

---

## RAPPORT DE PROJET

### *Projet d'électronique n°1 : ELECTRO-LIFT*

---

*Auteurs :*

Tom  
Robin  
Terence  
Laouïg

*Enseignant :*

M. KHOURY

GODARD  
QUERIAUX  
FOUCHIER  
ELEOUET

*En tant qu'équipe de quatre étudiants en école d'ingénierie, nous sommes plongés dans le développement d'ELECTRO-LIFT, un projet ambitieux qui repose sur la conception d'un système d'ascenseur commandé par FPGA. Sous la tutelle du Dr. Maxime SCHNEIDER, notre projet s'articule autour de l'élaboration d'une interface de contrôle capable de gérer un système d'ascenseur complexe, doté de huit étages, avec une attention particulière portée à l'étage présidentiel, qui requiert une séquence d'accès sécurisée.*

*Nos objectifs techniques s'alignent sur la réalisation d'une machine à états optimisée pour orchestrer le flux des passagers et intégrer des mesures de sécurité avancées, telles que la prévention des risques liés aux portes et la capacité de réponse aux urgences.*

*Ce projet s'inscrit dans notre parcours académique comme une application concrète des théories apprises, tout en mettant à l'épreuve notre maîtrise du VHDL et notre aptitude à la résolution de problèmes en ingénierie. La finalité est de présenter un prototype fonctionnel qui incarne notre savoir-faire technique et notre esprit collaboratif d'ingénieurs en formation.*

Nous attestons que ce travail est original, qu'il est le fruit d'un travail commun au 4 participants et qu'il a été rédigé de manière autonome.

Paris, le 20/10/2023

## Table des matières

<b>I. Objectif</b>	<b>4</b>
<b>II. Glossaire</b>	<b>4</b>
<b>A. Termes</b>	<b>4</b>
<b>B. Acronymes</b>	<b>4</b>
<b>III. L'équipe</b>	<b>4</b>
<b>A. Présentation de l'équipe</b>	<b>4</b>
<b>B. Organisation de l'équipe</b>	<b>4</b>
<b>C. Diagramme de GANTT</b>	<b>4</b>
<b>IV. Contexte et problématique</b>	<b>4</b>
<b>A. Contexte</b>	<b>4</b>
<b>B. Problématique</b>	<b>5</b>
<b>C. Spécifications techniques</b>	<b>5</b>
<b>V. Conception</b>	<b>5</b>
<b>A. Architecture fonctionnelle</b>	<b>5</b>
<b>B. Architecture matérielle</b>	<b>5</b>
<b>C. Architecture logicielle</b>	<b>5</b>
<b>VI. Développement</b>	<b>5</b>
<b>A. FS1 : Activer le mode maintenance</b>	<b>5</b>
<b>B. FS2 : Traiter les demandes</b>	<b>5</b>
<b>C. FS3 : Assurer la sécurité des passagers</b>	<b>5</b>
<b>D. FS4 : Déplacer les passagers</b>	<b>5</b>
<b>E. FS5 : Informer les passagers</b>	<b>5</b>
<b>VII. Tests et validation</b>	<b>5</b>
<b>A. FS1 : Activer le mode maintenance</b>	<b>5</b>
<b>B. FS2 : Traiter les demandes</b>	<b>5</b>
<b>C. FS3 : Assurer la sécurité des passagers</b>	<b>5</b>
<b>D. FS4 : Déplacer les passagers</b>	<b>5</b>
<b>E. FS5 : Informer les passagers</b>	<b>5</b>
<b>VIII. Bilan</b>	<b>6</b>
<b>A. État d'avancement</b>	<b>6</b>
<b>B. Pertinence de la solution technique</b>	<b>6</b>
<b>C. Bilan sur le travail d'équipe</b>	<b>6</b>
<b>IX. Sources</b>	<b>6</b>
<b>X. Annexes</b>	<b>6</b>

## I. Objectif

L'objectif de ce document est de présenter les directives d'un projet de conception d'un ascenseur numérique sur FPGA, nommé "ELECTRO-LIFT". Le document fournit une vision complète des fonctionnalités, des contraintes d'implémentation et des exigences associées à la réalisation du système d'ascenseur. Le lecteur y trouvera des détails sur le fonctionnement de l'ascenseur, notamment comment il se déplace entre les étages, comment il interagit avec les utilisateurs, et les mesures de sécurité intégrées.

## II. Glossaire

### A. Termes

Terme	Définition
Machine à états	Aussi connue sous le nom de machine d'états finis (MEF) ou automate fini est un modèle de comportement composé d'un nombre fini d'états, de transitions entre ces états et d'actions.
BDF	Un fichier de schéma utilisé par Intel Quartus Prime pour la conception de circuits avec une interface graphique. Il montre les composants et connexions dans un format visuel.
BSF	Un symbole de bloc pour Quartus Prime, représentant un sous-circuit encapsulé pour une utilisation dans des schémas bdf.

### B. Acronymes

Acronyme	Signification	Explication
FPGA	(Field-Programmable Gate array)	Un FPGA est un circuit intégré programmable qui peut être configuré après fabrication pour effectuer diverses fonctions logiques complexes en parallèle
VGA	(video graphic array)	
SMF/FSM	(State/finite machine File)	C'est une norme d'affichage informatique analogique qui définit une résolution spécifique de $640 \times 480$ pixels et une palette de 16 couleurs à l'origine.

## III. L'équipe

### A. Présentation de l'équipe



### Robin :

- *Compétences* : Codage VHDL, compilation ModelSim, simulation sur FPGA.
- *Qualités* : ATTARDé, Rigoureux, précis, méthodique et bon communicateur.

### Tom :

- *Compétences* : Quartus, BDF, machine à état, code VHDL simulation sur FPGA
- *Qualités* : Meticuleux, organisé, capable de travailler sous pression, bon en gestion de projets.

### Terence :

- *Compétences* : machine à état, simulation sur FPGA, rapport de projet.
- *Qualités* : Créatif, capable de trouver des solutions innovantes, esprit d'équipe, bon en résolution de problèmes.

### Laouïg :

- *Compétences* : Machine à état, simulation sur FPGA, compilation modelSim.
- *Qualités* : Analytique, orienté vers les détails, capable de travailler de manière autonome, bon en analyse de données.

## B. Organisation de l'équipe

Au sein de notre équipe de quatre étudiants en ingénierie, nous avons stratégiquement réparti les responsabilités pour maximiser l'efficacité et la productivité dans le cadre de notre projet ELECTRO-LIFT. Tom et Robin ont initié notre démarche en prenant en charge la fonction de maintenance FS1, établissant les fondations nécessaires pour assurer la gestion adéquate de l'ascenseur.

Dans le même temps, Laouïg et Terence se sont attelés aux défis techniques liés au traitement des demandes d'étages (FS2) et à l'acheminement des passagers (FS4). Leurs efforts ont été essentiels pour intégrer la logique de commande qui régit l'interaction utilisateur et la navigation entre les étages.

Par la suite, Tom a pris la responsabilité de renforcer la sécurité des passagers, en implémentant un système ingénieux à base de capteur à ultrasons qui empêche le départ de l'ascenseur en présence d'obstacles à proximité. Robin, quant à lui, a su relever le défi de communiquer efficacement avec les utilisateurs en affichant l'état actuel de l'ascenseur, un aspect crucial pour l'expérience utilisateur.

Enfin, Laouïg s'est consacré au module complémentaire du projet, enrichissant notre système par des fonctionnalités avancées d'information sur l'état de l'ascenseur, contribuant ainsi à une transparence et une interaction accrues avec l'utilisateur final.

Fusionner toutes ces fonctions distinctes en une seule solution cohérente représentait un véritable défi technique, compte tenu de la complexité intrinsèque de chaque module et des multiples conditions à respecter. Malgré cela, notre collaboration et notre détermination nous ont permis de relever ce défi avec brio, aboutissant à un système intégré qui incarne notre engagement envers l'innovation et l'excellence en ingénierie.

## C. Diagramme de GANTT

	Semaine 1	Semaine 2	Semaine 3	Semaine 4
Setup et familiarisation				
Contrôle d'étages				
Interface utilisateur				
Mode maintenance				
Traitement des demandes				
Sécurité des passagers				
Déplacement des passagers				
Informations aux usagers				
Etat du réseau				
Tests et intégration				
Optimisation et peaufinage				
Documentation et préparation à la soutenance				
Révision finale et tests				

*Figure 1 : Diagramme de Gantt*

## IV. Contexte et problématique

### A. Contexte

Les ascenseurs jouent un rôle crucial dans la facilitation de la mobilité verticale, en particulier dans les zones urbaines densément peuplées et les immeubles de grande hauteur. Ils ont des implications économiques, car ils permettent la construction de bâtiments plus hauts, optimisant ainsi l'utilisation de l'espace au sol. Sur le plan sociétal, les ascenseurs facilitent l'accès pour les personnes à mobilité réduite et contribuent à la conception inclusive des bâtiments.

La technologie des ascenseurs a évolué au fil du temps, depuis les premiers ascenseurs à vapeur du 19<sup>e</sup> siècle jusqu'aux ascenseurs modernes à commande électronique. La simulation et la modélisation numériques, comme celle décrite dans ce projet, sont des étapes récentes dans l'évolution de cette technologie, permettant de tester et de perfectionner les systèmes d'ascenseurs avant leur mise en œuvre réelle.

### B. Problématique

Notre projet d'électronique répond aux défis croissants posés par l'urbanisation rapide et la densification des villes. À mesure que les métropoles se développent et que la construction de bâtiments de grande hauteur devient courante, la nécessité d'une mobilité verticale efficace et fiable devient primordiale. L'ascenseur, dans ce contexte, n'est pas seulement un luxe, mais une nécessité pour assurer la fluidité du mouvement des personnes.

De plus, dans une société de plus en plus consciente des besoins des personnes à mobilité réduite, l'importance d'un ascenseur bien conçu et sûr est plus pertinente que jamais. Assurer l'accessibilité pour tous, quel que soit leur âge ou leur capacité physique, est une préoccupation majeure des sociétés modernes.

Enfin, la sécurité est devenue une préoccupation centrale dans tous les aspects de la vie quotidienne. Les incidents liés aux ascenseurs peuvent avoir des conséquences tragiques, d'où l'importance de garantir que chaque aspect, de la détection d'obstacles à la possibilité d'appeler les secours, soit impeccamment conçu.

Dans ce contexte, notre projet "ELECTRO-LIFT" s'efforce de concevoir un système d'ascenseur qui répond non seulement aux besoins fonctionnels, mais aussi aux attentes sociétales en matière de sécurité, d'accessibilité et d'interaction technologique.

## C. Spécifications techniques

- **Système d'ascenseur :** L'ascenseur doit pouvoir se déplacer séquentiellement d'un étage à l'autre, allant jusqu'à l'étage le plus élevé demandé, puis redescendre au rez-de-chaussée (RDC). Il doit montrer à tout moment à quel étage il se trouve.
- **Nombre d'étages :** Le système comporte huit étages, du RDC au 7<sup>e</sup> étage. Le dernier étage est l'étage présidentiel, accessible uniquement par le détenteur du code.
- **Mode Maintenance :** Ce mode peut être activé à tout moment en appuyant simultanément sur les boutons KEY0 et KEY1 deux fois de suite. En mode maintenance, l'ascenseur est bloqué à son étage actuel, et toutes les LEDs doivent clignoter à une fréquence de 1 Hz. La vitesse de l'ascenseur peut être ajustée avec un potentiomètre dans ce mode, entre 0,5 seconde et 5 secondes par étage.
- **Demandes d'arrêt :** Elles sont émises via des boutons poussoirs sur une breadboard (considéré comme l'intérieur de la cabine) et aussi sur les switchs directement sur la DE 10-Lite (considéré comme l'extérieur de la cabine). L'ascenseur affiche sa position actuelle à l'aide de LEDs sur la carte DE 10-Lite.
- **Restrictions d'accès :** Les utilisateurs dans la cabine peuvent se rendre de l'étage 0 (RDC) à l'étage 6. L'accès au 7<sup>e</sup> étage nécessite une séquence spécifique de touches. (KEY0 KEY 0 KEY1 KEY0 KEY0)
- **Mesures de sécurité :** La porte de l'ascenseur ne se ferme pas si un obstacle est détecté à moins de 50 cm, grâce à un capteur ultrason. Les passagers peuvent appeler les secours en appuyant simultanément sur les boutons KEY0 et KEY1.
- **Informations aux usagers :** Les demandes d'arrêt sont indiquées par des LEDs sur la breadboard, pilotées par des bascules JK. L'étage actuel est affiché sur l'afficheur HEX0, et une animation sur d'autres afficheurs montre l'état et la fermeture des portes, ainsi que la distance de l'obstacle que détecte le capteur ultrason. Un buzzer émet un bip à chaque passage d'étage.

## V. Conception

### A. Architecture fonctionnelle

- **Se déplacer séquentiellement entre les étages :** L'ascenseur doit pouvoir monter et descendre entre les étages de manière séquentielle, atteignant l'étage demandé le plus élevé avant de redescendre au RDC.
- **Afficher la position actuelle :** Le système doit indiquer en permanence à quel étage se trouve l'ascenseur, permettant aux utilisateurs d'en être informés.
- **Répondre aux demandes d'arrêt :** L'ascenseur doit pouvoir s'arrêter à un étage spécifique en réponse à une demande d'utilisateur.
- **Restreindre l'accès à certains étages :** Il doit y avoir une fonction permettant de limiter l'accès à des étages spécifiques, comme l'étage présidentiel, à moins qu'un code ou une séquence spécifique ne soit entré.
- **Assurer la sécurité des passagers :** L'ascenseur doit détecter les obstacles lors de la fermeture des portes et empêcher la fermeture si un obstacle est détecté. De plus, il doit fournir une option pour appeler les secours en cas d'urgence.
- **Entrer et sortir du mode maintenance :** Le système doit permettre aux utilisateurs d'activer un mode maintenance, qui offre des fonctionnalités supplémentaires comme l'ajustement de la vitesse de l'ascenseur.
- **Informier les utilisateurs :** L'ascenseur doit fournir des informations visuelles et sonores aux utilisateurs, notamment sur les demandes d'arrêt, l'état des portes, et le passage d'un étage.

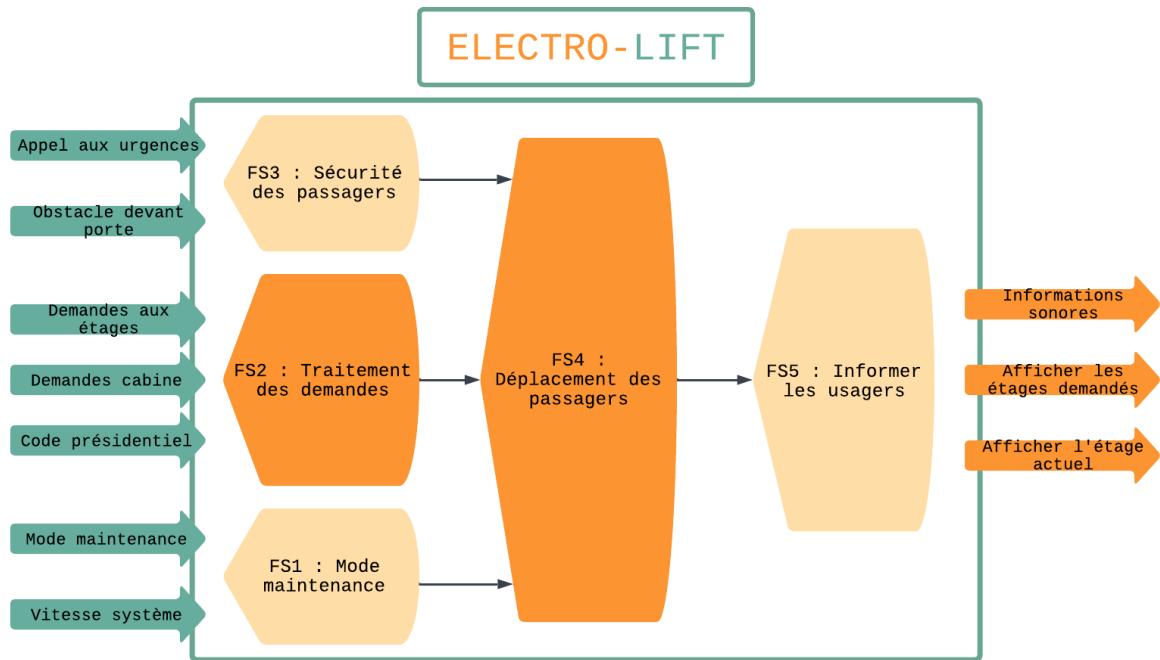
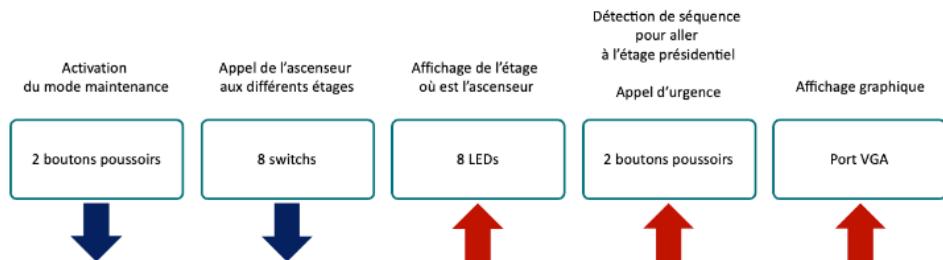


Figure 2 : Schéma circuit redressement double alternance

## B. Architecture matérielle

### Sur la DE 10-Lite : vue complète des différents étages



### Sur la breadboard : intérieur de la cabine

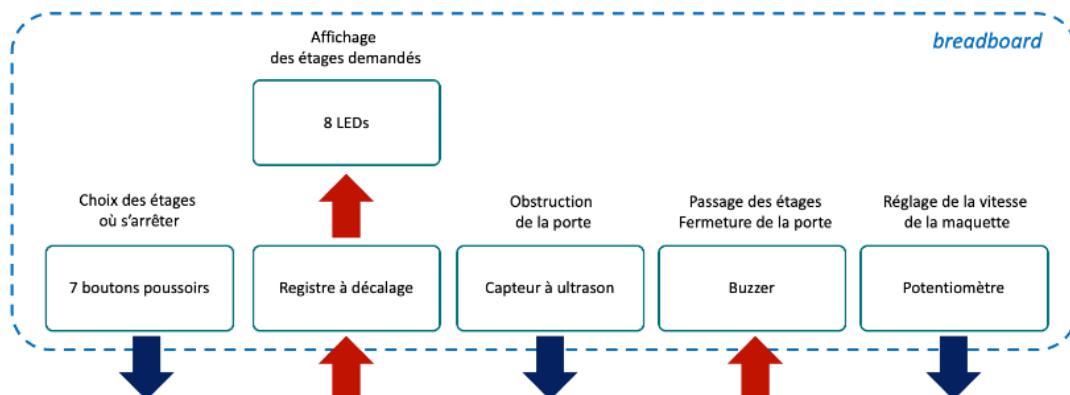


Figure 3 : Architecture logicielle demandée par le cahier des charges

- **FPGA (DE 10-Lite)** : C'est le cœur du projet. Un FPGA (Field-Programmable Gate Array) est un dispositif programmable qui permet de concevoir des circuits numériques personnalisés. Dans ce contexte, il est utilisé pour simuler le comportement et la logique de l'ascenseur.

- **LEDs** : Elles sont utilisées pour indiquer la position actuelle de l'ascenseur et d'autres informations. Par exemple, la LEDR0 pourrait indiquer le RDC, tandis que la LEDR7 pourrait indiquer le 7ème étage (étage présidentiel).

- **Boutons pousoirs sur une breadboard** : Ils sont utilisés pour émettre des demandes d'arrêt à différents étages. L'utilisation de boutons pousoirs rend l'interaction avec le système facile et intuitive.

- **Capteur à ultrasons** : Il sert à détecter les obstacles devant la porte de l'ascenseur. Si un obstacle est détecté à moins de 50 cm, la porte ne se ferme pas.

- **Potentiomètre** : Il est utilisé pour régler la vitesse de l'ascenseur en mode maintenance.

- **Bascules JK** : Ce composant est utilisé pour piloter les LEDs sur la breadboard et retenir leurs états.

- **Afficheurs HEX (HEX0 à HEX5)** : Ils servent à afficher visuellement l'étage actuel de l'ascenseur, l'état des portes et la distance de l'obstacle que détecte le capteur ultrason.

- **Buzzer** : Il émet un bip sonore pour notifier les passagers du passage à chaque étage ou d'une situation d'urgence.

Le FPGA est le centre de ce système. Il reçoit des informations venant des boutons pousoirs, du potentiomètre, et du capteur à ultrasons.

Les LEDs, les afficheurs HEX, le buzzer, et l'écran VGA sont pilotés par le FPGA pour fournir des sorties visuelles et sonores en fonction de la logique programmée.

Tous ces composants sont programmés et interconnectés grâce au logiciel Quartus Prime Lite. Quartus est un outil de conception pour les circuits FPGA et permet de développer, tester et programmer des designs numériques complexes. Dans le cadre de ce projet "ELECTRO-LIFT", Quartus Prime Lite joue un rôle essentiel pour assurer que chaque composant fonctionne et interagit comme prévu. De plus, une grande partie de la logique derrière le fonctionnement de l'ascenseur, notamment les transitions entre les étages, la gestion des appels d'étages, et les différentes fonctionnalités de sécurité et d'affichage, est réalisée grâce à des machines à état.

Les machines à état sont des modèles de conception qui permettent de représenter les différents états d'un système et les transitions entre ces états. Elles sont particulièrement utiles pour gérer la complexité d'un système comme un ascenseur, où de nombreuses actions et transitions dépendent de l'état actuel et des entrées utilisateurs.

## C. Architecture logicielle

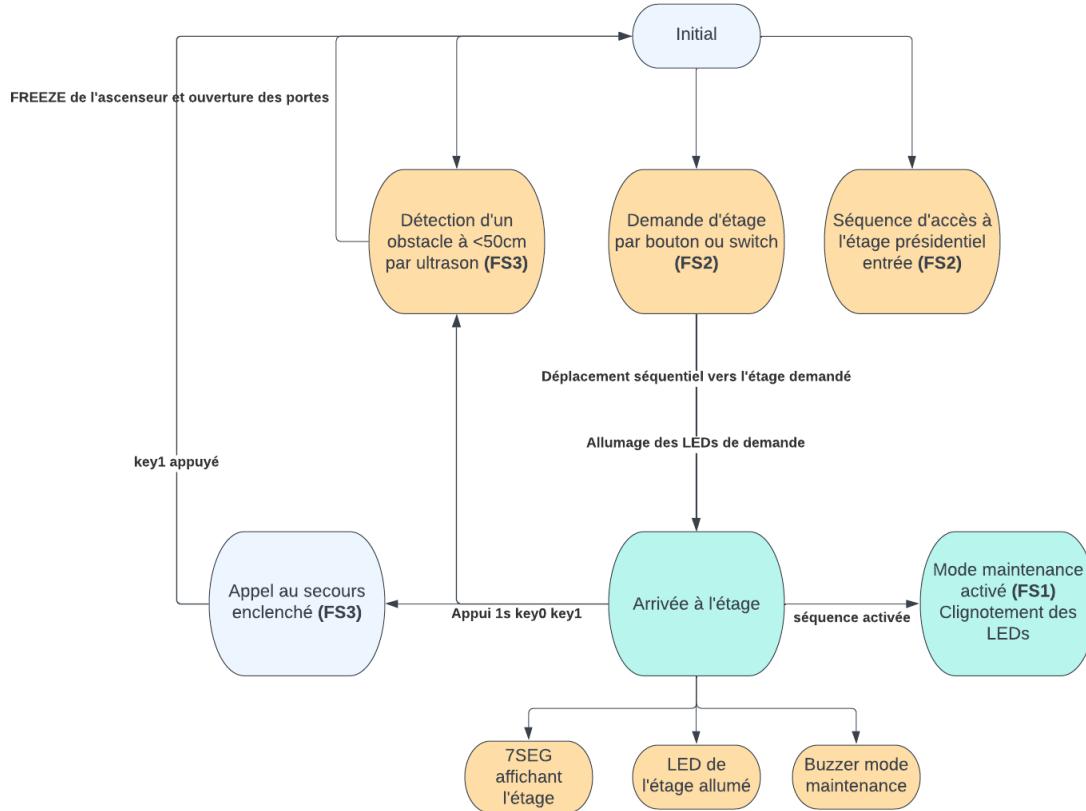


Figure 4 : Architecture logicielle du projet ELECTRO-LIFT

Notre projet a été conçu à l'aide de Quartus Prime Lite grâce à des fichiers .bsf .vhdl et .smf.

Tous nos fichiers et symboles sont centralisés dans notre fichier .bdf principal où tous nos modules sont connectés entre eux et liés à notre architecture matérielle. De plus, nous avons utilisé GitHub pour avancer en parallèle et en équipe, conservant notre avancée. Cela s'est avéré utile quand nous voulions développer une nouvelle branche du projet.

## VI. Développement

### A. FS1 : Mode maintenance

Le premier module concerne la mise en mode maintenance de notre ascenseur. L'utilisation de machine à état va grandement nous simplifier la vie dans la conception de module de type "automate", changeant d'état selon les différentes entrées de la machine.

Son implémentation va être donc centrée sur l'utilisation d'une machine à état avec deux entrées, KEY0 et KEY1 et simplement renvoyer en sortie **maintenance = 1** à l'état bas du deuxième appui simultané de KEY0 et KEY1.

Une fois arrivé à cet état **press2\_down** le mode maintenance est activé jusqu'à ce que le bouton KEY0 soit appuyé, revenant ainsi à l'état initial et répétant le cycle.

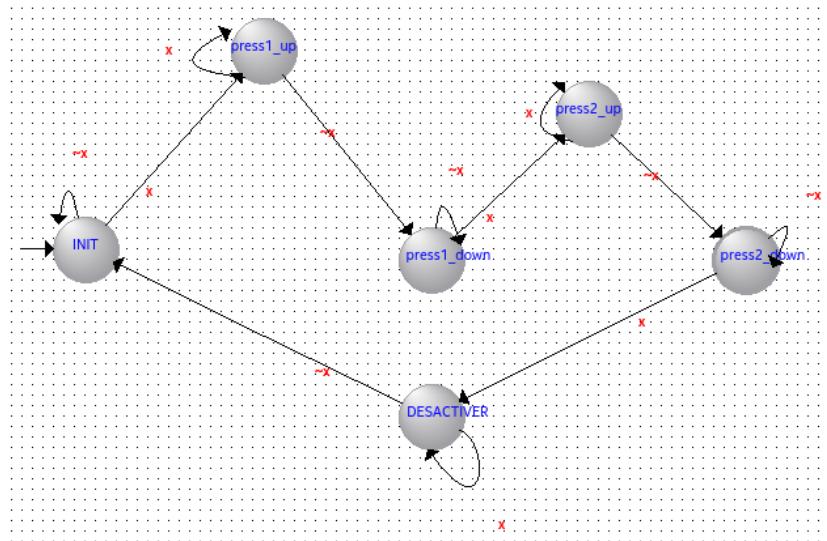


Figure 5 : SMF du module FS1

Ce fichier .smf, nommé "Maintenance", réagit à quatre signaux d'entrée: un signal de réinitialisation, une horloge, et deux signaux nommés KEY0 et KEY1. La réinitialisation force la machine à revenir à son état initial et met la sortie Z à zéro. La sortie Z change selon l'état de la machine, qui évolue en fonction des combinaisons des signaux des boutons KEY0 et KEY1. Lorsque la série de touches expliquée plus haut est saisie (deux fois KEY0 et KEY1 simultanément), le mode maintenance s'active ("press2\_down").

La machine à états utilise deux processus principaux: un pour changer l'état en fonction de l'horloge et un autre pour définir la logique de transition et de sortie basée sur l'état actuel et les entrées. En cas d'état indéfini, la machine signale une erreur et assigne une valeur indéterminée à la sortie Z.

Quand **maintenance** = 1 l'ascenseur va arrêter sa marche et toutes les LEDs de la breadboard (intérieur de l'ascenseur) et du FPGA (extérieur de l'ascenseur) vont se mettre à clignoter à intervalles réguliers d'une seconde (grâce à un diviseur d'horloge 1Hz).

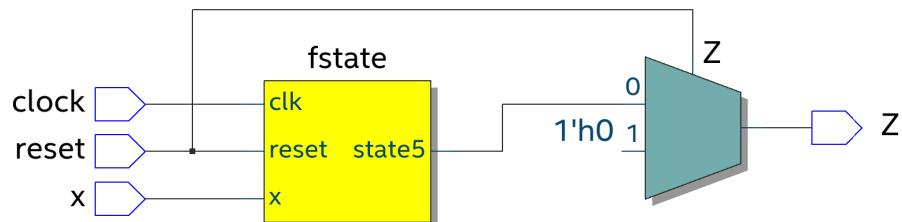


Figure 6 : RTL du mode maintenance (FS1)

Sur le RTL du FS1, on voit l'entrée "x" représentant KEY0 et KEY1 couplé par une porte AND. La clock permet de cadencer la DE 10-Lite à  $50 \times 10^6$  tour d'horloge par seconde, pour une réactivité extrêmement rapide. La sortie du composant "z" est l'activation ou non du mode maintenance.

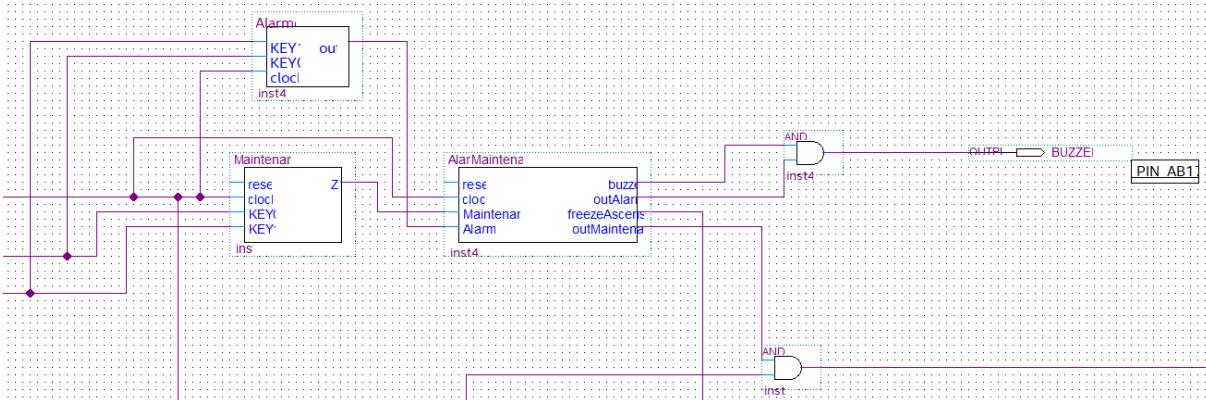


Figure 7 : BDF du mode maintenance (FS1)

Sur le fichier BDF de notre projet, on peut bien voir la sortie du composant “Alarme” et du composant “Maintenance” se connecter dans le composant “AlarMaintenance”.

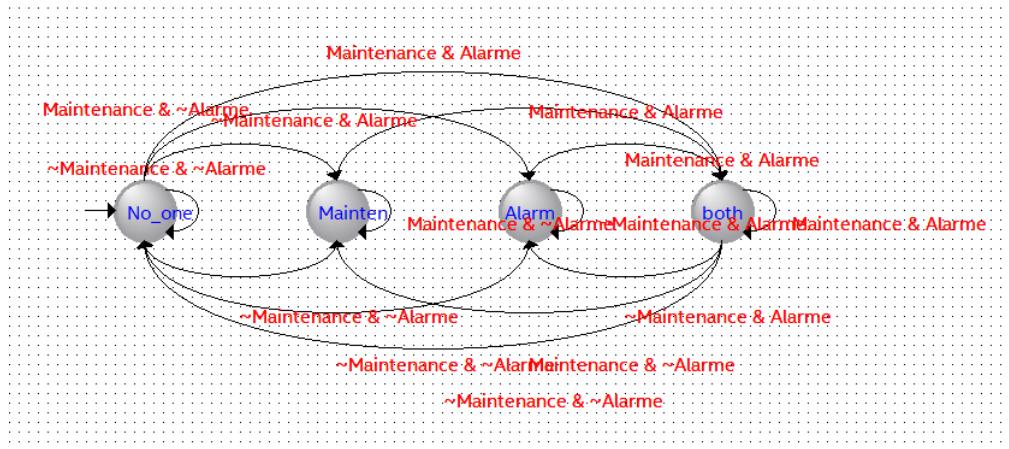


Figure 8 : SMF du AlarMaintenance (FS1)

Ce fichier .smf, nommé “AlarMaintenance” (et représenté dans le fichier BDF juste au dessus), définit un système de gestion d'alarme et de maintenance pour notre ascenseur, avec la logique nécessaire pour gérer différents états liés aux signaux d'alarme et de maintenance. L'entité AlarMaintenance prend en entrée des signaux pour la réinitialisation, l'horloge, un signal du mode Maintenance et un signal du mode Secours. Il y a quatre sorties: un buzzer, une sortie d'alarme, un signal pour bloquer le mouvement de l'ascenseur et une sortie de maintenance.

La machine à états finis à l'intérieur de l'entité a quatre états possibles :

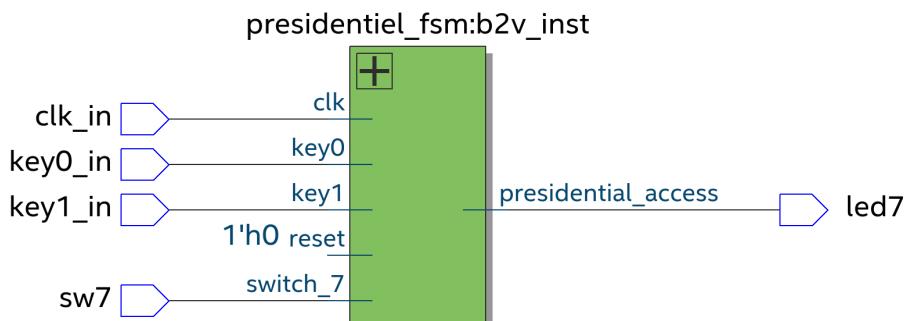
- Lorsque le système est réinitialisé, tout est mis à zéro et l'état est réglé sur state1.
- En fonctionnement normal, les transitions d'état se produisent en réponse aux conditions des signaux d'entrée de maintenance et d'alarme. Par exemple, si une alarme est active, mais pas de maintenance, le système passe à l'état qui active le buzzer et l'indication d'alarme, tout en gelant l'ascenseur.
- Si un signal de maintenance est reçu sans alarme, l'ascenseur est également gelé, mais sans l'activation du buzzer.
- Si les deux signaux sont actifs, un autre état est dédié pour gérer cette condition. Si aucun des deux signaux n'est présent, la machine revient à son état initial.

Les sorties sont ajustées pour refléter l'état actuel de la machine, en activant ou désactivant le buzzer, en signalant l'état de l'alarme et de la maintenance, et en contrôlant l'opération de l'ascenseur. Ce fichier est conçu pour empêcher de manière réfléchie l'activation simultanée de ces deux états critiques de l'ascenseur, assurant que les signaux appropriés sont envoyés en réponse aux conditions d'alarme et de maintenance pour la sécurité et la maintenance de l'ascenseur.

En cas d'état inattendu, le système signale une erreur et définit toutes les sorties à une valeur inconnue, indiquant un problème potentiel avec la logique de la machine à états ou les entrées reçues.

## B. FS2 : Traiter les demandes

Dans notre second module du projet, nous avons abordé la gestion des requêtes pour l'ascension ou la descente formulées tant depuis l'intérieur que l'extérieur de la cabine, y compris le processus d'autorisation pour accéder à l'étage présidentiel. Cette dernière fonction est pilotée par une machine à états finis (FSM) qui traite la séquence d'entrées spéciales via les boutons **key0** et **key1**. Nous avons opté pour une approche épurée qui allie fiabilité et efficacité : chaque pression de bouton correspondante à la séquence présidentielle est représentée par un état distinct au sein de la FSM.



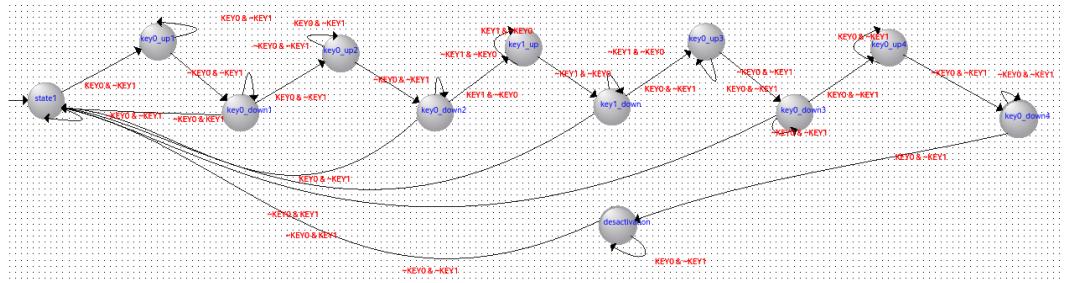


Figure 10 : FSM d'accès à l'étage présidentiel (FS2)

Le .smf "Président" établit un modèle de machine à états finis avec douze états distincts. Dans cette FSM, les états ne sont pas seulement des progressions linéaires mais permettent également des transitions conditionnelles entre eux.

Le comportement de la sortie "Z" est spécifiquement contrôlé à travers la FSM pour refléter des états significatifs, tels que l'activation du mode Présidentiel. Lorsqu'un état inattendu est atteint, la sortie "Z" est assignée à une valeur 'X', ce qui est une pratique courante pour indiquer une condition d'erreur ou un comportement imprévu dans le design des circuits numériques.

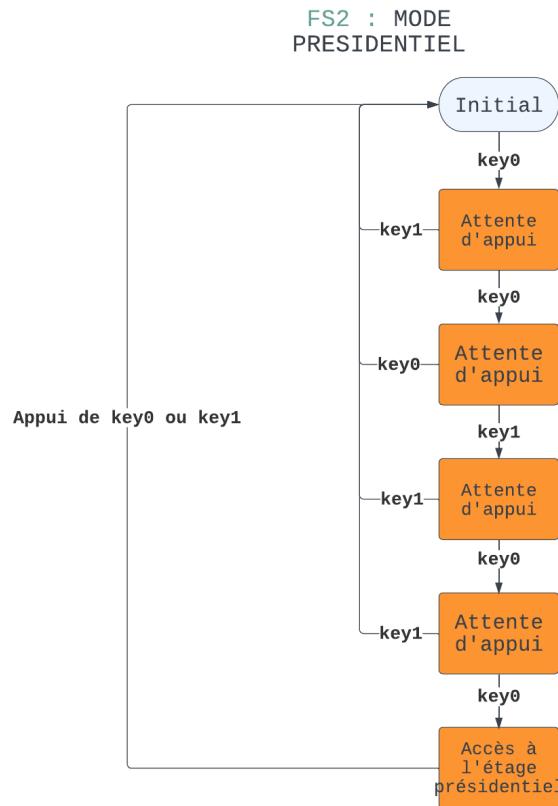


Figure 11 : Diagramme d'appel de l'étage présidentiel (FS2)

Dans ce diagramme d'états, on voit bien le rôle des boutons KEY0 et KEY1, ainsi que le retour à l'état initial si une touche n'est pas correcte pour l'activation du mode Présidentiel (KEY0 KEY0 KEY1 KEY0).

## C. FS3 : Sécurité des passagers

Dans notre module FS3 dédié à la sécurité des passagers d'un ascenseur, la partie de détection des obstacles par capteur ultrason joue un rôle crucial. Il est conçu pour interpréter les données d'un capteur à ultrasons et prendre des décisions en temps réel pour empêcher la fermeture des portes en présence d'un obstacle. Au démarrage, ou lors d'un signal de réinitialisation, le module se prépare en se positionnant dans un état d'attente, ou **idle**. Lorsqu'un signal est émis par le capteur à ultrasons, indiquant la réflexion d'une onde sur un objet, le module entre dans l'état **counting**.

Dans cet état, il effectue des calculs pour convertir le temps pendant lequel le signal "Echo" est haut en une distance, en utilisant la périodicité de l'horloge et une constante de division prédéfinie. Cette distance est ensuite comparée à un seuil déterminé, dans notre cas 50 cm, pour décider de l'action à prendre.

Si la distance calculée est inférieure à ce seuil, cela signifie qu'un obstacle est détecté à proximité immédiate des portes de l'ascenseur, et le module envoie alors un signal qui empêchera la fermeture des portes, conformément aux exigences de sécurité spécifiées dans notre cahier des charges, assurant ainsi la protection des passagers.

Vous avez ci-dessous la partie du BDF du projet dédiée au capteur à ultrasons et à l'affichage des portes sur 7SEG.

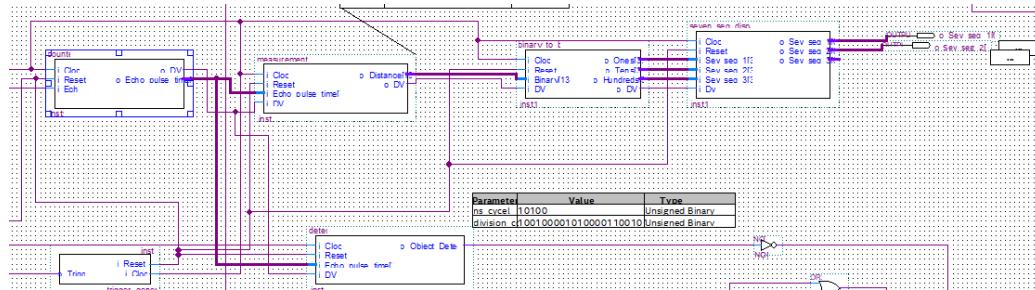


Figure 12 : BDF du capteur ultrasons (FS3)

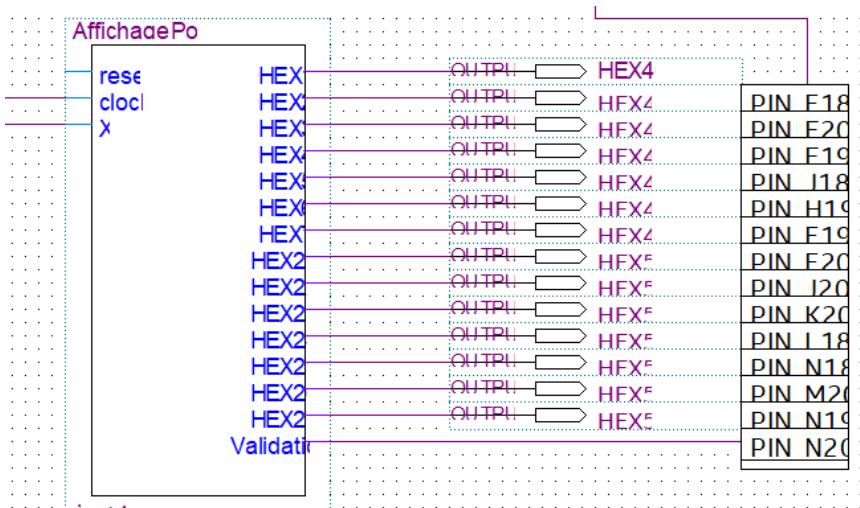


Figure 13 : BSF de l'affichage de l'état de la porte (FS3)

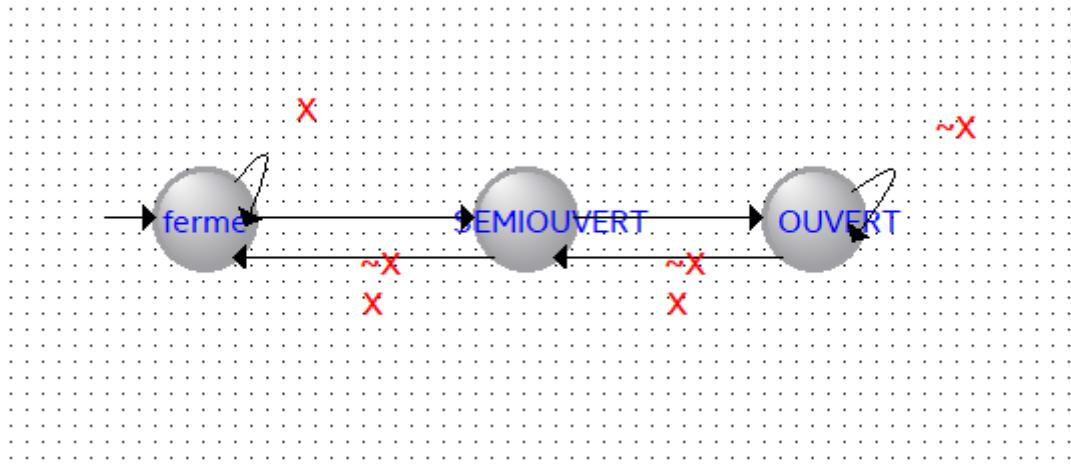


Figure 14 : SMF de l'affichage de la porte sur l'afficheur 7 segments (FS3)

La machine à état permet de contrôler l'ouverture ou la fermeture des portes de l'ascenseur à n'importe quel instant. Cette machine à état récupère en entrée une valeur de 0 ou de 1 en fonction de la distance calculée par le capteur. Lorsque la valeur en entrée passe de 1 à 0 nous changeons d'état et nous passons à l'état **semi-ouvert** pendant une période de 0,5 seconde que nous avons créé grâce à un diviseur d'horloge de 2 Hertz. Par la suite une fois que la période d'une demi seconde est passée nous passons à l'état suivant **ouvert**.

En sortie de ces états nous avons plusieurs valeurs. Il y a les sorties, qui permettent d'afficher l'état de la porte sur deux afficheurs 7 segments. Ces sorties se modifient à chaque changement d'état pour pouvoir actualiser l'affichage de la porte. Enfin, à chaque état nous avons une sortie de validation directement reliée au BSF ETAGE qui permet de stopper l'ascenseur si la distance entre le capteur ultrason et l'objet est inférieur à 50 cm.

### FS3 : Sécurité des passagers

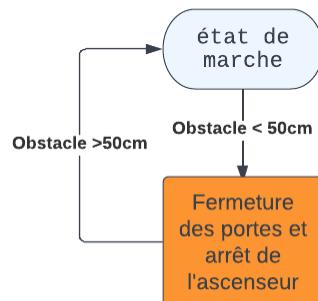


Figure 15 : Diagramme de sécurité des passagers (FS3)

## D. FS4 : Déplacer les passagers

Pour ce module qui concerne le déplacement des passagers, une machine à état est utilisée, prenant en entrée les données des switchs et boutons traités précédemment dans le FS2.

En fonction de ces entrées, les étages vont être parcourus séquentiellement, l'ascenseur va se rendre à l'étage souhaité. De plus, tant que le switch de l'étage en question est activé, l'ascenseur va rester à cet étage. Cela est possible grâce aux équations Verilog de notre state machin file.

Quand il n'y a plus aucune demande d'étage, l'ascenseur redescend automatiquement au rez-de-chaussée, toujours de façon séquentielle.

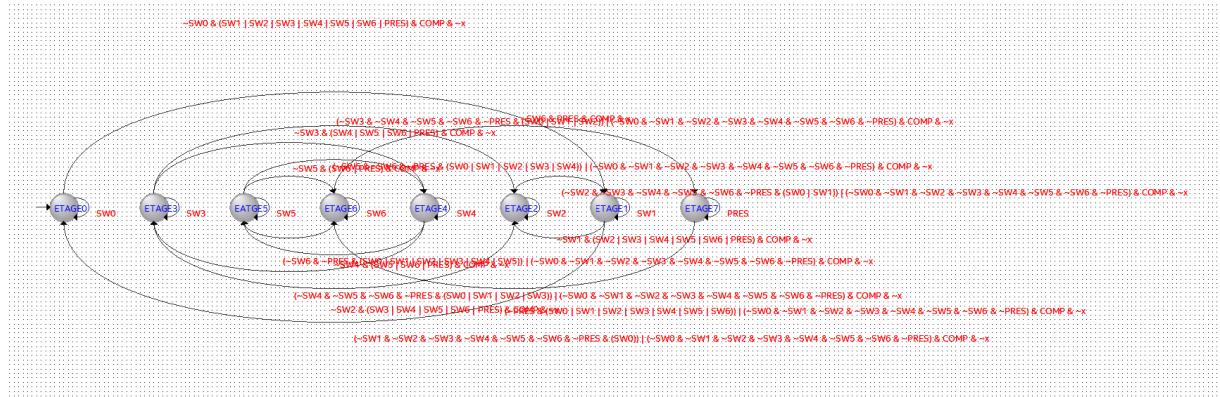


Figure 16 : SMF du déplacement de l'ascenseur de manière séquentielle à travers les étages (FS4)

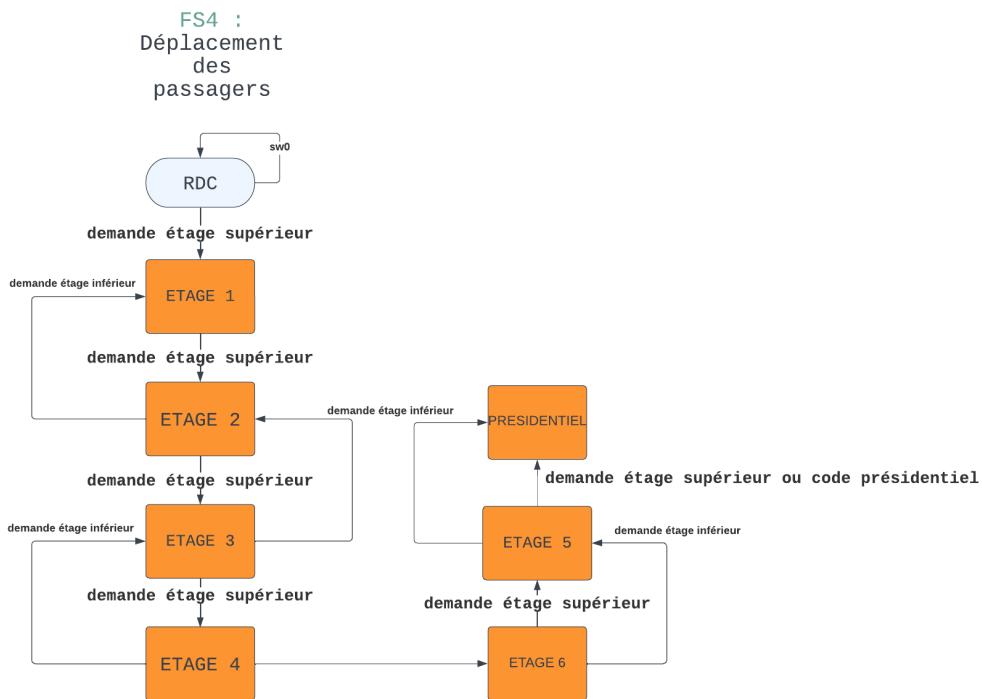


Figure 17 : Diagramme de déplacement séquentiel de l'ascenseur (FS1)

L'entité "ETAGE" gère les entrées sous forme d'interrupteurs pour la sélection des étages, un capteur de présence, une entrée de comparaison et un signal externe. Les sorties correspondent aux indicateurs d'étage, à un buzzer et à un indicateur additionnel.

Le système utilise les switchs de la DE 10-Lite pour permettre la sélection d'un étage, avec un affichage dédié pour chaque étage. Les transitions d'état sont conditionnées par les combinaisons d'entrées provenant des interrupteurs et du capteur de présence connecté au capteur ultrason. Le système peut se réinitialiser à l'étage de base sous certaines conditions et réagit à un reset en éteignant tous les indicateurs et le buzzer, signifiant un retour à l'état initial.

La machine est sensible à l'horloge pour les transitions d'état, assurant que les changements d'état ne se produisent qu'à des intervalles bien définis (1s par étage en valeur de base, mais elle peut changer en mode maintenance avec le potentiomètre) pour un fonctionnement prévisible et stable. L'indicateur LEDR7 est uniquement contrôlé dans l'état ETAGE7 (présidentiel), distinct des indicateurs d'étage standards.

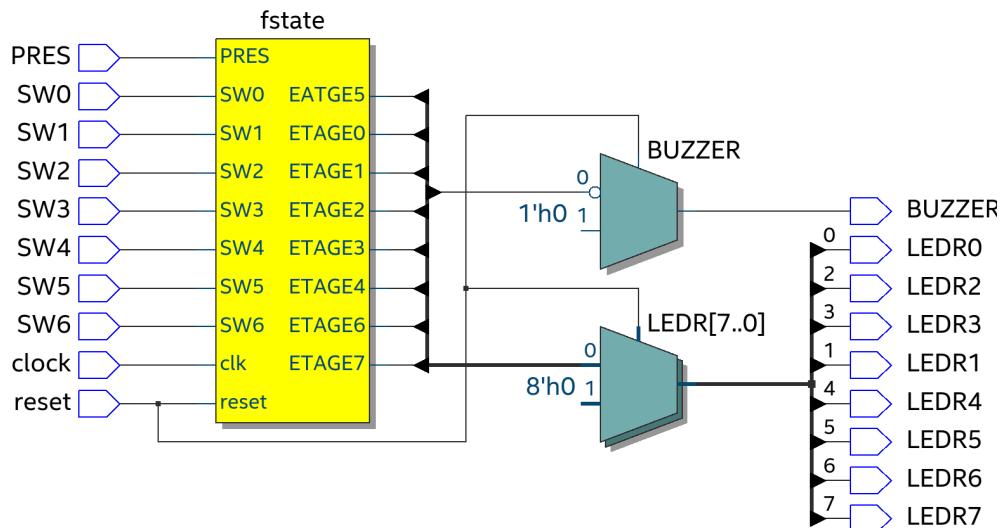


Figure 18 : RTL du FSM de déplacement des usagers (FS4)

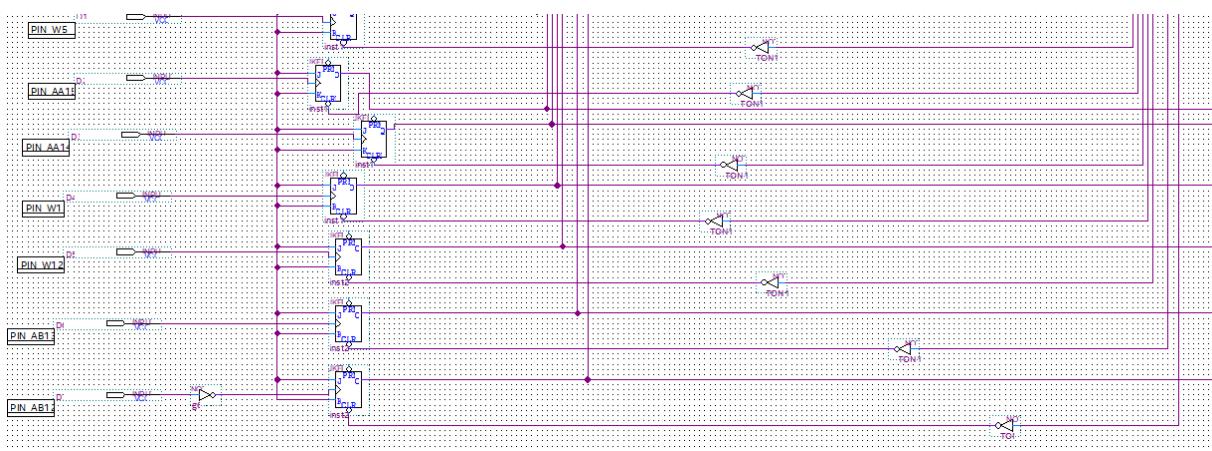


Figure 19 : BDF des bascules JK pour l'appel de l'ascenseur à l'aide des boutons poussoirs (FS4)

Les bascules JK sont des types de bascules bistables utilisées dans la conception de circuits numériques pour le stockage et le traitement de l'information binaire. À la différence des registres à décalage tels que le 74HC595, qui sont utilisés pour déplacer des bits en série à travers une chaîne de registres, les bascules JK peuvent être configurées pour effectuer une variété de fonctions logiques en plus du simple stockage de l'état.

Dans notre contexte, nous les avons utilisés à la place d'un registre à décalage. Les bascules JK peuvent être mises en chaîne d'une manière similaire pour réaliser une fonction de décalage. Chaque bascule JK a trois entrées principales : J (Set), K (Reset) et Clock. Lorsque l'horloge est activée, les entrées J et K déterminent l'action sur le bit stocké : conserver l'état, le mettre à 1, le mettre à 0, ou le basculer.

En résumé, notre utilisation de bascules JK au lieu de registres à décalage est une décision que nous avons prise afin de mener à bien notre projet, bien que nous n'ayons pas utilisé le registre à décalage.

## E. FS5 : Informer les passagers

Le module d'information des usagers a été l'un des plus simples à implémenter, car nous avons déjà les informations liées à notre système d'ascenseur (I/O). Pour les affichages des portes et de l'étage, nous avons utilisé des machines à états, très pratiques pour l'animation et le parcours séquentiel de données.

La machine à état va prendre en entrée les valeurs des 8 LEDs du FPGA. En fonction de la valeur de la LED allumée, on va avoir en sortie **ss0** à **ss7**, les segments que l'on veut allumer pour symboliser l'étage actuel de l'ascenseur.

On relie le composant sur le bdf aux différentes LEDs de l'ascenseur et à la clock 50 MHz afin d'avoir un bon rafraîchissement de l'état.

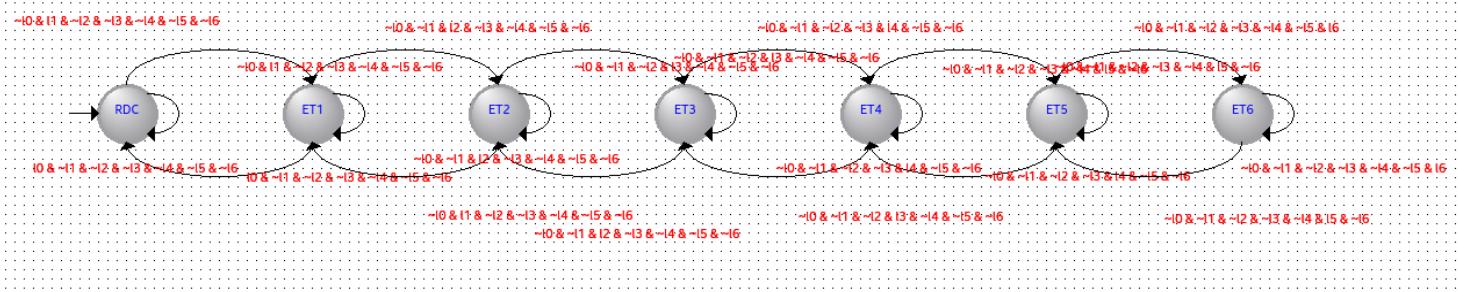


Figure 20 : FSM d'affichage des étages sur 7SEG (FS5)

L'affichage des étages sur l'afficheur 7SEG gère les états de l'affichage en fonction des signaux d'entrée qui représentent les requêtes d'étage ou les commandes de l'ascenseur. Lorsqu'un signal de réinitialisation est reçu, l'affichage est réinitialisé à un état de base.

Sinon, selon les signaux d'entrée et l'état actuel, l'affichage change pour indiquer l'étage correspondant, mais aussi l'étage 7 pour l'étage présidentiel. Des conditions logiques dans le code déterminent les transitions entre les états, ce qui affecte les segments allumés sur l'affichage pour représenter les chiffres correspondant aux étages de l'ascenseur. Le code assure que tous les segments et états non définis sont gérés de manière à éviter tout état indésirable ou indéfini.

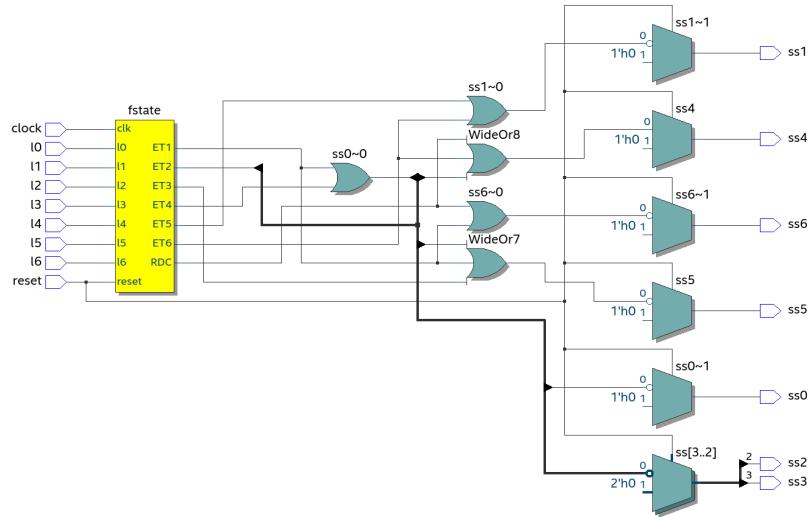


Figure 21 : RTL du FSM d'affichage des étages (FS5)

## VII. Tests et validation

### A. FS1 : Mode maintenance ✓

Pour la fonction FS1 dédié au développement d'un mode maintenance, les actions à réaliser étaient les suivantes :

- **(ET1.1)** : Rendre le mode maintenance activable à tout moment par une double pression simultanée sur les boutons KEY0 et KEY1.
- **(ET1.2)** : Faire clignoter toutes les LEDs à une fréquence de 1 Hz lorsqu'en mode maintenance.
- **(ET1.3)** : Permettre le réglage de la vitesse de déplacement de l'ascenseur via un potentiomètre, de 0,5 à 5 secondes par étage, tout en maintenant le temps de fermeture de porte constant.
- **(ET1.4)** : S'assurer que la vitesse de l'ascenseur ne change pas si le potentiomètre est ajusté hors du mode maintenance.
- **(ET1.5)** : Prendre en compte toute nouvelle valeur du potentiomètre ajustée pendant que le mode maintenance est inactif, lors de la réactivation suivante de ce mode.

Ces étapes garantiront que le mode maintenance peut être activé selon les besoins et que les réglages de vitesse de l'ascenseur sont ajustables de manière sûre et efficace.

Le code VHDL pour le mode maintenance utilise une machine à états finis pour détecter une séquence d'actions sur deux boutons, KEY0 et KEY1, afin de déclencher un mode de maintenance. Dans la simulation avec ModelSim, le bon fonctionnement est démontré par l'activation du signal Z à '1' quand les boutons sont pressés simultanément deux fois de suite, ce qui indique que la séquence d'activation est respectée. L'état state5 correspond à l'activation du mode maintenance. En cas de reset ou d'inactivité, la FSM retourne à l'état initial state1, tandis que des sorties indéterminées ou des erreurs d'état sont gérées avec des sorties 'X'. La simulation confirme que le système répond correctement à la séquence d'entrée prévue pour activer le mode maintenance.

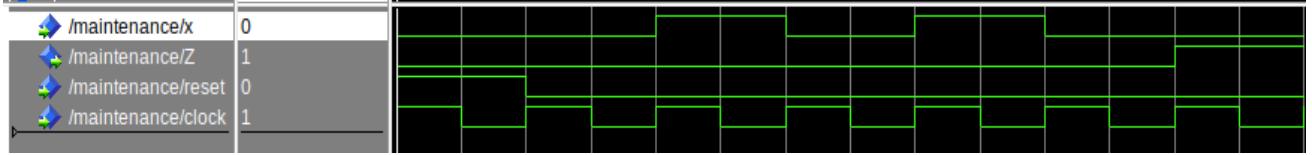


Figure 22 : Modelsim de mise en marche du mode Maintenance (FS1)

Dans la photo ci-dessous, toutes les LEDs sont allumées suite à l'accès au mode maintenance, clignotant toutes à une fréquence de 1Hz obtenue par un diviseur d'horloge sur Quartus.

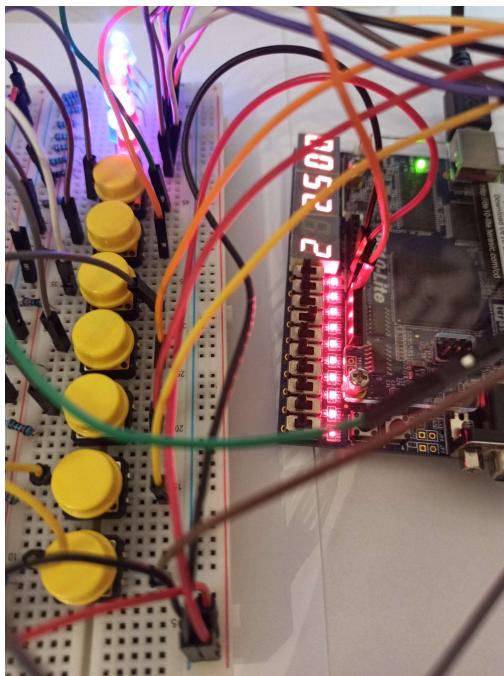


Figure 23 : Photo mode maintenance activé (FS1)

Pour conclure ce module, nous avons bien un accès à ce mode, bloquant l'ascenseur, comme le montre le Modelsim précédent. De surcroît, les LEDs vont toutes clignoter à une fréquence de 1Hz (Illustration ci-dessus). Cependant, la modification de la vitesse de l'ascenseur grâce au potentiomètre n'a pas été implémentée. Ainsi, ce premier module n'est que partiellement réalisé.

## B. FS2 : Traiter les demandes

Pour le module FS2, les actions à réaliser étaient les suivantes :

- **(ET2.1)** : Indiquer la position de l'ascenseur en temps réel à l'aide des LEDs LEDR0 à LEDR7 sur la carte DE 10-Lite.
- **(ET2.2)** : Équiper une breadboard de boutons poussoirs pour enregistrer les demandes d'arrêt à chaque étage.
- **(ET2.3)** : Limiter la sélection des étages possibles par les usagers dans la cabine de l'étage 0 (RDC) à l'étage 6.
- **(ET2.4)** : Rendre l'étage 7 (présidentiel) accessible uniquement par une séquence de touches définie (KEY0, KEY0, KEY1, KEY0, KEY0).

Dans la simulation ModelSim d'accès à l'étage présidentiel, on voit bien qu'à l'appui de la séquence **(key0 key0 key1 key0 key0)**, Z passe à 1 jusqu'à ce qu'un bouton soit ré-appuyé pour que la demande soit annulée

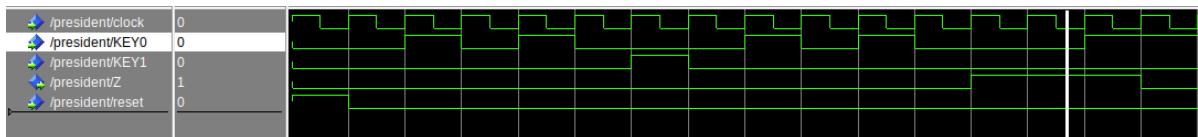
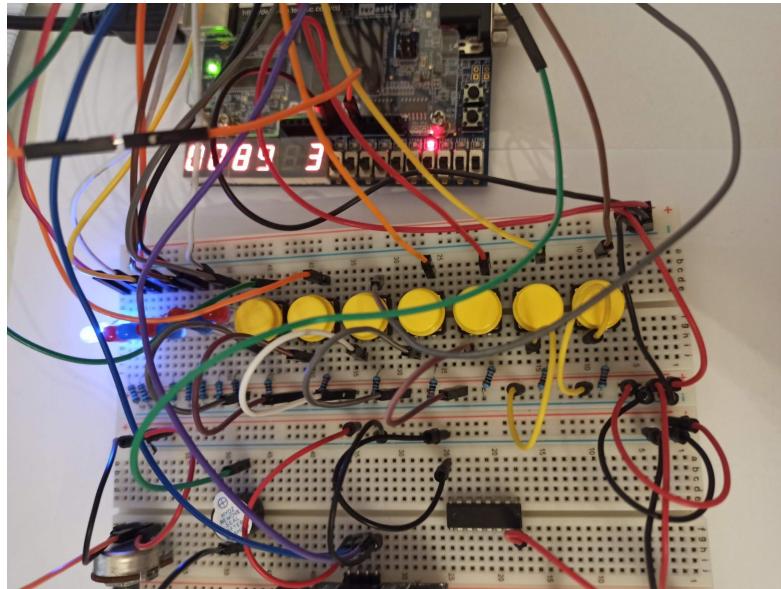


Figure 24 : Modelsim de l'accès à l'étage présidentiel (FS2)

Le module s'est avéré efficace pour l'affichage en temps réel de l'étage présidentiel. L'architecture bien conçue du système permet une interaction utilisateur intuitive grâce aux boutons KEY0 et KEY1, facilitant la demande d'arrêt à l'étage présidentiel.

En utilisant des boutons poussoirs pour entrer des séquences spécifiques, il transite entre différents états pour activer ou désactiver l'accès à cet étage. Un signal de sortie Z indique l'état actif de l'accès présidentiel, tandis que le système peut être réinitialisé à l'état initial pour la sécurité. La conception empêche l'accès non autorisé en n'autorisant l'étage présidentiel que via une séquence de touches validée.

La simulation ModelSim confirme que le module répond aux exigences de fonctionnement et de sécurité prévues, reflétant son aptitude à gérer les priorités d'accès aux différents étages de manière fiable et sécurisée.

Ainsi, notre module FS2 est totalement validé dans toutes ses fonctions.

## C. FS3 : Sécurité des passagers

Pour le module FS3 permettant d'assurer la sécurité des passagers, les actions à réaliser étaient les suivantes:

- **(ET3.1)** : Maintenir la porte de l'ascenseur ouverte si un obstacle est détecté à moins de 50 cm par le capteur à ultrasons.
- **(ET3.2)** : Permettre l'appel des secours en cas d'urgence en appuyant simultanément sur les boutons KEY0 et KEY1 pendant 1 seconde, ce qui bloque la progression de l'ascenseur et active une alarme sonore.
- **(ET3.3)** : Garantir que le temps de fermeture des portes soit constant et égal à deux secondes.

Ces directives contribuent à la fonction secondaire FS3, qui est essentielle pour garantir la sécurité des passagers à bord de l'ascenseur.

Dans les photos suivantes, la valeur inscrite sur les deux afficheurs 7SEG du milieu représente la distance à l'obstacle capté par l'ultrason.

Sur l'image de gauche, '00' représente les portes à l'état de fermeture et à droite '| |', les portes ouvertes.

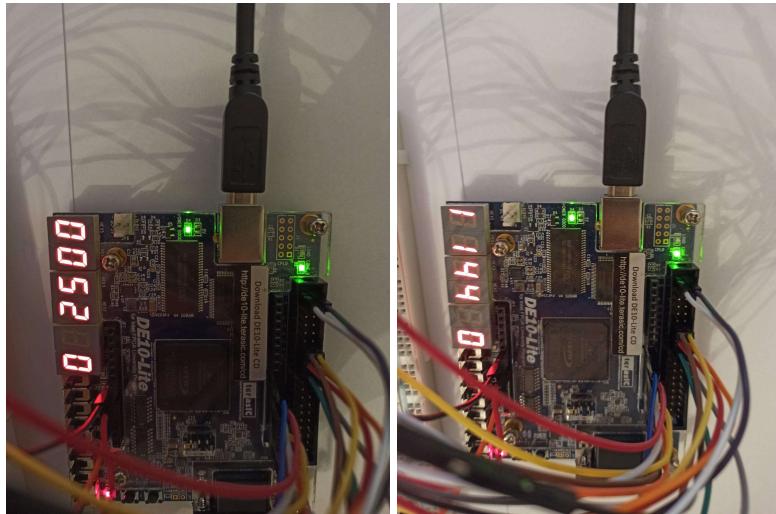


Figure 25 : Photos de la fermeture et de l'ouverture des portes à <50cm (FS3)

En conclusion de ce module, nous avons bien un arrêt de l'ascenseur quand un obstacle se trouve à moins de 50 cm du capteur à ultrason. De plus, le mode d'urgence s'active bien après l'appui simultané de key0 et key1 pendant une seconde, activant l'alarme sonore et bloquant l'ascenseur.

## D. FS4 : Déplacer les passagers ✓

Pour le module FS4 concernant le déplacement séquentiel de l'ascenseur, les actions à réaliser étaient les suivantes :

- **(ET4.1)** : Assurer un déplacement séquentiel des étages sans sauter d'étages
- **(ET4.2)** : Programmer l'ascenseur pour qu'il retourne au rez-de-chaussée (RDC) si aucun appel n'est en attente aux étages.
- **(ET4.3)** : Bloquer l'ascenseur à l'étage courant tant que le switch associé est activé.
- **(ET4.4)** : Initier la fermeture de la porte dès que le switch d'appel est désactivé.

Ces points garantiront que l'ascenseur déplace les passagers conformément aux contraintes spécifiées.

Pour cette simulation, on demande l'étage n°5 : On voit bien que les étages vont s'allumer séquentiellement jusqu'à l'étage n°5 sélectionné.

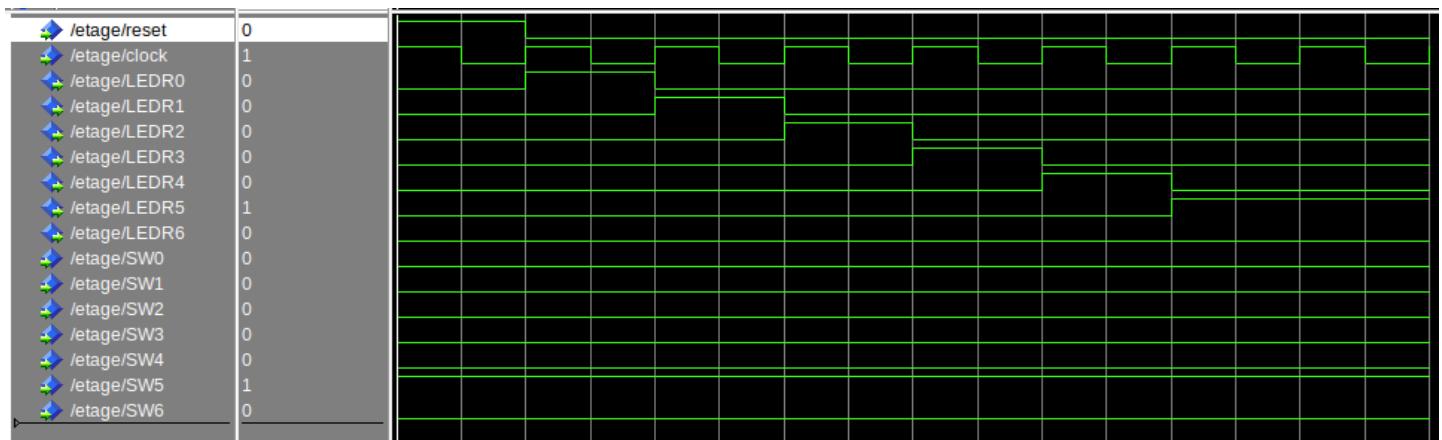
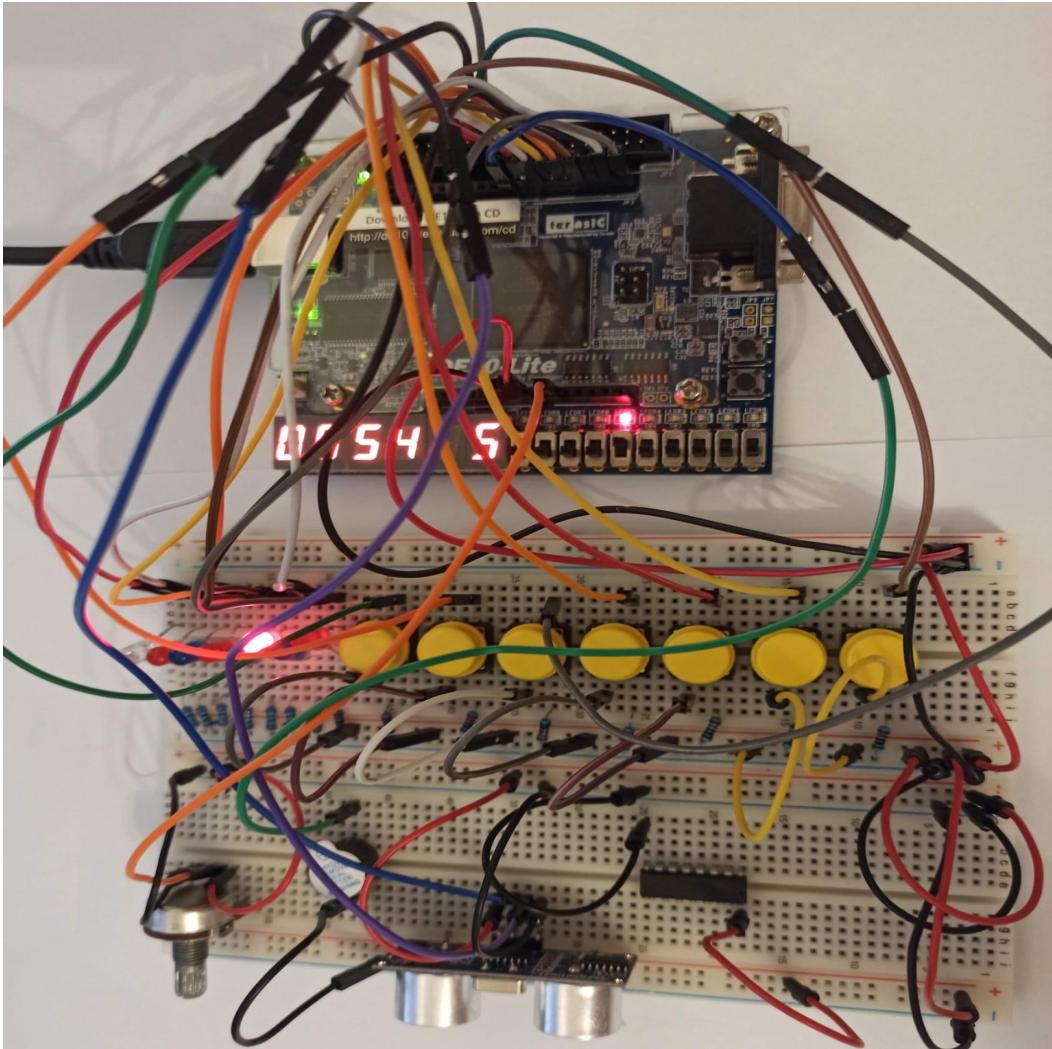


Figure 26 : Modelsim du déplacement séquentiel de l'ascenseur (FS4)

Pour résumer le fonctionnement du module VHDL décrit, celui-ci assure la gestion d'un système d'ascenseur en respectant les commandes d'arrêt à différents étages, telles qu'indiquées par les interrupteurs. Le comportement souhaité, tel que programmé dans ce module, comprend le mouvement séquentiel de l'ascenseur pour répondre aux demandes d'arrêt, la limitation d'accès à un étage présidentiel

spécial, et le retour automatique de l'ascenseur au rez-de-chaussée en l'absence de requêtes. Le buzzer et les indicateurs LED intégrés reflètent l'état actuel de l'ascenseur et signalent les actions à l'utilisateur. L'ensemble de ces fonctionnalités démontre la conformité du module avec les spécifications fournies et confirme sa validation.

## E. FS5 : Informer les passagers ✓

Pour le module FS5, les actions à réaliser étaient les suivantes :

- **(ET5.1)** : Afficher les demandes d'arrêt émises à partir des étages et de la cabine via des LEDs sur la breadboard.
- **(ET5.2)** : Contrôler les LEDs de la breadboard en utilisant un registre à décalage 74HC595.
- **(ET5.3)** : Présenter l'étage actuel de l'ascenseur sur l'afficheur HEX0.
- **(ET5.4)** : Créer une animation sur les afficheurs HEX2 à HEX5 pour visualiser l'état des portes (ouvertes/fermées).
- **(ET5.5)** : Activer le buzzer pour émettre un signal sonore à chaque passage d'étage.

Ces éléments garantissent que les usagers sont correctement informés de l'état de l'ascenseur et des demandes en cours.

Dans la simulation suivante, on peut voir que pour chaque étage où l'on se trouve, le code des segments que l'on veut afficher est présent.

Par exemple, quand l'ascenseur est au RDC (**I0 = 1**), tous les segments sont à **1** sauf le 6<sup>e</sup> est allumé, ce qui nous donne bien un '0' sur l'afficheur :

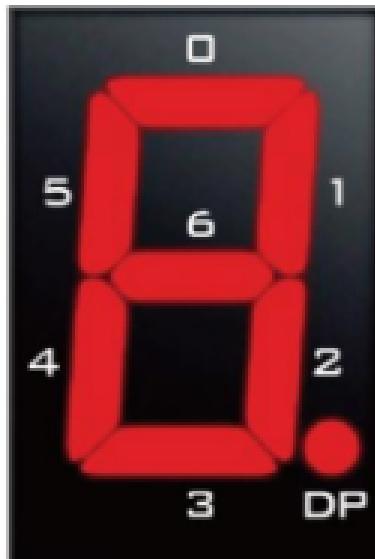


Figure 27 : numérotation des segments de notre 7SEG

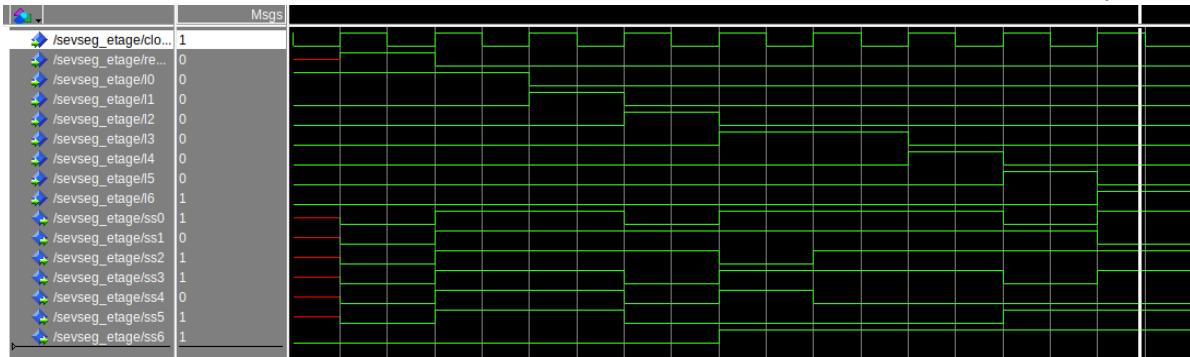


Figure 28: Modelsim de l'affichage 7SEG utilisateur (FS5)

Pour conclure ce module sur la gestion d'un ascenseur avec une breadboard, nous avons réussi à établir un système de signalement des demandes d'étages par LED et un affichage de l'étage courant via un afficheur 7 segments. Le son du buzzer à chaque passage d'étage a été intégré avec succès. Toutefois, l'intégration d'un contrôle des étages par le registre à décalage 74HC595 n'a pas été réalisée. Notre dispositif actuel utilise un fichier VHDL "SEVSEG\_ETAGE" qui traite les entrées des étages et commande l'affichage en conséquence, mais il pourrait être optimisé par l'emploi du registre à décalage pour une meilleure efficacité et extensibilité.

## VIII. Bilan

### A. État d'avancement

Nous sommes ravis d'annoncer que notre projet d'ascenseur a atteint avec succès les objectifs fixés pour chaque fonction clé. Le FS1 a mis en place un système de maintenance efficace, garantissant ainsi une durabilité et une disponibilité optimale de l'ascenseur. Le FS2 a facilité la gestion des demandes des usagers, assurant une expérience fluide et réactive. De plus, le FS3 a renforcé la sécurité des passagers en empêchant la fermeture des portes en présence d'objets à moins de 50 cm à l'aide d'un capteur à ultrasons.

Le FS4 a permis des transitions d'étages fluides en réponse aux demandes des usagers. De même, le FS5 a assuré une communication efficace avec les usagers, offrant des informations claires et précises aux différents étages et à l'intérieur de la cabine.

Cependant, malgré nos efforts, nous n'avons pas réussi à mettre en œuvre la fonction FS6 comme prévu initialement. Bien que cela représente un défi, nous sommes déterminés à poursuivre nos efforts pour surmonter ces difficultés et finaliser la mise en place du FS6. Notre équipe reste pleinement engagée à garantir que cette dernière fonction soit intégrée avec succès, afin d'améliorer davantage la surveillance et la gestion du système d'ascenseur, offrant ainsi une expérience utilisateur complète et satisfaisante.

### B. Pertinence de la solution technique

Bien que notre ascenseur réponde efficacement aux critères du cahier des charges, il présente certaines lacunes techniques, la gestion des séquences d'appui sur les boutons, trop similaires entre elles, soulève des problèmes nécessitant des ajustements pour atteindre un niveau de finition professionnelle et conforme à nos standards de qualité.

### C. Bilan sur le travail d'équipe

Au fil de ce premier projet d'envergure, nous avons appris la valeur de la cohésion d'équipe et la nécessité de progresser malgré les obstacles. Individuellement, nous avons acquis des compétences essentielles telles que la résolution de problèmes complexes, la gestion du stress et l'adaptation rapide aux imprévus, des atouts précieux pour nos futures recherches de stage. Le sentiment d'accomplissement ressenti après chaque problème résolu a été une source de motivation profonde.

Cependant, nous avons reconnu que notre organisation pourrait être améliorée, surtout en ce qui concerne la distribution du travail dans le temps. La tendance à concentrer une grande partie de l'effort dans la dernière ligne droite nous a appris l'importance d'une planification plus stricte et d'une répartition plus équilibrée du travail tout au long du projet.

Pour le prochain projet, notre plan d'action comprendra une planification plus détaillée dès le début, avec des échéances intermédiaires fixées pour maintenir une progression régulière. Nous mettrons également en place des revues de projet hebdomadaires pour surveiller notre avancement et ajuster notre stratégie en conséquence. Cette approche nous permettra d'aborder les défis de manière plus sereine et structurée, en veillant à optimiser notre productivité et à minimiser le stress lié aux délais.

### IX. Sources

Vous trouverez ci-dessous les documents utilisés et sites internet consultés pour développer notre projet :

<https://github.com/manishtanwar/Elevator-Controller/tree/master>

<https://www.spiceworks.com/tech/tech-general/articles/what-is-fsm/>

<https://www.youtube.com/watch?v=08YHxtSI3Bk&t=231s&pp=ygULZnNtIHF1YXJ0dXM%3D>

### X. Annexes

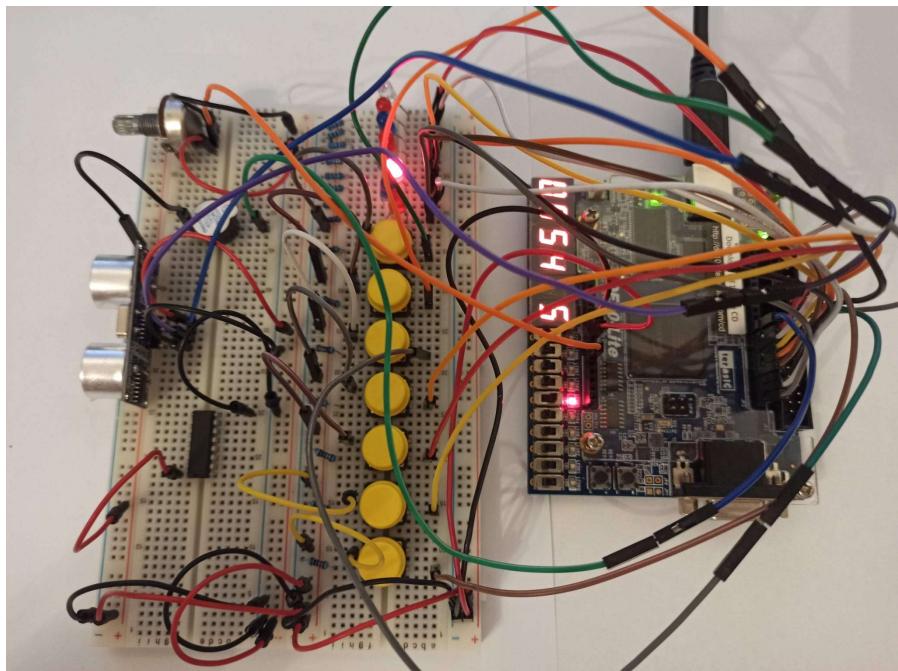


Figure 28 : Photo du projet complet