} + \overline{Q_4}) \cdot (\overline{Q_0} + \overline{Q_4}) \cdot (Q_0 + Q_1 + Q_2 + Q_4)$$

$$Q_4^+ = Q_4 \cdot (Q_0 + Q_1 + Q_2)$$

il ne reste que realiser le module sous quartus le cablage est donné par la fig



7.4 MCC

la table de verité traduisant la machine seauentiel est la suivante : les etats sont codee cette fois en codage 1 parmi 5 on en deduit les equation d'entrées suivantes pour les bascules D

Q4	Q3	Q2	Q1	Q0	K3	K2	K1	Q4+	Q3+	Q2+	Q1+	Q0+	S
0	0	0	0	1	0	X	X	0	0	0	0	1	0
0	0	0	0	1	1	X	X	0	0	0	1	0	0
0	0	0	1	0	1	X	X	0	0	0	1	0	0
0	0	0	1	0	0	X	X	0	0	1	0	0	0
0	0	1	0	0	1	X	X	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	X	X	1	0	1	0	0	0	0
0	0	1	0	0	X	1	X	0	0	0	0	1	0
0	1	0	0	0	X	X	1	0	1	0	0	0	0
0	1	0	0	0	X	X	0	1	0	0	0	0	0
1	0	0	0	0	X	X	X	1	0	0	0	0	1

$$Q_0^+ = Q_0 \cdot \overline{K_3} + Q_2 \cdot K_2$$

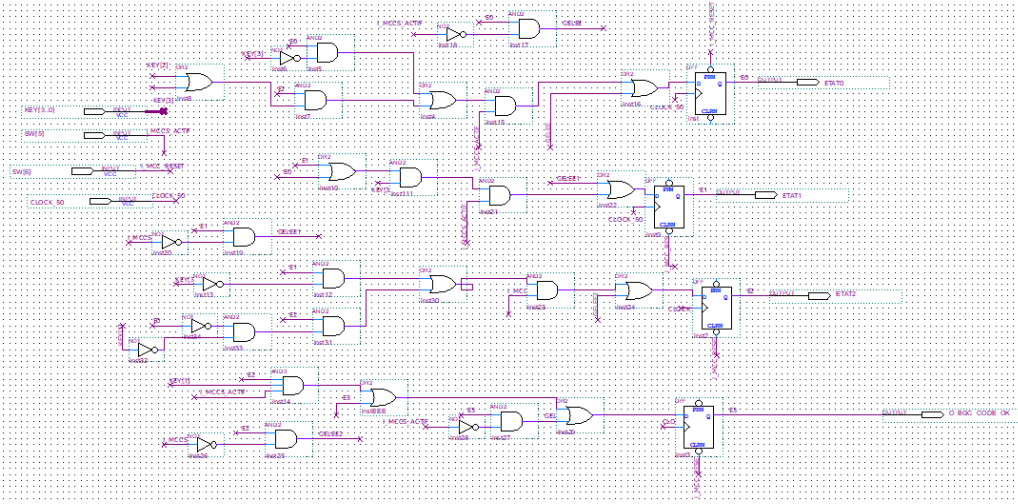
$$Q_1^+ = Q_0 \cdot K_3 + Q_2 \cdot K_3 + Q_1 \cdot K_3$$

$$Q_2^+ = Q_1 \cdot \overline{K_3} + Q_2 \cdot \overline{K_1} \cdot \overline{K_2} \cdot \overline{K_3}$$

$$Q_3^+ = Q_2 \cdot K_1 + Q_3 \cdot K_1$$

$$Q_4^+ = Q_4 + \overline{Q_3} \cdot \overline{K_1}$$

$$S = Q_4$$



7.5 MA20

la table de verité est :

	E4	E3	E2	E1	E0	D1	D0	U3	U2	U1	U0
20	1	0	1	0	0	1	0	0	0	0	0
19	1	0	0	1	1	0	1	1	0	0	0
18	1	0	0	1	0	0	1	1	0	0	0
17	1	0	0	0	1	0	1	0	1	0	0
16	1	0	0	0	0	0	1	0	1	0	0
15	0	1	1	1	1	0	1	0	1	0	0
14	0	1	1	1	0	0	1	0	1	0	0
13	0	1	1	0	1	0	1	0	0	1	0
12	0	1	1	0	0	0	1	0	0	1	0
11	0	1	0	1	1	0	1	0	0	1	0
10	0	1	0	1	0	0	1	0	0	1	0
9	0	1	0	0	1	0	0	1	0	0	1
8	0	1	0	0	0	0	1	0	0	1	0
7	0	0	1	1	1	0	0	0	1	1	0
6	0	0	1	1	0	0	0	0	1	1	0
5	0	0	1	0	1	0	0	0	1	0	1
4	0	0	1	0	0	0	0	0	1	0	1
3	0	0	0	1	1	0	0	0	0	1	1
2	0	0	0	1	0	0	0	0	0	1	0
1	0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0

les equatiions des unités et dizaines sont alors données par :

Unités :

$$u3 = \overline{E2} \cdot (E3 + E4) \cdot (\overline{E1} + E4) \cdot (E1 + \overline{E4})$$

$$u2 = (E2 + E4) \cdot (\overline{E1} + E4) \cdot (E1 + \overline{E3}) \cdot (\overline{E2} + \overline{E4})$$

$$u1 = (\overline{E2} + \overline{E3}) \cdot (\overline{E1} + \overline{E4}) \cdot (\overline{E1} + \overline{E3}) \cdot (E1 + E3 + E4)$$

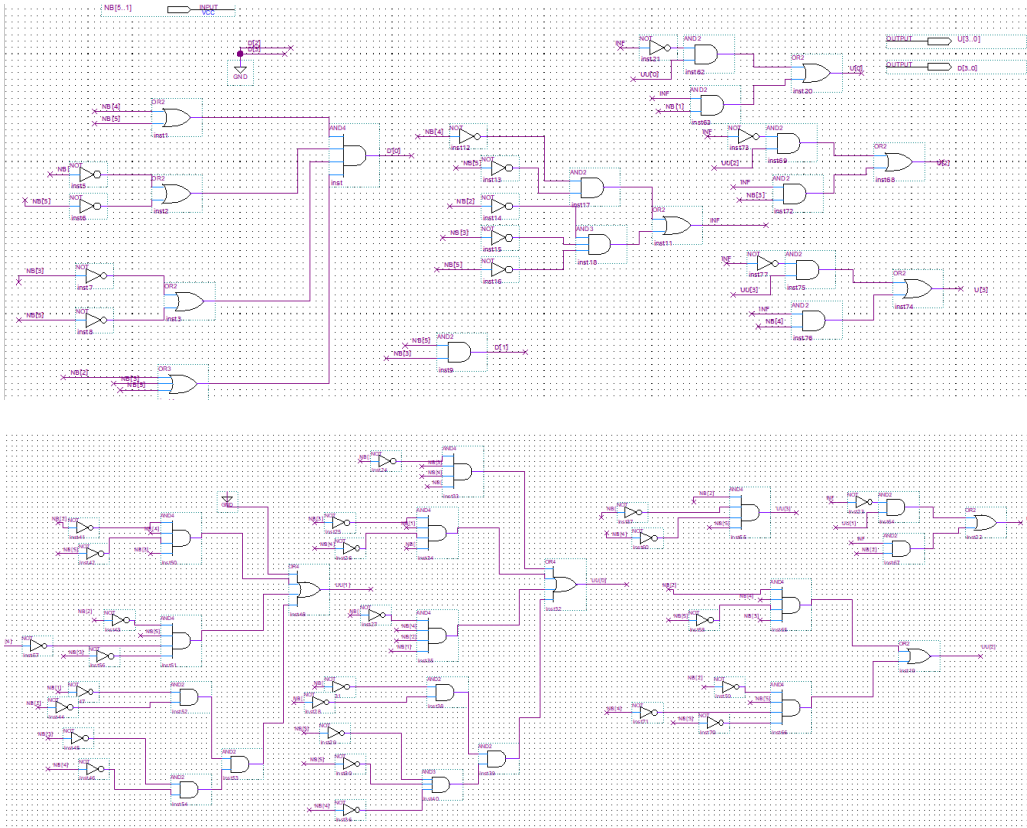
$$u0 = E0$$

Dizaines :

$$d1 = E4 \cdot E2$$

$$d0 = (E3 + E4) \cdot (\overline{E2} + \overline{E4}) \cdot (E1 + E2 + E4)$$

et pour l'implementation sous quartus on va cette fois ci essayer de faire autre chose que l'implementqtion des equqtions precedement etqblie (c'est juste pour changer de methode cabler les equations marche aussi peut etre meme mieux) on genere d'abbord un signal inf qui vaut 1 si le nombre est inferieur a 10 et 0 sinon alors si inf = 1 les unites du nombre sont le nombre lui meme et ces dizaines sont 0



7.6 MCS

la table de vérité : les equations :

E2	E1	E0	A	S[5]	S[4]	S[3]	S[2]	S[1]	S[0]
0	0	1	1	0	0	0	0	0	1
0	1	0	1	0	0	0	0	1	0
0	1	1	1	0	0	0	1	0	0
1	0	0	1	0	0	1	0	0	0
1	0	1	1	0	1	0	0	0	0
1	1	0	1	1	0	0	0	0	0

$$S[0] = A \cdot \overline{E2} \cdot \overline{E1} \cdot E0$$

$$S[1] = A \cdot \overline{E2} \cdot E1 \cdot \overline{E0}$$

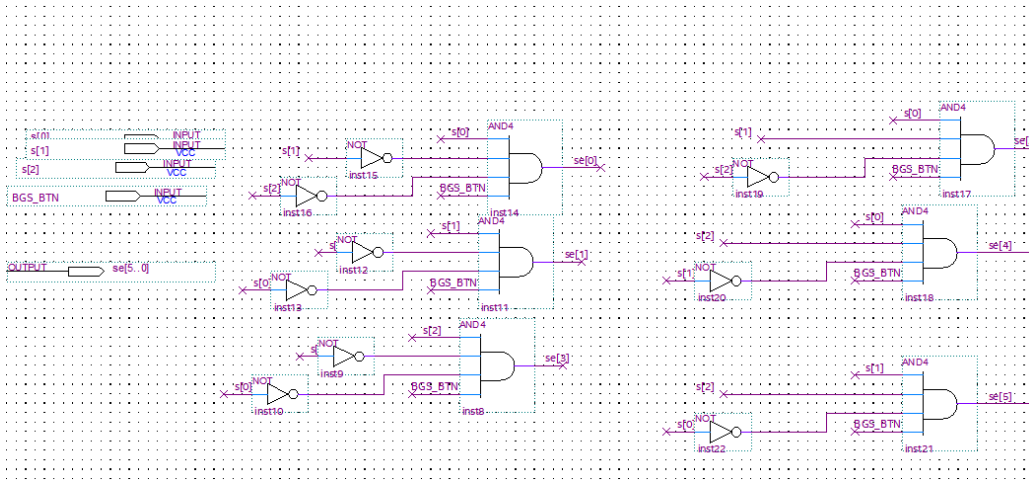
$$S[2] = A \cdot \overline{E2} \cdot E1 \cdot E0$$

$$S[3] = A \cdot E2 \cdot \overline{E1} \cdot \overline{E0}$$

$$S[4] = A \cdot E2 \cdot \overline{E1} \cdot E0$$

$$S[5] = A \cdot E2 \cdot E1 \cdot \overline{E0}$$

d'ou l'implementqtion quartus :



7.7 MPAV

P6	P5	P4	P3	P2	P1	C2	C1	C0	ER	V2	V1	V0	Choix_OK
X	X	X	X	X	0	0	0	1	1	0	0	1	1
X	X	X	X	X	1	0	0	1	0	0	0	1	1
X	X	X	X	0	X	0	1	0	1	0	1	0	1
X	X	X	X	1	X	0	1	0	0	0	1	0	1
X	X	X	0	X	X	0	1	1	1	0	1	1	1
X	X	X	1	X	X	0	1	1	0	0	1	1	1
X	X	0	X	X	X	1	0	0	1	1	0	0	1
X	X	1	X	X	X	1	0	0	0	1	0	0	1
X	0	X	X	X	X	1	0	1	1	1	0	1	1
X	1	X	X	X	X	1	0	1	0	1	0	1	1
0	X	X	X	X	X	1	1	0	1	1	1	0	1
1	X	X	X	X	X	1	1	0	0	1	1	0	1

on renseigne les entrées P0 jusqu'à P6(signaux presence de videoprojecteur) et C0 jusqu'à C2 (EMPLACEMENT CHOISI PAR L'UTILISTEUR) dans le logiciel de determination d'equation pour determiner l'equation du signal CHOIX-OK puis pour les bits du choix valide il suffit de reqliser une porte ou entre choix ok et le numero demande

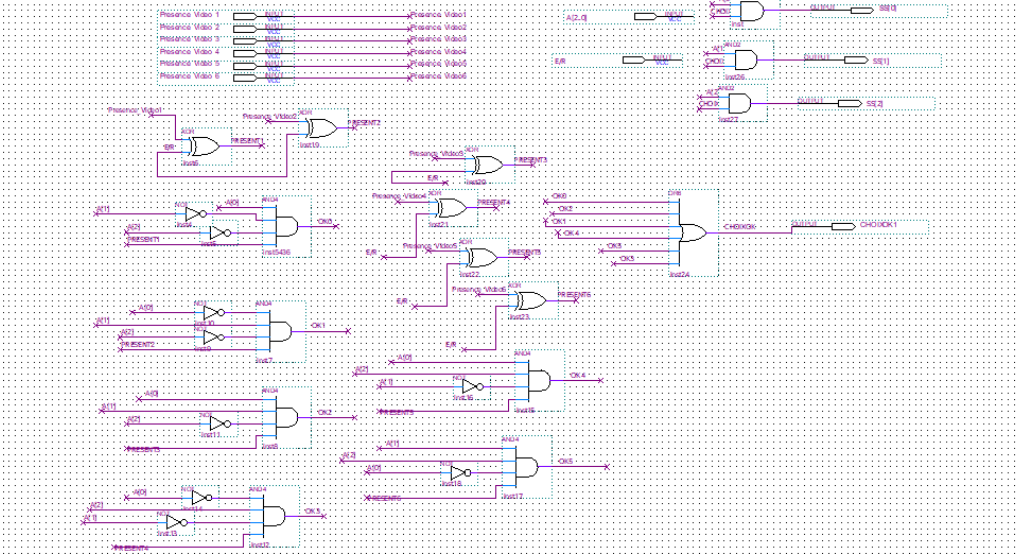


FIGURE 17 – $MPAV_C$

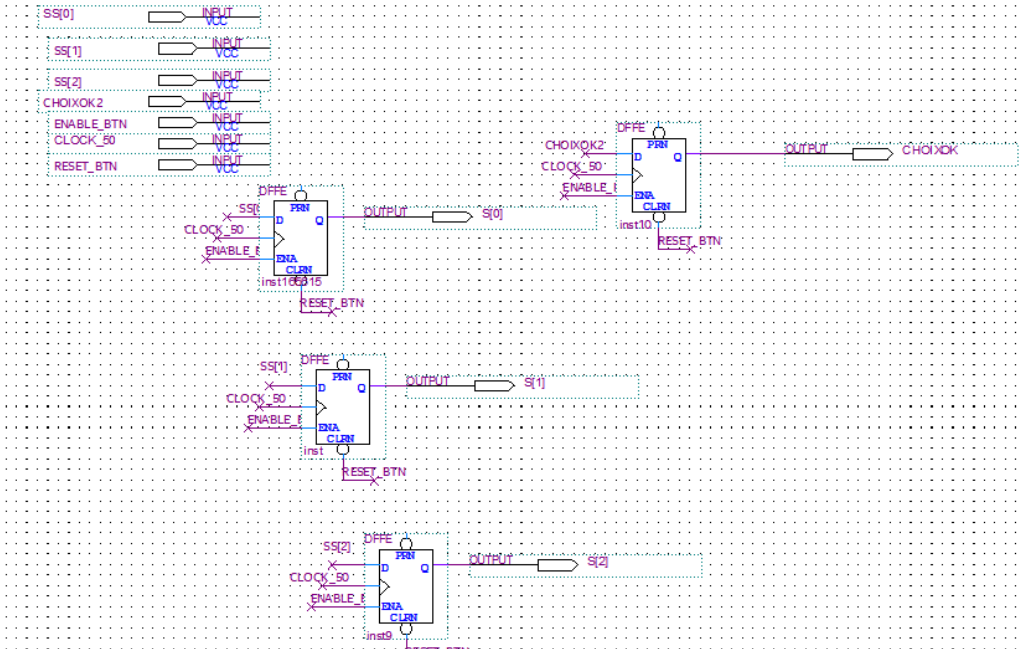


FIGURE 18 – $mpav_M$

7.8 validation intermédiaire des different modules

Au cours de nos six séances de BE, nous avons validé le fonctionnement de chaque module de manière indépendante en utilisant les composants de la carte DE1 comme entrées et sorties. Par exemple, pour le module MA7, nous simulons le nombre en entrée grâce aux interrupteurs et visualisons le résultat sur les afficheurs 7 segments. De même, pour le module MA20, nous affichons les unités sur un afficheur et les dizaines sur un autre. Pour les modules plus séquentiels comme MCS, nous utilisons les boutons-poussoirs comme entrée et les sorties sont les LEDs, où chaque LED représente un état, et cet état correspond au code correct. Nous avons suivi une approche similaire pour tous les modules créés, y compris le chef d'orchestre. Cette validation intermédiaire est cruciale, car une fois tous nos modules validés, il ne reste plus qu'à les mettre en relation, et l'application fonctionnera sûrement du premier coup.