الجمهورية الجزائرية الديمقراطية الشعبية
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR.YAHIA FARES DE MEDEA

FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

DEPARTEMENT DU GENIE ELECTRIQUE ET DE L'INFORMATIQUE

MÉMOIRE DE FIN D'ÉTUDES

Présenté pour l'obtention du :
Diplôme de Master en Génie Électrique
Option : Systèmes Intelligents et Communicants

Thème :

Smart Elevator :
Etude, conception et implementation VHDL d'un système intelligent de gestion d'un groupe d'ascenseurs



Proposé et Dirigé par :

Dr. BOUHEDDA Mounir.

Préparé et Présenté par:

Maasmi Amina et Tahraoui Fatima zohra.

Année Universitaire 2011-2012

« Les prouesses réalisées par des individus exceptionnels, grâce à leur art et à leur intelligence, tôt ou tard la technologie les rend possibles à tout le monde »

« Roland Topor »

Dédicace

Je dédie avec grand plaisir, ce modeste travail

À

Tous ce qui témoigne qu'il n'y a de Dieu qu'Allah et que

Mohamed est le prophète de DIEU.

À

Mes chers parents

Vous avez toujours été pour moi les parents idéaux, la lumière qui me guide dans les moments les plus obscures, et si j'en suis arrivé là ce n'est que grâce à vous mes parents adorés .Puisse dieu vous prêter longue vie et bonne santé afin que je puisse vous combler à mon tour.

À

Mes chères sœurs : Meriem, Zahra, Zineb et sa belle fille Noussaiba

Mes chers frères : Abderrahmane et Oussama

À

Ma chère amie et mon binôme : Fatima

Mes chères copines : Karima, Ryma, Souhila

Khaoula, Soumia, Hadjer

À

Tous mes collègues de la section de SIC

Tous ceux qui me sont chers et que j'ai omis de citer.

« Amina »



Dédicace

Je dédie ce travail :

À mes très chers parents « aissa, hamida »

Dont vos mérites, vos sacrifices et votre encourage, c'est à vous que je dois tout le profond amour que je vous porte pour les sacrifices que vous avez consenti pour ma réussite, Je vous remercie pour le soutien moral et l'encouragement que vous m'avez accordés et que dieu vous préservent bonne santé et longue vie.

À mes frères et mes sœurs

Hafsa, Mohamed, kenza, Aymen et la petite Doâe, Je vous souhaite tout le bonheur que vous méritez En souhaitant un brillant avenir.

À toi ma chère sœur ***Soumia*** *Vous Que dieux te protège pour moi et me protège pour toi.*

À ma famille

À ma copine et mon binôme Amina

Qui m'aide tout à longue de ce travail et soutient aux moments propices

À mes amies :

Karima, Ryma, Souhila, Soumia, Khaoula, Yasmina, Yamina

Et avec toute la joie, je le dédie à tout qui aime FATIMA.

«Fatima Zohra»

Remerciements

Au nom d'Allah le plus grand merci lui revient de nous avoir guidé vers le droit chemin, de nous avoir aidées tout au long de nos années d'étude.

Au Terme de notre travail, nous tenons à vous adresser Monsieur « Mounir Bouhedda » nos remerciements les plus vifs et notre gratitude. Nous avons trouvé les meilleures conditions pour réaliser notre travail sous votre encadrement. Nous vous remercions de votre gentillesse et de votre disponibilité

Nous remercions du fond du cœur nos chers parents de tout ce qu'ils ont fait pour nous.

Nous adressons également nos remerciements, à tous nos enseignants, qui nous ont donné les bases de la science.

Nous remercions très sincèrement, les membres de jury d'avoir bien voulu accepter de faire partie de la commission d'examinateur

ملخـــــص

الهدف من هذا المشروع هو تصميم نظام ذكي لتسيير مجموعة مصاعد("م" مصعد) في مبنى ب "ن" مستوى. لتحقيق هذه الغاية اقترحنا استعمال الدارة الرقمية القابلة للبرمجة FPGA من اجل تقليل متوسط زمن انتظار الركاب في الطوابق TAM, و استهلاك الطاقة من طرف المصاعد . هذا النظام يتفاعل مع كل الاحداث المرصودة في المحيط , و يأخذ افضل القرارات في الوقت الحقيقي . و يقوم ايضا ب : تحسين تسيير المقصورات (الغرف المتحركة في المصاعد) عن طريق توزيع الطلبات و اختيار خوارزميات مناسبة ,تقليل عدد المقصورات العاملة في الليل و في عطلة نهاية الاسبوع من اجل التقليل من استهلاك الطاقة , اعطاء الاولوية لطلبات الاشخاص المهمين و المعوقين المرصودة عن طريق بطاقاتهم الممغنطة . من اجل كل هذا اقترحنا بنية تتألف من نظام التحكم في مجموعة مصاعد SCGAمع شروحات معمقة على كيفية عمله , و انظمة تحكم محلية SCL,نظام لكل مصعد, تعمل وفق واحدة من ستة خوارزميات مقترحة. بعد تنفيذ البنية المقترحة في VHDL, أظهرت الاختبارات والتحليلات من خلال نتائج المحاكاة صلاحية النظام الذكي المقترح.

كلمات مفتاحية : مصعد , طاقة , FPGA , تسيير , ذكي , SCGA , SCL , TAM , VHDL

Résumé

Le but de ce projet est la conception d'un système intelligent de gestion d'un groupe de M ascenseurs dans un immeuble à N niveaux. Pour arriver à cette fin, un circuit numérique reconfigurable à base des FPGAs a été proposé dans le but de réduire le temps d'attente moyen (TAM) des passagers dans les étages et la consommation en énergie électrique des ascenseurs. Ce système réagit avec tous les évènements détectés dans l'environnement, et prendre les meilleures décisions en temps réel. En occurrence, optimiser la gestion des cabines par les affectation des appels ainsi que des choix des algorithmes approprié, réduire le nombre des cabines fonctionnelles pendant le week-end et les nuits afin de minimiser l'énergie, et donner la priorité aux appels des personnes importantes et handicapées détectés à travers leurs cartes magnétiques. Dans ce but, on a proposé une architecture composée d'un système de contrôle du groupe d'ascenseur (SCGA), avec des explications approfondies sur son fonctionnement**,** et des systèmes de contrôle locaux (SCLs),un pour chaque ascenseur ,qui peut fonctionner selon l'un des six algorithmes de dispatching proposés .Après une implémentation VHDL de toute l'architecture proposée, des tests et analyses à travers des résultats de simulations ont montré la validité du système intelligent proposé.

Mots clés: ascenseur, énergie, FPGA, gestion, intelligent, SCGA, SCL, TAM, VHDL.

Abstract

The purpose of this project is the design of an intelligent system of management of a group of M elevators in a building with N levels. To achieve this goal, a reconfigurable digital circuit based on the FPGAs has been proposed to reduce average waiting time (AWT) of passengers in the floors and the electric elevator energy consumption. This system reacts with all events detected in the environment, takes the best decisions in real time, and optimizes the management of cabins by the assignment of the appeals and choice of appropriate algorithms; reduce the number of functional cabin for the weekend and nights to minimize the energy, and give priority to the appeals of the important people and disabled detected through their magnetic cards. For this purpose, we proposed an architecture composed of elevator group control system (EGCS) with detailed explications on how it works, and local control systems (LCS) one for each elevator that can work according to one of the six dispatching algorithms proposed. After an implementation VHDL of the architecture proposed tests and analyses through results of simulations showed the validity of the proposed intelligent system.

Key words: elevator, energy, FPGA, management, Intelligent, EGCS, LCS, AWT, VHDL

Sommaire

Sommaire

Introduction générale.....2

Chapitre 1 : Présentation et description des ascenseurs

I. Introduction : 5

II. Définition : 5

III. Les catégories d'ascenseur 5

III.1 L'ascenseur hydraulique 6

- *Principe* 6
- *Description* 6
- *Energie* 7
- *Avantage* 7
- *Inconvénient* 7

III.2 L'ascenseur à traction à câbles 8

- ***Principe et description*** 8
- ***Energie*** 8
- ***Avantage*** 9
- ***Inconvénient*** 9

IV. Différentes composantes d'un ascenseur électrique 9

IV.1 Cabine d'ascenseur 10

IV.2 Porte de cabine 11

IV.3 Portes palières 11

IV.4 Boutons d'appels 11

IV.5 Boutons d'envois 12

IV.7 Charge utile 12

IV.8 Contrepoids 13

IV.9 Treuil 13

IV.10 Câble de traction 13

IV.11 Le moteur 13

IV. photocellules 14

V. Les critères du choix du type d'ascenseur 14

VI. Le système de contrôle d'un ascenseur (SCL) 15

VI.1 Collectif-haut / collectif –bas 16

VI.2 Sélectif –haut / Sélectif –bas 16

Sommaire

VI.3 Collectif-haut / Sélectif-bas 16
VI.4 Sélectif-haut /Collectif -bas 17
VI.5 Sélectif -Collectif /haut -bas 17
VI.6 Algorithme spécial 17
VII. Le système de contrôle d'un groupe d'ascenseur (SCGA) 17
VIII. Les circuits FPGA 18
IX. Conclusion 18

Chapitre 2 : Conception et Implémentation VHDL

I. Introduction 20
II. Présentation du hardware du système de contrôle global 20
III. Définition et affectation des entrées/sorties 21
IV. Principe général de fonctionnement du système de contrôle 26
V. L'architecture du circuit global 27
VI. Les composantes du système global 29
VI.1 Les mémoires RAM 29
VI.2 Le système de contrôle d'un groupe d'ascenseurs SCGA 29
VI.2.1 L'utilité du circuit SCGA 29
VI.2.2 L'entité SCGA 30
VI.2.3 Principe de fonctionnement du SCGA : 31
VI.2.4 L'architecture du circuit SCGA : 32
VI.2.5 Les composantes du système SCGA 33
VI.2.6 Exemple de simulation du circuit SCGA : 53
VI.3 Le système de contrôle local d'un ascenseur SCL : 57
VI.3.1 Intérêt du circuit SCL 57
VI.3.2 L'entité SCL 57
VI.3.3 Architecture du circuit SCL 58
VI.3.4 Principe de fonctionnement du circuit SCL 59
VI.3.5 Les composantes du circuit SCL 60
VI.3.5.1 Le circuit CH_CB 63
VI.3.5.1.1 Stratégie de fonctionnement 63
VI.3.5.1.2 Machine d'état 63
VI.3.5.1.3 Exemple de simulation 67
VI.3.5.2 Le circuit CH_SB 69
VI.3.5.2.1 Stratégie de fonctionnement 69

Sommaire

VI.3.5.2.2 Machine d'état 71

VI.3.5.2.3 Exemple de simulation 72

VI.3.5.3 Le circuit SH_CB 75

VI.3.5.3.1 Stratégie de fonctionnement 75

VI.3.5.3.2 Machine d'état 75

VI.3.5.3.3 Exemple de simulation 77

VI.3.5.4 Le circuit SH_SB 78

VI.3.5.4.1 Stratégie de fonctionnement 78

VI.3.5.4.2 Machine d'état 80

VI.3.5.4.3 Exemple de simulation 81

VI.3.5.5 Le circuit SC_HB 84

VI.3.5.5.1 Stratégie de fonctionnement 84

VI.3.5.5.2 Machine d'état 84

VI.3.5.5.3 Exemple de simulation 86

VI.3.5.6 Le circuit ALG_SPL 89

VI.3.5.6.1 Stratégie de fonctionnement 89

VI.3.5.6.2 Machine d'état 89

VI.3.5.6.3 Exemple de simulation 91

VI.3.6 Exemple de simulation du circuit SCL 94

V. Exemple de simulation du système global 97

VI. Conclusion 104

Conclusion générale 108

Références Bibliographique 111

Annexe

1- Les pseudos algorithmes des algorithmes de dispatching 111

1-1 Collectif haut/ Collectif bas 111

1-2 Collectif haut/ Sélectif bas 112

1-3 Sélectif haut/ Collectif bas 113

1-4 Sélectif haut/ Sélectif bas 114

1-5 Sélectif Collectif / haut bas 115

1-6 Algorithme spécial 117

2- Structure de la programmation VHDL du système de gestion 118

Liste d'abréviations

Liste des abréviations

FPGA	Field Programmable Gate Arrays
TAM	Temps d'Attente Moyen
SCGA	Système de Contrôle d'un Groupe D'ascenseurs
SCL	Système de Contrôle Local
VHDL	VHSIC Hardware Description Langage
IGA	Immeuble de Grande Hauteur
API	Automate Programmable Industriel
AWT	Avreag Waiting Time
EGCS	Elevator Groupe Control System
LCS	Local Control System
RAM	Random Access Memory

Liste des figures

Liste des figures

FigureI.1 : Principe de l'ascenseur hydraulique……6

FigureI.2 : Les composantes de l'ascenseur hydraulique…..……....7

FigureI.3 : Ascenseur à traction à câbles ……..……..8

FigureI.4 : Les composantes d'un ascenseur électrique.…….. 10

FigureI.5 : Cabine de l'ascenseur ……10

FigureI.6 : Les portes d'ascenseur……...……11

FigureI.7 : Les boutons d'appels……..……....11

FigureI.8 : Les boutons d'envois……..12

FigureI.9 : Les afficheurs …..……..12

FigureI.10 : Le moteur……..……...13

FigureI.11 : Le système barrage……...14

FigureI.12 : Les algorithmes de dispatching ……..……16

FigureI.13 : System de gestion d'un groupe d'ascenseur…….18

FigureII.1 : Exemple d'un groupe d'ascenseur de 3 à 3 niveaux…….21

FigureII.2 : Les entrées/sorties du système de contrôle global……..……..22

FigureII.3 : L'environnement du système de control global ……..……26

FigureII.4 : Schéma synoptique du circuit principal ……..……...28

FigureII.5 :L'architecture interne de la mémoire RAM ……..……...29

FigureII.6 : L'entité du circuit SCGA ……..……..31

FigureII.7 : L'organigramme de fonctionnement du circuit SCGA ……..……31

FigureII.8 : Schéma synoptique du circuit SCGA ……..……32

FigureII.9 : Architecture interne du registre_HB..……..…33

FigureII.10: Organigramme de fonctionnement du module de modification des RAMs 34

FigureII.11 : Entête du module de modification……..…….……...35

FigureII.12 : Entité du séquenceur……..……..…...36

Liste des figures

FigureII.13 : Machine d'état du séquenceur……..……..37

FigureII.14 : Exemple d'un registre_HB……..……..…..…37

FigureII.15 : Résultat de simulation du séquenceur..……..…….38

FigureII.16 : Entité d'extracteur d'appel……..39

FigureII.17 : Organigramme de fonctionnement d'extracteur d'appel……...…….….40

FigureII.18 : Exemple d'un registre_HB……..…...……..41

FigureII.19 : Exemple d'un tableau de direction des ascenseurs……..…….……41

FigureII.20 : Résultat de simulation du Modelsim……..……...41

FigureII.21 : Entité du circuit TAM……...……...42

FigureII.22 : Les cas de situation d'ascenseur par rapport à l'appel à servir……..44

FigureII.23 : Organigramme de fonctionnement du circuit TAM……..46

FigureII.24 : Architecture interne du circuit TAM…….……..…..…47

FigureII.25 : Résultat de simulation du scénario proposé « TAM »……..49

FigureII.26 : Le tableau des algorithmes de dispatching……..……. ..…..…50

FigureII.27 : Entité du circuit ALGO..……..…….50

FigureII.28 : Le découpage de la journée suivant le temps et le trafic…….……..51

FigureII.29 : L'arbre de décision suivant le temps…….....…….…51

FigureII.30 : L'arbre de décision suivant le type d'appel……..…...……..52

FigureII.31 : L'arbre de décision final……..…….……52

FigureII.32 : Résultat de simulation du circuit ALGO……..……..…...53

FigureII.33 : Résultat de simulation du scénario « partie 1 » du circuit SCGA..……...55

FigureII.34 : Résultat de simulation du scénario « partie 2 » du circuit SCGA..……...56

FigureII.35 : Entité du circuit SCL……..58

FigureII.36 : Architecture interne du circuit SCL..…….....59

FigureII.37 : Organigramme de fonctionnement du circuit SCL..……....60

FigureII.38 : Les entres/sorties d'algorithme de dispatching……..61

Liste des figures

FigureII.39 : Machine d'état d'algorithme collectif haut/collectif bas…….……..64

FigureII.40 : Les capteurs placés dans les portes palières……...…….….65

FigureII.41 : Interprétation du résultat de simulation du circuit CH_CB..…...…...…..70

FigureII.42 : Machine d'état d'algorithme collectif haut/sélectif bas…….……71

FigureII.43 : Interprétation du résultat de simulation du circuit CH_SB……...74

FigureII.44 : Machine d'état d'algorithme sélectif haut/collectif bas……..……...75

FigureII.45 : Interprétation du résultat de simulation du circuit SH_CB…....…….….79

FigureII.46 : Machine d'état d'algorithme sélectif haut /sélectif bas……..80

FigureII.47 : Interprétation du résultat de simulation du circuit Sh_SB…….…….……83

FigureII.48 : Machine d'état d'algorithme sélectif collectif/haut bas……...…...85

FigureII.49 : Interprétation du résultat de simulation du circuit SC_HB…….……...88

FigureII.50 : Machine d'état d'algorithme algorithme spécial……...…….….90

FigureII.51 : Interprétation du résultat de simulation du circuit ALG_SPL…..……..93

FigureII.52 : Interprétation du résultat de simulation du circuit SCL…….…..……96

FigureII.53 : Interprétation du résultat de simulation du system de contrôle avec le scénario1……...……..……...99

FigureII.54 : Interprétation du résultat de simulation du system de gestion avec le scénario 2 (partie1)…….……...……..…...102

FigureII.55 : Interprétation du résultat de simulation du system de gestion avec le scénario 2 (partie2)……...….……..103

FigureAx.56 : Structure des programmes VHDL…….……..…….…..120

Liste des Tableaux

Liste des tableaux

Tableau II.1 : Entrées du système de gestion et leur utilité ……...…… 23

Tableau II.2 : Sorties du système de gestion et leur signification ……..…… ..25

Introduction Générale

Introduction générale

À l'ère de la vitesse, nous savons tous que le temps et l'énergie sont devenus de plus en plus précieux. Jour après jour, basant sur l'évolution technologique, l'être humain cherche à inventer des moyens qui facilitent la vie quotidienne et économisent le plutôt possible ces deux critères, surtout dans les endroits où nous faisons la plupart de nos activités.

Le transport vertical est le transport en commun le plus utilisé dans le monde, on parle de 8,5 millions d'ascenseurs fonctionnent, un total auquel viennent s'ajouter 500 000 nouvelles installations chaque année [8],il est devenu un élément familier et incontournable de nos immeubles d'habitation, de nos administration, bureaux, des musées, aéroports, hôpitaux etc…,il est intervenu avec le développement rapide de la construction de la ville et l'apparition d'immeubles de grande hauteur (IGA),qui font l'usage d'un ou plusieurs ascenseurs un aspect important et indispensable. Pour cet effet, le transport vertical devient l'un des premiers domaines que la technologie a appliqué ses évolutions afin d'augmenter sa performance et de diminuer sa consommation en énergie.

Avec l'invention du premier ascenseur électrique par Werner Siemens et Hulstie en 1880 [9], l'industrie de l'ascenseur était véritablement née, les évolutions industrielles cherche à développer le côté matériel des ascenseurs afin de mieux répondre aux besoins d'une urbanisation croissante : L'intégration des moteurs puissants, les perfections du système de sécurité, la généralisation des portes coulissantes automatiques, et tous les autres enrichissements du côté matériel rend l'ascenseur un équipement de luxe , mais très coûteux du coté énergie .Pour cet effet, le développement est passé au côté software, afin de construire un système de gestion qui exploite ces évolutions, et optimise le plutôt possible la consommation énergétique. À l'arrivé des transistors cette approche est devenue réalisable mais la naissance des microprocesseurs et l'épanouissement de l'informatique ont révolutionné le fonctionnement des ascenseurs, et fait entrer les systèmes de gestion dans une nouvelle ère c'est « l'ère de l'électronique»

Les chercheurs commencent à construire des systèmes de gestion du trafic dans les immeubles a un seul ascenseur à base de l' Automates Programmables Industriels (API), qui est une unité de contrôle dédiée à la gestion d'un cycle qui évolue en fonction d'événements extérieurs , cette technique est bien amélioré avec l'intégration de l'électronique de puissance, qui permet le contrôle direct des moteurs à courant alternatif par la variation de tension et de fréquence. Mais à l'apparition de la notion du groupe d'ascenseur , provoquée par la croissance de la hauteur des immeubles, la gestion simple des automates ne peut pas être

satisfaite, le comportement complexe du groupe d'ascenseur exige la technologie de chercher à construire des systèmes de gestion rendant les ascenseurs, Plus rapides, moins gourmands en énergie, plus spacieux et plus accessibles aux personnes âgées et handicapées. Grâce à l'apport de la microinformatique, la conception des systèmes intelligents de gestion du trafic sera possible afin d'atteindre des degrés plus élevés d'opération autonome pour assurer une haute performance même dans des conditions défavorables, les ascenseurs seront encore plus intelligents pour favoriser un trafic optimal et le rendre plus fluide. Et avec l'exploitation de l'évolution de l'industrie du circuit intégré, cette performance sera agrandie par l'intégration de ce système complexe dans une seule carte dédié pour la gestion des ascenseurs, qui consomme le minimum de la puissance, comme les cartes FPGA

Ce travail de mémoire a pour objectif de concevoir et implémenter un système intelligent de gestion de M ascenseurs dans un immeuble à N niveaux, intégré dans un circuit FPGA, composé du système de contrôle d'un groupe d'ascenseur SCGA ,et des systèmes de contrôle locaux SCLs un pour chaque ascenseur, afin d'optimiser le temps d'attente moyen des usagers et la consommation énergétique des ascenseurs, ce système doit réagir d'une maniéré correcte et en temps réel après chaque évènement provenant de l'environnement , il doit réduire la consommation énergétique le plutôt possible , n'oublions pas l'obligation du distinct des appels prioritaires des responsable et des personnes handicapées à travers les capteurs des cartes magnétiques , et les servir directement .

La démarche adoptée dans ce travail est comme suit :

- Le premier chapitre présente brièvement les ascenseurs, leurs types et leurs composantes, avec un aperçu sur les algorithmes de dispatching.
- Le deuxième chapitre traite la problématique, on donne des explications approfondies sur l'architecture proposée et les algorithmes de dispatching utilisés, avec des validations de bon fonctionnement des circuits construits à partir des résultats des simulations fournies par Modelsim de Modeltech
- Enfin on clôture ce document avec une conclusion générale et des perspectives

Chapitre 1 :
Présentation et description des ascenseurs

I. Introduction :

La circulation verticale des personnes est devenue aujourd'hui l'un des paramètres essentiels de la qualité de vie dans les bâtiments et les gratte-ciels, qu'ils sont devenus une solution pour une expansion verticale, ce qui conduit à la nécessite d'inventer ou de créer un élément ou un dispositif pour se déplacer entre plusieurs étages, afin de faciliter la circulation des personnes. Les ascenseurs et les escaliers mécaniques deux inventions importants pour répondre à ces besoins [1].

Dans ce sens, on va présenter dans ce chapitre l'ascenseur dans le but de répondre aux questions suivantes : c'est quoi un ascenseur ? Et quels est ses types et ses composantes ? Et comment le contrôler ?

II. Définition :

Un ascenseur est un moyen de transport sur un axe vertical, et comme tous les moyens de transport il sera soumis aux mêmes lois du capacité et temps de transport, d'attente, de saturation, de sécurité, etc. ...

Cet appareil qui dessert des niveaux définis à l'aide d'une cabine, destinés au transporte verticale des personne et des marchandises dans des immeubles à niveaux multiples, On distingue techniquement trois types d'appareils [2] :

- Les ascenseurs destinés à l'usage unique des personnes.
- Les monte charges destinés aux personnes et marchandises.
- Les monte-charge industriels destinés aux marchandises uniquement.

III. Les catégories d'ascenseur:[4]

On distingue deux grandes familles d'ascenseur :

- Les ascenseurs à traction à câble.
- Les ascenseurs hydrauliques.

En général, ces deux types utilisent l'énergie électrique pour déplacer verticalement la cabine d'ascenseur, cependant, les ascenseurs hydrauliques sont nettement moins utilisés que les ascenseurs à treuil.

III.1 L'ascenseur hydraulique :

- ***Principe :***

Comme toute machine hydraulique, la pompe met sous pression l'huile qui pousse le piston hors du cylindre vers le haut. Lorsque la commande de descente est programmée, la vanne (by-pass) de la pompe permet de laisser sortir l'huile du cylindre vers le réservoir, comme montre la figure I.1 ,

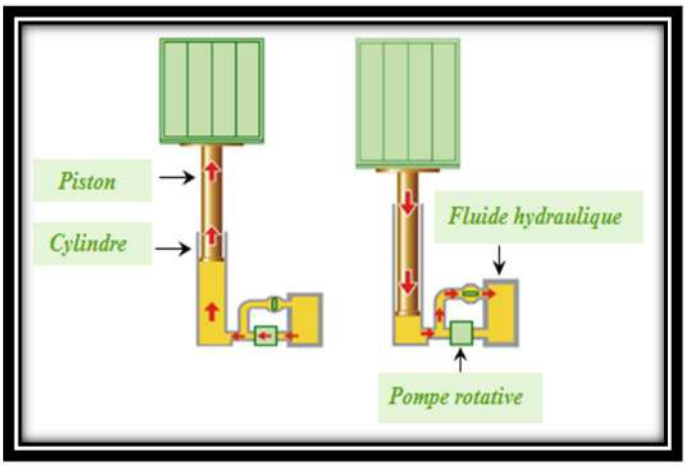


Fig. I.1. Principe de l'ascenseur hydraulique

- ***Description :***

En général, les ascenseurs hydrauliques sont utilisés pour satisfaire des déplacements relativement courts de l'ordre de 15 à 18 m maximums. Plusieurs modèles existent sur le marché, on en cite :

- À cylindre de surface.
- À cylindre enterré.
- Télescopiques à cylindre de surface.

Ils se composent principalement (figure I.2):

- d'une cabine.
- des guides.
- d'un ensemble pistons-cylindres hydrauliques placé sous la cabine de l'ascenseur.
- d'un réservoir d'huile.
- d'un moteur électrique accouplé à une pompe hydraulique.
- d'un contrôleur.

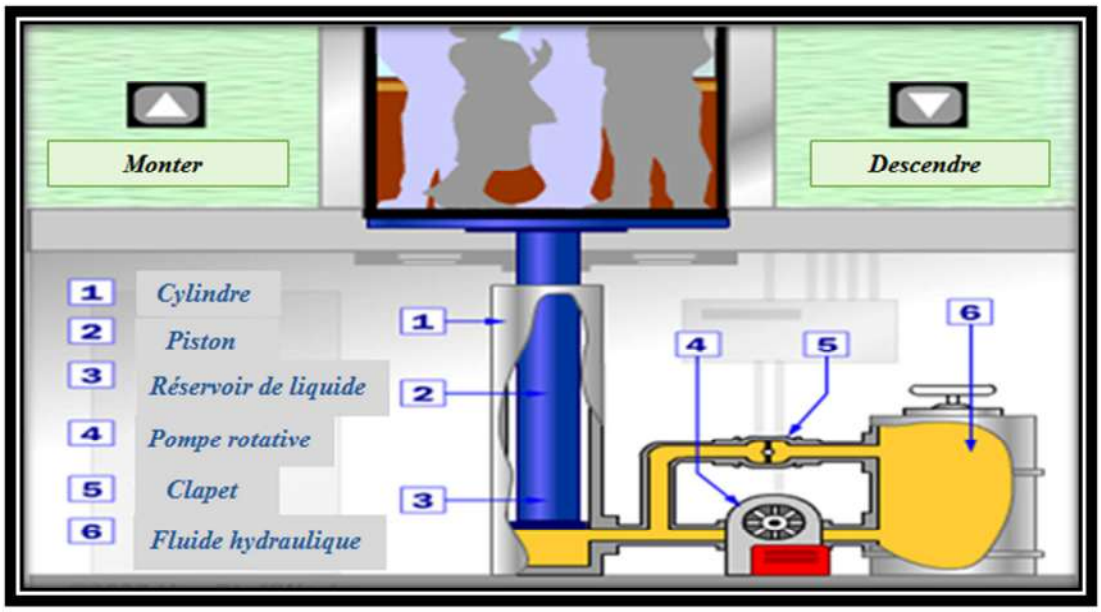


Fig. I.2. Les composantes de l'ascenseur hydraulique.

Les différents modèles permettent de tenir compte des critères de place, de hauteur d'immeuble à desservir vu que la hauteur possible est limitée, stabilité de sol et de sous-sol, risque de pollution par rapport au sol.

- ***Energie :***

Énergétiquement parlant, les ascenseurs hydrauliques posent un problème dans le sens où il n'y a pas de contrepoids qui équilibre la cabine comme dans les systèmes à traction à câble.

- ***Avantage :***

- Précision au niveau du déplacement (mise à niveau).
- Charge importante.
- réglage facile de la vitesse de déplacement.
- ne nécessite pas de local de machinerie.
- implantation facile dans un immeuble existant.

- ***Inconvénient :***

- course verticale limitée à une hauteur entre 15 à 18m.
- la sécurité incendie compliquée à cause de la quantité importante d'huile.
- risque de pollution des sous-sols.
- consommation énergétique importante.
- l'absence de contrepoids provoque un surdimensionnement, des consommations et des appels de puissance importants (à charge et à vitesse égales, il faut de l'ordre de 3 fois plus de puissance).
- une vitesse réduite.

III.2 L'ascenseur à traction à câbles :

- ***Principe et description:***

Les ascenseurs à traction à câbles sont les types d'ascenseurs les plus fréquemment utilisés, notamment dans les bâtiments tertiaires, l'entraînement des ascenseurs est assuré par un moteur électrique , ils se différencient entre eux selon le type de motorisation :

- à moteur-treuil à vis sans fin.
- à moteur-treuil planétaire.
- à moteur à attaque directe (couramment appelé "**Gearless**" ou sans treuil)

Un ascenseur à treuil se compose essentiellement d'une cabine, d'un contrepoids, des câbles reliant la cabine au contrepoids, des guides, et d'un système de traction au-dessus de la cage de l'ascenseur comme montre la figure I.3.

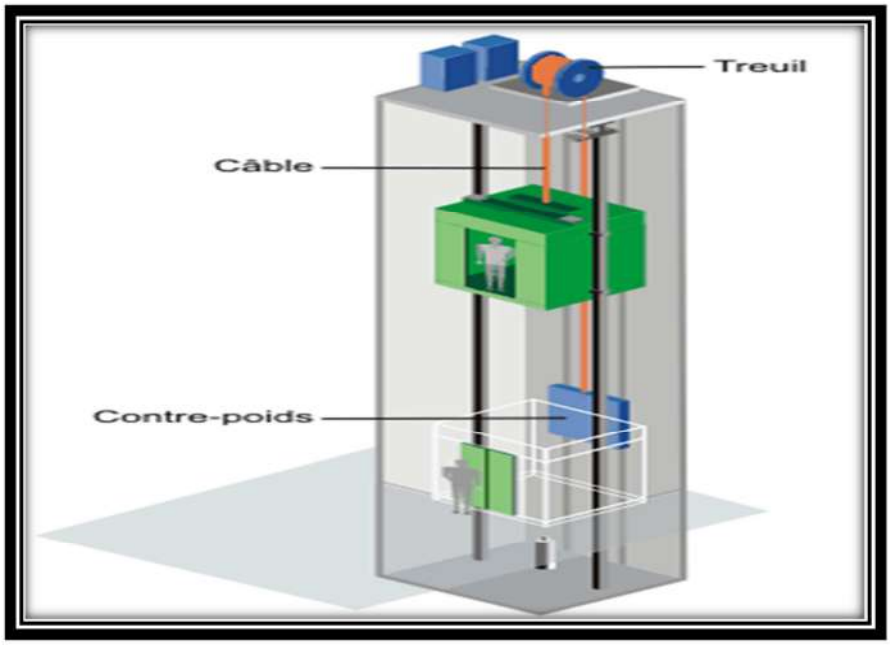


Fig. I.3. Ascenseur à traction à câbles

- ***Energie :***

Les ascenseurs à traction à câbles sont plus intéressants que les ascenseurs hydrauliques dans le sens où le contrepoids réduit fortement, quel que soit le type de motorisation, les consommations et les courants de démarrages sont

réduits par rapport aux ascenseurs hydrauliques (à charge et à vitesse égales, la puissance est réduite d'un facteur 3).

- ***Avantage :***
  - absence de limite de hauteur du bâtiment suivant le type de motorisation précision au niveau de la vitesse et du déplacement.
  - rapidité de déplacement.
  - grande plage de variation de vitesse.
  - efficacité énergétique importante.
  - pas de souci de pollution.
  - la réduction des coûts de dimensionnement de l'installation électrique.
  - la maîtrise des chutes de tension sur le réseau.
  - la limitation des consommations et des appels de puissance.

- ***Inconvénient :***
  - En version standard (treuil à réducteur), nécessite un local de machinerie en toiture.
  - exigence très importante sur l'entretien.
  - Nécessité de tenir compte du poids de la cabine, des câbles, du contrepoids, de la structure de la salle des machines, et des équipements de la salle des machines. Le poids total repose sur la structure du bâtiment (colonne ou mur de gaine porteur renforcé) et se reporte au niveau des fondations.
  - Peut imposer un volume construit inesthétique visible sur le toit.
  - problème d'accessibilité.
  - compacité de la gaine réduite par la présence de la cabine et du contrepoids et, par conséquent, réduction de la surface utile dans les étages du bâtiment.

IV. Différentes composantes d'un ascenseur électrique : [1] [4]

On se limite aux composants d'un ascenseur à treuil car il est le cas qui va être traité dans notre projet, la figure I.4 montre tous ses composants

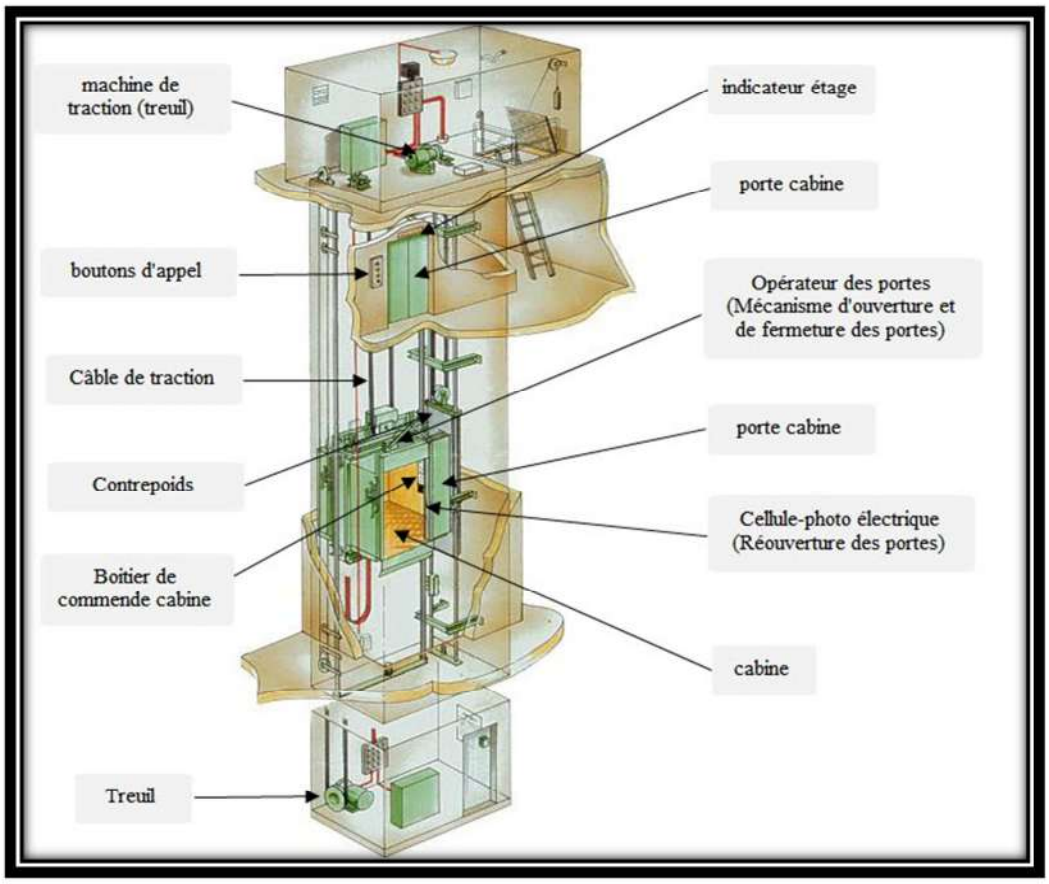


Fig. I.4. Les composantes d'un ascenseur électrique

IV.1 Cabine d'ascenseur :

Elément composé d'un plancher, de parois et d'un toit destiné à accueillir les personnes et les marchandises (La partie visible de l'ascenseur), Cet élément est inséré et fixé dans un cadre appelé suspension cabine (figure I.5) .



Fig. I.5. Cabine de l'ascenseur

IV.2 Porte de cabine :

Porte à fermeture généralement automatique destinée à confiner l'utilisateur dans la cabine pendant le déplacement de celle-ci, lui interdisant tous contact avec les parties extérieures de la cabine.

IV.3 Portes palières :

C'est la porte externe de l'ascenseur (figure I.6). Chaque ascenseur est équipé d'autant de porte palière que de nombre d'étage. Elles peuvent être battantes et commandées manuellement ou automatiques et coulissantes (à ouverture centrale ou latérale). Elles doivent être équipées d'un dispositif empêchant leurs ouvertures si la cabine n'est pas sur le niveau et bloquant le départ pendant leur ouverture.



Fig. I.6. Les portes d'ascenseur

IV.4 Boutons d'appels:

On nomme boutons d'appels les boutons installés aux paliers (figure I.7), ils peuvent être communs si on utilise plusieurs ascenseurs



Fig. I.7. Les boutons d'appels.

IV.5 Boutons d'envois:

Les boutons d'envois sont installés dans la cabine (figure I.8).



Fig. I.8. Les boutons d'envois

IV.6 Les afficheurs :

Dans chaque étage, on trouve des indicateurs de direction de la cabine (haut ou bas) ainsi de numéro d'étage (figure I.9)



Fig.I.9. les afficheurs

IV.7 Charge utile:

Capacité maximum en Kg qu'une cabine d'ascenseur peut contenir. Au-delà de cette capacité, le système de traction n'est plus en mesure de contrôler le déplacement et l'arrêt correct de la cabine. Dans certains cas de surcharge exagérée, des blocages intempestifs peuvent se produire. La charge utile est une indication qui doit obligatoirement figurer dans la cabine.

IV.8 Contrepoids:

Elément destiné à contre balancer le poids de la suspension cabine augmenté de la moitié de la charge utile. Celui-ci est constitué d'une suspension métallique contenant des gueuzes en fonte destinées à l'alourdir. Lorsque la cabine d'ascenseur monte, le contrepoids descend.

IV.9 Treuil :

Machine composée d'un dispositif de freinage et d'un moteur et destinée à actionner les câbles de traction de l'ascenseur.

IV.10 Câble de traction :

Câble généralement en acier, destiné à suspendre la cabine au contre poids et à déplacer celle-ci.

IV.11 Le moteur :

Le moteur le plus couramment employé est actuellement le moteur asynchrone à double cage (figure I.10), Ce choix à des avantages faible (y compris pour des fortes puissances), très robuste (pas de contact) et bon rendement.

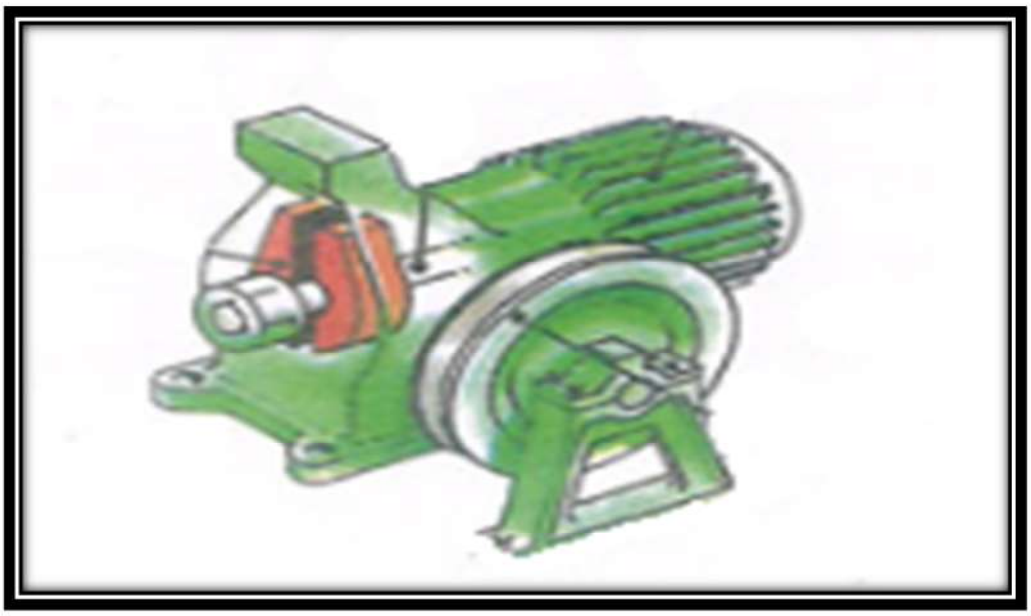


Fig.I.10. le moteur

IV. photocellules :

Elles sont destinées à la détection d'objets de toute nature, l'utilisateur a le choix entre trois systèmes de base :

- Le système barrage (figure I.11)
- Le système reflex.
- Le système de proximité.

Dans les portes d'ascenseur on utilise le système barrage, pour les longues portes et objets réfléchissants .Ce système de détection contient un émetteur et récepteur séparés permet de détecter tout objet interrompant le faisceau lumineux (les personnes).donc la porte ne ferme pas.

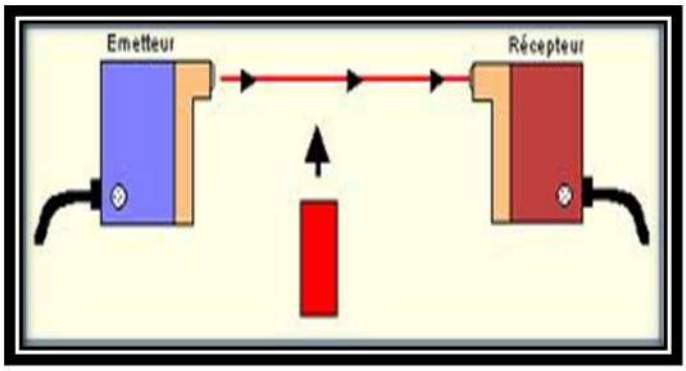


Fig. I.11. Le système barrage

V. Les critères du choix du type d'ascenseur : [5]

En général, les dépenses énergétiques des ascenseurs ne sont pas la priorité des gestionnaires de bâtiments tertiaires. En effet, la préoccupation première reste avant tout : emmener un maximum de monde en toute sécurité et avec un maximum de confort, on retrouve des critères de choix:

- ***constructifs :***

Hauteur de bâtiment, espace disponible au niveau des étages, possibilité de placer une salle des machines au sommet de la gaine, stabilité du terrain de sécurité.

- ***organisationnels*** **:**

Comme le type de fonction du bâtiment, son occupation et son type de fonctionnement en garantissant une performance de confort et de trafic (rapport vitesse/charge)

- ***énergétiques*** **:**

Basées essentiellement sur la consommation et les appels de puissance de la motorisation.

VI. Le système de contrôle d'un ascenseur (SCL) :

De nombreux modernes ascenseurs sont contrôlés par un système local, qui doit traiter toutes les informations pertinentes et tourner correctement le moteur pour déplacer la cabine dans la position correcte. Pour ce faire, ce système a besoin de connaître au moins deux choses :

- Où les gens veulent aller ??
- Où la cabine d'ascenseur est ??

Il est très facile de découvrir où les gens veulent aller ,les boutons de l'ascenseur et les boutons dans chaque étage sont tous raccordés au SCL, quand quelqu'un appuie ces boutons, il enregistre cette demande, et à partir des conditions du trafic on peut choisir un algorithme , parmi les algorithmes de dispatching, pour servir la demande enregistrée, et l'exploiter pendant une certaine période, jusqu'à ce que l'état du trafic soit à nouveau modifié.

Les algorithmes de dispatching sont les aspects les plus importants dans les systèmes de contrôle des ascenseurs, ces algorithmes peuvent réduire le temps d'attente moyen de passager ***TAM*** (***AWT*** : ***A****verage* ***w****aiting* ***t****ime*) jusqu'à une certaine valeur, et réduire aussi la consommation électrique du système d'ascenseur. Ici nous mettons en œuvre six types d'algorithmes de dispatchings (figure I.12).

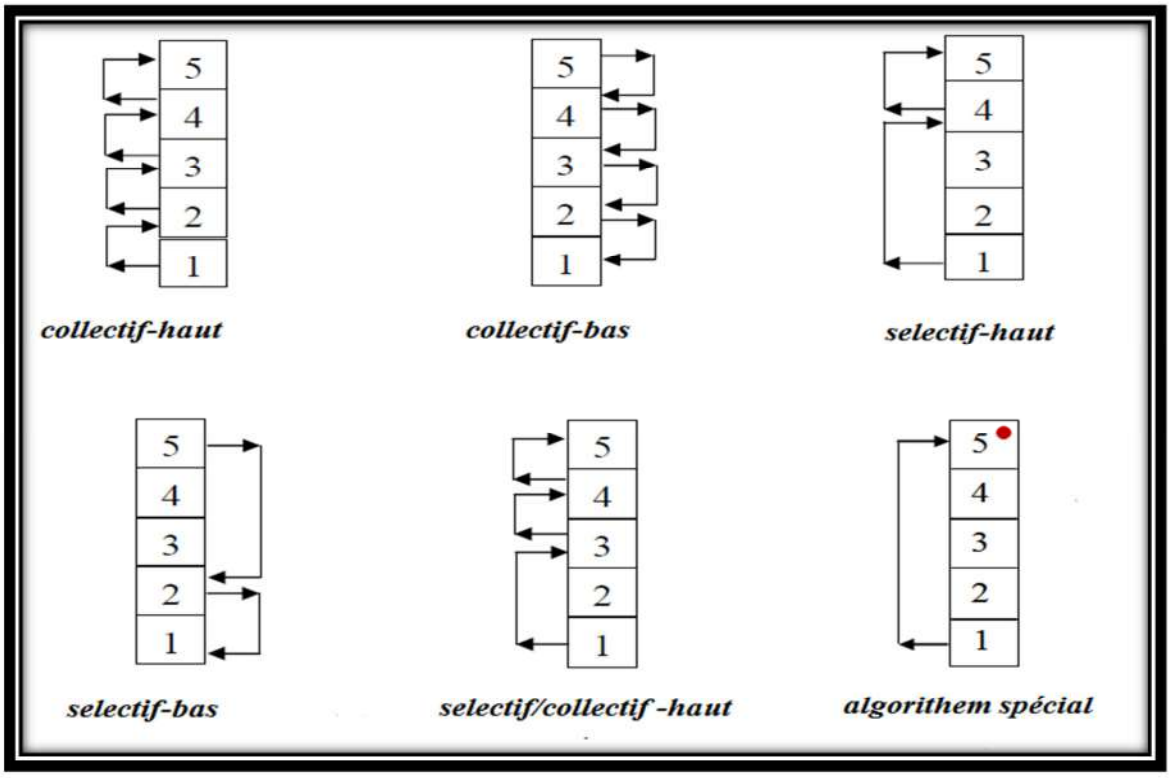


Fig. I.12. Les algorithmes de dispatching

VI.1 Collectif-haut / collectif –bas :

Mettant en œuvre le principe collectif dans les deux sens de déplacement haut et bas, l'ascenseur doit servir toutes les demandes ayant la même direction de son déplacement, et qui ne sont pas encore dépassées. S'il n'y a pas d'autres demandes dans son chemin, il doit changer son sens de déplacement s'il existe des demandes dans la direction opposée, sinon il devient en état d'attente.

VI.2 Sélectif –haut / Sélectif –bas :

Dans le mode sélectif l'ascenseur doit servir la demande correspondante à l'étage le plus bas (en cas de montée) et le plus haut en cas de descente, et quand il atteint l'étage choisi, il ne répond à aucune autre demande avant qu'il termine tous les sélections des passagers à l'intérieur de la cabine.

VI.3 Collectif-haut / Sélectif-bas :

Cet algorithme est adapté lorsque le trafic en direction du haut est majoritaire, l'ascenseur doit servir les demandes effectuées dans le plus petit temps possible selon le

principe collectif, et il adapte le mode sélectif dans le sens opposé, pour diminuer le nombre de stop lors de sa descente.

VI.4 Sélectif-haut /Collectif -bas :

Le même principe d'algorithme précédant (collectif-haut/sélectif-bas) mais on inverse le chemin. Cet algorithme est adapté lorsque le trafic en direction du bas est majoritaire.

VI.5 Sélectif -Collectif /haut -bas :

Cet algorithme est approprié lorsque le trafic des passagers est égal dans les deux directions haut et bas, lors de son démarrage l'ascenseur choisit la première demande par le mode sélectif, ensuite il termine son déplacement par le mode collectif pour servir les sélections et les demandes en attentes, et la même chose pour le sens opposé.

VI.6 Algorithme spécial :

Cet algorithme est spécialisé pour les demandes prioritaires détectées par des capteurs des cartes magnétiques placés dans chaque étage, cette priorité est donnée aux responsables et aux personnes handicapées. A la détection d'un appel spécial, l'ascenseur doit le servir directement en ignorant toutes les autres demandes en attentes sauf les sélections venus à l'intérieur de la cabine pour la vider, il ne les prend en compte qu'a après amène la personne spéciale son l'étage désiré

VII. Le système de contrôle d'un groupe d'ascenseur (SCGA)

Avec l'apparition d'immeubles de grande hauteur l'utilisation de plusieurs ascenseurs est devenue plus importante et ses qualités du service sont de plus en plus élevées.

Afin d'améliorer leurs performances le déplacement de ces ascenseurs est géré par un system de contrôle d'un groupe d'ascenseur **SCGA**. C'est un système de commande qui doit sélectionner l'ascenseur le plus approprié à la demande d'un ou plusieurs passagers et détermine le meilleur algorithme à exécuter dans chaque **LCS,** comme montre la figure I.13

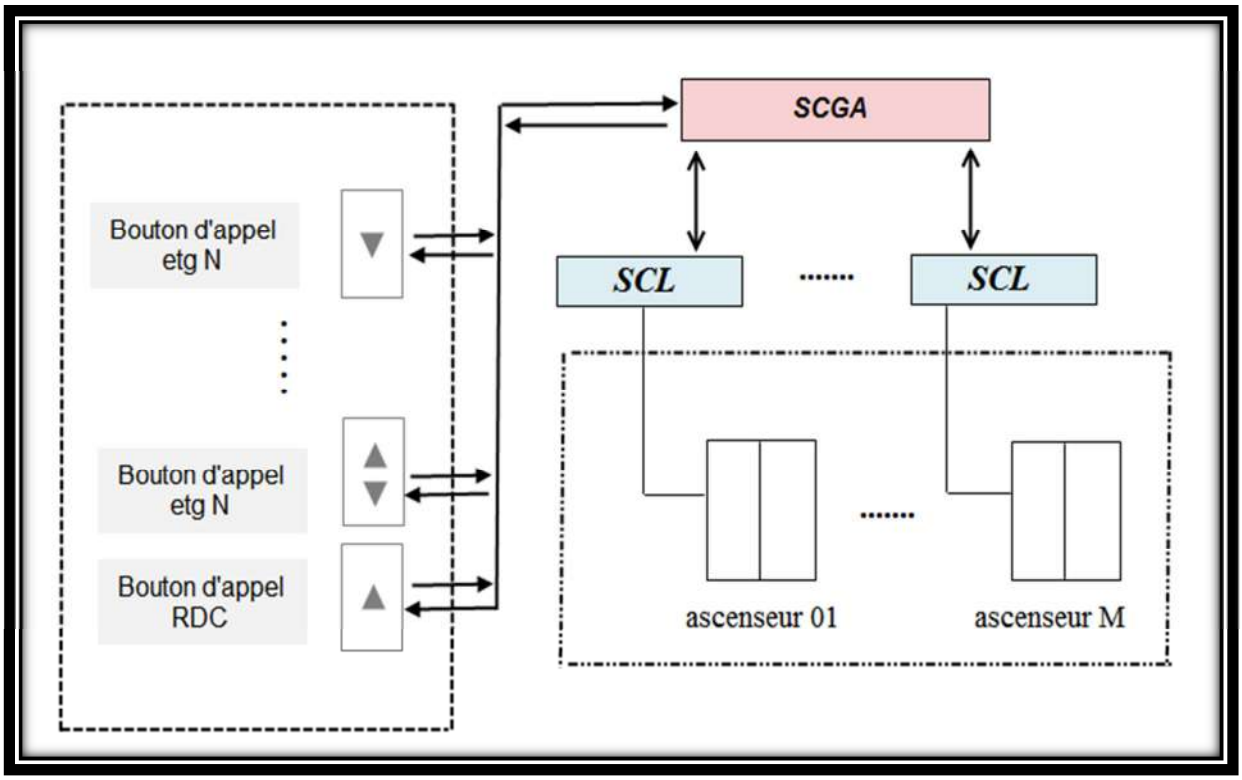


Fig. I.13. System de gestion d'un groupe d'ascenseur

VIII. Les circuits FPGA:[6]

Les FPGAs sont des circuits à architecture programmable qui ont été inventés par la société XILINX en 1985. Ils sont entièrement reconfigurables et ne demandent donc pas de fabrication spéciale en usine, ni des systèmes de développement coûteux ; ceci permet de les reprogrammer à volonté afin d'accélérer notablement certaines phases de calculs.

Un autre avantage de ces circuits est leur grande souplesse qui permet de les réutiliser à volonté dans des algorithmes différents en un temps très court (quelques millisecondes).

IX. Conclusion :

Dans ce chapitre nous avons présenté une vue générale sur les ascenseurs en faisant la lumière sur les composantes principales d'un ascenseur électrique, ainsi que les systèmes de contrôle d'un ou plusieurs ascenseurs.

CHAPITRE 2
Conception et implémentation VHDL

I. Introduction

Afin de fournir des services pratiques et confortables de transport entre les étages d'immeuble pour les passagers, un système de gestion global est installé. Ce système doit être capable d'interagir avec tous les évènements extérieurs, et de prendre les meilleures décisions pour optimiser le déplacement des cabines des ascenseurs. Il doit être capable de gérer plusieurs demandes des utilisateurs simultanées, et de choisir les meilleurs algorithmes, afin de servir le plus rapidement toutes les demandes.

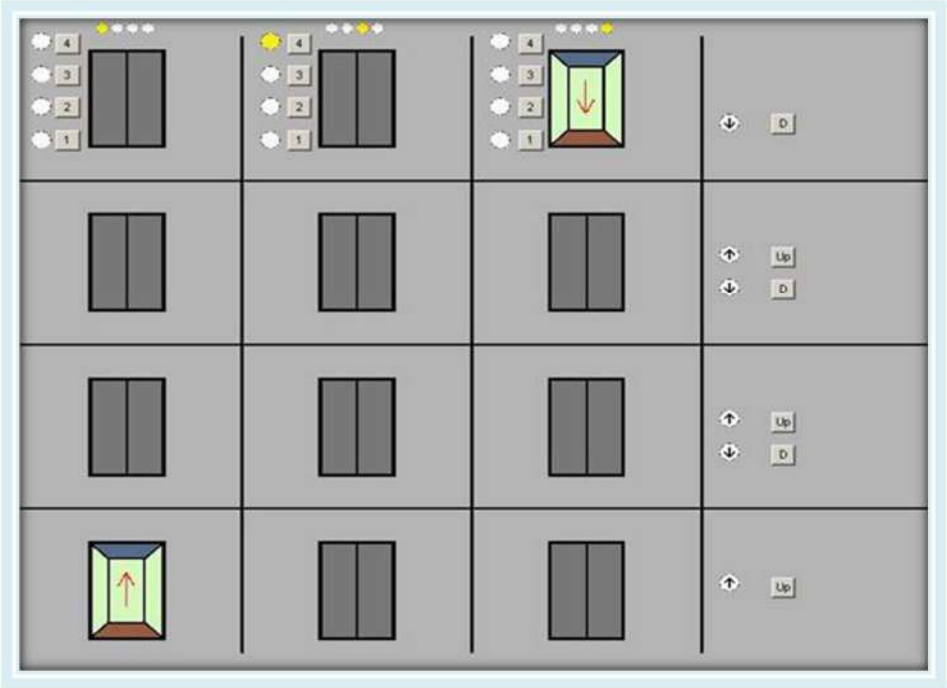
Dans ce chapitre nous allons décrire en détail le fonctionnement de ce système, et on va proposer une architecture complète satisfaisant tous ses objectifs, puis on va l'implémenter sur un langage matériel de haut niveau« VHDL », avec des tests de simulation pour valider le bon fonctionnement.

II. Présentation du hardware du système de contrôle global

Notre projet permet de construire un système de contrôle gérant **M** ascenseurs à **N** niveaux, en effet une description des entrées/sorties s'avère importante :

- ✓ Sur la face avant de chaque ascenseur des boutons poussoirs d'appel à chaque niveau permettant d'appeler la cabine d'un parmi les M ascenseur (figure II.1).
- ✓ Des boutons de commande à l'intérieur de chaque cabine permettant de sélectionner l'étage désiré.
- ✓ La montée et la descente de chaque ascenseur sont assurées par la rotation d'un moteur.
- ✓ Les niveaux des cabines sont détectés par des capteurs de présence dans chaque étage.
- ✓ L'ouverture et la fermeture des portes sont assurées par un vérin de translation.
- ✓ La détection d'ouverture et de fermeture d'une porte se fait grâce à des détecteurs de fin de course.
- ✓ Les ascenseurs doivent fonctionner d'une manière optimale par rapport au temps. Aux positions indiquées par les touches sélectrices dans les cabines ou bien dans les étages, les portes des étages doivent s'ouvrir et se fermer, pour des raisons de sortie des passagers, par commande d'un contrôleur de temps et des interrupteurs de fin de course. Des empêchements de la fermeture doivent être signalés par des barrages photoélectriques et entrainer que la porte s'ouvre de nouveau.

- ✓ La surcharge des cabines et les arrêts d'urgence sont des entrées qu'il faut prendre en compte.
- ✓ Dans les weekends et la nuit il n'y a qu'un seul ascenseur qui fonctionne, cela est assuré par les 2 entrées « jour et temps ».



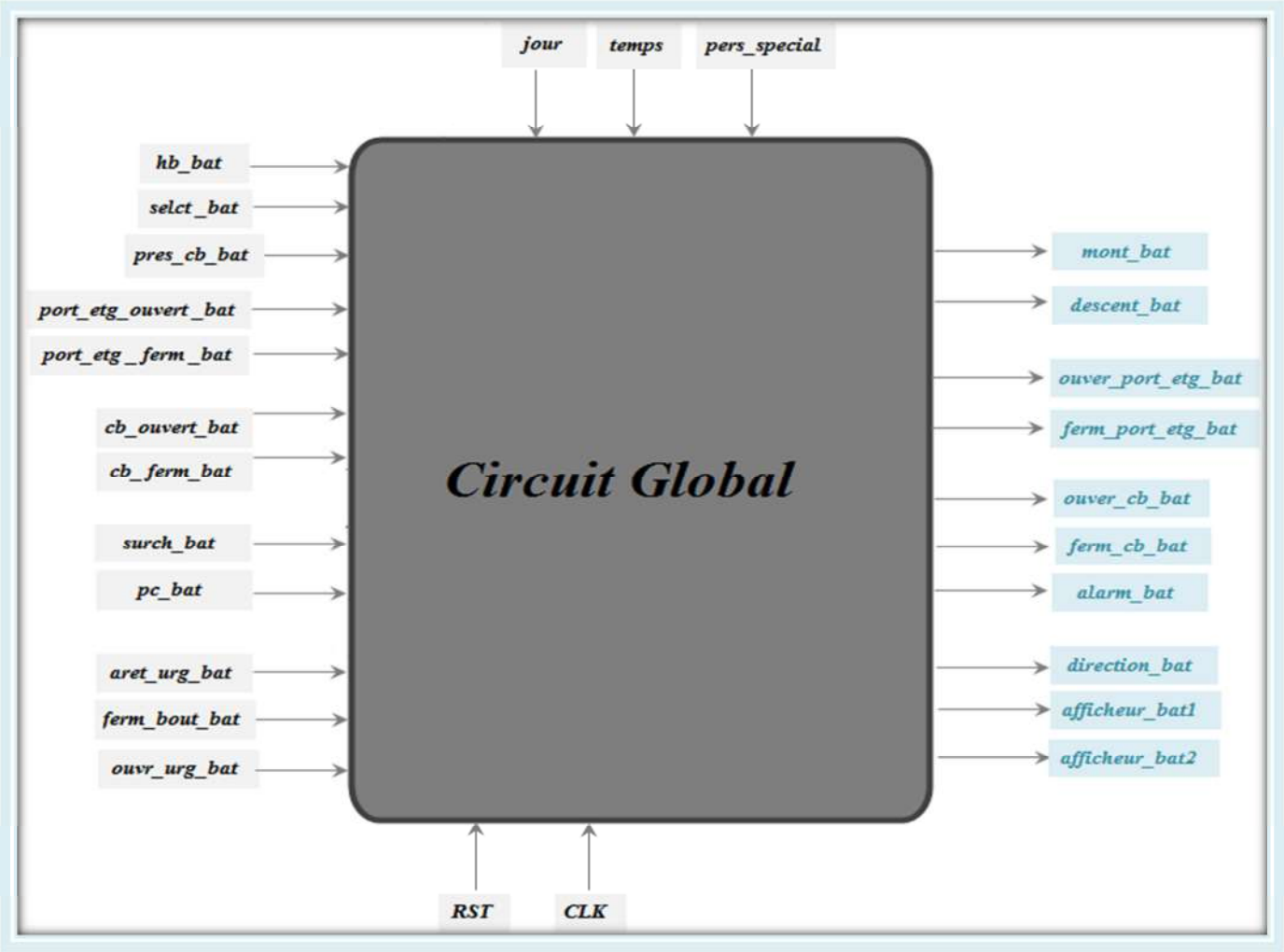
II.1 : exemple d'un groupe d'ascenseur de 3 à 3 niveaux

III. Définition et affectation des entrées/sorties

Dans cette partie on va donner les symboles des entrées/sorties (figure II.2) de notre système de contrôle qui seront utilisés par la suite dans toutes les explications ainsi que les programmes VHDL avec :

N : nombre des étages

M : nombre des ascenseurs



FigII.2 : les entrées/sorties du système de contrôle global

C'est un système à 17 entrées et 10 sorties, en général les entrées sont des détecteurs placés afin de détecter les changement d'état des :« cabines des ascenseurs , des portes étages et cabines» , avec des capteurs de « temps »et de « jour », n'oublions pas les détecteurs des « cartes magnétiques » pour les utilisateurs spéciaux (responsable, personne handicapée) , notant que les sorties sont en réalité des commandes envoyées aux actionneurs « des moteurs, des portes ,des sonnettes d'alarme ainsi que des afficheurs placés dans chaque étage ».Le tableau II.1 donne en détail chaque entrée et son utilité :

Tableaux II.1 : Entrées du système de gestion et leur utilité

Symbole de l'entrée	*Type*	*description*
hb_bat	*Matrice : 2*N*	*les boutons des appels hauts (ligne 1 de la matrice) et bas (ligne 2 de la matrice) de tous les étages (N colonnes)*
selct_bat	*Matrice : M*N*	*Les boutons des sélections des étages désirés (N étages) dans chaque cabine parmi les M.*
pres_cb_bat	*Matrice : M*N*	*Les Capteurs de présence de chaque ascenseur placés dans les N étages*
port_etg_ouvert_bat	*Matrice : M*N*	*Les Capteurs d'ouverture de toutes les portes étages*
port_etg_ferm_bat	*Matrice : M*N*	*Les Capteurs de fermeture de toutes les portes étages*
cb_ouvert_bat	*Vecteur : M*	*Les Capteurs d'ouverture des portes cabines de M ascenseurs*
cb_ferm_bat	*Vecteur : M*	*Les Capteurs de fermeture des portes cabines de M ascenseurs*
surch_bat	*Vecteur : M*	*Les Capteurs de surcharge de M ascenseurs*
pc_bat	*Vecteur : M*	*Les Capteurs de photo cellule pour M ascenseurs*

aret_urg_bat	*Vecteur : M*	*Les boutons d'arrêt d'urgence pour M ascenseurs*
ferm_bout_bat	*Vecteur : M*	*Les boutons de fermeture des portes dans les M ascenseurs*
ouvr_urg_bat	*Vecteur : M*	*Les boutons d'ouverture des portes dans les M ascenseurs*
temps	*Entier*	*Horloge externe pour l'indication d'heure*
jour	*Entier*	*Il nous indique le jour par un numéro entre 0 et 6 (0=samedi, … ,6=vendredi)*
pers_special	*Std_logic*	*Un capteur d'utilisation descartes spéciales pour les responsables et les personnes handicapées*
rst	*Std_logic*	*Bouton poussoir d'alimentation et de réinitialisation*
clk	*Std_logic*	*horloge*

Le tableau II.2 explicite chaque commande aussi que sa signification :

Tableaux II.2 : Sorties du système de gestion et leur signification

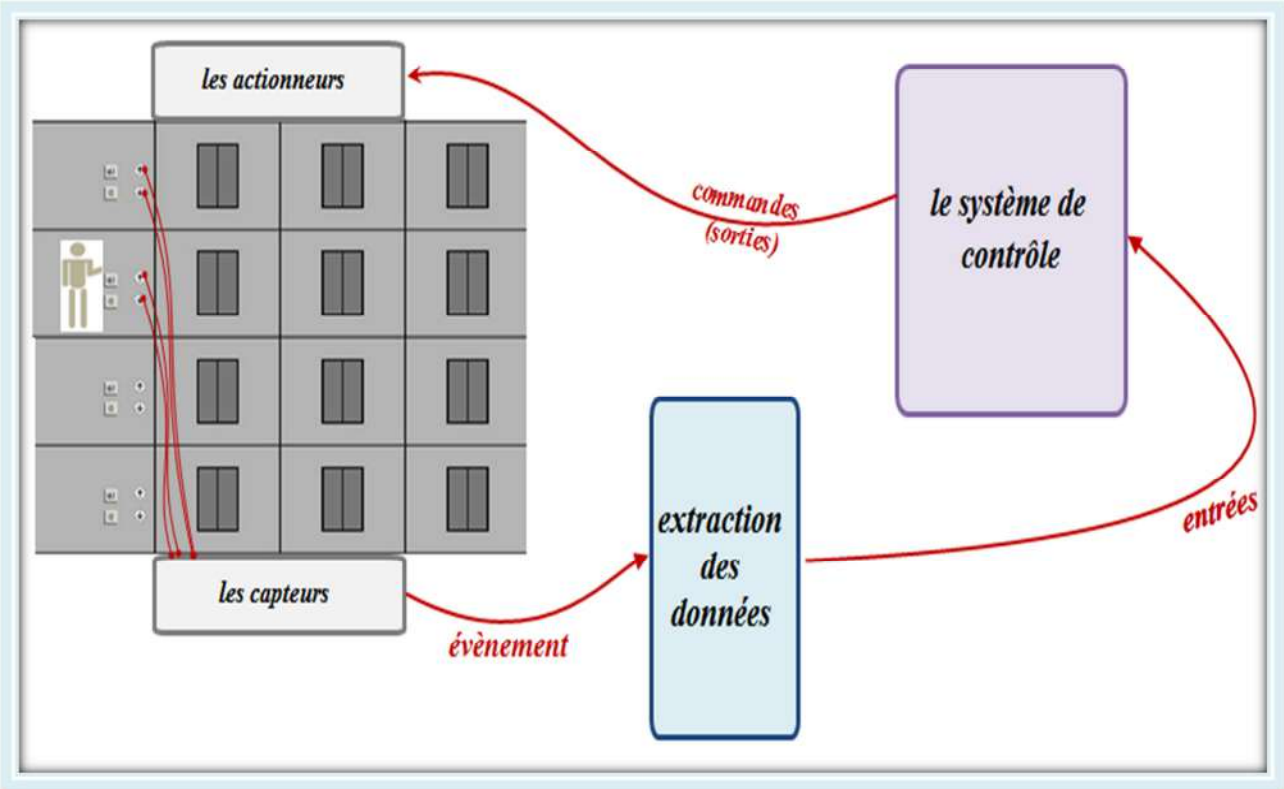
Symbole de la sorties	***Type***	***Description***
mont_bat	*Matrice : M*N*	*Les Commandes de monter i étages (0<i<N) pour les M ascenseurs*
descent_bat	*Matrice : M*N*	*Les Commandes de descendre i étages (0<i<N) pour les M ascenseurs*
ouver_port_etg_bat	*Matrice : M*N*	*Les Commandes d'ouverture de portes étages dans chaque niveau pour chaque ascenseur*
ferm_port_etg_bat	*Matrice : M*N*	*Les Commandes de fermeture des porte étages dans chaque pour chaque ascenseur*
ouver_cb_bat	*Vecteur : M*	*Les Commandes d'ouverture des portes cabines des ascenseurs*
ferm_cb_bat	*Vecteur : M*	*Les Commandes de fermeture des portes cabines des ascenseurs*
alarm_bat	*Vecteur : M*	*Les commandes de Lancement de l'alarme en cas de surcharge pour chaque ascenseur*
direction_bat	*Vecteur : M*	*La Direction de mouvement (vers le haut ou vers le bas) de chaque ascenseur*
afficheur_bat1afficheur_bat1	*Matrice : M*7*	*Deux afficheurs de sept segments pour chaque ascenseur indiquant son emplacement*

IV.Principe général de fonctionnement du système de contrôle

A partir des évènements effectués dans l'immeuble, un extracteur de données doit fournir au système de gestion toutes les entrées nécessaires pour son fonctionnement (les entrées qui sont détaillées précédemment), ce système doit réagir suivant la nature de l'évènement détecté, et le type de la demande arrivée :

- ✓ il cherche à servir la demande dans le moins TAM possible si l'évènement est un appel simple (haut ou bas), contrairement à une demande spéciale (responsable ou personne handicapée) qu'il doit la servir directement.
- ✓ à chaque sélection il s'occupe d'amener les passagers aux étages désirés.
- ✓ Il doit répondre aux demandes d'ouverture et de fermeture des portes.

Tout ça se déroule à partir des commandes envoyées aux actionneurs pour que les ascenseurs mettent à jour leurs états comme montre la figure II.3 :



FigII.3 : l'environnement du système de control global

V. L'architecture du circuit global

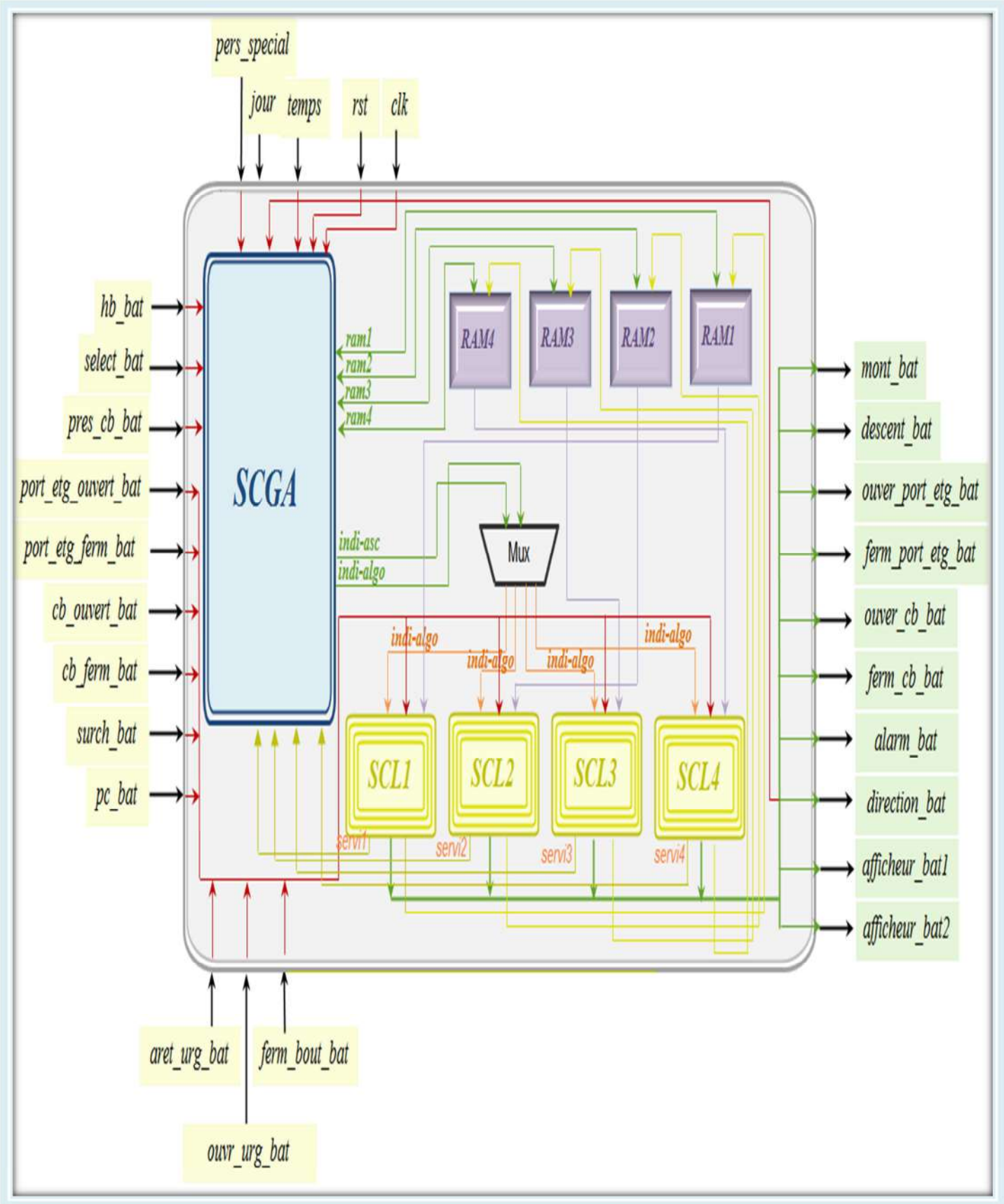
Pour réaliser ce travail, le système de gestion comporte deux modules importants : un serveur qui est le système de contrôle de groupe d'ascenseur **SCGA** et M systèmes de contrôle local **SCL** (un par chaque ascenseur), le système comporte aussi M mémoires internes de type **RAM** pour stocker l'état actuel de chaque ascenseur et les demandes à servir.

- Le **SCGA** permet de choisir l'ascenseur et l'algorithme de dispatching les plus appropriés à la demande (appel haut ou bas/simple ou prioritaire) effectuée suivant des critères prédéfinis , puis il doit sauvegarder la demande dans la RAM d'ascenseur choisi , et donner à son SCL le nouveau algorithme sélectionné .

- Chaque sélection d'étage désiré est sauvegardée directement par le **SCGA** dans la RAM appropriée à l'ascenseur où la demande est effectuée.

- Chaque **SCL** contrôle un ascenseur, il permet de servir toutes les demandes enregistrées au niveau de sa RAM par des commandes de montée ou de descente et la commande d'ouverture et de fermeture des portes, suivant l'algorithme de dispatching choisi par le **SCGA**.

Pour mieux comprendre le principe de fonctionnement du système intelligent de gestion, on prend par exemple un immeuble à 20 étages et 4 ascenseurs, on a besoin donc d'un SCGA avec 4 RAM et 4 SCL comme c'est illustré dans la figure II.4



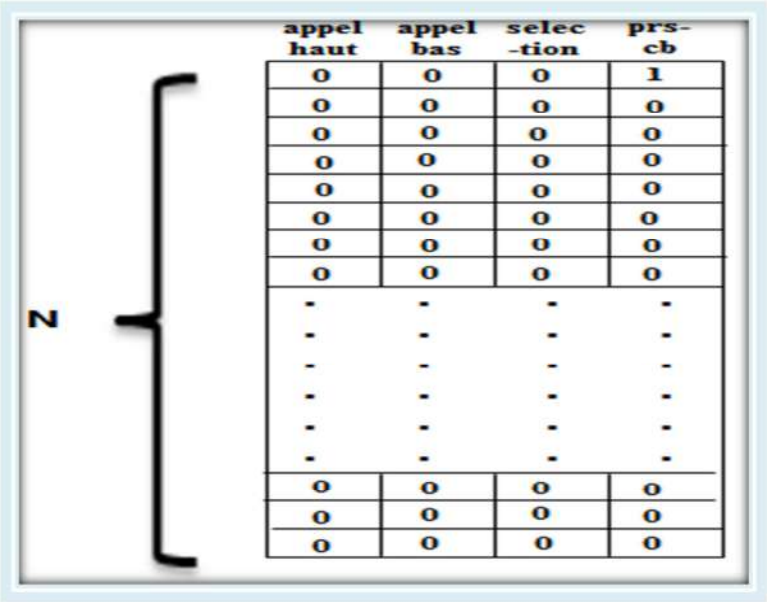
FigII.4 : Schéma synoptique du circuit principal

VI.Les composantes du système global

VI.1 Les mémoires RAM

Chaque ascenseur a sa propre RAM, elle permet de sauvegarder son état ainsi que les demandes à servir. Ce Sont des matrices des capteurs de types ***std_logic*** et de taille **N*4** avec :

- **N :** le nombre des étages
- Les 4 colonnes sont successivement : ***appels haut***, ***appels bas***, ***sélections (les étages désiré)***, ***présence _cabine***. Comme montre la figure II.5



FigII.5 : l'architecture interne de la mémoire RAM

En initialisant toutes les cases de la matrice à '0', car il y'a aucun évènement détecté sauf la présence des cabines des ascenseurs au RDC, ce qui permet de remplir la case (0,4) par '1' dans toutes les RAM. Après ils peuvent être modifié par le SCGA en cas de détection des appels hauts ou bas ou des sélections et à chaque nouvel état de la cabine, ou par les SCL's après chaque service des demandes.

VI.2 Le système de contrôle d'un groupe d'ascenseurs SCGA

VI.2.1 L'utilité du circuit SCGA

C'est le noyau du système de gestion, il est activé à chaque demande effectuée pour choisir l'ascenseur avec l'algorithme de dispatching le plus approprié pour la servir le plus tôt possible. Il

permet aussi de mettre à jour les RAMs par les appels, les sélections et les présences cabines détectées.

De nombreuses méthodes ont été proposées afin de mettre en œuvre des stratégies de contrôle pour le SCGA, afin d'améliorer leur performances dans des situations différentes de trafic .Comme par exemple les SCGA basée sur la logique floue, les réseaux de neurones, les algorithmes génétiques, les chaînes de Markov etc... . [7]

L'utilisation de ces méthodes augmente la performance de fonctionnement des ascenseurs, mais la difficulté se pose sur le passage par la modélisation de ses méthodes avec un langage comme MATLAB de MATHWORKS, puis son implémentation avec un langage machine comme VHDL , qui prennent beaucoup de temps qu'on n'avait pas , malgré qu'on a essayé de le faire, c'est pour laquelle on a pensé à utiliser une architecture simple pour le système SCGA qui assure le bon fonctionnement des ascenseurs avec des bonnes performances, et on a abouti à une architecture qu'on va l'effectuer un zoom dans la partie suivante .

VI.2.2 L'entité SCGA

Pour une réaction juste et rapide, le **SCGA** a besoin de suivre tous les évènements détectés autour du bâtiment (figureII.6):

- ✓ les appels hauts et bas effectués au niveau des étages à partir la matrice ***hb_bat***
- ✓ les sélections à l'intérieur des ascenseurs, et les présences cabines, à partir des deux matrice ***selct_bat*** et ***pres_cb_bat*** successivement
- ✓ le temps (l'heure), et la direction de mouvement de tous les ascenseurs
- ✓ la disponibilité des ascenseurs à partir des deux signaux :
  - ***unbl*** : pour connaitre les ascenseurs allumés
  - ***servi*** : pour connaitre les ascenseurs réservés pour les appels prioritaires
- ✓ l'historique de chaque ascenseur à partir des 4 RAMs
- ✓ le type d'appel (simple ou prioritaire) et bien sur le signal d'horloge ***clk*** et le signal d'initialisation ***rst***

Sa réaction peut être représentée par l'envoi du ***numéro d'ascenseur*** et le ***numéro d'algorithme*** choisis pour l'appel à traiter, (il nous donne son type et son numéro étage où il est effectué), avec la mise à jour des RAMs modifiées

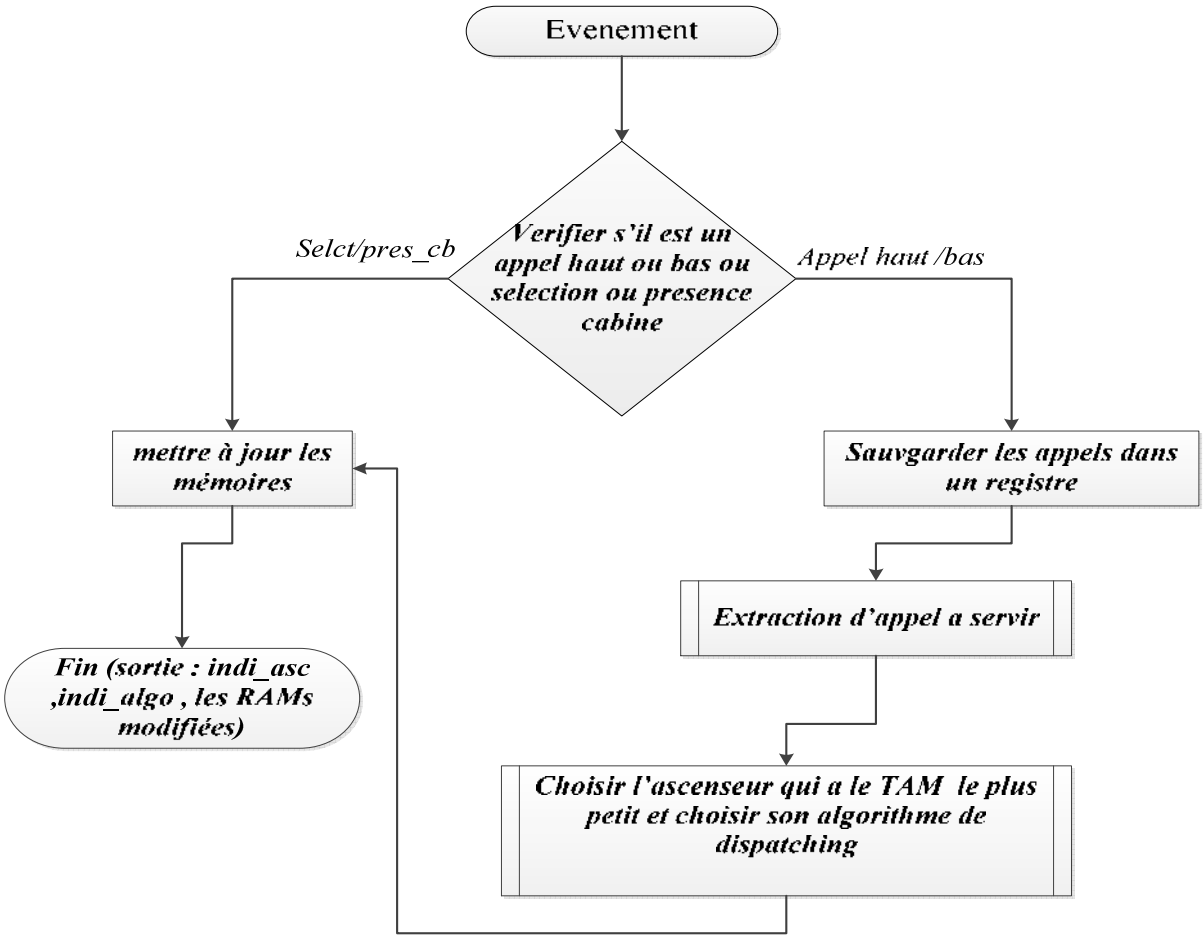
```
entity scga is
      port (  hb_bat : in matrice_HB ;
              selct_bat,pres_cb_bat : in matrice_SP ;
              pers_special,rst,clk : in std_logic ;
              temps : in integer ;
              direct : in std_logic_vector (0 to M-1) ;
              ram1,ram2,ram3,ram4 : in matrice_SP ;
              unabl : in std_logic ;
              servi1,servi2,servi3,servi4 : in std_logic ;

              indi_asc : out integer range 0 to M ;
              indi_algo : out integer range 0 to k ;
              indi_apel : out integer range 0 to N-1;
              typ_apel  : out integer range 0 to 1;
              ram1_out,ram2_out,ram3_out,ram4_out : out matrice_SP) ;
end scga;
```

FigII.6 : l'entité du circuit SCGA

VI.2.3 Principe de fonctionnement du SCGA :

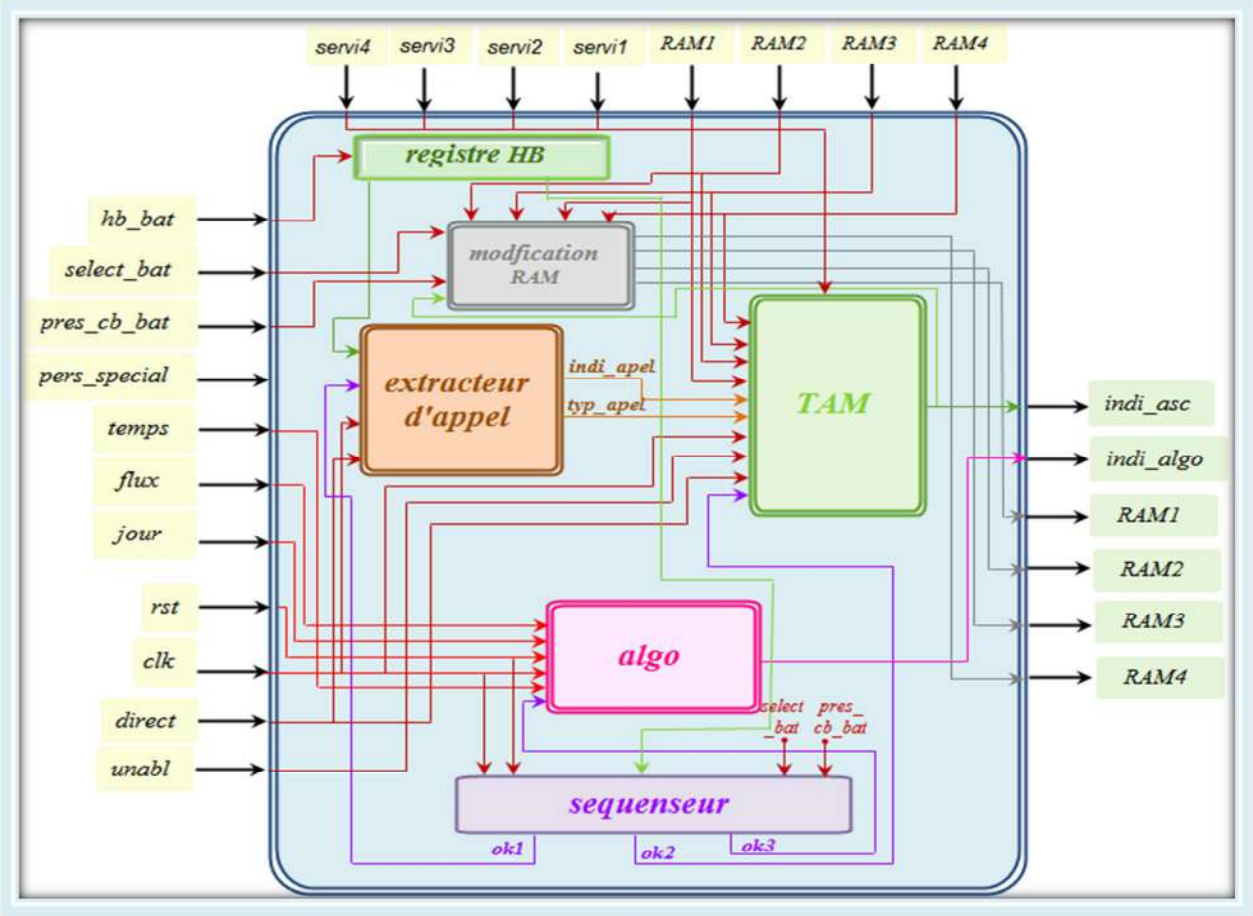
Le principe général de fonctionnement de système SCGA est expliqué par l'organigramme représenté par la figure II.7 :



FigII.7.l'organigramme de fonctionnement du circuit SCGA

VI.2.4 L'architecture du circuit SCGA :

Pour assurer le déroulement des tâches proposées dans l'organigramme précédant notre SCGA est composé de 6 modules importants donnés par la figure II.8 :



FigII.8.Schéma synoptique du circuit SCGA

À partir de la nature d'évènement détecté il fonctionne comme suit :

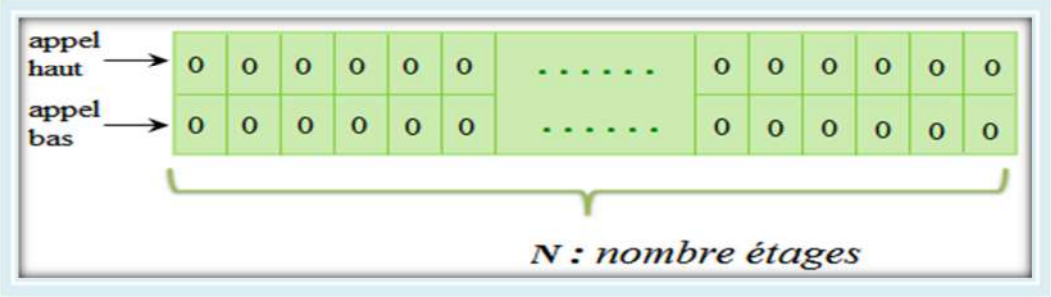
- A la détection d'un appel ,notre **SCGA** le sauvegarde directement dans un registre (registre HB) et à partir **d'un extracteur d'appel** la demande à servir est sélectionnée en envoyant les informations sur son type (haut ou bas) et le numéro d'étage où il est effectué au circuit de choix d'ascenseur(**TAM**) , ce dernier doit nous fournir le numéro d'ascenseur choisi parmi les ascenseurs libres de service de priorité (il ne faut pas perturber un ascenseur réservé pour un appel d'un directeur ,ou une personne handicapé) , et en d'autre part le circuit de choix de l'algorithme de dispatching (**algo**) fonctionne en parallèle afin de nous donner le numéro d'algorithme sélectionné , à la fin notre système fait la mise à jour des RAMs par **un module de modification** qui doit placer l'appel dans la RAM d'ascenseur choisi.

- Si on a une sélection ou un évènement de présence des cabines le système ne fait que la mise à jour des RAM par le module de modification
- Ce déroulement séquentiel des tâches est assuré par le circuit **séquenceur** à partir des signaux de réveils envoyés au circuit approprié à chaque fois.

VI.2.5 Les composantes du système SCGA

VI.2.5.1 Registre _HB :

L'arrivée de plusieurs appels haut ou bas en même temps ne doit pas perturber le système de contrôle qui traite une autre demande, et de même il faut les sauvegarder pour ne pas les perdre afin de les traiter ultérieurement une après l'autre, c'est la raison pour laquelle on a pensé à un regroupement des demandes en tant que registre, c'est le **registre_HB** crée dans ce but (figure II.9) :



FigII.9.architecture interne du registre_HB

C'est une simple matrice de type ***matrice_HB*** composée de deux lignes : une pour les appels haut et l'autre pour les appels bas, à l'initialisation toutes les cases sont à '0 ', à la détection d'un nouveau appel la case correspondante (typ_appel, indi_appel) remplie par '1' telle que : ***typ_appel*** :'0'= haut / '1'= bas, ***indi_appel*** : numéro d'étage ou l'appel est effectué

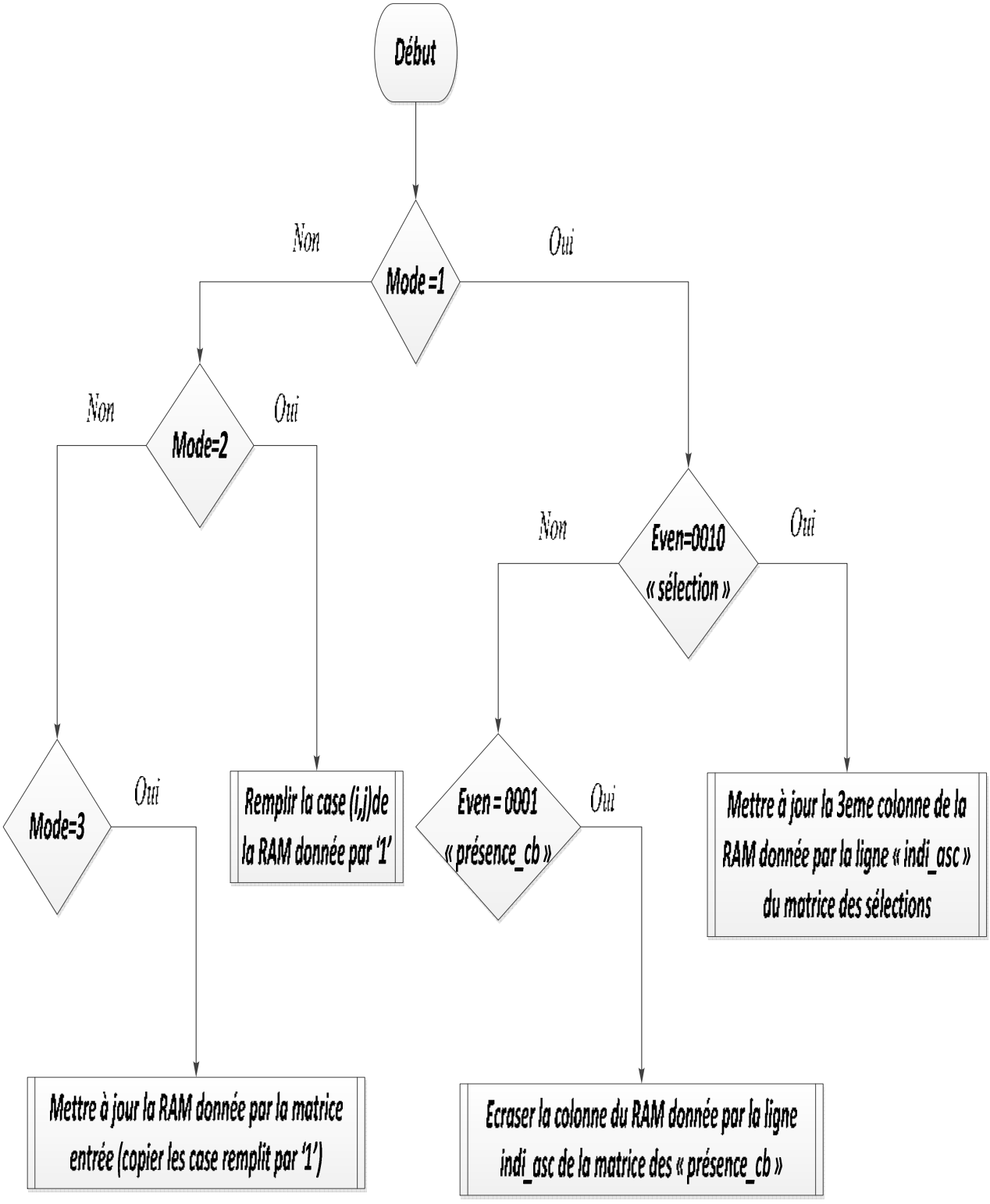
VI.2.5.2 Module de modification des RAMs :

La RAM représente l'état actuel de chaque ascenseur, elle nécessite une mise à jour en temps réel pour le fonctionnement juste du système de contrôle, pour cette raison on a créé la procédure **modification_ram**, qui est une simple procédure à 3 mode suivant les entrées reçues:

- Modification d'une colonne ***(mode 1)***
- Modification d'une case ***(mode 2)***
- Modification d'une matrice ***(mode 3)***

VI.2.5.2.1 Principe de fonctionnement du module de modifications :

Le principe de fonctionnement de la procédure **modification_ram** est montré par l'organigramme de la figure II.10:



FigII.10 : organigramme de fonctionnement du module de modification des RAMs

VI.2.5.2.2 L'entête du module de modification :

La figure II.13 donne l'entête du module de modification

```
procedure modification_ram ( variable ram_asc : in matrice_sp ;
                             signal indi_asc : in integer range 0 to M ;
                             variable mode : in integer range 0 to M-1 ;
                             signal i : in integer range 0 to 1 ;
                             signal j : in integer range 0 to N-1 ;
                             variable even : in std_logic_vector (0 to M-1) ;
                             signal data_in : in matrice_SP;
                             variable data_out : out matrice_SP ) ;
```

FigII.11.Entête du module de modification

- ***ram_asc*** : de type matrice_sp, c'est l'historique d'ascenseur qui nous allons le mettre à jour
- ***indi_asc*** : numéro d'ascenseur => entier [0, M-1]
- ***mode*** : de type entier, on a 3 modes :
  - ***mode=1*** : modification d'une colonne de la matrice donnée
  - ***mode=2*** : modification d'une case de la matrice donnée
  - ***mode=3*** : modification complète de la matrice donnée
- ***i, j :*** les indices de la case à modifier si on est dans le mode 2
- ***even*** : un variable qui nous indique le type des données entrées '**data_in**' : « 0010 » pour la matrice des sélections, « 0001 » pour la matrice des présences des cabines.
- ***data_in*** : matrice des nouvelles données

VI.2.5.2.3 L'utilisation du module de modification :

Comme illustre le schéma synoptique du SCGA montré dans la figure II.10, le module de modification est utilisé dans deux cas :

- ***Evènement au niveau de sélections/présences cabines :***

  A la détection des nouvelles sélections ou des nouveaux états des cabines les RAMs des ascenseurs concernées doivent être mise à jour par l'appel de la procédure ***modification-ram*** donnant : l'historique du mémoire d'ascenseur (ram), le num_asc,mode=1, data_in = la matrice des sélections/ la matrice des présences cabines, le SCGA appelle la procédure quatre fois pour les quatre ascenseurs.

- ***ascenseur choisi par le circuit TAM :***

  A la sélection d'ascenseur pour la demande traité par le module TAM, l'appel doit être sauvegardé dans la RAM d'ascenseur choisi pour être servi, ce stockage est effectué par l'appel de la procédure ***modification-ram*** donnent seulement: indi_asc (sortie du TAM), mode =2, i et j (i : appel haut/bas, j : numéro d'étage où la demande est effectué).

Remarque : ce module de modification est aussi utilisé en troisième mode au niveau du SCL, plus de détails vont être données dans les parties suivantes

VI.2.5.3 Le séquenceur :

VI.2.5.3.1 L'entité séquenceur

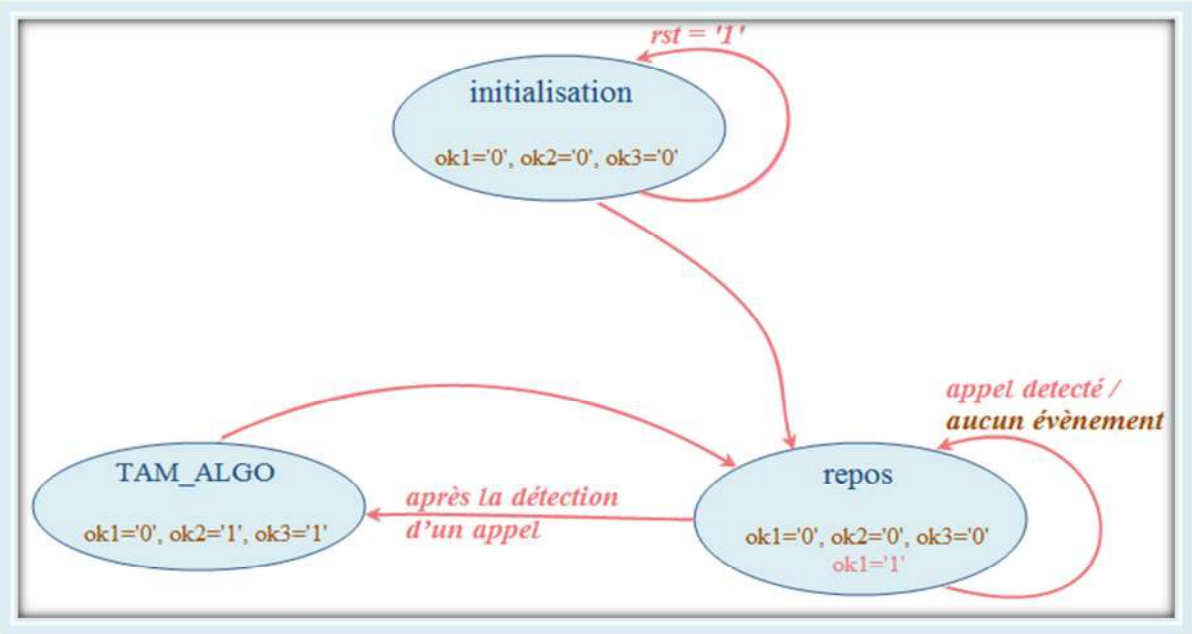
c'est un circuit important pour le bon fonctionnement du système , il gère la collaboration entre les différents module de SCGA , par 3 signaux des réveils (un par modules), il a besoin que du registre d'appel, du signal d'horloge , et du signal de réinitialisation « rst » , il donne en sortie les valeurs des trois signaux de réveils : ok1,ok2,ok3 pour les trois circuit : l'extracteur d'appel , TAM, ALGO . L'entité peut être aussi donnée par la figure II.12 :

```
entity sequenceur is
        port ( clk , rst :  in std_logic ;
                reg_HB :  in matrice_HB;
               ok1,ok2,ok3 : out std_logic );
end sequenceur ;
```

FigII.12 : Entité du séquenceur

VI.2.5.3.2 Le principe de fonctionnement du séquenceur :

Son principe de fonctionnement est illustré par la machine d'état représenté par la figure II.13



FigII.13.Machine d'état du séquenceur

En initialisant notre système de gestion (alimentation ou initialisation avec un bouton poussoir ***rst***) , le séquenceur met les trois signaux de réveils en ' 0' (aucun module fonctionne) , il reste comme ça dans l'état repos jusqu'à la détection d'un appel enregistré dans le registre d'appel ***reg_hb*** , dans le même cycle il réveille l'extracteur d'appel par ***ok 1='1'*** , et dans le cycle suivant quand ce dernier nous a fourni le type d'appel et son indice le séquenceur passe à l'état TAM_AlGO pour réveiller le module ***TAM*** et ***ALGO*** en parallèle afin de calculer l'indice ascenseur et indi _algo et il passe à l'état repos en attendant un autre appel détecté.

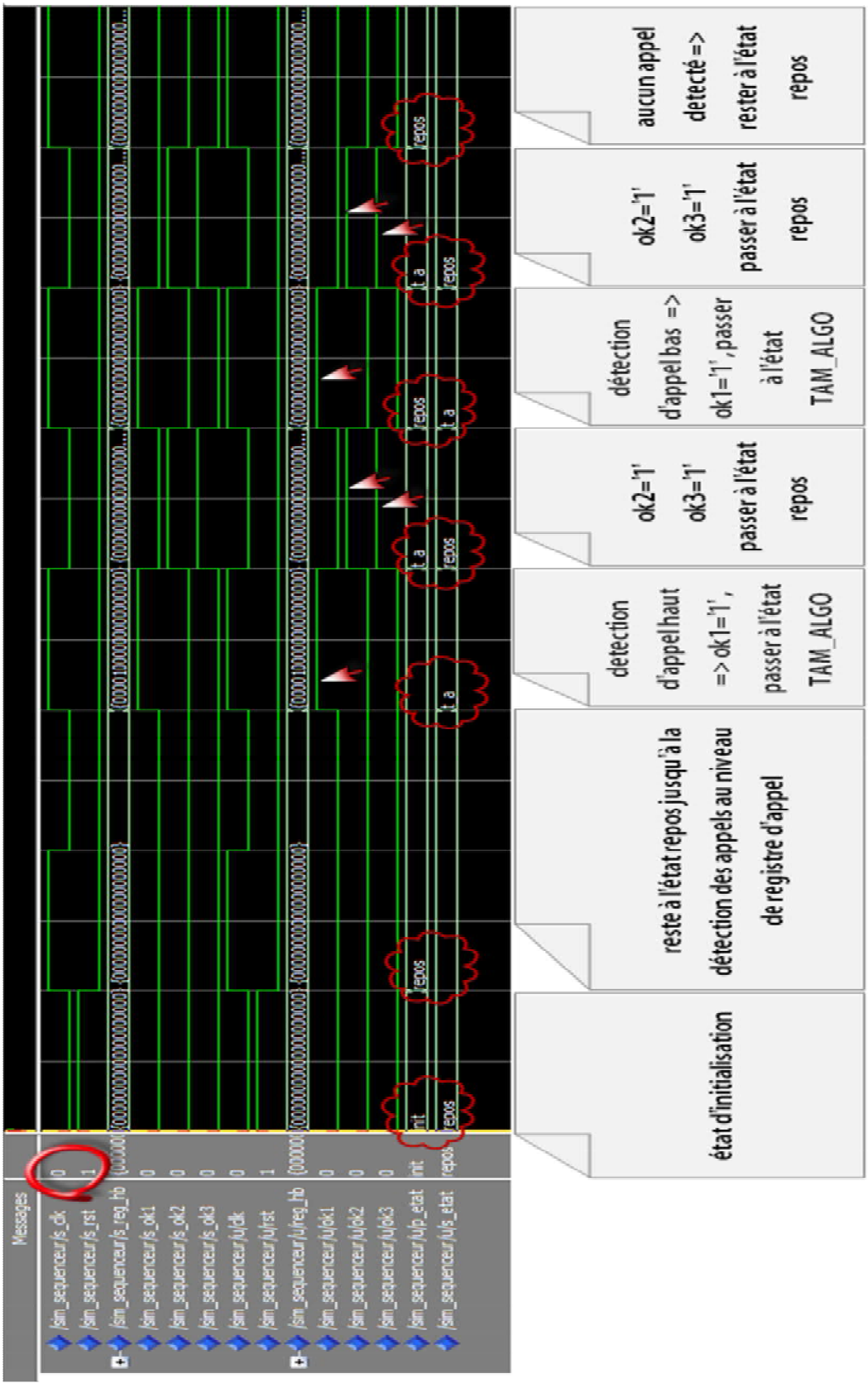
VI.2.5.3.3 Exemple de simulation :

Pour montrer le bon fonctionnement du séquenceur, on utilise par exemple un registre avec un appel haut au 4éme étage et appel bas au 17éme étage (figure II.16)

0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

FigII.14 : Exemple d'un registre_HB

La figure II.15 est le résultat de simulation de séquenceur :



FigII.15 : Résultat de simulation du séquenceur

VI.2.5.4 L'extracteur d'appel :

C'est le module qui assure le choix d'appel qui convient pour optimiser le nombre de déplacement des cabines des ascenseurs, il joue un rôle important dans la gestion intelligente de notre system de gestion, c'est lui qui choisit l'appel à servir à chaque fois, il est réveillé à chaque détection des appels au niveau de ***registre d'appel.***

VI.2.5.4.1 L'entité extracteur appel :

Pour que l'extracteur d'appel fonctionne, il a besoin d'un signal de réveil envoyé par le séquenceur, avec le registre d'appel et le tableau des direction de toutes les ascenseur , et bien sur le signal d'horloge , en sortie il fournit les information sur l'appel sélectionné pour le servir (le type d'appel ainsi que le numéro d'étage où il est effectué) (figure II.16) .

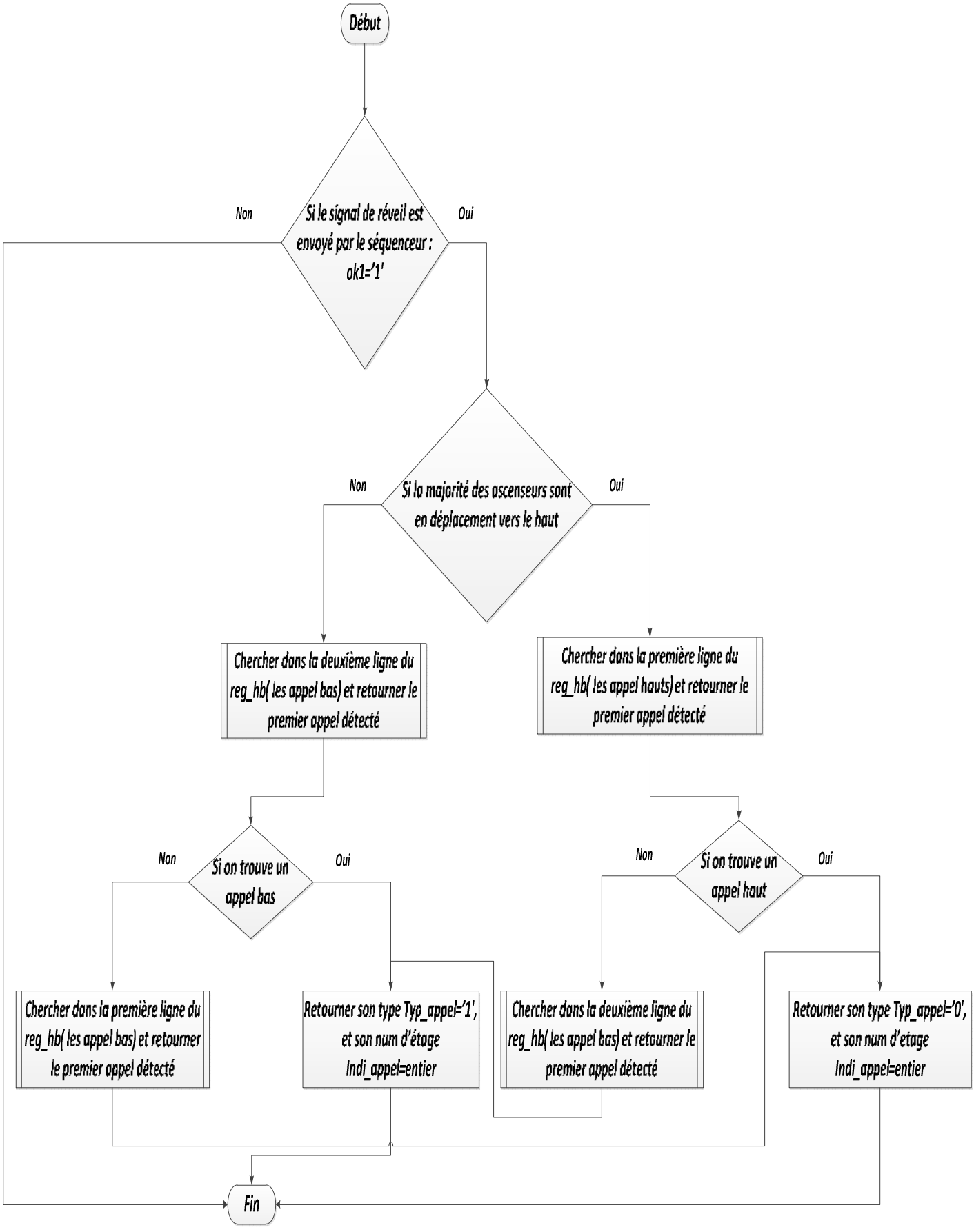
```
entity extracteur_appel  is

    port ( ok2 : in std_logic;
           reg_hb : in matrice_HB;
           direct : in std_logic_vector (0 to M-1);
           clk : in std_logic;
           indi_apel : out integer range 0 to N-1;
           typ_apel : out integer range 0 to 1);
end extracteur_appel;
```

FigII.16.Entité d'extracteur d'appel

VI.2.5.4.2 Principe de fonctionnement d'extracteur d'appel

L'organigramme de la figure II.17 décrit le principe de fonctionnement de ce module :



FigII.17 : Organigramme de fonctionnement d'extracteur d'appel

VI.2.5.4.3 Exemple de simulation :

Le meilleur exemple pour tester le fonctionnement d'extracteur_appel est de le donner un registre avec 2 appels : haut et bas (on utilise le même registre utilisé précédemment dans l'exemple de séquenceur),et en suppose que la majorité des ascenseurs sont en déplacement vers le bas (figure II.18):

0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

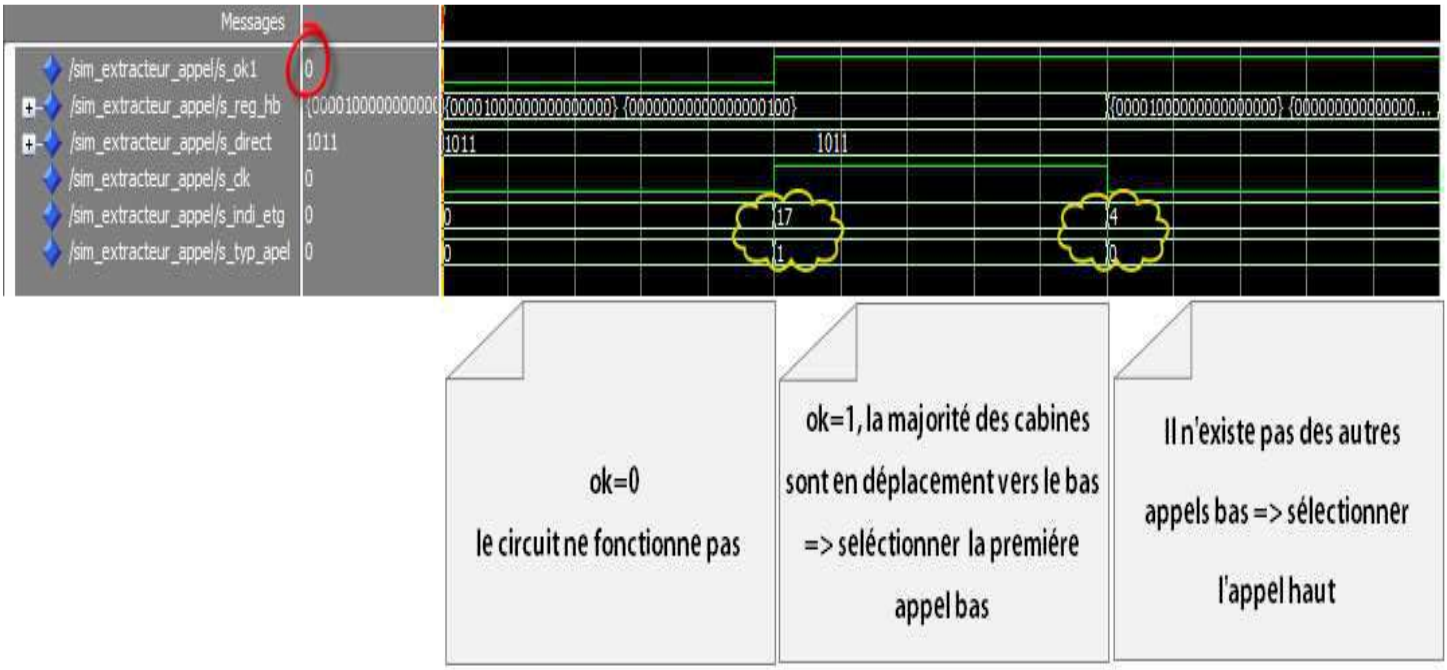
FigII.18. Exemple d'un registre_HB

1	0	1	1

'0':déplacement vers le haut
'1':déplacement vers le bas

FigII.19 : Exemple d'un tableau de direction des ascenseurs

Le résultat de simulation donné par la figure II.20 :



FigII.20.Résultat de simulation du Modelsim

VI.2.5.5 Le circuit de choix d'ascenseur « TAM »

Ce module représente avec le circuit **ALGO** le cœur de system SCGA, il s'occupe de choisir l'ascenseur le plus approprié pour servir la demande détecté, afin d'optimiser le temps d'attente moyen qui est son critère de choix, ***«il fait le calcul du TAM de chaque ascenseur disponible et celui qui correspondant au minimum est sélectionné ».***

VI.2.5.5.1 L'entité du circuit TAM

Pour que ce circuit peut fonctionner il a besoin d'un signal de réveil venus de séquenceur (***ok2***), et les quarte ***RAMs*** des ascenseurs, avec des informations sur l'appel à servir (son ***type*** et son ***numéro*** d'étage où il est effectué) fournies par l'extracteur d'appel, n'oublions pas le vecteur de ***direction*** de mouvement de tous les ascenseurs, avec un signal assuré leur alimentation (***unabl***) et bien sur le signal d'***horloge***.Une autre entrée « ***pers_special*** » très importante pour distinguer les demandes spéciales ,lors de la détection de ce signal, les TAM 's sont calculés d'une autre maniéré pour servir le plus rapidement possible les demande prioritaires, même dans les autres cas simples , il a besoin de connaitre les ascenseurs qui sont réservés pour servir les appels spéciaux(utilité du entrées ***servi1,servi2,servi3,servi4 : servi i='1' => ascenseur i réservé)*** pour ne pas les perturber, ils ne sont pas pris en compte .A la fin, ce module fourni le numéro d'ascenseur correspondant au TAM minimal, c'est le plus approprié pour servir l'appel détecté (figure II.21) .

```
entity TAM  is
  port ( ok2 : in std_logic;
         ram1,ram2 ,ram3,ram4 : in matrice_sp ;
         indi_apel : in integer range 0 to N-1 ;
         typ_apel : in integer range 0 to 1;
         direct : in std_logic_vector (0 to M-1) ;
         clk : in std_logic ;
         unabl : in std_logic ;
         pers_special : in std_logic ;
         servi1,servi2,servi3,servi4 : in std_logic;
         indi_asc : out integer range 0 to M ) ;
end TAM;
```

FigII.21. Entité du circuit TAM

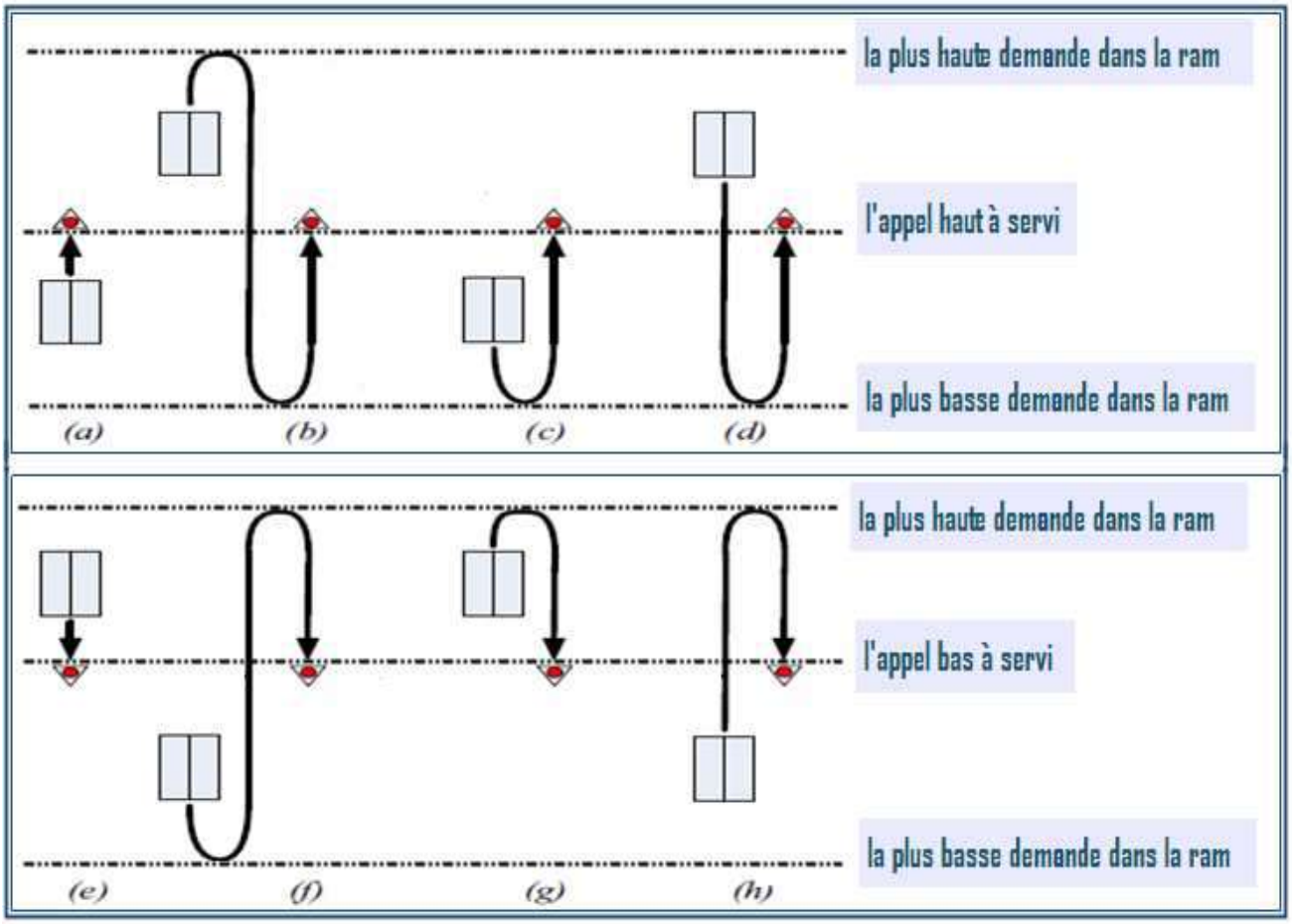
VI.2.5.5.2 C'est quoi le TAM ?

C'est la durée moyenne, en secondes, qu'un passager passe en attente pour un ascenseur, mesurée entre l'instant que le système de contrôle détecte un appel jusqu'à l'instant où le passager peut entrer dans la cabine. Il est calculé par la formule suivante:

$$TAM = nbr_{stop} \times (T_{ouv} + T_{ferm} + T_{att} + T_{acc} + T_{des}) + dist(app,cb) \times T_{etg}$$

Où :

- ***nbr**stop* : il regroupe tous les appels (haut et bas) et les sélections existant entre la position de la cabine d'ascenseur (qui on va calculer son TAM) et la position de l'appel à servir, bien sûr qui sont sauvegardés dans sa ram, ça c'est on a des simples demandes , mais il n'est que le nombre des sélections enregistré en cas de détection du signal « pers_special » (dans les demandes prioritaires les autres appel sont ignoré ,on ne prend pas en compte que les sélection pour vider la cabine afin de servir l'appel spécial)
- T_{ouv}: Temps d'ouverture des portes.
- T_{ferm} : Temps de fermeture des portes.
- T_{att} : Temps d'attente de la cabine dans l'étage où il est arrêté
- T_{acc} : Temps d'accélération de la cabine lors de son démarrage.
- T_{dec} : Temps de décélération de la cabine lors de son arrivé.
- T_{etg} : Temps de passage de l'ascenseur d'un étage à l'autre.
- ***dist (app,cb)***: c'est le nombre de tous les étages visités par la cabine afin d'arriver à l'appel à servir, passant par toutes les demandes (appels hauts, bas et sélections) trouvées dans son chemin , cette distance est calculé à travers une petite comparaison entre la position de la cabine d'ascenseur **(P_{cab})** et sa direction de déplacement (vers le haut ou vers le bas) , avec la position (**(P_{app})**)et le type d'appel à servir , cette comparaison est expliqué clairement par la figure II.21 :



FigII.22.Les cas de situation d'ascenseur par rapport à l'appel à servir

À travers la position d'ascenseur et la position d'appel à servir on distingue 4 cas pour chaque type d'appel (les cas « a b c d » pour un appel haut / « e f g h » pour un appel bas), dans ce qui suit on va expliquer chaque cas en donnant l'expression de calcul de la distance ***dist (app,cb) :***

- ***le cas « a » :***

  L'appel haut à servir est au-dessus d'ascenseur qui est en déplacement vers le haut, et aucune demande reste en attente, alors la distance n'est que le nombre des étages entre les deux positions (la position de la cabine et le numéro d'étage où l'appel est effectué)

$$dist(app, cb) = P_{app} _ P_{asc}$$

- ***le cas « b » :***

  L'appel haut à servir est au-dessous d'ascenseur qui est en déplacement vers le haut pour servir des demandes en attente, mais il y a aussi des demandes en attente dans le sens opposé , alors que la distance est : le nombre des étages entre la position de la cabine et la

position du plus haute demande en attente (P_{hig}) , en ajoutant le nombre des étage entre cette dernière et la plus basse demande (P_{low}) , avec le nombre des étages entre cette dernière et la position d'appel à servir

$$dist(app, cb) = (P_{hig} - P_{asc}) + (P_{hig} - P_{low}) + (P_{app} - P_{low})$$

- ***le cas « c , d » :***

deux cas similaires, l'appel haut à servir situé au sens opposé du direction de déplacement d'ascenseur sauf que dans le premier cas « c » elle est au-dessus et le deuxième cas « d »elle est au-dessous de la cabine , et il ne reste que les demandes dans sa direction , alors la distance est tout simplement le nombre des étages entre la position de la cabine et la position d'appel plus bas (P_{low}) plus le nombre des étages entre ce dernier et la position d'appel à servir .

$$dist(app, cb) = (P_{asc} - P_{low}) + (P_{app} - P_{low})$$

- ***le cas « e , f , g , h » :***

Sont successivement les mêmes cas que les quatre précédents « a, b, c, d » mai avec un appel bas à servir, les expressions de calcul du distance sont alors :

  - ***« e » :***

$$dist(app, cb) = (P_{asc} - P_{app})$$

  - ***« f » :***

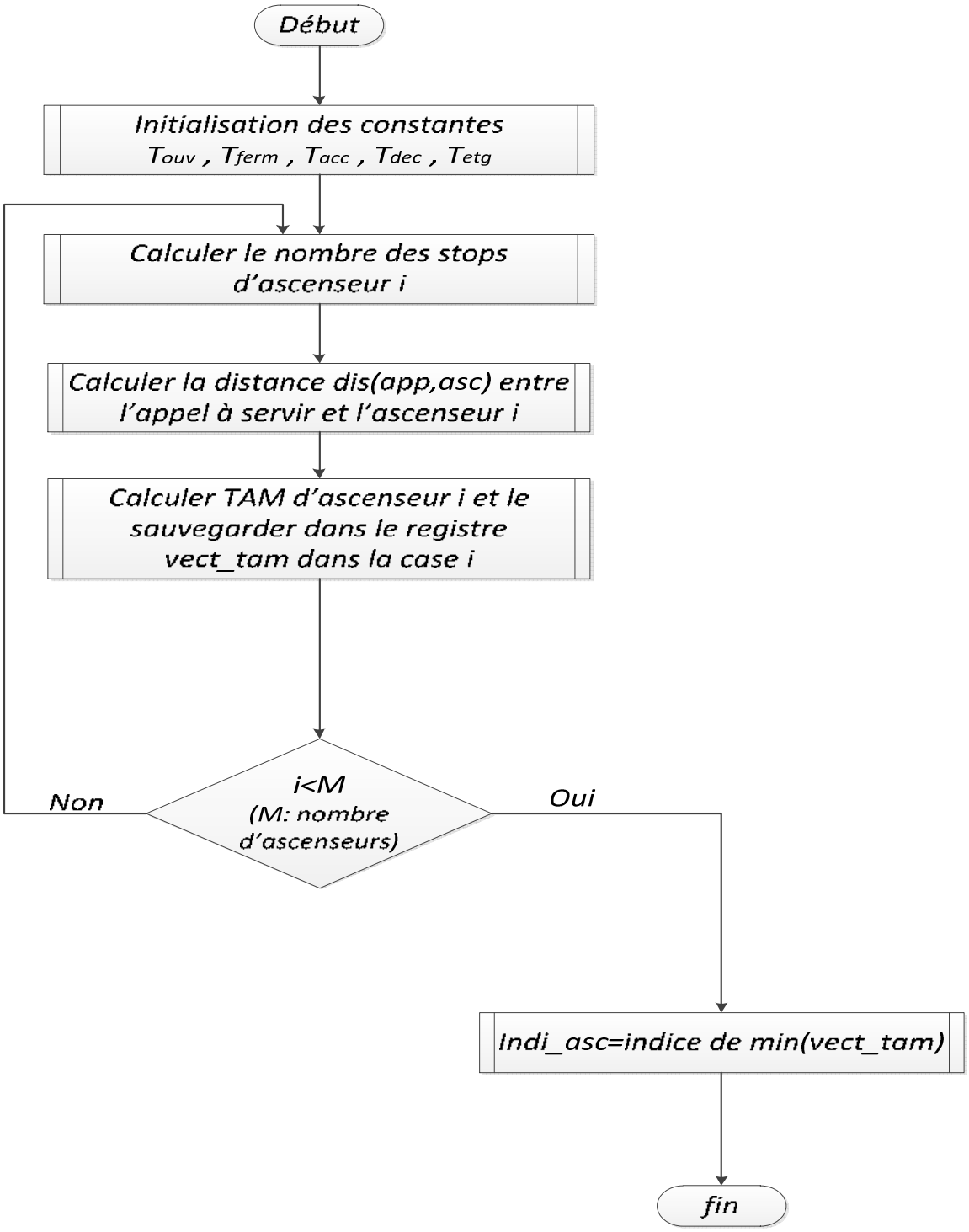
$$dist(app, cb) = (P_{asc} - P_{low}) + (P_{hig} - P_{low}) + (P_{hig} - P_{app})$$

  - ***« g , h » :***

$$dist(app, cb) = (P_{hig} - P_{asc}) + (P_{hig} - P_{app})$$

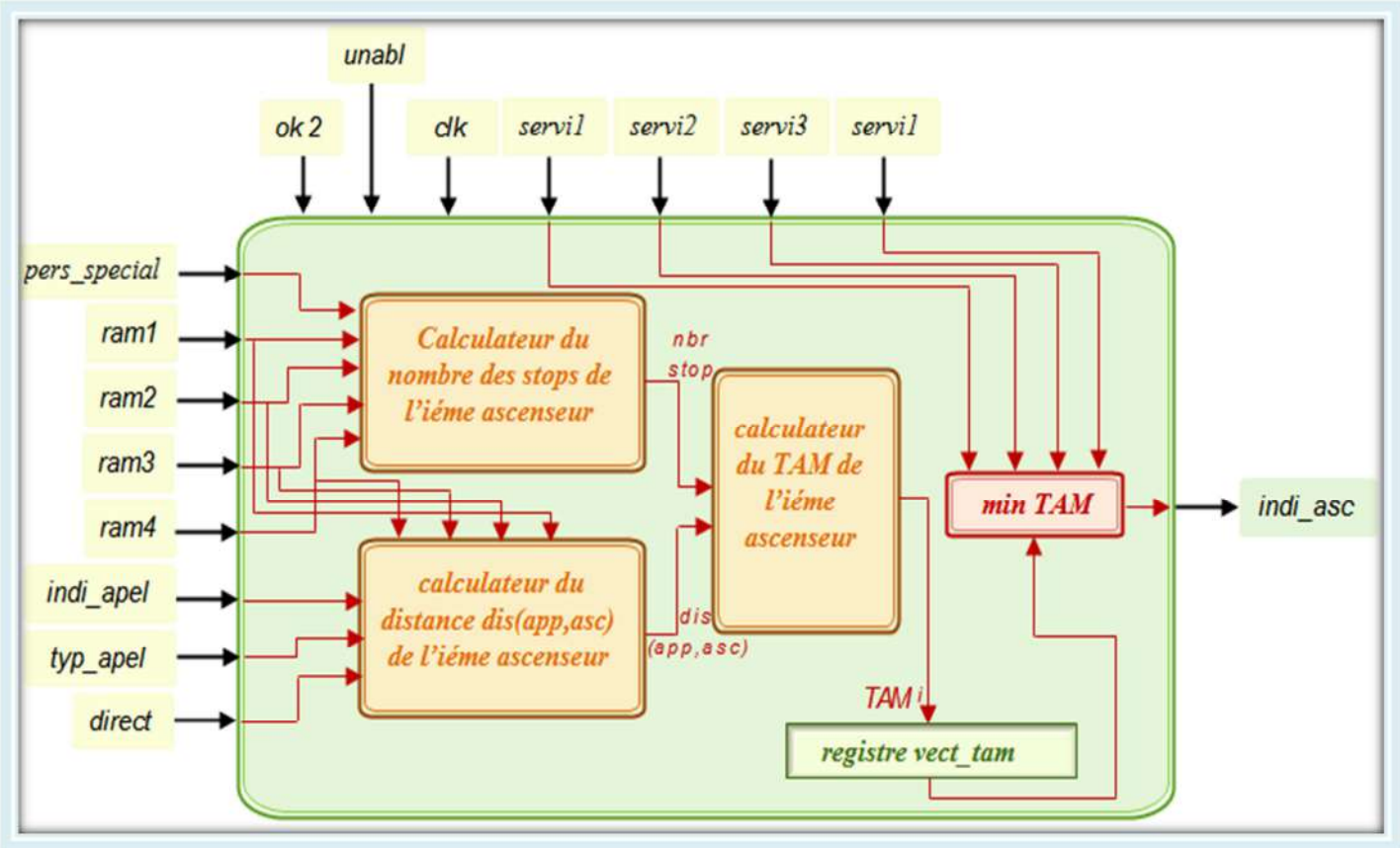
VI.2.5.5.3 L'organigramme de fonctionnement

A partir de ce qui est expliqué précédemment ce module se charge de calculer le TAM de chaque ascenseur afin de choisir le minimum parmi eux, passant par les étapes montrées par l'organigramme de la figure II.22 :



FigII.23. Organigramme de fonctionnement du circuit TAM

L'architecture illustrée par la figure II.23 représente les différents blocs du circuit TAM



FigII.24.Architecture interne du circuit TAM

VI.2.5.5.4 Exemple de simulation

Afin de comprendre le fonctionnement de ce circuit on va proposer un simple scénario et voir le résultat de simulation du circuit TAM, mais avant, il faut connaitre que les constats sont initialisés au début du programme par des valeurs réel :

- ✓ $T_{ouv} = T_{ferm} = 2$ s
- ✓ $T_{att} = 6$ s
- ✓ $T_{acc} = T_{dec} = 1$ s
- ✓ $T_{etg} = 6$ s

Scénario :

1- On suppose qu'on a 4 ascenseurs en fonctionnement dans un immeuble de 20 étages, telle que les situations des ascenseurs sont :
   - ***Ascenseur 1*** : est au RDC et en déplacement vers le haut, « appel haut au 3ème étage, sélection au 1ier et 6ème étages » sont en attente
   - ***Ascenseur 2*** : est au 6ème étage et en déplacement vers le haut, « appel haut au 8ème étage, appel bas au 2ème étage, sélection au 9ème étage » sont en attente
   - ***Ascenseur 3*** : est au 3ème étage et en déplacement vers le bas, « sélection au RDC » est en attente
   - ***Ascenseur 4*** : est au 13ème étage et en déplacement vers le bas, « appel bas au 1ier étage, sélection au 9ème et 12ème étage » sont en attente

2- Le circuit TAM est désactivé jusqu'à l'arrivée du signal de réveil (ok2) après la détection d'un appel haut au 6ème étage.
3- Il est réveillé une autre fois par un appel bas au 8ème étage.
4- Et finalement, il est réveil par un appel haut spécial d'une personne handicapée au 2ème étage.

La figure II.24 est le résultat de simulation : à droite c'est le résultat de simulation donné par Modelism et à gauche son interprétation par un schéma d'explication.

VI.2.5.6 Le circuit de choix d'algorithme de dispatching « ALGO »

Le choix du meilleur ascenseur pour servir un appel joue un rôle important dans l'optimisation du TAM, mais l'arrivée le plus rapidement possible à l'étage ou l'appel est effectué augmente cette optimisation ,n'oublions pas l'importance de diminuer le plus possible le temps d'amener les passagers aux étages désirés et son effet sur l'augmentation du performance du système de gestion , ces optimisations sont assurés par le bon choix de l'algorithme de dispatching pour servir cet appel , c'est exactement l'utilité de la construction d'un circuit de choix d'algorithme de dispatching dans notre system de gestion .

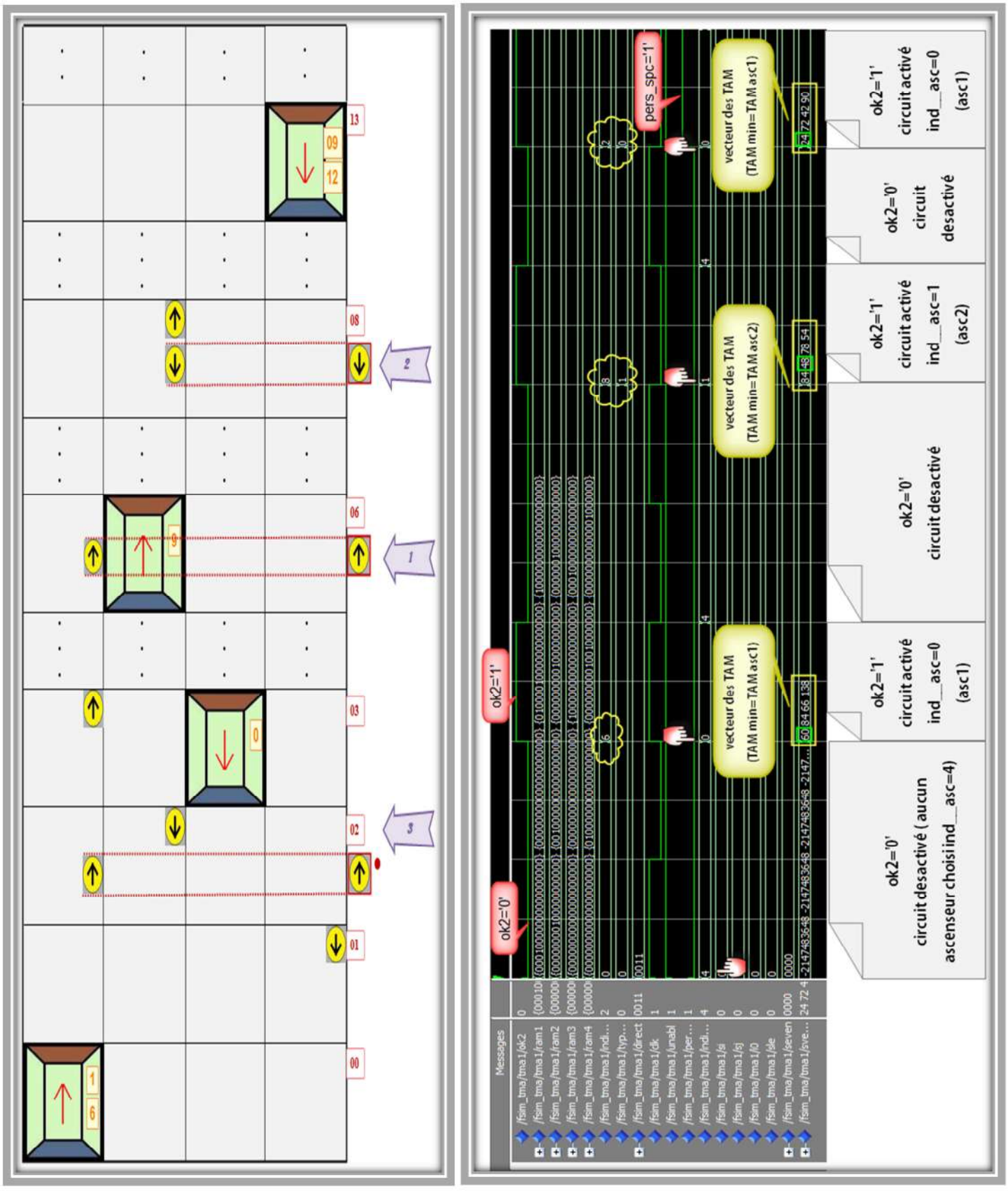
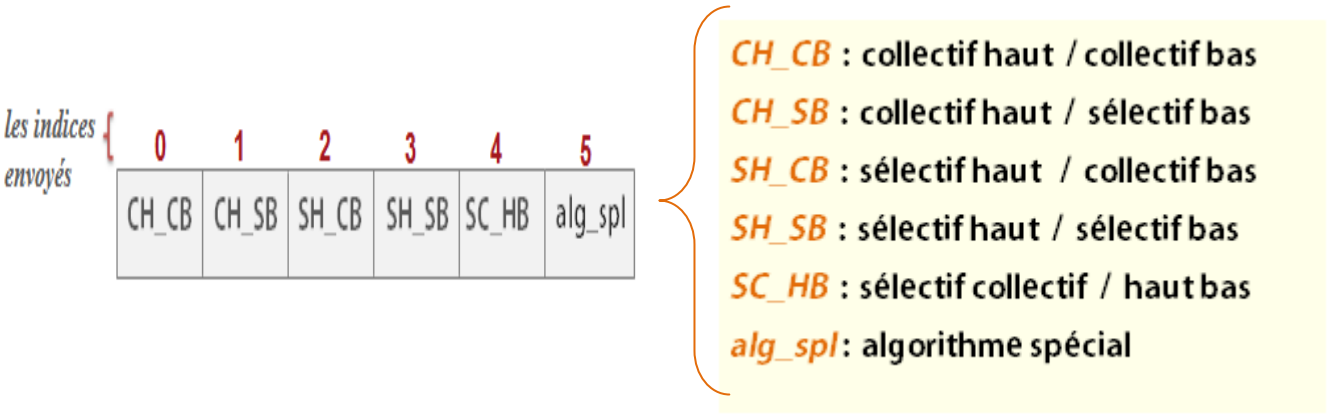


Fig II.24 : Résultat de simulation du scénario proposé « TAM »

VI.2.5.6.1 L'entité du circuit ALGO

C'est un circuit réveillé à chaque arrivé du signal ***ok3*** à partir du séquenceur, il se base dans son choix sur 3 critères importants : le ***temps***, le ***type*** d'appel détecté, la nature de la demande si elle est simple ou spéciale (personne normale, responsable ou personne handicapé :c'est une information fournis par le signal « ***pers_special*** » figureII.25), en sortie il nous donne ***l'indice d'algorithme*** choisi parmi les six algorithmes de dispatching comme illustre le tableau suivant :



FigII.26. Le tableau des algorithmes de dispatching

La figure II.26 donne l'entité du circuit ALGO:

```
entity algo is
   port ( ok3 : in std_logic ;
          temps : in integer ;
          pers_special,clk : in std_logic;
          typ_apel : in integer range 0 to 1;
          indi_algo: out integer range 0 to k);
end algo;
```

FigII.27 .Entité du circuit ALGO

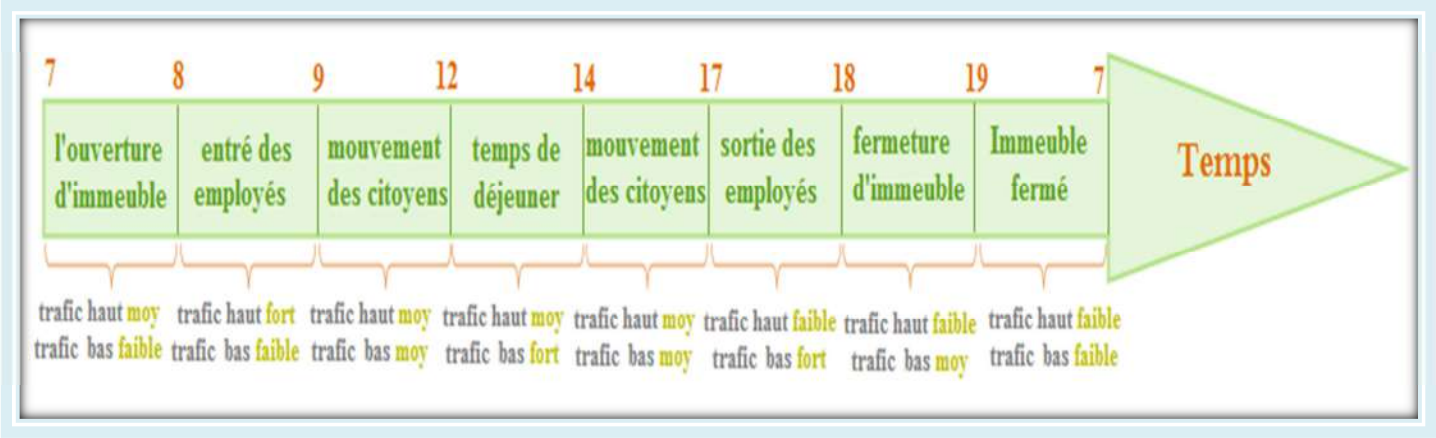
VI.2.5.6.2 L'arbre de décision :

Nous avons imaginé une simple stratégie pour choisir l'algorithme de dispatching basant sur les trois entrées : type d'appel, temps et pers_special :

1- A l'arrivée d'une demande spéciale l'algorithme « algo_special » d'indice '5' est choisi directement, car c'est le seul qui traite les demandes spéciaux parmi les six algorithmes de dispatching

✓ Dans les autres cas (les simples demandes) le choix d'algorithme se base sur les deux critères restant : le temps et le type, telle que :

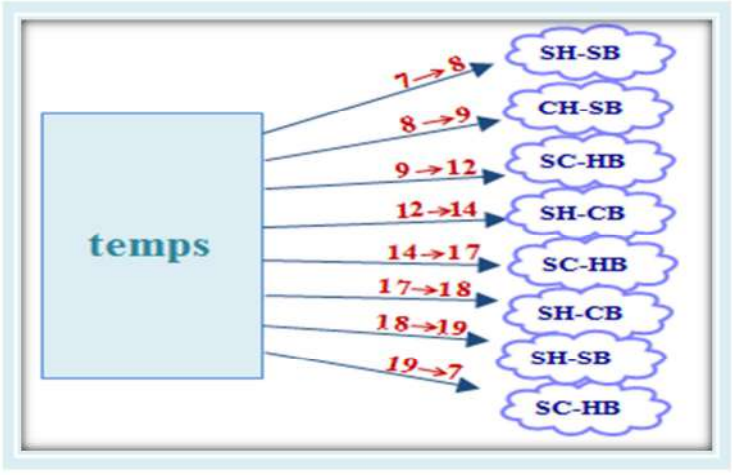
- ***Le type*** : 0 =>un appel haut , 1 => appel bas
- ***Le temps*** : on a une simple horloge indiquant le temps, selon elle et à partir le trafic générale remarqué habituellement dans l'immeuble, on peut découper la journée comme illustre la figure II.29:



FigII.28.Le découpage de la journée suivant le temps et le trafic

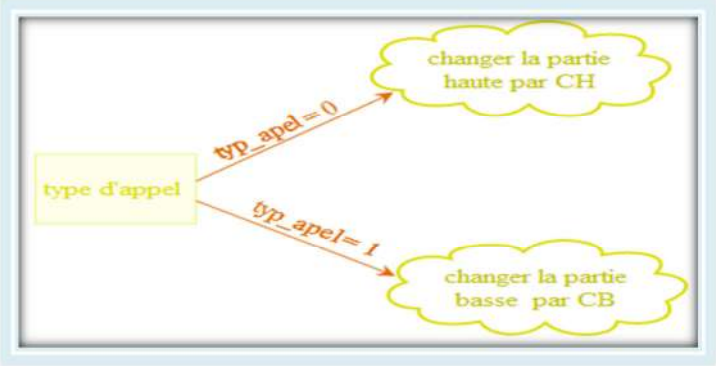
✓ Dans ce cas notre stratégie de choix est composée de deux étapes :

2- Le choix général suivant la situation du trafic indiqué par le temps, comme illustre la figure II.28 :



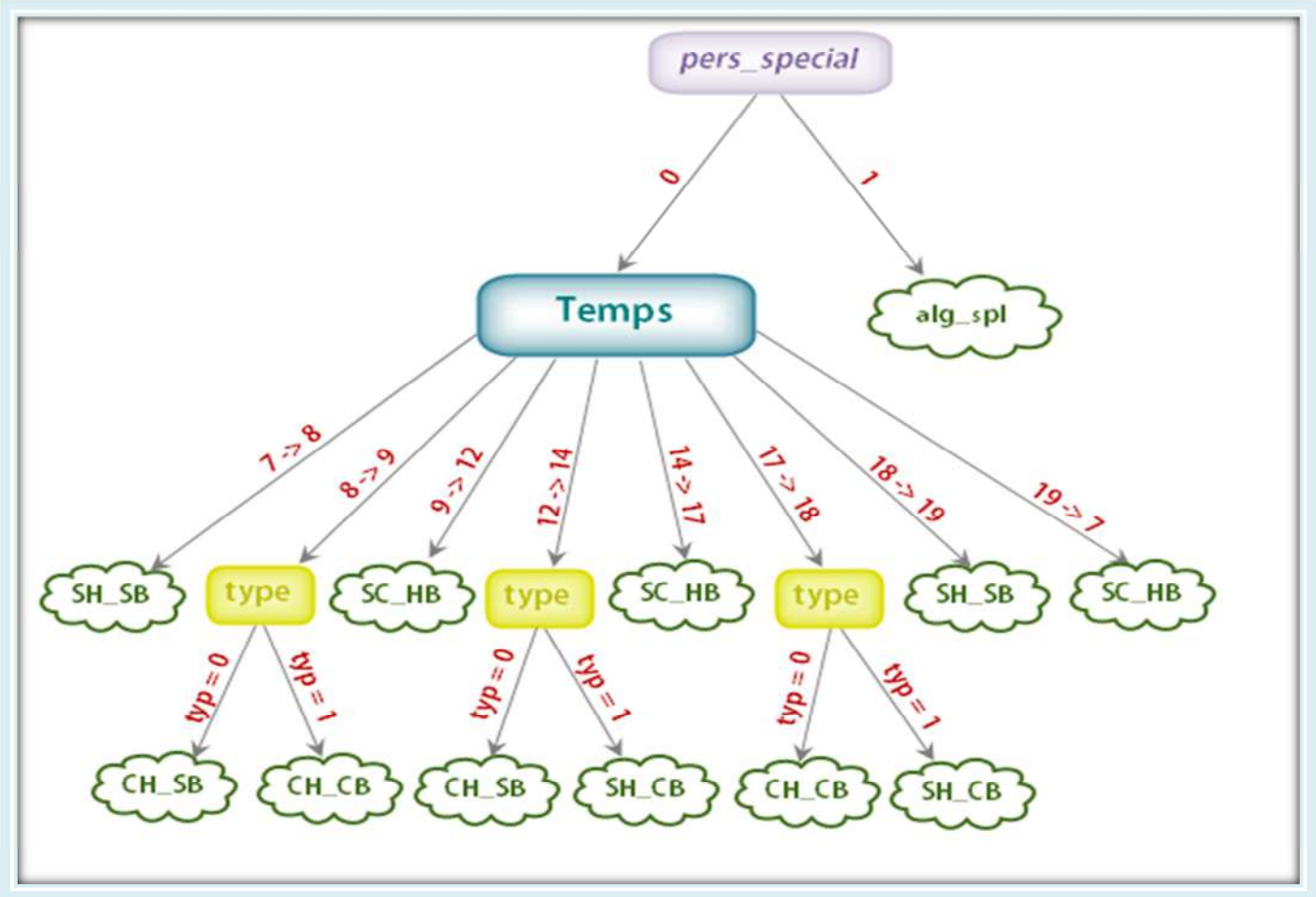
FigII.29. L'arbre de décision suivant le temps

3- La précision du choix suivant le type d'appel détecté dans des certains moments :



FigII.30. L'arbre de décision suivant le type d'appel

Basant sur les critères vus dans les point 1 ,2 ,3 ; notre stratégie finale de choix d'algorithme de dispatching à chaque détection d'un appel est illustrée par l'arbre de décision représenté par la figure II.30 :



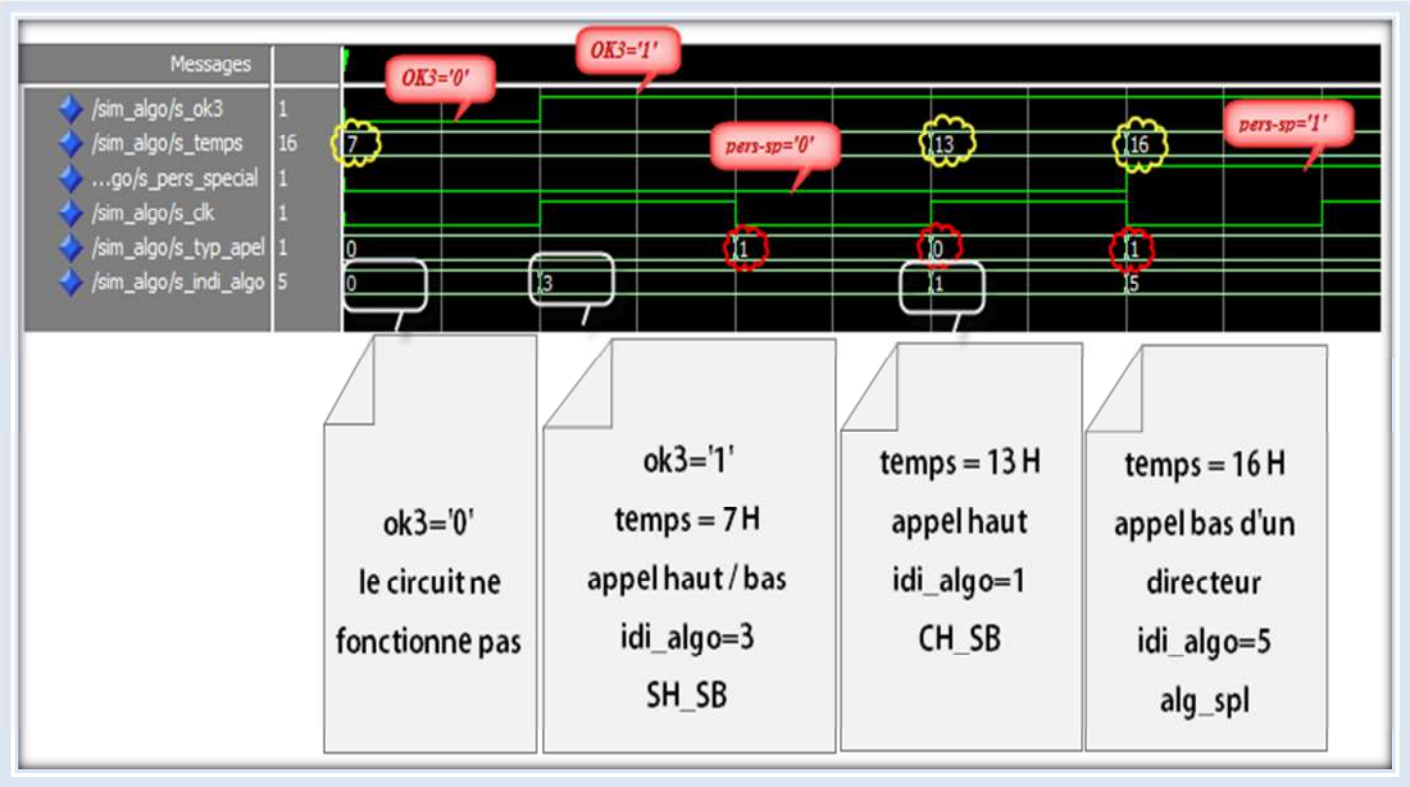
FigII.31. L'arbre de décision final

VI.2.5.6.3 Exemple de simulation :

On prend par exemple :

- Appel bas à 7 h
- Appel haut à 13h
- Appel bas d'un directeur à 16 h

Le résultat de simulation du circuit ALGO est donné par la figure II.31



FigII.32. Résultat de simulation du circuit ALGO

VI.2.6 Exemple de simulation du circuit SCGA :

Dans les parties précédentes nous avons démontré le bon fonctionnement de chaque composante du circuit SCGA, il suffit donc d'assurer la collaboration entre eux par un simple exemple :

Scénario : dans le même immeuble de 20 étages avec 4 ascenseurs on suppose que :

☞ A 10 H, les états des ascenseurs sont :

- ***Ascenseur 1*** : au ***RDC***, en mouvement vers le haut, « appel haut au $3^{ème}$ étage, sélection au 1^{ier} étage »

- ***Ascenseur 2*** : au $6^{ème}$ étage, en mouvement vers le haut, « appel haut au $8^{ème}$ étage, appel bas au $2^{ème}$ étage, sélection au $9^{ème}$ étage »
- ***Ascenseur 3***: au $3^{ème}$ étage, en mouvement vers le bas, « sélection au ***RDC*** »
- ***Ascenseur 4***: au $13^{ème}$ étage, en mouvement vers le bas, « appel bas au 1^{ier} étage, sélection au $9^{ème}$, $12^{ème}$ étage, réservé pour un appel spécial »

- ***A 10h01*** :
  - sélection dans la cabine d'ascenseur1 du $6^{ème}$ étage
  - l'ascenseur 2 est au $5^{ème}$ étage
- ***A 10 h02*** :
  - Appel haut au $7^{ème}$ étage

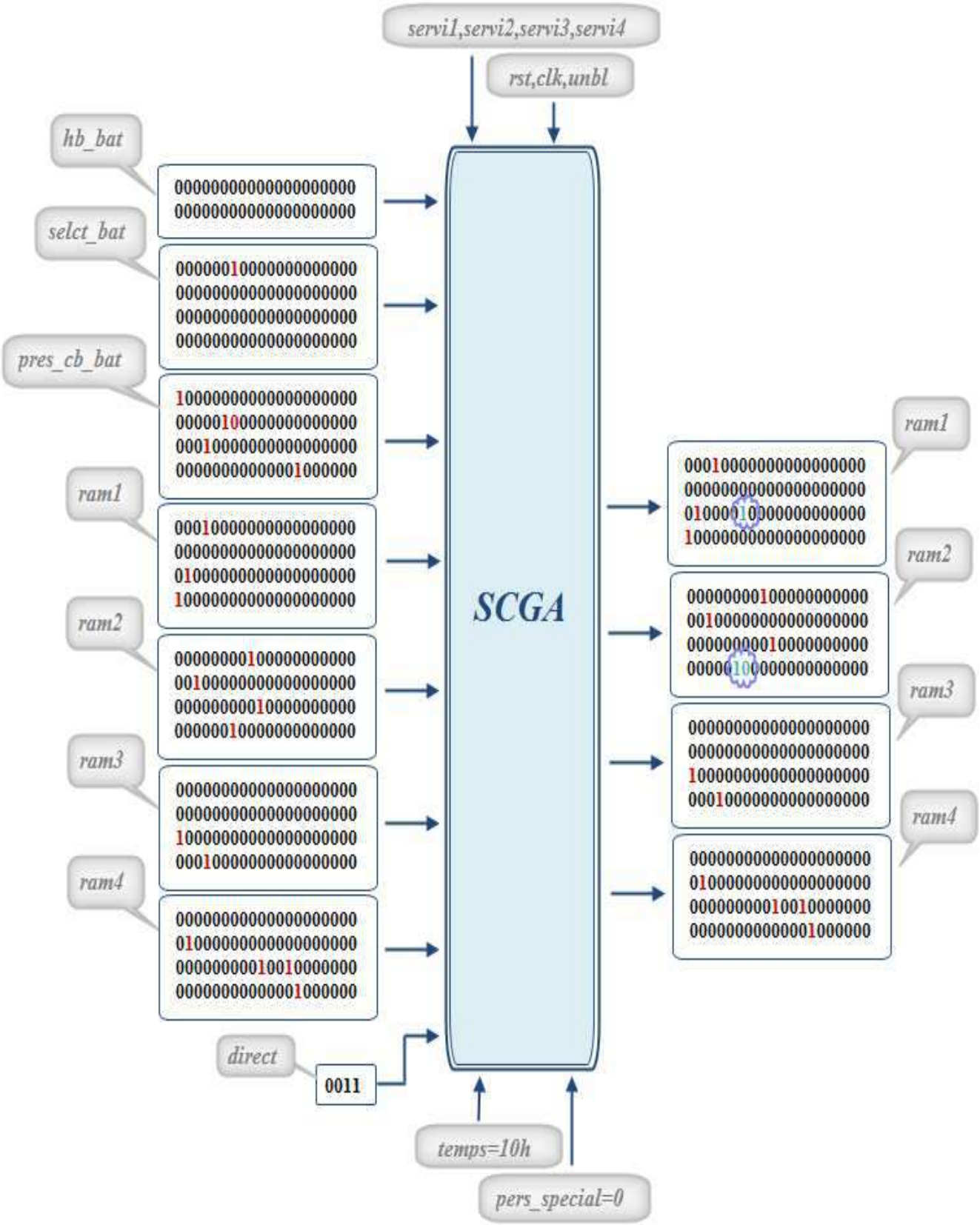
Remarque : à cause de la difficulté de copier le résultat fourni par le Modelism (image de grande taille parce que c'est une simulation pour un immeuble de 20 étages et 4 ascenseurs) on va expliquer le résultat de simulation par des schémas plus clairs.

Résultat :

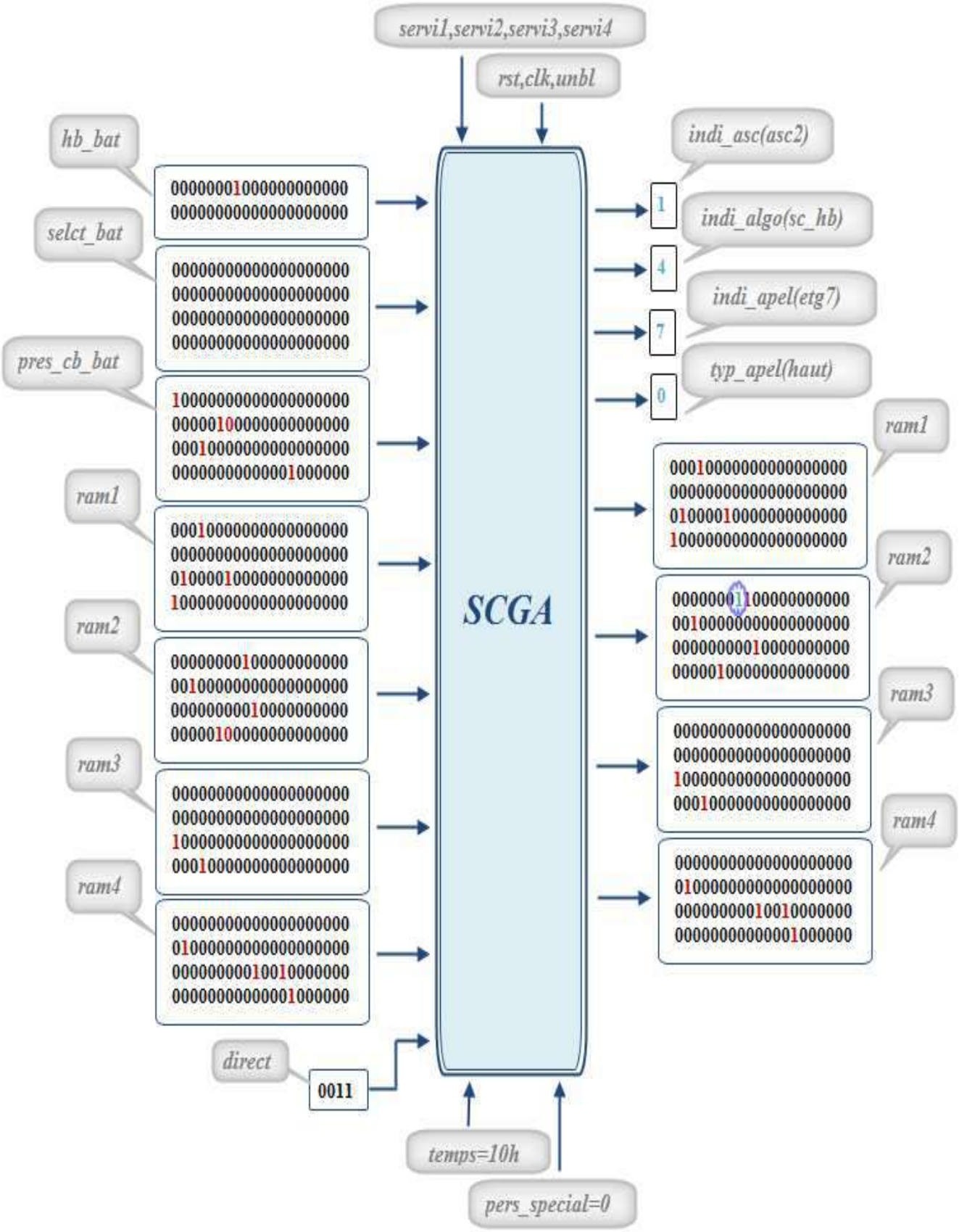
Les réactions du système SCGA selon les résultats de simulation fournis par Modelsim du scénario proposé sont :

1- Comme montre la figure II.32, à 10h01 les entrées du système SCGA sont les matrices et les vecteurs qui représentent les changements détecté dans l'environnement (les '1' en rouge) concernant la sélection arrivée « asc1 : $6^{ème}$ étage », la présence de toutes les cabines, et l'état actuel de chaque ascenseur « RAM's ».Le SCGA a placé la sélection dans la RAM d'ascenseur 1, et mis à jour la présence d'ascenseur 2 dans sa RAM.

2- La figure II.33 illustre la réaction du système SCGA après la détection d'un appel haut au $7^{ème}$ étage (le '1' en rouge dans la matrice hb_bat), il a choisi l'ascenseur et l'algorithme les plus appropriés pour servir l'appel détecté, puis l'a placé dans la RAM d'ascenseur choisi



FigII.33. Résultat de simulation du scénario « partie 1 » du circuit SCGA



FigII.34. Résultat de simulation du scénario « partie 2 » du circuit SCGA

VI.3 Le système de contrôle local d'un ascenseur SCL :

VI.3.1 Intérêt du circuit SCL

On peut considérer les ***SCLs*** comme les bras du système de gestion global, ils sont conçus pour appliquer les décisions fournis par le SCGA aux ascenseurs, afin de servir toutes les demandes en attentes enregistrées sur sa RAM, il se charge d'envoyer des commandes de :

- ***« monter/descendre »*** aux moteurs
- ***« ouvrir/fermer »*** aux portes étages et portes cabine
- ***« activer/ désactiver »*** aux sonnettes d'alarme
- « ***les situations des cabines »*** aux afficheurs des étages et des directions d'ascenseurs placés dans chaque étage.

C'est un module fondamental, à une grande importance dans la gestion juste et intelligente des ascenseurs, les SCLs installés dans l'immeuble ont leurs propres environnements, et ils fonctionnent en parallèle d'une manière indépendante.

Pour exploiter le maximum les choix intelligents du SCGA, chaque SCL peut fonctionner selon 6 algorithmes de dispatching différent, chacun prend en compte toutes les situations qui peuvent apparaître habituellement dans les immeubles, par exemple :

- la surcharge des cabines des ascenseurs
- le temps d'ouverture et de fermeture des portes
- le temps d'attente d'entrée et de sortie des passagers
- les demandes indiquées par les touches d'ouverture et de fermeture des portes
- la possibilité d'apparitions des empêchements de la fermeture.

VI.3.2 L'entité SCL

A partir de ce qui vient d'être expliqué précédemment dans l'architecture du system de gestion globale, on peut constater que chaque ascenseur installé dans l'immeuble est associé par son propre système de contrôle local SCL, et chaque SCL a besoin de détecter tous les évènements effectués dans son environnement, afin de réagir d'une manière juste et en temps réel. Il a besoin donc de suivre :

- ✓ L'activation d'ascenseur par le signal « ***unbl »*** (figure II.36)
- ✓ L'algorithme choisi pour cet ascenseur à partir « ***indi_algo »*** envoyé par le SCGA

- ✓ Les capteurs d'ouverture et de fermeture de toutes les portes étages situées dans le chemin de déplacement de cet ascenseur
- ✓ Les capteurs d'ouverture et de fermeture de la cabine d'ascenseur
- ✓ Le capteur de surcharge et photo cellule de présence, placés dans la cabine
- ✓ Les capteurs liés aux boutons d'arrêt d'urgence, et les boutons d'ouverture et de fermeture des portes d'ascenseurs
- ✓ L'état actuel de l'environnement d'ascenseur, fourni par la RAM envoyée à partir le système global
- ✓ En cas de détection d'un appel prioritaire (responsable, personne handicapé), il a besoin de connaitre l'indice d'appel, et le type (haut / bas) pour le servir directement
- ✓ Les signaux d'horloge (***CLK)*** et remise à zéro (***RST)***

La réaction du circuit SCL est représenté par l'envoie des commandes expliquées dans la partie précédente, plus des informations fournies au system de gestion global concernant :

- ⇨ la ***direction*** de mouvement de cet ascenseur
- ⇨ Le nouvel état d'ascenseur, à partir de la RAM modifiée « ***matrice_nouv*** »envoyée
- ⇨ Le signal « ***servi»*** qui indique si cet ascenseur est libre de service prioritaire ou non, pour ne pas le perturber si oui.

```
entity scl is
    port ( unabl : in std_logic ;
           indi_algo : in integer range 0 to k ;
           port_etg_ouvert,port_etg_ferm : in std_logic_vector (0 to N-1) ;
           cb_ouvert,cb_ferm,surch,pc,aret_urg,ferm_bout,ouv_urg : in std_logic ;
           rst,clk : in std_logic ;
           rasci : in matrice_SP ;
           id_ap : in integer range 0 to N-1;
           typ_apel : in integer range 0 to 1;

           ouver_cb,ferm_cb,direction,alarm : out std_logic ;
           mont,descent,ouver_port_etg,ferm_port_etg : out std_logic_vector(0 to N- 1);
           matrice_nouv : out matrice_SP;
           afficheur1, afficheur2 : out std_logic_vector (6 downto 0);
           servi : out std_logic) ;
end scl ;
```

Fig II.35 entité du circuit SCL

VI.3.3 Architecture du circuit SCL

Pour mieux comprendre le principe de fonctionnement du SCL, il faut tout d'abord bien connaitre son architecture interne.

C'est un circuit simple composé de 6 modules, chacun représente un algorithme de dispatching, plus un module d'adaptation des entrées/sorties (les entrés d'algorithme sélectionné, et les sorties du système de contrôle SCL), les deux principes fondamentaux appliqués dans ce circuit sont:

- Un module seulement qui fonctionne selon un algorithme parmi les 6 algorithmes de dispatching, leur entrées sont les mêmes que le circuit SCL, et ce dernier adapte ses sortie à partir des sorties fournies par l'algorithme qui fonctionne en utilisant le module d'adaptation.
- Lors du changement d'algorithme, et avec les entrées provenant à partir l'environnement, le nouvel algorithme a besoin de connaitre l'état final d'algorithme précédant, à partir des données fournies par celui-ci concernant l'indice appel à traiter, et l'état où la machine d'état est arrêtée.

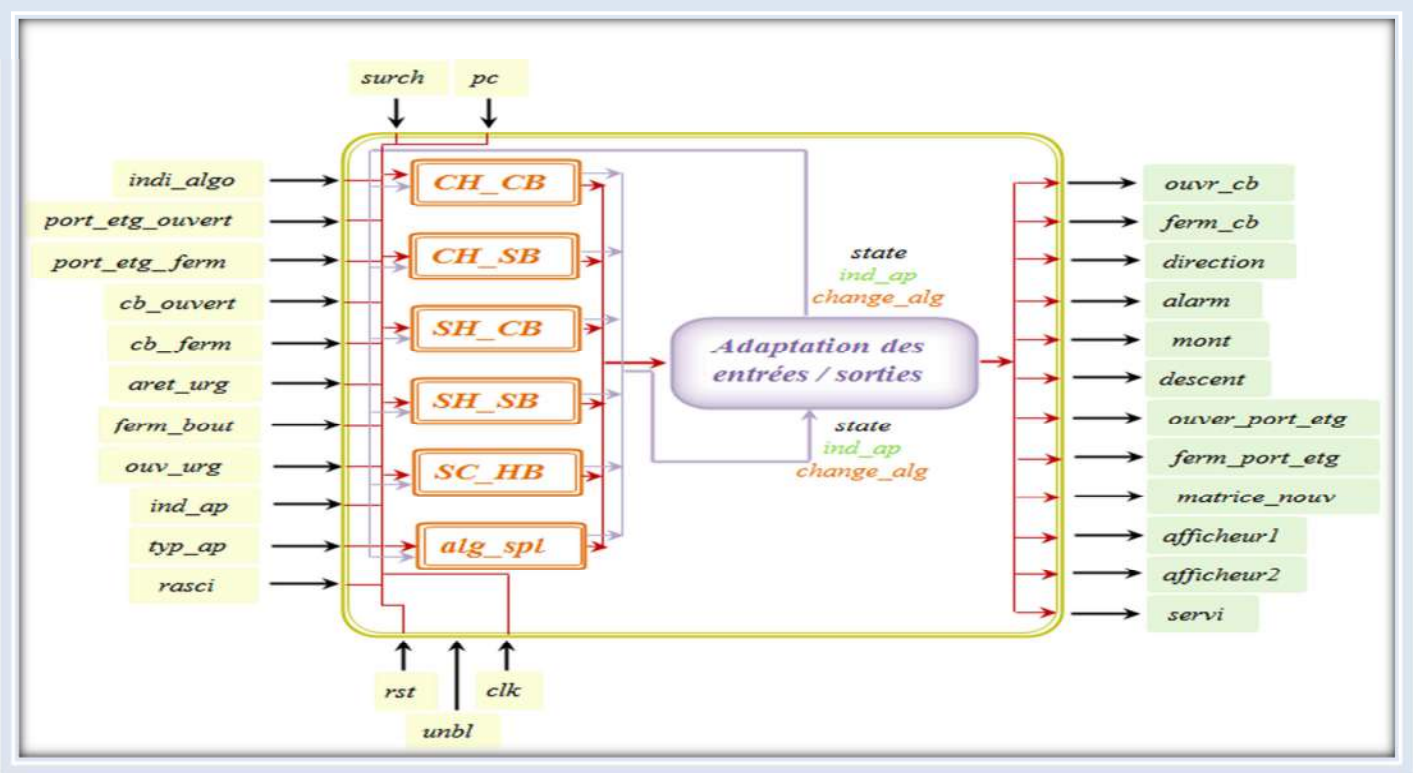


Fig II.36 Architecture interne du circuit SCL

VI.3.4 Principe de fonctionnement du circuit SCL

La figure II.38 représente l'organigramme des taches réalisées par le circuit SCL :

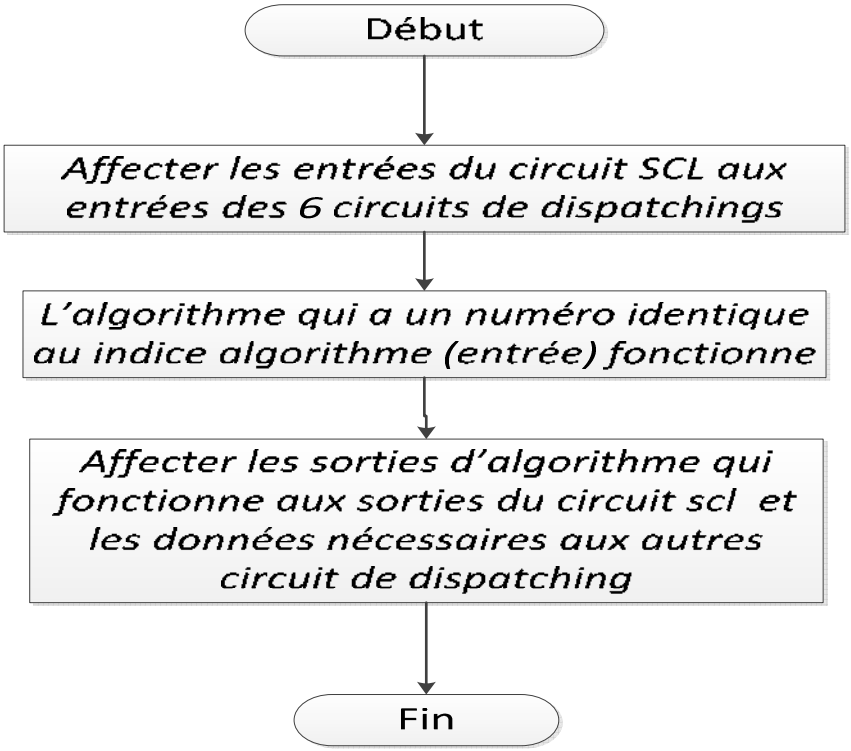


Fig II.37 l'organigramme de fonctionnement du circuit SCL

VI.3.5 Les composantes du circuit SCL

A part le circuit d'adaptation des entrées/sorties, les six circuits restants sont des implémentations des algorithmes de dispatching qu'on a déjà expliqué leur principe de fonctionnement dans le chapitres I , ce sont des différentes stratégies de contrôle, choisies suivant des critères précis, et appliquées afin de répondre aux demandes des utilisateurs le plus rapidement possible.

Comme on a expliqué précédemment, les entrées/sorties sont les mêmes pour tous les algorithmes de dispatching indiquées par le schéma de la figure II.39

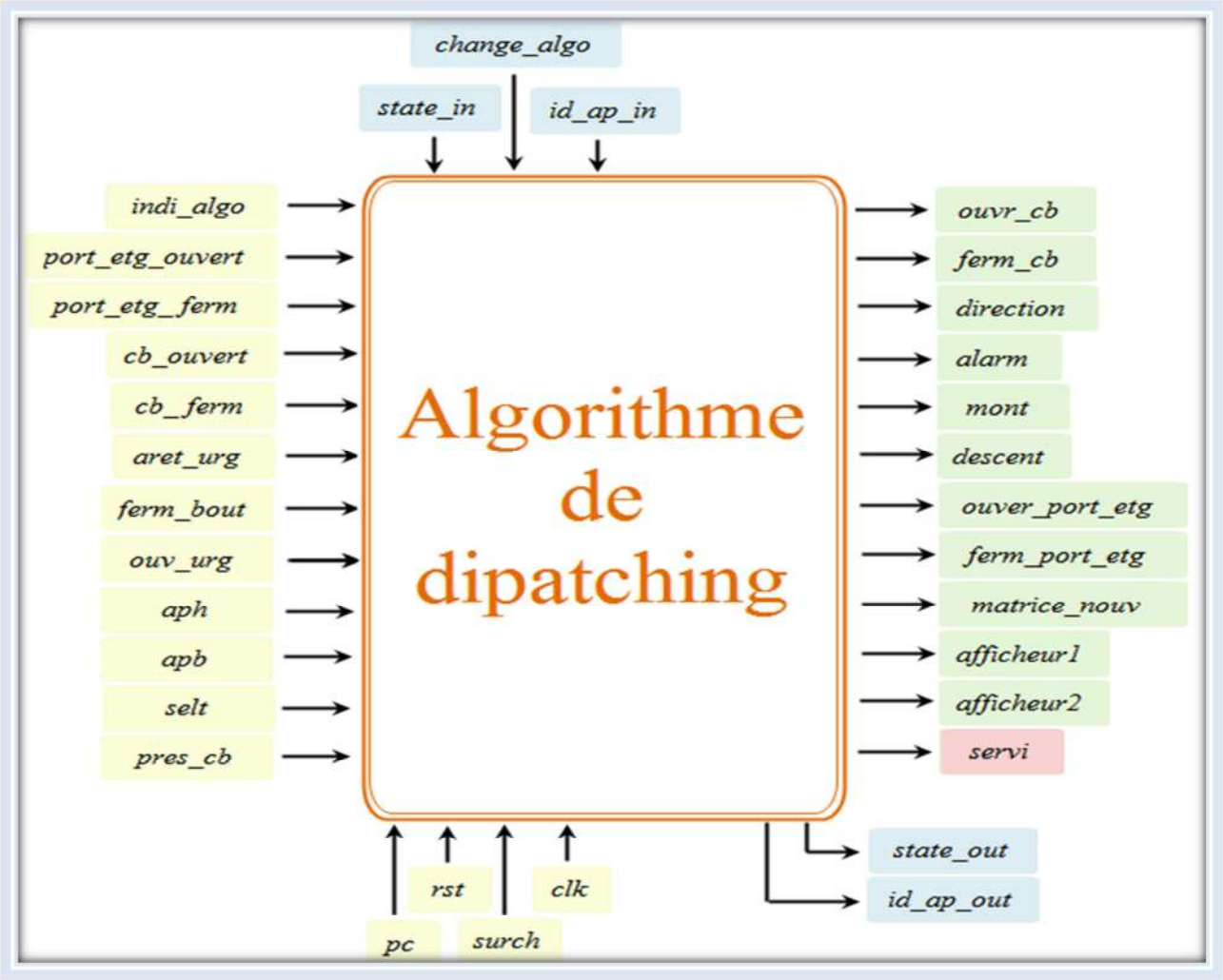


Fig II.38 les entres/sorties d'algorithme de dispatching

a) Entrées :

- Les mêmes entrées du circuit SCL sauf que la RAM est décomposée en 4 vecteurs :
  - La ligne 1 de la RAM est affectée au vecteur des appels hauts « ***aph*** »
  - La ligne 2 de la RAM est affectée au vecteur des appels bas « ***apb*** »
  - La ligne 3 de la RAM est affectée au vecteur des sélections « ***selt*** »
  - La ligne 4 de la RAM est affectée au vecteur des présences cabines « ***pres_cb*** »
- Entrées donnant le dernier état d'environnement avant le changement d'algorithme pour commencer à partir de celui-ci.

b) Sorties :

- Les mêmes sorties que le circuit **SCL**, plus les sorties de sauvegarde d'état d'environnement en cas du changement d'algorithme, qui sont les deux signaux « ***state_out*** : l'état d'arrêt dans la machine d'état, ***id_ap_out*** : l'indice d'appel à traiter »

- Sortie « ***servi*** » ne se trouve que dans le 6ème algorithme ***alg_spl***. Lors de son fonctionnement, ce signal est mis à **'1'** pour indiquer que cet ascenseur est entrain de servir une demande spéciale et il ne faut pas le perturber, sinon il est mis à '0'.

Remarques importantes :

- ✓ Dans les implémentations VHDL des algorithmes de dispatching réalisés, on a travaillé sur les deux fronts du signal d'horloge, le **front montant** pour décider l'état suivant à l'aide des informations envoyés par les capteurs , et le **front descendant** pour envoyer les commandes aux actionneurs des ascenseurs.
- ✓ A chaque commande envoyée, l'ascenseur ne termine son travail qu'après la confirmation d'implémentation juste de son ordre par les actionneurs, à partir des capteurs dispersés dans l'immeuble. ***Exemple*** : après l'envoi du commande de montée par 3 étages (par exemple), l'ascenseur n'envoie pas la commande d'ouverture qu'après la confirmation du montée de la cabine étage par étage, à partir des capteur du présence cabine ,la même chose après l'envoi du commande d'ouverture et de fermeture des portes , il ne passe pas à l'état suivante qu'après être sûr que les portes sont ouvertes ou fermées, par les capteurs des portes des étages et de la cabine .
- ✓ A cause de la contrainte précédente (la confirmation de la bonne marche d'ascenseur après chaque commande) , la programmation d'une simulation pour assurer le bon fonctionnement de chaque algorithme sera difficile, car il faut donner après chaque commande l'état de tous les capteurs d'ascenseur appropriés par l'imagination, et une petite erreur peut provoquer un blocage du programme, et ça ne passe jamais dans la réalité avec l'existence des capteurs réels liés au system de contrôle SCL ,c'est pour ça, qu'on a essayé d'assurer la bonne marche de tous les circuit par des scénarios courts et simples.
- ✓ concernant les machines d'états expliquées dans les parties suivantes, Il y a des états communes entres tous les algorithmes de dispatchings comme l'état S2 et S3 d'ouverture et de fermeture des portes, et même pour les autres états, le principe de leur fonctionnement reste le même, ce qui diffère c'est la stratégie de choix des appels à servir en premier, c'est pour ça et afin de ne pas réciter les même principes à chaque fois , nous avons bien expliqué les états dans le premier algorithme CH_CB , et on a cité ce qui diffère dans les autres .
- ✓ Dans chaque algorithme il y a un module de détection du signal de surcharge, pour envoyer la commande d'alarme aux sonnettes, pour que les passagers soient informés au problème.
- ✓ on a montré la réaction juste de nos circuits suivant des scénarios réels implémentés par des programmes de simulation en VHDL , mais on n'a pas pu copier les résultats de simulation

fourni par Modelsim, par ce qu'ils sont des images de très grande taille , alors dans la dernière partie de chaque algorithme de dispatching , on a cité le scénario proposé puis la réaction d'ascenseur pas à pas, selon notre programme, fournie par le Modelsim , et on a interprété les résultats des simulations par des images pour mieux comprendre son fonctionnement.

VI.3.5.1 Le circuit CH_CB

VI.3.5.1.1 Stratégie de fonctionnement

Le circuit ***CH_CB*** est une implémentation d'algorithme de dispatching ***collectif haut /collectif bas***. Il permet de servir toutes les demandes ayant la même direction de déplacement d'ascenseur, et qui ne sont pas encore dépassées, de point de vue implémentation VHDL ça veut dire que :

- Si l'ascenseur est en déplacement vers le haut, il doit servir tous les appels hauts et les sélections détectées aux étages qui ne sont pas encore dépassés, ce sont les '1' trouvés au niveau des deux vecteurs **aph** et **selt,** dans les cases supérieures à l'indice de présence de la cabine d'ascenseur
- Si l'ascenseur est en déplacement vers le bas, il doit servir tous les appels bas et les sélections détectées aux étages qui ne sont pas encore dépassés, ce sont les '1' trouvés au niveau des deux vecteurs **apb** et **selt,** dans les cases inférieures à l'indice de présence de la cabine d'ascenseur

VI.3.5.1.2 Machine d'état

Afin d'appliquer les principes expliqués précédemment et pour réaliser une implémentation VHDL de la stratégie proposée, on a pensé à construire la machine d'état illustrée dans la figure II.40 :

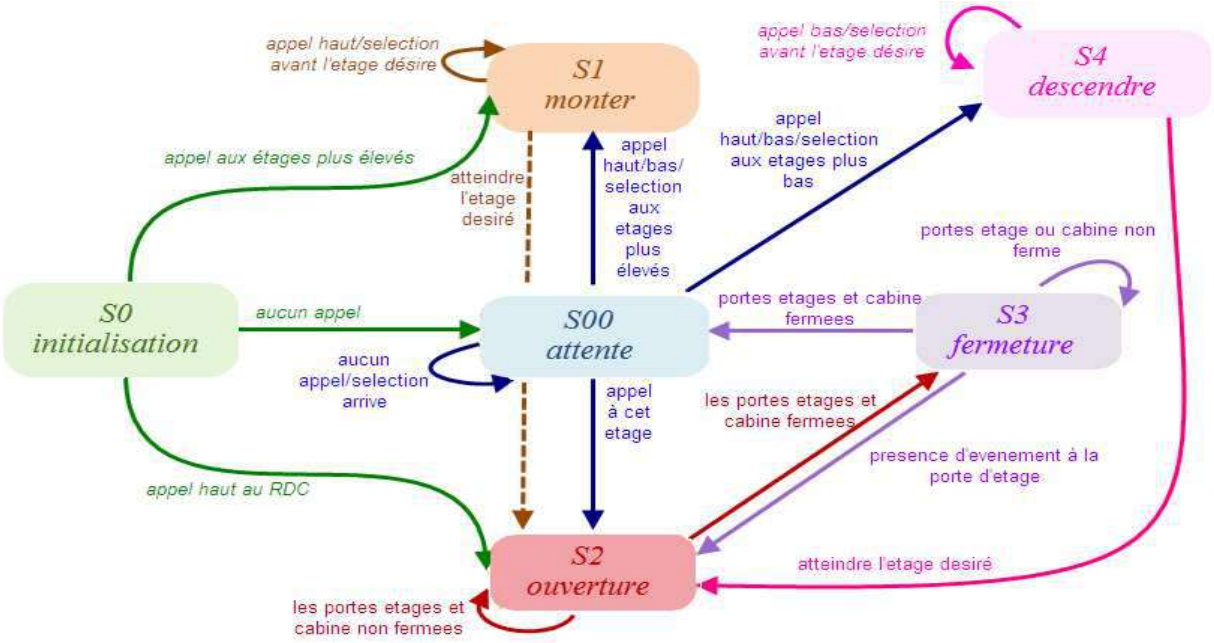


Fig II.39 machine d'état d'algorithme collectif haut/collectif bas

❖ ***S0 (état d'initialisation) :***C'est l'état d'ascenseur lors de son alimentation ou son initialisation ***« rst='1 »,*** la cabine se trouve normalement au RDC. À partir de cet état et suivant l'évènement détecté, on peut aller aux 3 états possibles (après la confirmation que toutes les portes sont fermées) :

- ☞ À l'état d'ouverture des portes (passant par l'état de monter par '0 'étage) : si on a une demande au RDC ***« aph(0) ='1' »***
- ☞ À l'état de montée : si on a au moins une demande (haut/ bas) aux étages les plus élevés « ***aph(i) ='1' ou apb(i)='1' tq 0<i<N*** », en envoyant l'indice d'appel à servir
- ☞ À l'état d'attente : si aucun évènement n'est détecté

❖ ***S1 (état de déplacement vers le haut) :*** Dans cet état, une commande de montée est envoyée au moteur d'ascenseurs. À partir de vecteur de sortie « ***mont »*** de taille N (N : nombre des étages d'immeuble),on peut commander le moteur de monter i étages à la fois, il suffit d'envoyer un '1' dans la $i^{ème}$ case de ce vecteur « ***mont(i)='1' »***, gardant la capacité de contrôler l'existence d'une autre demande dans son trajet, pour la servir immédiatement, seulement avec un nouveau calcule de i , qui est la différence entre l'étage d'appel à servir et l'étage d'existence de cabine . Nous ne passons pas à l'état d'ouverture des portes que si on atteint l'étage d'appel à servir après l'arrêt automatique de la cabine.

Exemple :

- on suppose qu'un appel est détecté à l'état S0 au 3ème étage, alors
  L'indice d'appel venu = 3.
- Envoyer la commande : ***mont(3)='1'***
- La cabine commence à monter
- Lorsqu'elle atteint le premier étage, un appel haut au 2ème étage est détecté
- Changement de la commande : ***mont(1)='1'***
- La cabine monte un étage et s'arrêté (au 2ème étage)
- Passer à l'état d'ouverture des portes

❖ ***S2 (état d'ouverture des portes) :***les portes palières prennent un délai de temps pour s'ouvrir et se fermer suivant un mécanisme de coulissement , et il faut prendre ce mécanisme en compte pour que notre system puisse gérer correctement leurs mouvement, c'est exactement l'utilité d'utilisation de deux signaux pour chaque porte, en indiquant son état (capteurs d'ouverture et de fermeture (figure II.41)) et deux autres pour commander l'action à faire (actions d'ouverture et de fermeture) , on les regroupe dans des vecteurs de taille N (un vecteur pour chaque signal) pour toutes les portes trouvées dans le trajet d'ascenseur (une porte dans chaque étage) :

- port_etg_ouvert : vecteurs des capteurs d'ouvertures des portes
- port_etg_ferm : vecteurs des capteurs de fermeture des portes
- ouver_port_etg : vecteurs des commandes d'ouverture des portes
- ferm_port_etg : vecteurs des commandes de fermeture des portes

n'oublions pas la porte simple de la cabine, représenté par deux capteurs ***(cb_ouver ,cb_ferm)*** et deux actionneurs ***(ouver_cb***, ***ferm_cb***)

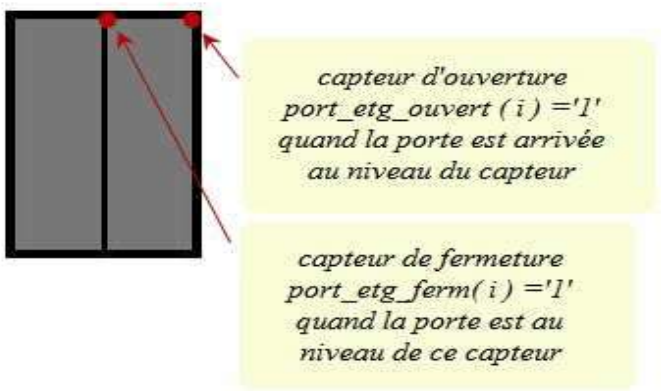


Fig II.40 les capteurs placés dans les portes palières

Implémentation du contrôle du mécanisme des portes :

En initialisation et lors de déplacement d'ascenseur, toutes les portes des étages sont fermées c'est bien que tous les capteurs d'ouverture donnent '0', et les capteurs de fermeture donnent '1' ; par conséquent, l'ouverture et la fermeture des portes passent par les étapes suivantes (exemple la porte d'étage 3) :

- lors de la montée d'ascenseur :

port_etg_ouvert (3) = '0' ; ***port_etg_ferm (3) ='1'***

- L'ascenseur atteint l'étage 3 :

Envoyer la commande d'ouverture :

De la porte d'étage : ***ouver_port_etg(3) ='1'***

Et la porte cabine : ***ouver_cb='1'***

- La porte commence à s'ouvrir (temps d'ouverture)

port_etg_ouvert (3) = '0' ; ***port_etg_ferm (3) ='0' ; cb_ouver='1' ;***

- La porte est ouverte

port_etg_ouvert (3) = '1' ; ***port_etg_ferm (3) ='0'***

- Attendre Le délai d'entrée des passagers (un compteur représenté par un tableau std_logic de taille 3 : « 000 …… 111 »)

❖ ***S3 (état de fermeture des portes) :*** Après le passage du délai d'attente d'entrées des passagers à la cabine, ou après la détection d'une demande manuelle de fermeture (bouton de fermeture) ,la porte doit se fermer de la même manière d'ouverture expliqué dans S2, sauf qu'il faut prendre en compte le contrôle du surcharge de la cabine, et l'existence des empêchements de fermeture (photo cellule), ou l'arrivée d'une demande manuelle d'ouverture (bouton d'ouverture), qui causent la réouverture des portes d'ascenseurs de nouveau (aller à l'état S2) . La fermeture des portes passe par les étapes suivantes :

- Envoyer la commande de fermeture : ***ferm_port_etg(3) ='1' , ferm_cb='1'***
- La porte commence à se fermer (le temps de fermeture)

port_etg_ouvert (3) = '0' ; ***port_etg_ferm (3) ='0' ;***

- La porte est fermée

port_etg_ouvert (3) = '0' ; ***port_etg_ferm (3) ='1' ; cb_ferm='1' ;***

Après la fermeture des portes on passe à l'état d'attente S00 pour décider l'état suivant.

❖ ***S4 (état de déplacement vers le bas) :*** Dans cet état une commande de descente est envoyée au moteur d'ascenseurs. À partir du vecteur de sortie « ***descent»*** de taille N (N : nombre des étages d'immeuble),on peut commander le moteur de descendre i étages à la fois, il suffit d'envoyer un '1' dans la $i^{ème}$ case de ce vecteur « ***descent(i)='1' »***, gardant la capacité de contrôler l'existence d'une autre demande dans son trajet, pour la servir immédiatement, seulement avec un nouveau calcule de i, qui est la différence entre l'étage d'existence de cabine et l'étage d'appel à servir . Nous ne passons pas à l'état d'ouverture des portes que si on atteint l'étage d'appel sélectionné après l'arrêt automatique de la cabine.

Exemple :

- on suppose qu'un appel est détecté à l'état d'attente S00 au 1ére étage, et la cabine est au $6^{ème}$ étage alors :

  L'indice d'appel venue = 6
- Envoyer la commande : ***descent (5)='1'***
- La cabine commence à descendre
- Lorsqu'elle atteint le $4^{ème}$ étage, sélection au $2^{ème}$ étage est détectée
- Changement de la commande : ***descent (2)='1'***
- La cabine descente par 2 étages et s'arrête (au $2^{ème}$ étage)
- Passer à l'état d'ouverture des portes

❖ ***S00 (état d'attente) :*** C'est l'état de repos d'ascenseur, il se met à cet état après chaque fermeture des portes, ou s'il n'y a aucune demande détectée, il reste à cet état jusqu'à l'arrivé d'un nouvel événement, et on peut aller à 3 états possibles selon son emplacement :

☞ A l'état S1 s'il y a des demandes aux étages plus hauts

☞ A l'état S4 s'il y a des demandes aux étages plus bas

☞ A l'état S2 d'ouverture des portes si la demande est à l'étage où la cabine se trouve

VI.3.5.1.3 Exemple de simulation

Pour assurer le bon fonctionnement de l'ascenseur avec le mode ***collectif haut / collectif bas*** on a imaginé un scénario qui montre la stratégie de sélection des demandes à servir

Scénario :

☞ L'ascenseur est au RDC en état repos (la porte cabine et toutes les portes étages sont fermées)

☞ **Appel haut** au **RDC** est arrivé

- ⇨ L'ascenseur ouvre les portes
- ⇨ Attendre un délai pour que les passagers entrent
- ⇨ La fermeture des portes

☞ Lors de la fermeture des portes un passager veut monter à la cabine, il empêche la fermeture

- ⇨ les portes s'ouvrent de nouveau
- ⇨ Attendre un délai pour que les passagers entrent
- ⇨ La fermeture des portes

☞ Les passagers à l'intérieur de la cabine veulent aller aux **4ème** et **8ème** étage

- o La cabine commence à monter, et quand elle atteint le 1ier étage un **appel haut** au **2ème** étage arrive
- o La cabine monte un autre étage
  - ⇨ Les portes s'ouvrent pour servir l'appel haut (2ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes
- o La cabine monte 2 étages
  - ⇨ Les portes s'ouvrent pour servir la sélection (4ème étage)
  - ⇨ Attendre un délai pour que les passagers sortent
  - ⇨ La fermeture des portes, et l'arrivé d'un **appel bas** au **5ème** étage
- o La cabine monte par 4 étages (il ne sert pas l'appel bas)
  - ⇨ Les portes s'ouvrent pour servir la sélection (8ème étage)
  - ⇨ Attendre un délai pour que les passagers sortent
  - ⇨ La fermeture des portes
- o La cabine commence à descendre par 3 étages pour servir l'appel bas

☞ Lorsque l'ascenseur est dans le 7ème étage un **appel haut** au **6ème** étage est arrivé

- o La cabine termine sa descente au 5ème étage et il ne sert pas l'appel haut
  - ⇨ Les portes s'ouvrent pour servir l'appel (5ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes

☞ Les passagers à l'intérieur de la cabine veulent aller au **4ème** étage

- La cabine descend par un étage
  - ⇨ Les portes s'ouvrent pour servir la sélection ($4^{ème}$ étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes

☞ Aucun autre appel bas resté, l'ascenseur doit changer son sens de déplacement pour servir l'appel haut resté selon les mêmes principes expliqués lors de son déplacement vers le haut.

☞ Le résultat de simulation de ce scénario est interprété par la figure II.42

VI.3.5.2 Le circuit CH_SB

VI.3.5.2.1 Stratégie de fonctionnement

Le circuit ***CH_SB*** est une implémentation d'algorithme de dispatching : ***collectif haut /sélectif bas***, il conserve les mêmes principes du mode collectif (expliqué précédemment) lors de son déplacement vers le haut, et dans le sens opposé , il doit servir la demande la plus haute ,de point de vue numéro d'étage , en premier ; après qu'il atteint l'étage choisi, il ne répond à aucune demande avant qu'il termine toutes les sélections des passagers à l'intérieur de la cabine, de point de vue implémentation VHDL ça veut dire que :

- Si l'ascenseur est en déplacement vers le haut, il doit servir tous les appels hauts et les sélections détectées aux étages qui ne sont pas encore dépassés, ce sont les '1' trouvés au niveau des deux vecteur **aph** et **selt,** dans les cases supérieures à l'indice de présence de la cabine d'ascenseur
- Si l'ascenseur est en déplacement vers le bas, il doit servir le plus haut appel bas détecté, c'est l'indice le plus grand dans le vecteur **apb** qui contient un '1', puis servir tous les indices qui contient un '1' dans le vecteur **selt**

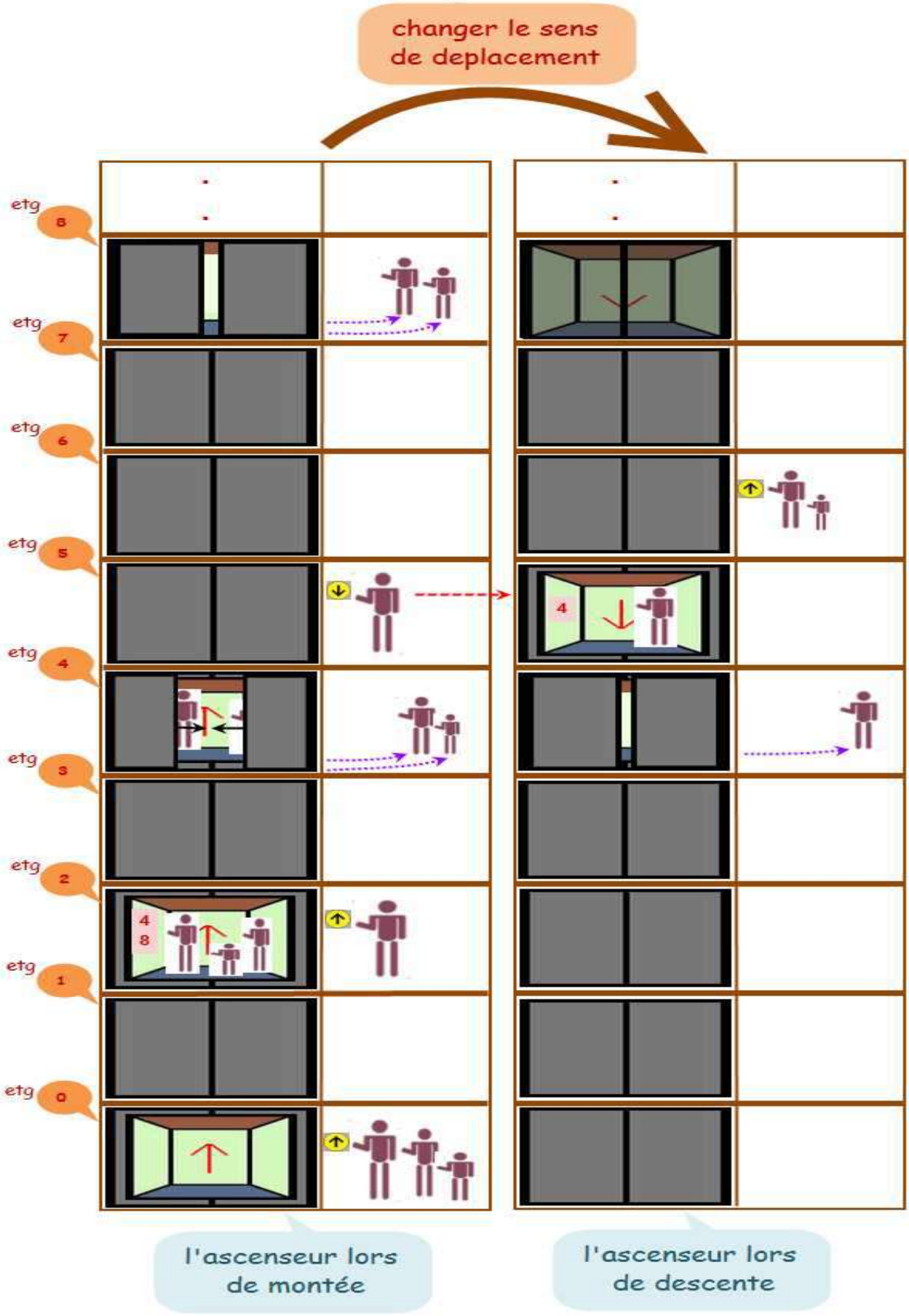


Fig II.41 interprétation du résultat de simulation du circuit CH_CB

VI.3.5.2.2 Machine d'état

Le fonctionnement du circuit CH_SB est illustré par la machine d'état représentée dans la figure II.43 :

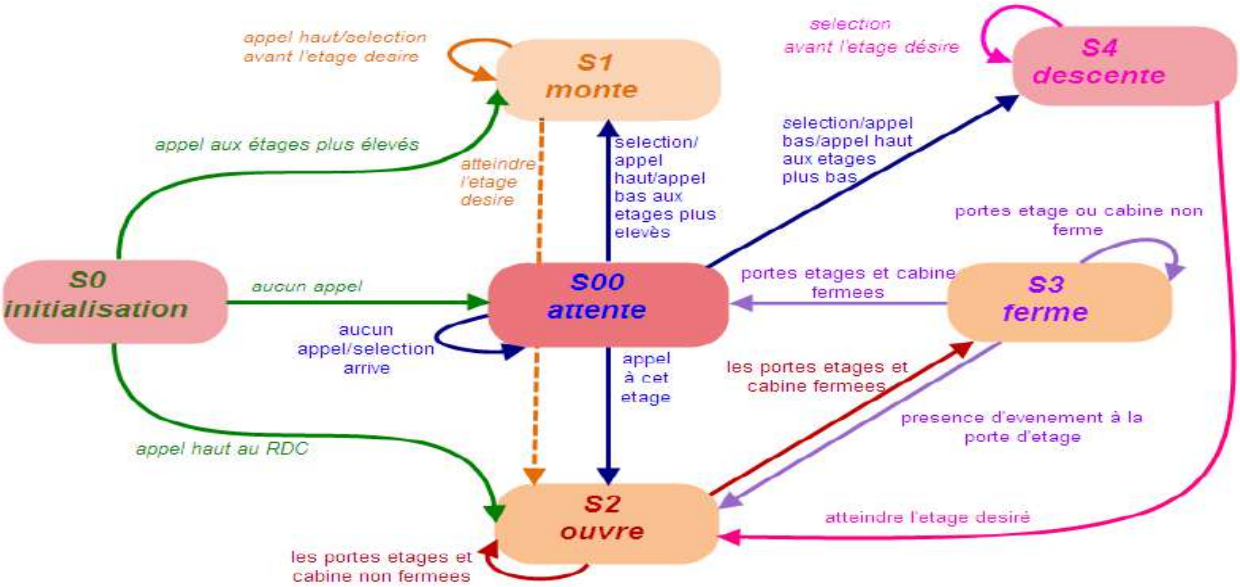


Fig II.42 machine d'état d'algorithme collectif haut/sélectif bas

❖ ***S0 (état d'initialisation) :*** De même que l'algorithme précédant, à partir de cet état on peut aller à 3 états possibles :

☞ À l'état d'ouverture des portes : si on a une demande au RDC ***« aph(0) ='1' »***

☞ À l'état de montée :

- si on a un appel aux étages plus élevés « ***aph(i) ='1'*** »
- si aucune demande haut est détectée, chercher le plus haut appel bas (le premier à partir dernier étage)

☞ À l'état d'attente : si aucun évènement n'est détecté

❖ ***S1, S2, S3:***Le même principe expliqué précédemment (partie VI.3.5.1.3)

❖ ***S4 (état de déplacement vers le bas) :*** Le même principe de descente (expliqué dans la partie VI.3.5.1.3), sauf que lors de sa descente, l'ascenseur ne prend en compte que les sélections des passagers à l'intérieur de la cabine, c.-à-d. il ne sert pas immédiatement les appels détectés dans son trajet.

❖ ***S00 (état d'attente) :*** C'est l'état de repos d'ascenseurs, à partir de cet état et suivant l'évènement détecté et le sens de déplacement d'ascenseur, on peut aller à:

☞ *A l'état S1* :

- si l'ascenseur est en déplacement vers le haut, et il y a des appels hauts ou des sélections aux étages plus élevés
- sinon chercher des appels bas aux étages plus élevés, et prend le premier à partir du dernier étage
- si l'ascenseur est en déplacement vers le bas, et il ne reste que des demandes aux étages plus hauts

☞ *A l'état S4 :*

- si l'ascenseur est en déplacement vers le bas, et il y a des sélections aux étages plus bas
- sinon chercher des appels bas et prend l'indice le plus haut
- sinon chercher des appels hauts aux étages plus bas

☞ A l'état S2 d'ouverture des portes si la demande et à l'étage où se trouve la cabine.

VI.3.5.2.3 Exemple de simulation

Pour assurer le bon fonctionnement de l'ascenseur avec le mode ***collectif haut / sélectif bas*** on a imaginé un scénario qui montre la stratégie de sélection des demandes à servir

Scénario :

☞ L'ascenseur est au RDC en état repos (la porte cabine et toutes les portes étages sont fermées)

☞ **Appel haut** au **RDC** est arrivé

⇨ L'ascenseur ouvre les portes

⇨ Attendre un délai pour que les passagers entrent

⇨ La fermeture des portes

☞ Lors de la fermeture des portes l'état de surcharge est détecté à la cabine

⇨ L'alarme est déclenchée

⇨ les portes s'ouvrent de nouveau

⇨ Attendre jusqu'à la régulation du problème (sortie de quelque passagers) puis recommencer de calcul du délai d'attendre

⇨ La fermeture des portes

☞ Les passagers à l'intérieur de la cabine veulent aller au **4ème** et **8ème** étage

o La cabine commence à monter et quand elle atteint le 1ier étage un **appel haut** au **2ème** étage arrive

- La cabine monte un autre étage
  - ⇨ Les portes s'ouvrent pour servir l'appel haut (2ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes
- La cabine monte 2 étages
  - ⇨ Les portes s'ouvrent pour servir la sélection (4ème étage)
  - ⇨ Attendre un délai pour que les passagers sortent
  - ⇨ La fermeture des portes, et l'arrivé d'un **appel bas** au **5ème** étage

- La cabine monte 4 étages (il ne sert pas l'appel bas), et quand elle atteint le 6ème étage un **appel haut** au **8ème** étage est détecté, et au niveau de **7ème** étage un **appel bas** au même étage est arrivé
  - ⇨ Les portes s'ouvrent pour servir la sélection et l'appel haut (8ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes
- La cabine descend d'un étage pour servir l'appel bas (il ne reste pas d'autre sélections)
  - ⇨ Les portes s'ouvrent (7ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes

☞ Les passagers à l'intérieur de la cabine veulent aller au 3ème étage

- La cabine descend 4 étage (il ne sert pas l'appel bas sur son trajet au niveau du 5ème étage, les sélections sont prioritaires)
  - ⇨ Les portes s'ouvrent pour servir la sélection (3ème étage)
  - ⇨ Attendre un délai pour que les passagers sortent
  - ⇨ La fermeture des portes

☞ Aucune autre sélection est en attente, l'ascenseur doit changer son sens de déplacement : aucun appel haut arrivé alors il doit servir l'appel bas qui reste au 5ème étage.

☞ Le résultat de simulation de ce scénario est interprété par la figure II.44

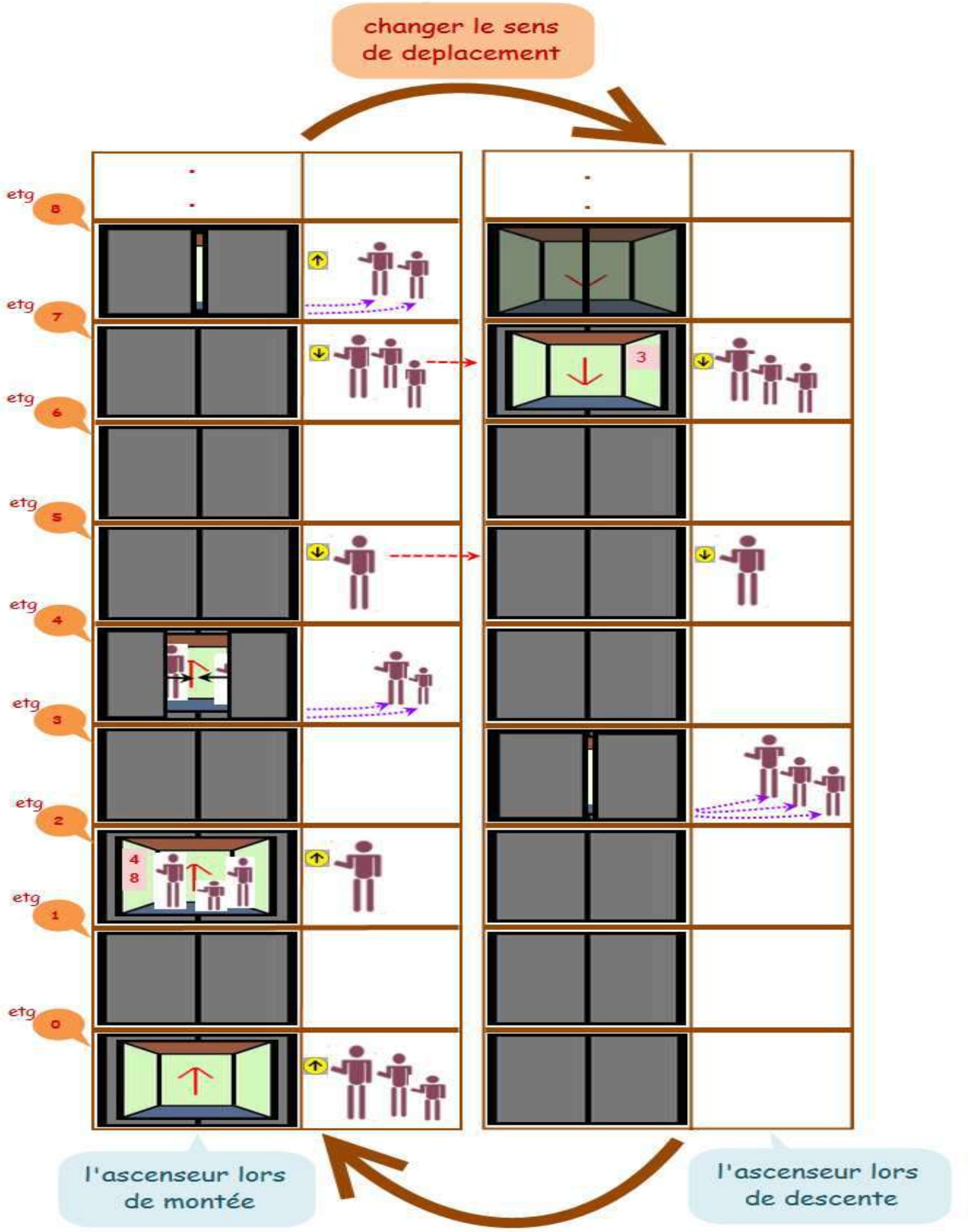


Fig II.43 interprétation du résultat de simulation du circuit CH_SB

VI.3.5.3 Le circuit SH_CB

VI.3.5.3.1 Stratégie de fonctionnement

Le circuit ***SH_CB*** est une implémentation d'algorithme de dispatching : ***sélectif haut /collectif bas***, il fonctionne contrairement au algorithme précédant (CH_SB).Lors de son déplacement vers le haut l'ascenseur travaille dans le mode sélectif, et dans les sens opposé au mode collectif, de point de vue implémentation VHDL ça veut dire que :

- Si l'ascenseur est en déplacement vers le haut, il doit servir le premier appel haut détecté à partir du RDC, c'est l'indice le plus petit dans le vecteur **aph** qui contient un '1', puis sert tous les indices contenant un '1' dans le vecteur **selt.**
- Si l'ascenseur est en déplacement vers le bas, il doit servir tous les appels bas et les sélections détectées aux étages qui ne sont pas encore dépassés, ce sont les '1' trouvés au niveau des deux vecteur **apb**et **selt,** dans les cases inferieurs à l'indice de présence de la cabine d'ascenseur

VI.3.5.3.2 Machine d'état

Le fonctionnement du circuit SH_CB est illustré par la machine d'état représentée par la figure II.45 :

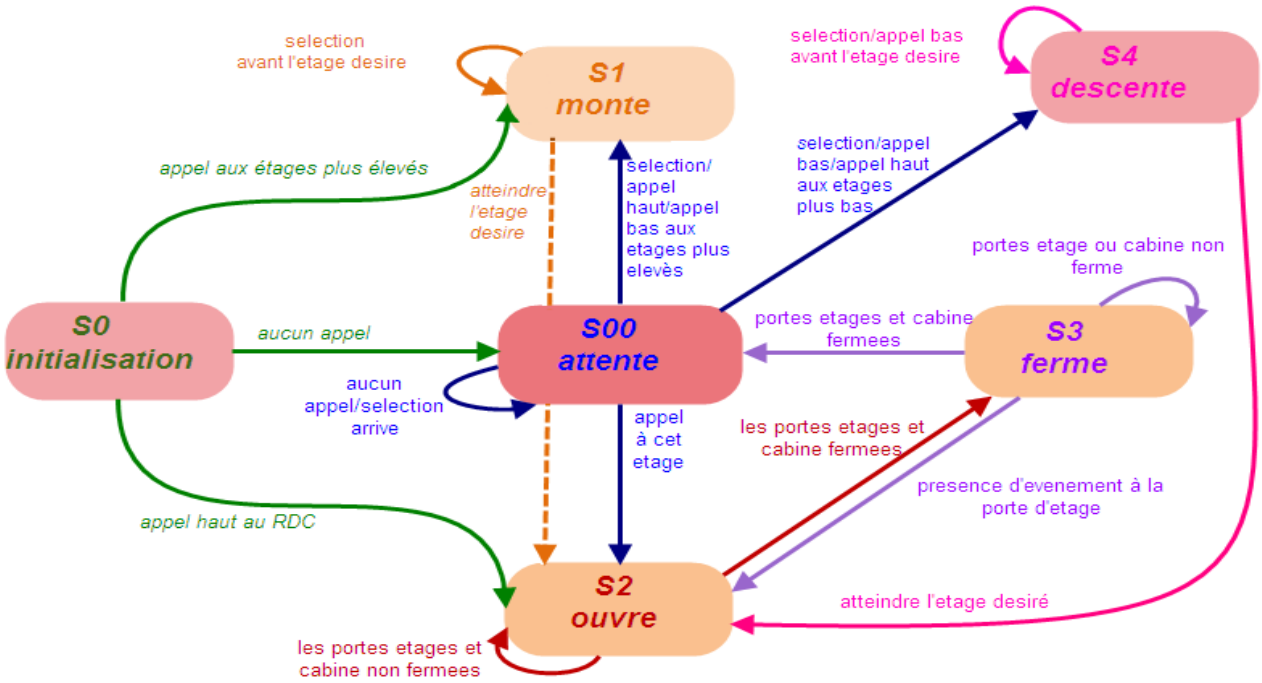


Fig II.44 machine d'état d'algorithme sélectif haut/collectif bas

❖ ***S0 (état d'initialisation) :*** la cabine se trouve au RDC. A partir de cet état et suivant l'évènement détecté on peut aller à 3 états possibles :

☞ À l'état d'ouverture des portes : si on a une demande au RDC ***« aph(0) ='1' »***

☞ À l'état de montée :

- si on a un appel haut aux étages plus élevés « ***aph(i) ='1'***», envoyer l'indice du premier appel trouvé à partir du RDC
- si aucune demande haut est détecté, chercher les appels bas ou les sélections aux étages plus élevés

☞ À l'état d'attente : si aucun évènement est détecté

❖ ***S1 :*** Même principe de montée (expliqué dans la partie VI.3.5.1.3), sauf que les appels qui peuvent être servis immédiatement sont les sélections seulement.

❖ ***S2, S3:***Le même principe expliqué précédemment (la partie VI.3.5.1.3)

❖ ***S4 (état de déplacement vers le bas) :*** Le même principe de descente, sauf que lors de sa descente, l'ascenseur sert immédiatement tous les appels bas et les sélections détectées sur son trajet.

❖ ***S00 (état d'attente) :***C'est l'état de repos d'ascenseurs, à l'arrivé d'un nouvel évènement, et selon l'emplacement et le sens de déplacement d'ascenseur, on peut aller à :

☞ *A l'état S1* :

- si l'ascenseur est en déplacement vers le haut, et il y a des sélections aux étages plus hauts
- sinon chercher des appels hauts aux étages les plus élevés, et prendre le premier à partir du RDC
- si l'ascenseur est en déplacement vers le bas, et il ne reste que des demandes aux étages plus hauts

☞ *A l'état S4 :*

- si l'ascenseur est en déplacement vers le bas, et il y a des sélections ou des appels bas aux étages plus bas

☞ A l'état S2 d'ouverture des portes si la demande et à l'étage où la cabine se trouve

VI.3.5.3.3 Exemple de simulation

Pour assurer le bon fonctionnement de l'ascenseur avec le mode sélectif haut /collectif bas on a imaginé un scénario qui montre la stratégie de sélection des demandes à servir

Scénario :

- ☞ L'ascenseur est au RDC en état repos (la porte cabine et toutes les portes étages sont fermées)
- ☞ **Appel haut** au **RDC** est arrivé
  - ⇨ L'ascenseur ouvre les portes
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes

- ☞ Lors de la fermeture des portes le bouton d'ouvrir d'urgence est appuyé
  - ⇨ les portes s'ouvrent de nouveau
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes
- ☞ Les passagers à l'intérieur de la cabine veulent aller au **4ème et 8ème** étage
  - o La cabine commence à monter et quand elle atteint le 1ier étage un **appel haut** au **2ème** étage arrive
  - o La cabine monte par 3 étages (ne sert pas l'appel haut parce que il y a des sélections en attente)
    - ⇨ Les portes s'ouvrent pour servir la sélection (4ème étage)
    - ⇨ Attendre un délai pour que les passagers entrent
    - ⇨ La fermeture des portes, et l'arrivé d'un **appel bas** au **5ème** étage
  - o La cabine monte par 4 étages (il ne sert pas l'appel bas), et quand elle atteint le 6éme étage un **appel bas** au **8ème** étage est détecté
    - ⇨ Les portes s'ouvrent pour servir la sélection et l'appel bas (8ème étage)
    - ⇨ Attendre un délai pour que les passagers entrent
    - ⇨ La fermeture des portes
- ☞ Les passagers à l'intérieur de la cabine veulent aller au **3ème** étage
  - o La cabine descend 3 étages
    - ⇨ Les portes s'ouvrent pour servir l'appel bas (5ème étage)
    - ⇨ Attendre un délai pour que les passagers entrent
    - ⇨ La fermeture des portes

☞ Une autre **sélection** au **2ème** étage

- La cabine descendre par 2 étages
  - ⇨ Les portes s'ouvrent pour servir la première sélection (3ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes
- La cabine descend d'un étage
  - ⇨ Les portes s'ouvrent pour servir la deuxième sélection qui a en même temps un appel haut (2ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes

☞ Le résultat de simulation de ce scénario est interprété par la figure II.46

VI.3.5.4 Le circuit SH_SB

VI.3.5.4.1 Stratégie de fonctionnement

Le circuit ***SH_SB*** est une implémentation d'algorithme de dispatching : ***sélectif haut / sélectif bas***, l'ascenseur fonctionne en mode sélectif dans les deux sens, lorsqu'un appel est sélectionné, il ne sert aucun autre après avoir servir tous les sélections(les étages désirés sélectionné par les gens qui sont à l'intérieur de la cabine), de point de vue d'implémentation VHDL ça veut dire que :

- Si l'ascenseur est en déplacement vers le bas, il doit servir le premier appel bas détecté à partir du dernier étage, c'est l'indice le plus grand dans le vecteur **apb** qui contient un '1', puis servir tous les indices qui contient un '1' dans le vecteur **selt**
- Si l'ascenseur est en déplacement vers le haut, il doit servir le premier appel haut détecté à partir du RDC, c'est l'indice le plus petit dans le vecteur **aph** qui contient un '1', puis servir tous les indices qui contient un '1' dans le vecteur **selt**

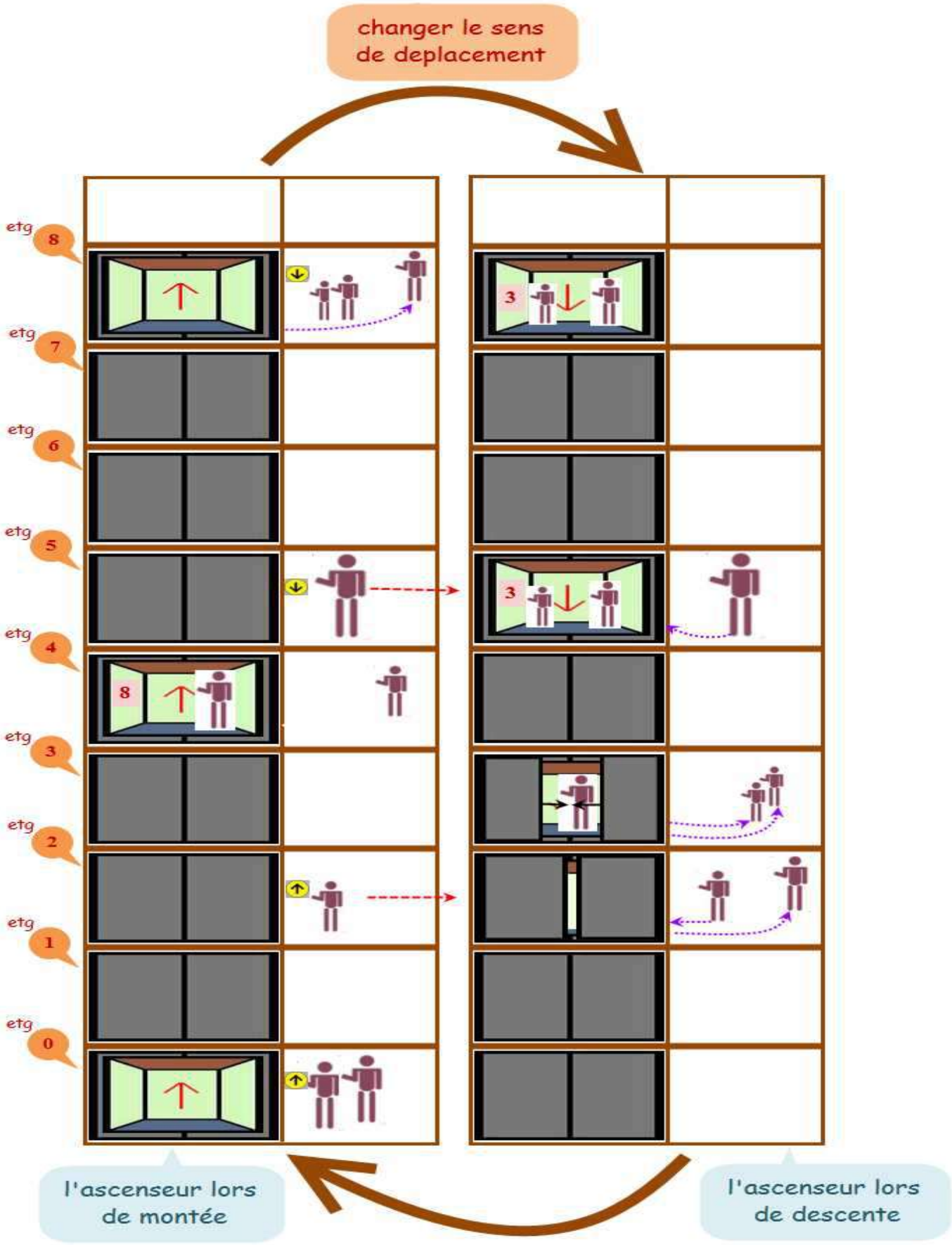


Fig II.45 interprétation du résultat de simulation du circuit SH_CB

VI.3.5.4.2 Machine d'état

Le fonctionnement du circuit SH_SB est illustré par la machine d'état représentée par la figure II.47 :

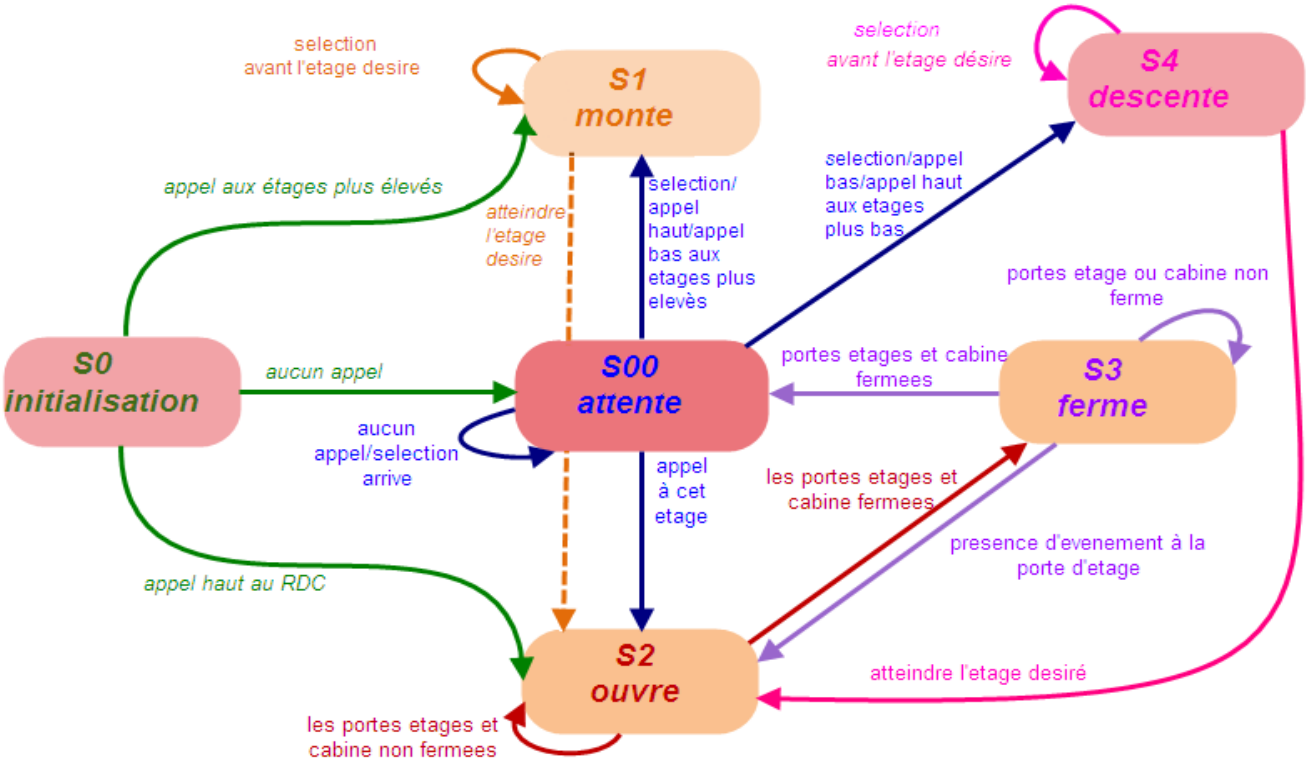


Fig II.46 machine d'état d'algorithme sélectif haut /sélectif bas

❖ ***S0 (état d'initialisation) :*** À partir de cet état et selon l'évènement détecté on peut aller aux 3 états possibles :

☞ À l'état d'ouverture des portes : si on a une demande au RDC ***« aph(0) ='1' »***

☞ À l'état de montée :

- si on a un appel haut aux étages plus élevés « ***aph(i) ='1'***», envoyer l'indice du premier appel trouvé à partir du RDC
- sinon chercher un appel bas et sélectionner le premier à partir le dernier étage

☞ À l'état d'attente : si aucun évènement n'est détecté

❖ ***S1, S4 :*** Même principe de monter et de descendre, sauf que l'ascenseur ne prend en compte que les sélections entre l'étage où la cabine se trouve, et l'étage d'appel à servir

❖ ***S2, S3:***Le même principe expliqué précédemment (la partie VI.3.5.1.3)

❖ ***S00 (état d'attente) :*** C'est l'état de repos d'ascenseurs, suivant l'évènement détecté il passe à un parmi les états suivant :

☞ *A l'état S1* :

- s'il y a des sélections aux étages plus hauts, en cas de déplacement vers le haut d'ascenseur
- sinon chercher des appels hauts aux étages les plus élevés et prendre le premier à partir du RDC
- sinon chercher des appels bas aux étages les plus élevés et prendre le premier à partir du dernier étage
- si l'ascenseur est en déplacement vers le bas et il ne reste aucune demande dans son trajet, au contraire de sens opposé (détection des demandes aux étages élevés), il doit changer son sens et choisir l'appel à servir de la même manière

☞ *A l'état S4 :*

- s'il y a des sélections aux étages plus bas, en cas de déplacement vers le bas d'ascenseur
- sinon chercher des appels bas aux étages les plus bas et prend le premier à partir du dernier étage
- sinon chercher des appels hauts aux étages plus bas et prend le premier à partir du RDC
- si l'ascenseur est en déplacement vers le haut et il ne reste aucune demande dans son trajet, au contraire de sens opposé (détection des demandes aux étages bas), il doit changer son sens et choisir l'appel à servir de la même manière

☞ A l'état S2 d'ouverture des portes si la demande et au l'étage où se trouve la cabine

VI.3.5.4.3 Exemple de simulation

Pour assurer le bon fonctionnement de l'ascenseur avec le mode ***sélectif haut / sélectif bas*** on a imaginé un scénario qui montre la stratégie de sélection des demandes à servir

Scénario :

☞ L'ascenseur est au RDC en état repos (la porte cabine et toutes les portes étages sont fermées)

☞ **Appel haut** au **RDC** est arrivé

⇨ L'ascenseur ouvre les portes

⇨ Attendre un délai pour que les passagers entrent

☞ Avant que le délai soit achevé un bouton de fermeture est appuyé

⇨ Les portes se ferment directement

☞ Les passagers à l'intérieur de la cabine veulent aller au **4ème et 8ème** étage

- La cabine commence à monter et quand elle atteint le 1ier étage un **appel haut** au **2ème** étage arrive
- La cabine monte par 3 étages (ne sert pas l'appel haut parce qu'il y a des sélections en attente)
  - ⇨ Les portes s'ouvrent pour servir la sélection (4ème étage)
  - ⇨ Attendre un délai pour que les passagers sortent
  - ⇨ La fermeture des portes, et l'arrivé d'un **appel bas** au **5ème étage**
- La cabine monte par 4 étages (il ne sert pas l'appel bas), et quand elle atteint le 6éme étage un **appel bas** au **8ème** étage est détecté
  - ⇨ Les portes s'ouvrent pour servir la sélection et l'appel bas (8ème étage)
  - ⇨ Attendre un délai pour que les passagers sortent
  - ⇨ La fermeture des portes

☞ Les passagers à l'intérieur de la cabine veulent aller au 3ème étage

- La cabine commence à descendre et quand elle atteint le 7ème étage une sélection au 6éme étage est détectée
  - ⇨ Les portes s'ouvrent pour servir la sélection (6ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes
- La cabine descendre par 3 étages (elle ne sert pas l'appel bas au 5ème étage)
  - ⇨ Les portes s'ouvrent pour servir la 2ème sélection (3ème étage)
  - ⇨ Attendre un délai pour que les passagers sortent
  - ⇨ La fermeture des portes
  - ⇨ La cabine descendre par 1 étage pour servir l'appel haut au 2ème étage, puis lors de sa descente elle sert l'appel bas du 5ème étage si elle est vide

☞ Le résultat de simulation de ce scénario est interprété par la figure II.48

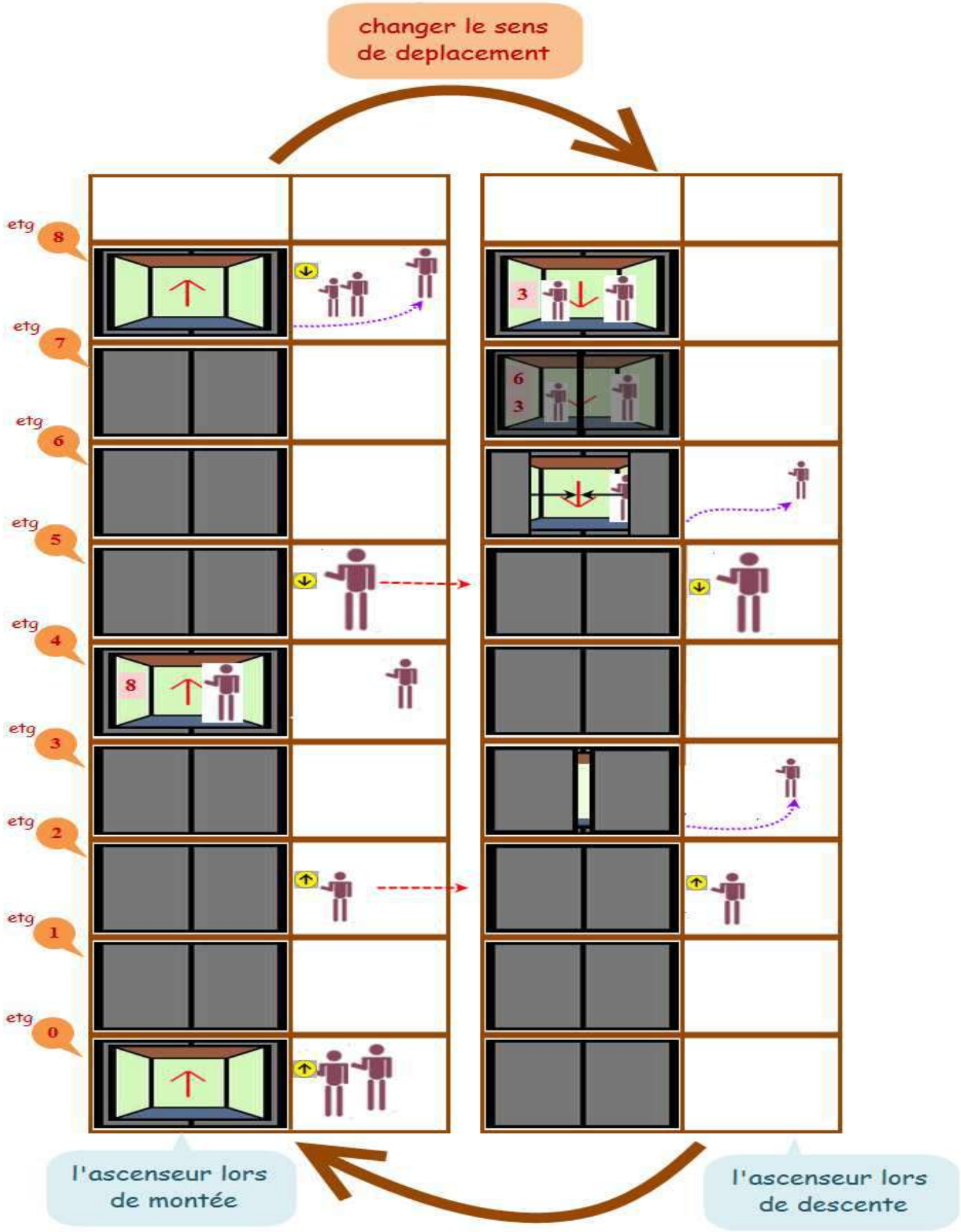


Fig II.47 interprétation du résultat de simulation du circuit Sh_SB

VI.3.5.5 Le circuit SC_HB

VI.3.5.5.1 Stratégie de fonctionnement

Le circuit **SC_HB** est une implémentation d'algorithme de dispatching ***sélectif collectif _haut bas***, l'ascenseur travail en mode sélectif pour servir le premier appel détecté a le même sens de son déplacement, puis en mode collectif pour terminer son trajet, de point de vue d'implémentation VHDL ça veut dire que :

- Si l'ascenseur est en déplacement vers le haut, il doit servir le premier appel haut détecté à partir du RDC, c'est l'indice le plus petit dans le vecteur **aph** qui contient un '1', puis il doit changer son mode au mode collectif pour servir tous les appels haut et les sélections aux étages qui ne sont pas encore dépassés, ce sont les '1' trouvés au niveau des deux vecteurs **aph** et **selt ,** dans les cases supérieures à l'indice de présence de la cabine d'ascenseur
- De même en cas de déplacement vers le bas, il doit servir le premier appel bas détecté à partir du dernier étage, c'est l'indice le plus grand dans le vecteur **apb** qui contient un '1', puis il doit changer son mode au mode collectif pour servir tous les appels bas et les sélections aux étages qui ne sont pas encore dépassés, ce sont les '1' trouvés au niveau des deux vecteurs **apb** et **selt ,** dans les cases inférieures à l'indice de présence de la cabine d'ascenseur .

VI.3.5.5.2 Machine d'état

Le fonctionnement du circuit SH_SB est illustré par la machine d'état représentée par la figure II.49 :

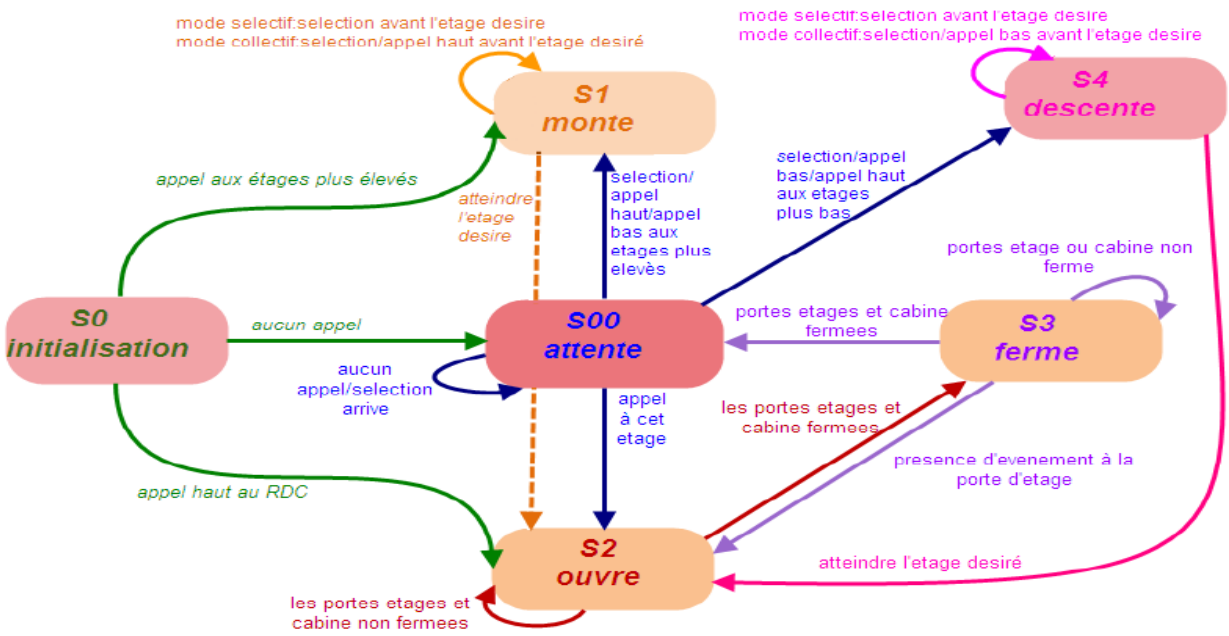


Fig II.48 machine d'état d'algorithme sélectif collectif/haut bas

❖ ***S0 (état d'initialisation)*** À partir de cet état et suivant l'évènement détecté on peut aller aux 3 états possibles suivants:

☞ À l'état d'ouverture des portes : si on a une demande au RDC ***« aph(0) ='1' »***

☞ À l'état de montée :

- si on a un appel haut aux étages plus élevés « ***aph(i) ='1' »*** (en envoyant l'indice du premier appel trouvé à partir du RDC)
- sinon chercher un appel bas et sélectionner le premier à partir le dernier étage

☞ À l'état d'attente : si aucun évènement est détecté

❖ ***S1, S4 :***Avant qu'il atteigne le premier appel choisi pour le servir, l'ascenseur doit ignorer tous les appels intermédiaires qui viennent lors de son déplacement, et quand il les sert ,il change sa stratégie du choix des appels intermédiaires à servir suivant son sens de déplacement , si il est en déplacement vers le haut il ne prend en compte que les sélections et les appels haut , contrairement au sens opposé qu'il doit servir tous les appels bas et les sélections arrivés dans son trajet avant de servir la demande sélectionnée

❖ ***S2, S3:***Le même principe expliqué précédemment (la partie VI.3.5.1.3)

❖ ***S00 (état d'attente) :***C'est l'état de repos d'ascenseurs, suivant l'évènement détecté il passe à un parmi les états suivant

☞ *A l'état S1* :

- s'il y a des sélections ou appels haut aux étages les plus hauts,

- sinon chercher des appels bas aux étages les plus élevés

☞ *A l'état S4 :*

- s'il y a des sélections ou appels bas aux étages plus bas,
- sinon chercher des appels hauts dans les étages bas

☞ A l'état S2 d'ouverture des portes si la demande et au l'étage ou la cabine se trouve

VI.3.5.5.3 Exemple de simulation

Pour assurer le bon fonctionnement de l'ascenseur avec le mode sélectif collectif/ haut bas on a imaginé un scénario qui montre la stratégie de sélection des demandes à servir

Scénario :

☞ L'ascenseur est au RDC en état repos (la porte cabine et toutes les portes étages sont fermées)

☞ **Appel haut** au **3ème** est arrivé

- La cabine commence à monter et quand elle atteint le 1ier étage un **appel haut** au **2ème étage** arrive
- Elle termine sa montée par 2 étages
  - ⇨ L'ascenseur ouvre les portes pour servir l'appel haut (3ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes

☞ Les passagers à l'intérieur de la cabine veulent aller au **4ème et 8ème** étage

- La cabine monte par 1 étage
  - ⇨ Les portes s'ouvrent pour servir la sélection (4ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes, et l'arrivé d'un **appel bas** au **5ème étage**
- La cabine commence à monter pour servir la deuxième sélection, et quand elle atteint le 5éme étage un **appel haut** au **6ème étage** est détecté
  - ⇨ L'ascenseur s'arrête et les portes s'ouvrent pour servir l'appel haut (6ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes
- La cabine monte par 2 autres étages pour servir la deuxième sélection étage
  - ⇨ Les portes s'ouvrent (8ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes

- La cabine commence à descendre pour servir l'appel bas du 5ème étage et quand elle atteint le 7ème étage un appel bas au 6ème étage est détecté arrive
- Elle termine sa descente par 2 étages en ignorant l'appel bas intermédiaire
  - ⇨ Les portes s'ouvrent pour servir l'appel bas (5ème étage)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes

☞ Les passagers à l'intérieur de la cabine veulent aller au **3ème** étage

- La cabine descendre par 2 étages
  - ⇨ Les portes s'ouvrent pour servir la sélection (3ème étage)
  - ⇨ Attendre un délai pour que les passagers sortent
  - ⇨ La fermeture des portes
  - ⇨ Aucune autre demande vers le bas n'est détectée, l'ascenseur doit servir les appels restés dans les sens opposé (haut) de la même manière

☞ Le résultat de simulation de ce scénario est interprété par la figure II.50

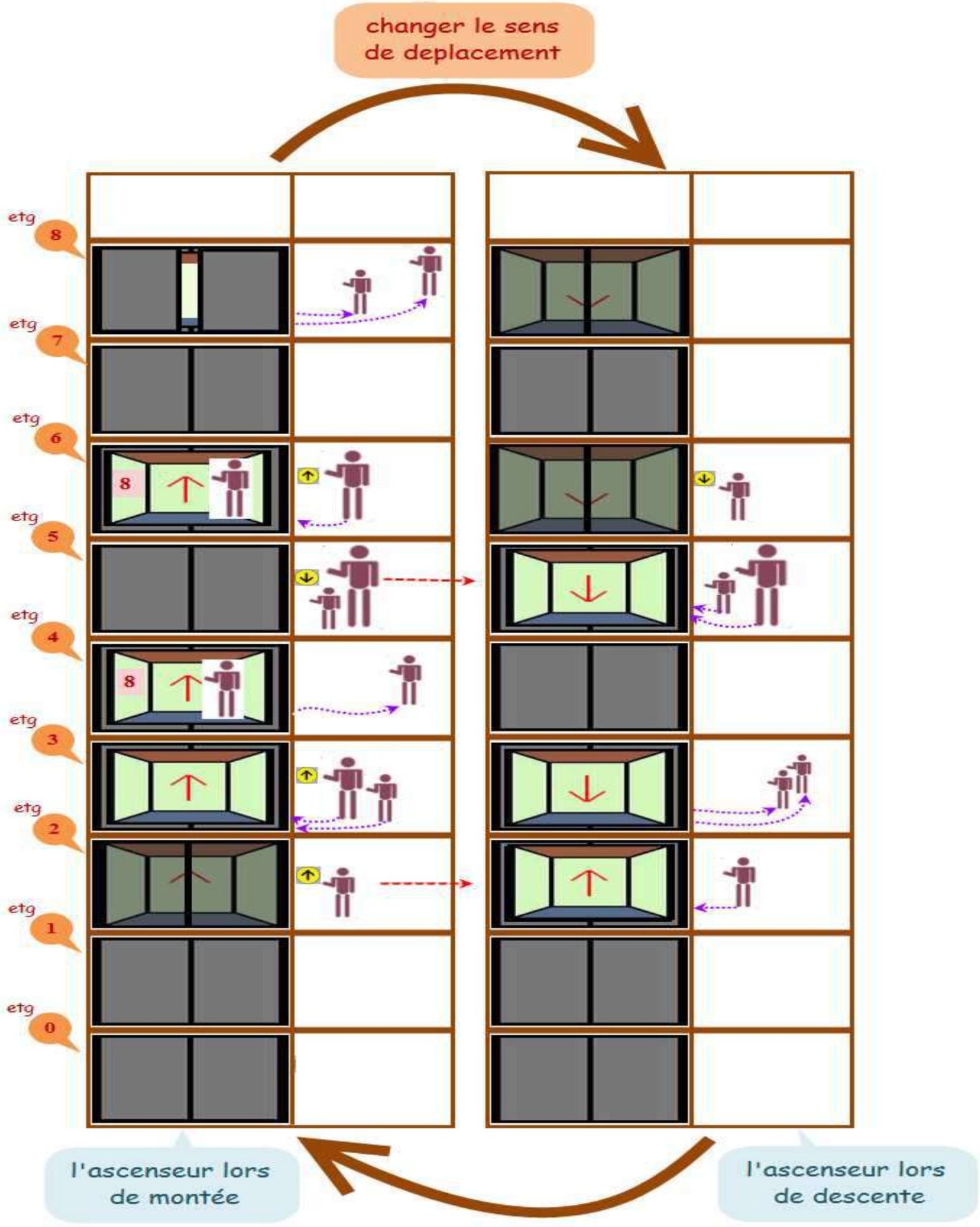


Fig II.49 interprétation du résultat de simulation du circuit SC_HB

VI.3.5.6 Le circuit ALG_SPL

VI.3.5.6.1 Stratégie de fonctionnement

Le circuit **ALG_SPL** est une implémentation d'algorithme de dispatching ***algorithme spécial*** construit pour traiter les demandes prioritaires. A la détection d'un appel spécial d'un responsable ou d'une personne handicapée, l'ascenseur doit la servir directement en ignorant toutes les autres demandes venues à l'extérieur de la cabine, il ne sert que les sélection effectuées à l'intérieur du cabine, afin de la vider le plutôt possible avant qu'il atteint l'étage ou l'appel prioritaire est effectué , et il ne prend en compte les appels externes qu'a après avoir amène la personne spéciale à l'étage désiré ,de point de vue implémentation VHDL ça veut dire que :

- Quand la demande spéciale est arrivée et quel que soit le mode et le sens de déplacement d'ascenseur, cet algorithme est activé par le SCL en donnant le numéro d'étage(***indice d'appel***) ou l'appel prioritaire est effectué, comparant avec l'étage où la cabine se trouve, l'ascenseur doit monter si l'appel situé aux étages plus élevés , ou descendre si il est aux étages plus bas
- Apres que l'ascenseur atteint l'étage de la personne spéciale il ne sert que les sélections venues de l'intérieur de la cabine, ce sont tous les ***'1'*** trouvés dans le vecteur **selt**
- Pendant le fonctionnement de cet algorithme, un signal doit être envoyé au système de gestion globale indiquant que cet ascenseur est réservé à un appel prioritaire pour ne pas le perturber « le signal ***servi*** », et après que la cabine est vidé et la personne spéciale est amené au étage désiré, ce signal est annulé pour remettre l'ascenseur au mode normal de fonctionnement (avec l'algorithme CH_CB).

VI.3.5.6.2 Machine d'état

Afin d'appliquer les principes expliqués précédemment et pour réaliser une implémentation VHDL de la stratégie proposée, on a deviné à construire la machine d'état illustrée dans la figure II.51 :

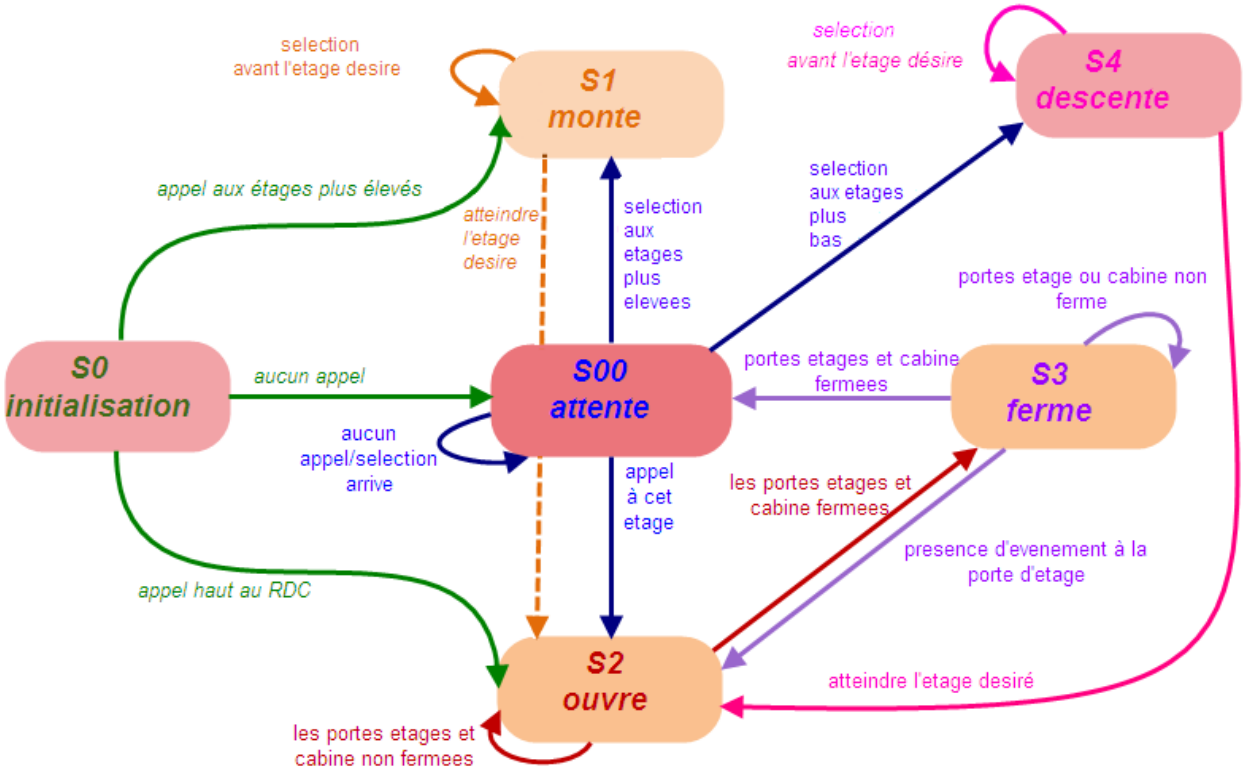


Fig II.50 machine d'état d'algorithme algorithme spécial

❖ ***S0 (état d'initialisation) :*** il est différent à tous les états S0 des autres algorithmes de dispatchings, il représente dans cet algorithme l'état de départ pour servir l'appel prioritaire, à partir l'indice d'appel spécial envoyé par le système de contrôle local d'ascenseur ,en comparant avec l'étage d'existence de la cabine , la décision est pris comme suivant :

☞ À l'état d'ouverture des portes : si on a une demande au RDC ***(aph(0) ='1')***

☞ À l'état de montée : si l'étage d'appel spécial > étage de la cabine

☞ À l'état de descente : si l'étage d'appel spécial < étage de la cabine

❖ ***S1, S4 :***L'ascenseur doit vider la cabine le plutôt possible avant qu'il atteint l'étage du personne spécial, c'est pour ça, lors de sa montée ou sa descente, il ne sert que les demandes des gens qui sont à l'intérieur de la cabine, et qu'ils veulent aller aux étages situés entre l'étage de la cabine et l'étage d'appel spécial

❖ ***S2, S3:***Le même principe expliqué précédemment (la partie VI.3.5.1.3)

❖ ***S00 (état d'attente) :*** Dans cet état, l'ascenseur chargé de servir les sélections pour vider la cabine afin de servir l'appel spécial. De même, après qu'il sert l'appel spécial il s'occupe d'amener la personne au étage désiré, c'est bien qu'on peut aller à partir cette état à :

☞ *A l'état S1* :

- s'il y a des sélections aux étages plus élevés avant qu'il atteint l'étage d'appel spécial, ou bien si la personne spéciale qui est à l'intérieur de la cabine veut aller à un étage plus haut

☞ *A l'état S4 :*

- s'il y a des sélections aux étages plus bas avant qu'il atteint l'étage d'appel spécial, ou bien si la personne spéciale qui est à l'intérieur de la cabine veut aller à un étage plus bas

VI.3.5.6.3 Exemple de simulation

Pour assurer le bon fonctionnement de l'ascenseur avec le mode sélectif collectif/ haut bas on a imaginé un scénario qui montre sa stratégie de sélection des demandes à servir

Scénario :

☞ On suppose que l'ascenseur fonctionne en mode CH_CB, et la cabine est au 3éme étage

☞ Un appel bas d'un responsable est détecté au 8ème étage, avec un autre appel bas simple détecté au 5ème étage

- Le mode d'ascenseur est changé directement par le SCL au ALG_SPL, en envoyant l'indice d'appel spécial=8 et le type=1 (appel bas), le signal « servi='1' » est envoyé au system global;
- L'ascenseur commence à monter en ignorant toutes les demandes externes (ne sert pas l'appel de 5ème étage), et quand elle atteint le 5^{ier} étage une **sélections** au **6ème** étage arrive
- il termine sa montée par 1 étage
  - ⇨ L'ascenseur ouvre les portes pour servir la sélection (6ème étage)
  - ⇨ Attendre un délai pour que les passagers sortent
  - ⇨ La fermeture des portes
- il termine sa montée par 2 étages
  - ⇨ L'ascenseur ouvre les portes pour servir l'appel du responsable (8ème étage)
  - ⇨ Attendre un délai pour qu'il entre
  - ⇨ La fermeture des portes

☞ Le responsable veut aller au RDC

- L'ascenseur commence à descendre au RDC, et quand il atteint le 4ème étage un ***appel bas*** au ***2ème*** étage est détecté
- il termine sa descente par 4 étages (il ne sert pas l'appel bas du 2ème étage)
  - ⇨ L'ascenseur ouvre les portes pour servir la sélection du directeur (RDC)
  - ⇨ Attendre un délai pour que les passagers entrent
  - ⇨ La fermeture des portes

☞ L'ascenseur termine de servir l'appel spécial, l'envoie du signal ***servi*** est arrêté, et le SCL remettre l'ascenseur au mode CH_CB pour servir les appels en attente (appel au 5ème étage, appel bas 2ème étage)

☞ Le résultat de simulation de ce scénario est interprété par la figure II.52

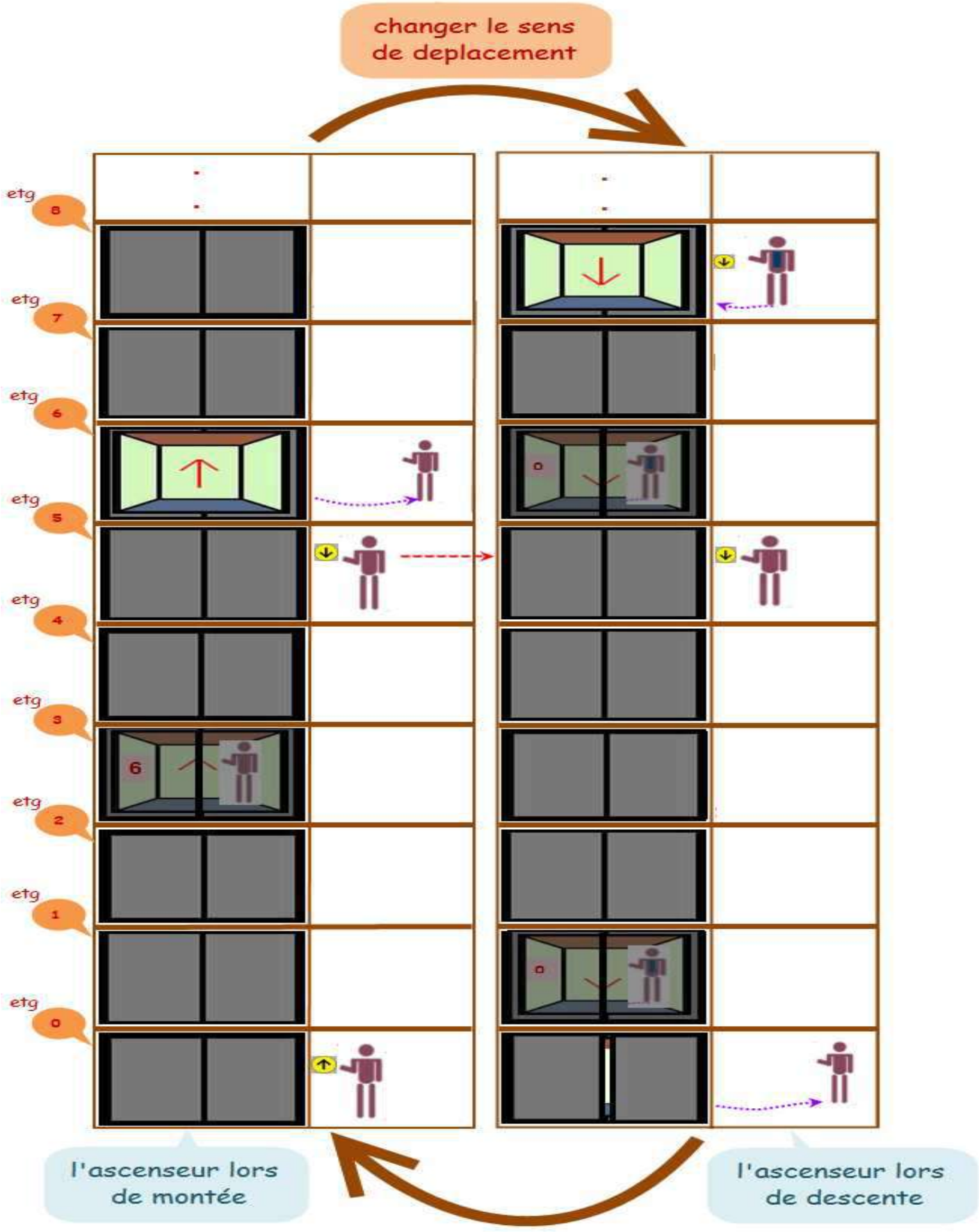


Fig II.51 interprétation du résultat de simulation du circuit ALG_SPL

VI.3.6 Exemple de simulation du circuit SCL

Dans les parties précédentes, nous avons démontré le bon fonctionnement d'ascenseur avec tous les algorithmes de dispatching proposés, il suffit donc d'assurer l'absence du conflit et sa bonne marche en cas du changement d'algorithme lors de son travail

Scénario : dans le même immeuble de 20 étages avec 4 ascenseurs, on va essayer de montrer la bonne marche d'ascenseur 1 avec l'algorithme de dispatching ***CH_CB***, puis changer le au ***ALG_SPL*** lors de la détection d'une demande prioritaire d'une personne handicapée, et après qu'il l'amène au étage désiré, le mode de travail d'ascenseur change au algorithme ***SH_CB***, pour servir les demandes en attente :

- ☞ L'ascenseur est en état repos, la cabine se trouve au RDC, toutes les portes des étages et de la cabine sont fermées
- ☞ Après 2 secondes un **appel haut** au **2ème** étage est arrivé (sélectionner l'algorithme ***CH_CB***)
  - o L'ascenseur monte 2 étages (pour servir l'appel haut)
    - ⇨ Ouverture des portes (2ème étage)
    - ⇨ Attendre un délai pour l'entrée des passagers
    - ⇨ Fermeture des portes (2ème étage)

- ☞ Les passagers à l'intérieur de la cabine veulent aller au **3ème** étage, et en même temps un **appel haut** d'une **personne handicapée** est détecté au **5ème** étage (changer le mode de fonctionnement d'ascenseur au ***Alg_spl***)
  - o L'ascenseur monte par 1 étage (pour servir la sélection et vider la cabine)
    - ⇨ Ouverture des portes (3ème étage), un **appel bas** au **4ème** étage est arrivé
    - ⇨ Attendre un délai pour la sortie des passagers
    - ⇨ Fermeture des portes (3ème étage)
  - o L'ascenseur monte de 2 étages pour servir l'appel de la personne handicapée
    - ⇨ Ouverture des portes (5ème étage)
    - ⇨ Attendre un délai pour l'entrée de la personne
    - ⇨ Fermeture des portes (5ème étage)
- ☞ La personne handicapée veut aller au **6ème** étage
  - o L'ascenseur monte par 1 étage (pour servir la sélection)
    - ⇨ Ouverture des portes (6ème étage)
    - ⇨ Attendre un délai pour la sortie de la personne

⇨ Fermeture des portes (6ème étage)

☞ L'ascenseur est servi la demande prioritaire, son mode de fonctionnement est changé au ***SH_CB*** pour servir les demandes en attentes

- L'ascenseur descend de 2 étages (pour servir l'appel bas du 4ème étage)
  - ⇨ Ouverture des portes (4ème étage)
  - ⇨ Attendre un délai pour l'entrée des passagers
  - ⇨ Fermeture des portes (4ème étage)

☞ Les passagers à l'intérieur de la cabine veulent aller au RDC

- L'ascenseur descente par 4 étages (pour servir la sélection)
  - ⇨ Ouverture des portes (RDC)
  - ⇨ Attendre un délai pour la sortie des passagers
  - ⇨ Fermeture des portes (RDC)

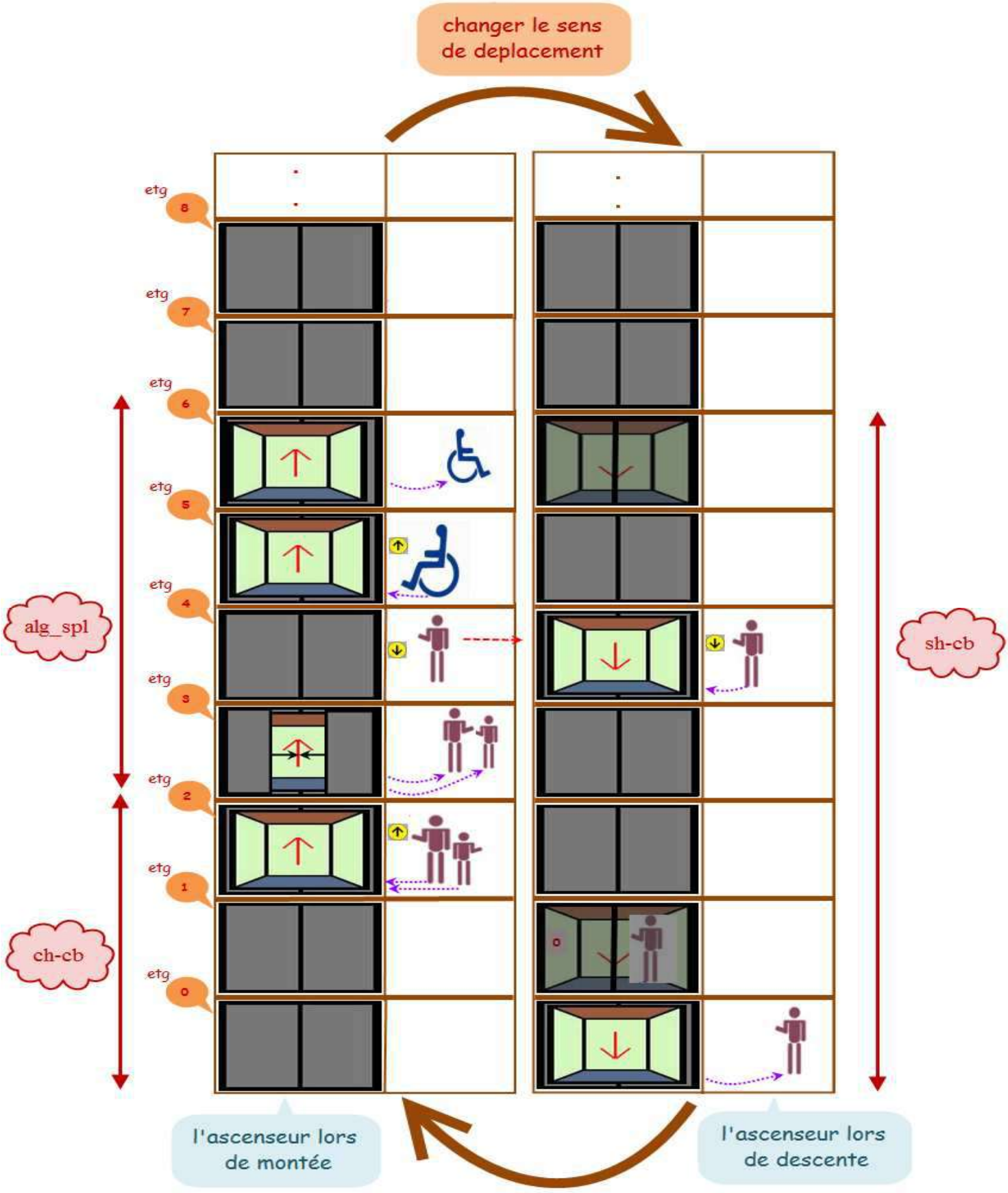


Fig II.52 interprétation du résultat de simulation du circuit SCL

Remarque : lors du changement d'algorithme pendant le fonctionnement de l'ascenseur, le nouveau algorithme doit terminer à partir l'état ou le précédant s'arrête, par exemple si le changement du mode se produit après l'ouverture des portes, le nouveau algorithme doit terminer la fermeture avant qu'il décide la demande suivante à servir

V. Exemple de simulation du système global

Apres la confirmation que le system de contrôle du groupe d'ascenseur SCGA, et les systèmes de contrôle locaux SCL's réagissent d'une manière juste et en temps réel, il ne reste que l'assurance de la collaboration entre eux, et l'absence du conflit d'accès aux mémoires communes, afin de gérer correctement et d'une manière intelligente n'importe quel groupe d'ascenseur dans un immeuble à N niveaux, pour but de diminuer le plus possible le temps d'attende des passager l'arrivée des ascenseurs.

Pour cela, et à cause de la difficulté du programmation d'un fichier de simulation introduit par l'obligation de suivre tous les capteurs repartie dans l'environnement avant et après chaque commande envoyé par le système de contrôle global , nous avons pensé à programmer 2 scénarios différents qui montrent la bonne marche des ascenseurs dans des différentes périodes et des diverses situations .Comme on a fait avec tous les scénarios précédant, on va citer la réaction du programme pas à pas ,suivie par une interprétation des résultats avec des images imitent ce qui se passe en réalité (exemple d'immeuble à 20 étages avec 4 ascenseurs)

Scenario 1 :

- ***Mardi à 9H du matin*** : Les 4 ascenseurs sont au RDC, toutes les portes cabines et les portes des étages sont fermées
- ***à 9:05*** des personnes au **RDC** demandent l'ascenseur
  - le system choisi ***l'ascenseur 1*** avec l'algorithme ***collectif haut/sélectif bas***
- les gens à l'intérieur de la cabine1 veulent aller au **4ème** et **8ème** étage
  - l'ascenseur1 commence à monter par 4 étages pour servir la première sélection
- quand l'ascenseur1 atteint le 2ème étage, un **appel haut** est détecté au **1ier** étage
  - le system choisi ***l'ascenseur2*** avec l'algorithme ***collectif haut/sélectif bas*** pour servir l'appel haut du 1ier étage
- Lors de la fermeture des portes (d'ascensuer1 au 4éme étage et l'ascensur2 au 1ier étage), un **appel bas** au **5ème** étage est arrivé, et la personne dans l'ascenseur 2 veut aller au **3ème** étage

- Le system choisi ***l'ascenseur3*** avec l'algorithme ***collectif haut/collectif bas*** pour servir l'appel bas du 5ème étage, et l'ascenseur3 commence à monter par 5 étages
- L'ascenseur 1 et l'ascenseur3 termine leur montée afin de servir les sélections du 8ème étage et 3ème étage successivement

☞ Les gens dans la cabine d'ascenseur3 veulent descendre au **3ème** étage

- Les autres ascenseurs (1,3) sont en état d'attente après la terminaison de toutes les demandes

☞ Quand l'ascenseur3 atteint le 4éme étage (dans son trajet au 3ème étage) un **appel bas** au **7ème** étage est arrivé

- Le system choisi ***l'ascenseur1*** avec l'algorithme ***collectif haut/collectif bas*** pour servir l'appel bas du 7ème étage, il descend par un étage pour le servir

☞ La personne du 7ème étage veut aller au **RDC**

- L'ascenseur descend 7 étages pour servir la sélection

☞ Aucun nouvel évènement détecté, tous les ascenseurs sont en l'état d'attente

Remarque :

L'interprétation du résultat de simulation du système globale avec le scénario proposé est imité dans l'image du figure II.54, la séquence des évènements est citée par des numérotations croissance (exp : le premier évènement est l'arrive d'appel RDC numéroté par 1)

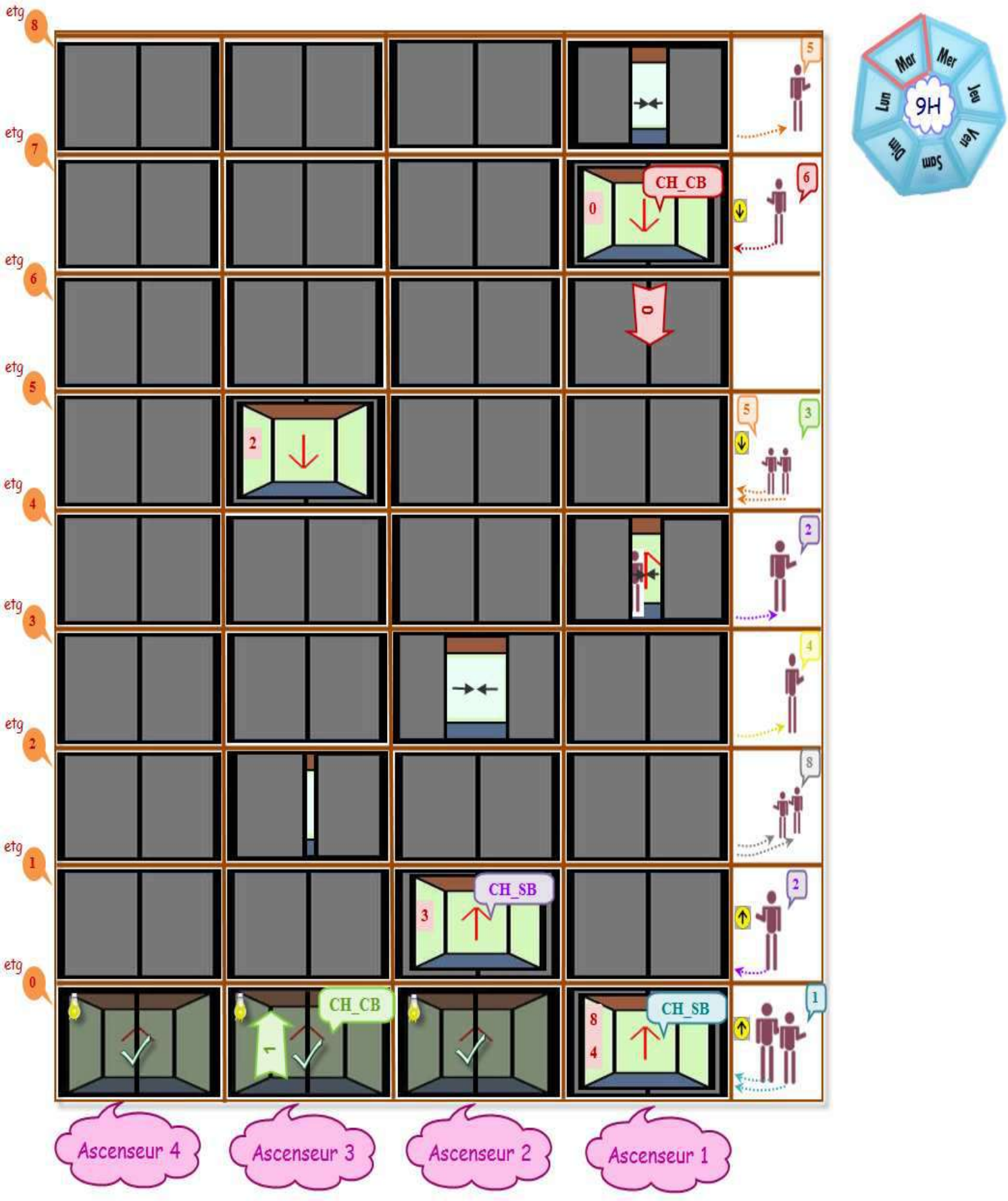


Fig II.53 interprétation du résultat de simulation du system de contrôle avec le scénario 1

Scenario 2 :

☞ ***Jeudi à 6H du matin*** : Les 4 ascenseurs sont au RDC, seulement l'ascenseur 1 qui fonctionne, toutes les portes cabines et les portes des étages sont fermées

☞ ***à 6:10*** un **appel haut** au **1ier** étage est arrivé

- le system choisi « ***l'ascenseur 1 »*** avec l'algorithme « ***sélectif collectif/haut bas »***

☞ l'ascenseur1 commence à monter par 1 étage et quand la cabine atteint le 1ier étage, un autre **appel haut** au **4ème** étage est détecté

- le system choisi le même ***ascenseur (***le seul ascenseur qui fonctionne***)*** avec le même ***algorithme de dispatching***

☞ la personne (du 1ier étage) à l'intérieur de la cabine veut aller au **6ème** étage

- l'ascenseur commence à monter par 3étages pour servir l'appel haut du 4ème étage dans son chemin
- l'appel haut est servi, la deuxième personne veut aller au **7ème** étage
- l'ascenseur termine sa montée pour servir la première sélection (6ème étage) puis monte un autre étage pour servir la 2ème sélection (7ème étage)

☞ ***à 13H*** : tous les ascenseurs sont disponibles, l'ascenseur 1 est au 7ème étage et les autres sont au RDC

☞ Un appel bas d'une personne handicapé est arrivé au **8ème** étage arrivé

- le system choisi « ***l'ascenseur1 »*** avec l'algorithme « ***algorithme spécial »***
- l'ascenseur monte par 1 étage pour servir la demande

☞ le handicapé à l'intérieur de la cabine 1 veut aller au **RDC**

- l'ascenseur commence à descendre pour l'amener à l'étage désiré

☞ quand l'ascenseur1 atteint le 6ème étage, **3 appel haut** (**RDC**, **2ème** étage, **5ème** étage) et un **appel bas** au **3ème** étage sont arrivés au même temps

- pour ***l'appel haut du RDC*** : le system choisi « ***l'ascenseur2 »*** avec l'algorithme « ***collectif haut/sélectif bas»***
- pour ***l'appel haut du 2ème étage*** le system choisi « ***l'ascenseur3 »*** avec l'algorithme « ***collectif haut/sélectif bas»***
- pour ***l'appel haut du 5ème étage*** le system choisi « ***l'ascenseur1 »*** et il le laisse fonctionner avec le même algorithme

- pour ***l'appel bas du 3ème étage*** le system choisi « ***l'ascenseur4 »***avec l'algorithme « ***sélectif haut/collectif bas»***
- les portes d'ascenseur 2 s'ouvrent pour servir l'appel du RDC
- l'ascenseur 3 commence à monter afin de servir l'appel du 2ème étage
- l'ascenseur 4 commence à monter pour atteint l'étage 3 d'appel bas

☞ les gens du RDC (à l'intérieur de la cabine2) veulent aller au **1ier** et **4ème** étage

- l'ascenseur 2 monte par un étage est sert la première sélection
- l'ascenseur 1 amène le responsable au RDC et passe en état d'attente
- l'ascenseur 3 atteint le 2ème étage et sert l'appel haut
- l'ascenseur 4 atteint le 3ème étage et sert l'appel bas

☞ la personne du 2ème étage (à l'intérieur de la cabine3) veut monter au 4ème étage

- l'ascenseur 2 termine sa montée afin de servir la deuxième sélection du 4ème étage

☞ les gens du 3ème étage (à l'intérieur de la cabine4) veulent descendre au RDC

- l'ascenseur 3 commence à monter par 2 étages
- l'ascenseur 4 commence à descendre par 3 étages

☞ les ascenseurs amènent les passagers aux étages désirés ; aucune autre demande n'est détectée, ils passent en état d'attente (ascenseur 1 => RDC, ascenseur2=> 7ème étage, ascenseur 3 => 4ème étage, ascenseur 4=>RDC).

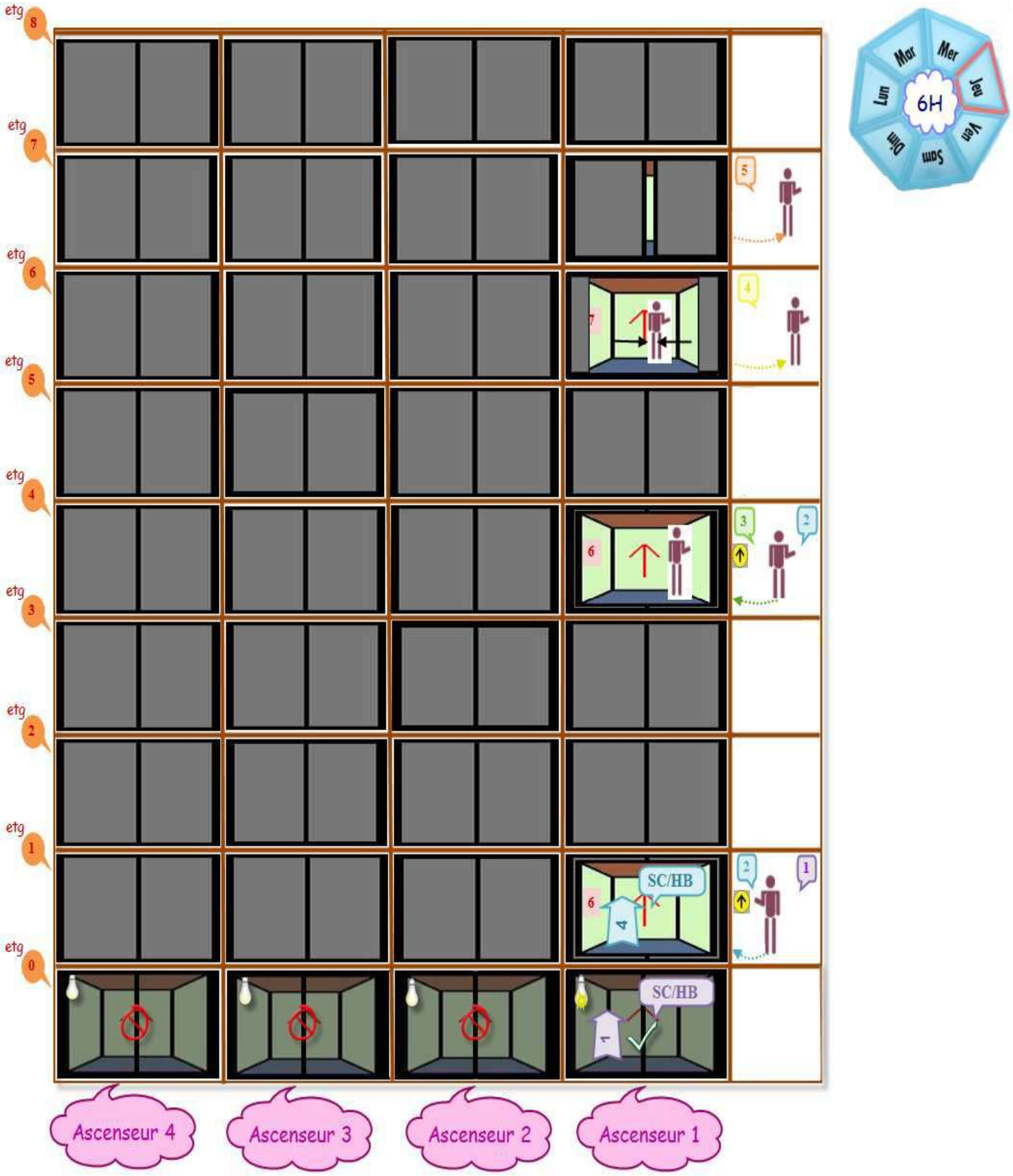


Fig II.54 interprétation du résultat de simulation du system de gestion avec le scénario 2 (partie1)

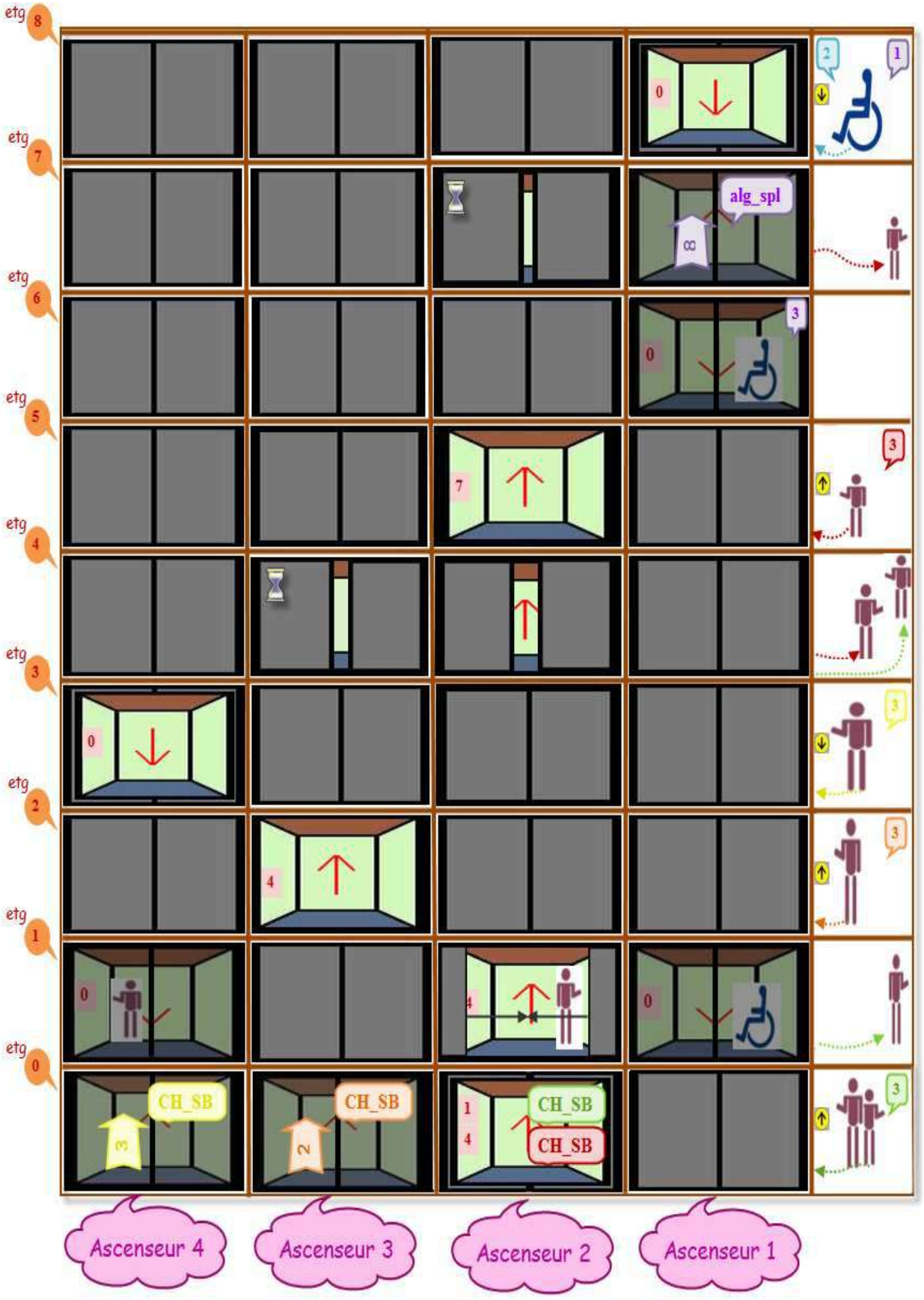




Fig II.55 interprétation du résultat de simulation du system de gestion avec le scénario 2 (partie2)

VI. Conclusion

Dans ce chapitre, on a proposé une architecture du système de gestion d'un groupe d'ascenseurs permet d'accomplir tous ces objectif, ce système qui compose d'un SCGA et des SCL's, réagit correctement et en temps réel avec tous les évènements détectés dans son environnement, il peut distinguer les appels des personnes importants et handicapés et les servir directement, et à partir ses stratégie de choix intelligentes il minimise le TAM des usagers et la consommation énergétique des ascenseurs. Après une implémentation VHDL de l'architecture proposée, des tests et des simulations ont montré la validité du système intelligent proposé.

Conclusion Générale

Conclusion générale

La nouvelle technologie des cartes dédiées à la gestion intelligente du trafic rend les ascenseurs plus fonctionnels, plus rapides, plus performant et moins consommant l'énergie. Dans ce projet de fin d'études, on a proposé une architecture d'un système intelligent autonome, capable de gérer M ascenseurs dans un immeuble à N niveaux, d'une manière juste et en temps réel, il réagit selon des stratégies qui diminuent le nombre de déplacement des cabines le plutôt possible afin d'optimiser le temps d'attente des usagers et la consommation énergétique des ascenseurs, il permet de :

- Activer un seul ascenseur dans les moments du faible trafic comme les weekends et les nuits
- Utilisation de 6 algorithmes de dispatching différents, sélectionnés suivant le trafic et la situation des appels à servir
- prendre en compte toutes les évolutions industrielles intégrées dans les ascenseurs : le mécanisme de coulissement des portes, les capteurs des empêchements des fermetures, les capteurs de surcharge des cabines, les boutons d'urgence, et surtout la nouvelle technologie des capteurs des cartes magnétiques pour les responsables et les personnes handicapées , afin de leur servir directement avec une stratégie spéciale .
- Perfectionner le système de sécurité par des vérifications effectuées avant chaque commande envoyée par exemple : la montée des cabines ne peut être réalisée qu'après la confirmation des fermetures de toutes les portes étages et les portes cabines
- La rapidité de la réaction du système de choix d'ascenseur et d'algorithme de dispatching approprié pour une demande détectée, qui ne consomme qu'un cycle d'horloge, et ses effet importants sur la rapidité du système de gestion global, et la réalisation de la notion du « temps réel »

Afin d'implémenter ces notions d'intelligence et de la rapidité du system de gestion complexe avec un langage machine de haut niveau « VHDL », on a rencontré plusieurs difficultés, et on n'a pas pu les résoudre facilement:

- ☞ Trouver une architecture du système (système de contrôle d'un groupe d'ascenseur SCGA) contenant plusieurs modules qui fonctionnent en parallèle pour donner le numéro d'ascenseur et le numéro d'algorithme le plus approprié à un appel détecté, et qu'il doit réagir dans un cycle d'horloge afin de conserver la notion du « temps réel »

- ☞ Le problème d'accès à la mémoire commune, contenant l'état d'ascenseur, entre le système SCGA et le system SCL, et la possibilité d'apparition du conflit lors de la modification en parallèle
- ☞ Le changement dynamique et souple d'algorithme de dispatching dans n'importe quel moment sans provoquer un blocage du système
- ☞ Penser toujours à une architecture générale pour que notre système soit adaptable à n'importe quel groupe d'ascenseur (n'importe quel nombre d'étages avec n'importe quel nombre de cabines).

Finalement, ce projet a été très bénéfique où il nous a donné l'occasion à toucher essentiellement à deux domaines très intéressant : avoir une idée détaillée sur le contrôle des moyens de transport vertical, et développer des architectures intelligentes et complexes de gestion de trafic et les adapter à la programmation VHDL

Comme perspective à notre travail, on propose pour le futur la synthèse de notre système afin de l'intégrer dans une carte FPGA (faute de temps, on n'a pas pu l'avoir) et de prendre en compte :

- ✓ la nouvelle approche inventée aujourd'hui concernant la sélection des étages désirés à l'instant de l'affectation de l'appel d'un ascenseur, qui permet l'intégration de la gestion de la destination à notre system, afin de regrouper les usagers ayant la même destination, et donc de réduire le nombre d'arrêts des cabines. Ceci a pour conséquence, pour l'usager, d'atteindre encore beaucoup plus rapidement son étage de destination
- ✓ d'intégrer les nouvelles propositions des systèmes SCGA basant sur la logique floue et les réseaux de neurone afin d'augmenter l'intelligence de ses stratégies de choix
- ✓ augmenter la précision de choix des algorithmes de dispatching en prenant en compte autres critères comme le flux dans chaque étage et le flux général d'immeuble détecté à partir des capteurs spéciaux

Références Bibliographiques

Références bibliographiques

[1] BERNARD QUIGNARD, « Guide pratique ascenseur et sécurité », (EYROLLES), 2éme édition, 2000.

[2] M.SAIDI & R.RIHANI, « Commande par FPGA d'un moteur pas à pas » mémoire d'ingénieur, Institut Supérieur des Etudes Technologique de Nabeul, Tunisie, Juin 2006.

[3] PFE «Etude d'un ascenseur commandé par automate programmable » Université sidi Mohammed ben Abdallah, Fès, Maroc, 2006/2007.

[4] R.ABLOUI & Y.YAHIAOUI & M.MEZHOUD, PFE «Etude d'un ascenseur avec la réalisation de son armoire » Université YAHIA FARES de MEDEA, octobre 1994.

[5] M.FEKHAR, PFE «Implémentation d'un algorithme de contrôle d'un ascenseur R+3 sur circuit FPGA » Université YAHIA FARES de MEDEA, 2010/2011.

[6] Clive "Max" Maxfield, The Design Warrior's Guide to FPGA,Elsevier and Newnes,2004.

[7] R.PATJOSHI, «Design and implementation of embedded based elevator control system »Masterof Technology in VLSI Design and Embedded System,Rourkela 2010

[8] Site wed : www.ascenseur-eco.fr , « Système de gestion environnementalDurable aujourd'hui et demain», publié le 2/3/2010, dernière consultation:01/09/2012.

[9] Site web : http://www.ascenseurs.fr , «L'Ascenseur : une longue histoire», dernière consultation:01/09/2012.

Annexe

Annexe

1- Les pseudos algorithme des algorithmes de dispatching

1-1 Collectif haut/ Collectif bas

1- en alimentation de l'ascenseur la cabine est au RDC :

- S'il y a un appel haut au RDC alors **servie** l'appel **(ouverture et fermeture)**.
- Sinon : chercher s'il y a des appels hauts ou Bas aux étages plus haut :

  ⇨ si oui **monté** la cabine vers l'étage d'appel (aller à l'étape 2).

  ⇨ sinon rester en **attente** au RDC (aller à l'étape 4)**.**

2- Déplacement vers le haut:

a. S'il y a une sélection ou un appel Haut dans cet étage alors arrêteet servie la demande **(ouverture et fermeture)** puis retourner à l'état d'**attente** (l'étape 4).

b. Sinon tester s'il y a des sélections ou des appels hauts ou des appels bas dans les étages les plus élevés :

   ⇨ si oui retourner à l'étape (2).

   ⇨ Sinon aller à c de 2.

c. tester s'il y a des sélections ou des appels hauts ou des appels bas dans les étages les plus bas :

   ⇨ si oui **descendre** et aller à l'étape 3.

   ⇨ sinon, rester en **attente** à cet étage (aller à l'étape 4)**..**

3- Déplacement vers le bas:

a. S'il y a une sélection ou un appel Bas dans cet étage alors arrête et **servie** la demande **(ouverture et fermeture)** puis retourner à l'état d'**attente** (l'étape 4)**.**

b. Sinon tester s'il y a des sélections, des appels dans les étages plus bas.

   ⇨ si oui retourner à l'étape 3.

   ⇨ sinon aller à c de 3.

c. cherche s'il y a des sélections ou des appels aux étages plus hauts :

   ⇨ **si oui monte** et aller à l'étape 2.

   ⇨ sinon, rester en **attente** à cet étage (aller à l'étape 4)**.**

4- Rester en attente :

a. S'il y un appel haut ou bas aux étage plus élevés aller à l'étape 2.

b. Sinon : S'il y a un appel haut ou bas aux étages plus bas aller à l'étape 3.

c. Sinon aller à l'étape 4.

1-2 Collectif haut/ Sélectif bas

1. **en alimentation de l'ascenseur la cabine est au RDC :**
   - S'il y a un appel haut au RDC alors **servie** l'appel **(ouverture et fermeture)**.
   - Sinon : cherche s'il y a des appels hauts ou Bas aux étages plus haut :
     ⇨ si oui **monté** la cabine vers l'étage d'appel (aller à l'étape 2).
     ⇨ sinon resté en **attente** au RDC (aller àl'étape 4)**.**
2. **déplacement vers le haut :**
   a. S'il y a une sélection ou un appel Haut dans cet étage alors arrête et **servie** la demande puis retourner à l'état d'**attente** (aller à l'étape 4).
   b. Sinon tester s'il y a des sélections ou des appels hauts ou des appels bas dans les étages les plus élevés :
   ⇨ si oui retourner à l'étape (2).
   ⇨ Sinon aller à l'étape 2.
   c. tester s'il y a des sélections, des appels bas ou des appels hauts ou dans les étages les plus bas :
   ⇨ si oui **descente** (aller à l'étape 3).
   ⇨ sinon, rester en **attente** à cet étage (aller à l'étape 4)**.** .
3. **déplacement vers le bas :**
   a. Testé s'il y a des sélections dans les étages plus bas et avant l'étage d'appel choisit àservir.
   ⇨ Si oui : **descente** à l'étage de la nouvelle sélection trouvée (retourné à l'étape 3).
   ⇨ Sinon :
   ✓ si l'ascenseur atteint l'étage désiré, alors servie la demande puis aller à (b de 3).
   ✓ Sinon reste à l'étape 3.
   b. Cherche s'il y a des appels bas aux étages plus bas :
   ⇨ si oui **descente** à l'étage d'appel (retourné à l'étape 3).
   ⇨ sinon aller à c de 3.
   c. Cherche s'il y a des appels hauts aux étages plus bas :
   ⇨ si oui **descente** à l'étage d'appel (retourné à l'étape 3).
   ⇨ Sinon aller à d de 3**.**
   d. Cherche s'il y a des appels hauts ou bas aux étages plus élevés.
   ⇨ Si oui monte à l'étage d'appel (aller à l'étape 2).
   ⇨ Sinon resté en **attente** (aller à l'étape 4).

4. **Rester en attente :**

   a. S'il y un appel haut ou bas aux étage plus élevés aller à l'étape 2.

   b. Sinon **a**, S'il y a un appel haut ou bas aux étages plus bas, aller à l'étape 3.

   c. Sinon resté à l'étape 4.

1-3 Sélectif haut/ Collectif bas

1. **en alimentation de l'ascenseur la cabine est au RDC :**

   - S'il y a un appel haut au RDC alors **servie** l'appel **(ouverture et fermeture)**.
   - Sinon : chercher s'il y a des appels hauts ou Bas aux étages plus haut :
     - ⇨ si oui **monté** la cabine vers l'étage d'appel (aller à l'étape 2).
     - ⇨ sinon resté en **attente** au RDC (aller à l'étape 4**).**

2. **déplacement vers le haut :**

   a. Testé s'il y a des sélections dans les étages plus haut et avant l'étage d'appel choisit à servir.
     - ⇨ Si oui **monte**à l'étage de la nouvelle sélection trouvée (retourné à l'étape 2).
     - ⇨ Sinon :
       - ✓ si l'ascenseur atteint l'étage désiré, alors servie la demande puis retourner à (b de 2).
       - ✓ Sinon retourné à l'étape 2.

   b. Cherche s'il y a des appels hauts aux étages plus élevées :
     - ⇨ si oui **monte** (aller à l'étape 2).
     - ⇨ sinon aller à c de 2.

   c. Cherche s'il y a des appels bas aux étages plus élevées :
     - ⇨ si oui **monte** (aller à l'étape 2).
     - ⇨ Sinon aller à d de 2**.**

   d. Cherche s'il y a des appels haut ou bas aux étages plus bas.
     - ⇨ Si oui **descente** (aller à l'étape 3).
     - ⇨ Sinon resté en **attente** (aller à l'étape 4).

3. **déplacement vers le bas :**

   a. S'il y a une sélection ou un appel Bas dans cet étage alors arrêteet **servie** la demande puis retourner à l'état d'**attente.**

   b. Sinon tester s'il y a des sélections, des appels dans les étages plus bas.
     - ⇨ si oui retourner à l'étape 3.
     - ⇨ sinon aller à c de 3.

   c. cherche s'il ya des sélections ou des appels aux étages plus hauts :
     - ⇨ **si oui monte** et aller à l'étape 2.

⇨ sinon, rester en **attente** à cet étage (aller à l'étape 4)**.**

4. Rester en attente :

a. S'il y un appel haut ou bas aux étages plus élevés aller à l'étape 2.

b. Sinon **a**, S'il y a un appel haut ou bas aux étages plus bas, aller à l'étape 3.

c. Sinon rester à l'étape 4.

1-4 Sélectif haut/ Sélectif bas

1. en alimentation de l'ascenseur la cabine est au RDC :

- S'il y a un appel haut au RDC alors servie l'appel **(ouverture et fermeture)**.
- Sinon : chercher s'il y a des appels hauts ou Bas aux étages plus haut :

⇨ si oui **monté** la cabine vers l'étage d'appel (aller à l'étape 2).

⇨ sinon rester en **attente** au RDC (aller à l'étape 4)**.**

2. déplacement vers le haut :

a. Testé s'il y a des sélections dans les étages plus haut et avant l'étage d'appel choisit à servir.

⇨ Si oui **monte**à l'étage de la nouvelle sélection trouvée (retourné à l'étape 2).

⇨ Sinon :

✓ si l'ascenseur atteint l'étage désiré, alors servie la demande puis retourner à (b de 2).

✓ Sinon retourné à l'étape 2.

b. Cherche s'il y a des appels hauts aux étages plus élevées :

⇨ si oui **monte** (aller à l'étape 2).

⇨ sinon aller à c de 2.

c. Cherche s'il ya des appels bas aux étages plus élevées :

⇨ si oui **monte** (aller à l'étape 2).

⇨ Sinon aller à d de 2**.**

d. Cherche s'il y a des appels haut ou bas aux étages plus bas.

⇨ Si oui **descente** (aller à l'étape 3).

⇨ Sinon resté en **attente** (aller à l'étape 4).

3. déplacement vers le bas :

a. Testé s'il y a des sélections dans les étages plus bas et avant l'étage d'appel choisit àservir.

⇨ Si oui : **descente** à l'étage de la nouvelle sélection trouvée (retourné à l'étape 3).

⇨ Sinon :

✓ si l'ascenseur atteint l'étage désiré, alors servie la demande puis aller à (b de 3).

✓ Sinon reste à l'étape 3.

b. Cherche s'il y a des appels bas aux étages plus bas :

⇨ si oui **descente** à l'étage d'appel (retourné à l'étape 3).

⇨ sinon aller à c de 3.

c. Cherche s'il y a des appels hauts aux étages plus bas :

⇨ si oui **descente** à l'étage d'appel (retourné à l'étape 3).

⇨ Sinon aller à d de 3**.**

d. Cherche s'il y a des appels hauts ou bas aux étages plus élevés.

⇨ Si oui monte à l'étage d'appel (aller à l'étape 2).

⇨ Sinon resté en **attente** (aller à l'étape 4).

4. Rester en attente :

a. S'il y un appel haut ou bas aux étage plus élevés aller à l'étape 2.

b. Sinon **a**, S'il y a un appel haut ou bas aux étages plus bas, aller à l'étape 3.

d. Sinon aller à l'étape 4.

1-5 Sélectif Collectif / haut bas

1. en alimentation de l'ascenseur la cabine est au RDC :

- S'il y a un appel haut au RDC alors **servie** l'appel **(ouverture et fermeture)**.
- Sinon : chercher s'il y a des appels hauts ou Bas aux étages plus haut.

⇨ si oui **monté** la cabine vers l'étage d'appel (aller à l'étape 2).

⇨ sinon rester en **attente** au RDC (aller à l'étape 4)**.**

2. déplacement vers le haut :

- **si le mode est sélectif alors :**

Testé s'il y a des sélections dans les étages plus haut et avant l'étage d'appel choisit à servir.

⇨ Si oui **monte**à l'étage de la nouvelle sélection trouvée (retourné à l'étape 2).

⇨ Sinon monte l'ascenseur à l'étage désiré et servie la demande puis retourner à l'étape 2 (mode collectif).

- Si le mode est collectif alors:

a. S'il y a une sélection ou un appel haut dans cet étage alors arrête et **servie** la demande puis retourner à l'état d'**attente** (aller à l'étape 4).

b. Sinon tester s'il y a des sélections ou des appels hauts ou des appels bas dans les étages les plus élevés :

⇨ si oui retourner à l'étape (2).

⇨ Sinon aller à (c de 2).

c. tester s'il y a des sélections, des appels bas ou des appels hauts dans les étages les plus bas :

⇨ si oui **descendre** et aller à l'étape 3 (mode sélectif).

⇨ sinon, rester en **attente** à cet étage (aller à l'étape 4)**.**

3. déplacement vers le bas :

- **si le mode est sélectif alors:**

Testé s'il y a des sélections dans les étages plus haut et avant l'étage d'appel choisit à servir.

⇨ Si oui **monte**à l'étage de la nouvelle sélection trouvée (retourné à l'étape 3).

⇨ Sinon monte l'ascenseur à l'étage désiré et servie la demande puis retourner à l'étape 3 (mode collectif).

- **si le mode est collectif alors:**

a. S'il y a une sélection ou un appel Bas dans cet étage alors arrête et **servie** la demande puis retourner à l'état d'**attente.**

b. Sinon tester s'il y a des sélections, des appels dans les étages plus bas.

⇨ si oui retourner à l'étape 3(mode sélectif).

⇨ sinon aller à c de 3.

c. cherche s'il y a des sélections ou des appels aux étages plus hauts :

⇨ si oui **monte** et aller à l'étape 2.

⇨ sinon, rester en **attente** à cet étage (aller à l'étape 4)**.**

4. Rester en attente :

a. S'il y un appel haut ou bas aux étage plus élevés aller à l'étape 2.

b. Sinon **a**, S'il y a un appel haut ou bas aux étages plus bas, aller à l'étape 3.

e. Sinon rester à l'étape 4.

1-6 Algorithme spécial

1. en alimentation de l'ascenseur la cabine est au RDC :

- S'il y a un appel haut au RDC alors **servie** l'appel **(ouverture et fermeture)**.
- Sinon : chercher s'il y a des appels hauts ou Bas aux étages plus haut.
  - ⇨ si oui **monté** la cabine vers l'étage d'appel (aller à l'étape 2).
  - ⇨ sinon rester en **attente** au RDC (aller à l'étape 4)**.**

2. déplacement vers le haut :

a. Testé s'il y a des sélections dans les étages plus haut et avant l'étage d'appel choisit à servir.

- ⇨ Si oui **monte**à l'étage de la nouvelle sélection trouvée (retourné à l'étape 2).
- ⇨ Sinon :
  - ✓ si l'ascenseur atteint l'étage désiré, alors servie la demande puis retourner à (b de 2).
  - ✓ Sinon retourné à l'étape 2.

b. Sinon resté en **attente** d'une autre demande d'un personne spécial (aller à l'étape 4).

5. déplacement vers le bas :

a. Testé s'il y a des sélections dans les étages plus bas et avant l'étage d'appel choisit à servir.

- ⇨ Si oui : **descente** à l'étage de la nouvelle sélection trouvée (retourné à l'étape 3).
- ⇨ Sinon :
  - ✓ si l'ascenseur atteint l'étage désiré, alors servie la demande puis aller à (b de 3).
  - ✓ Sinon reste à l'étape 3.

b. Sinon resté en **attente** autre demande d'un personne spécial (aller à l'étape 4).

3. Rester en attente :

a. S'il y un appel haut ou bas aux étage plus élevées aller à l'étape 2.

b. Sinon **a**, S'il y a un appel haut ou bas aux étages plus bas, aller à l'étape 3.

c. Sinon rester à l'étape 4.

2- Structure du programmation VHDL du système de gestion :

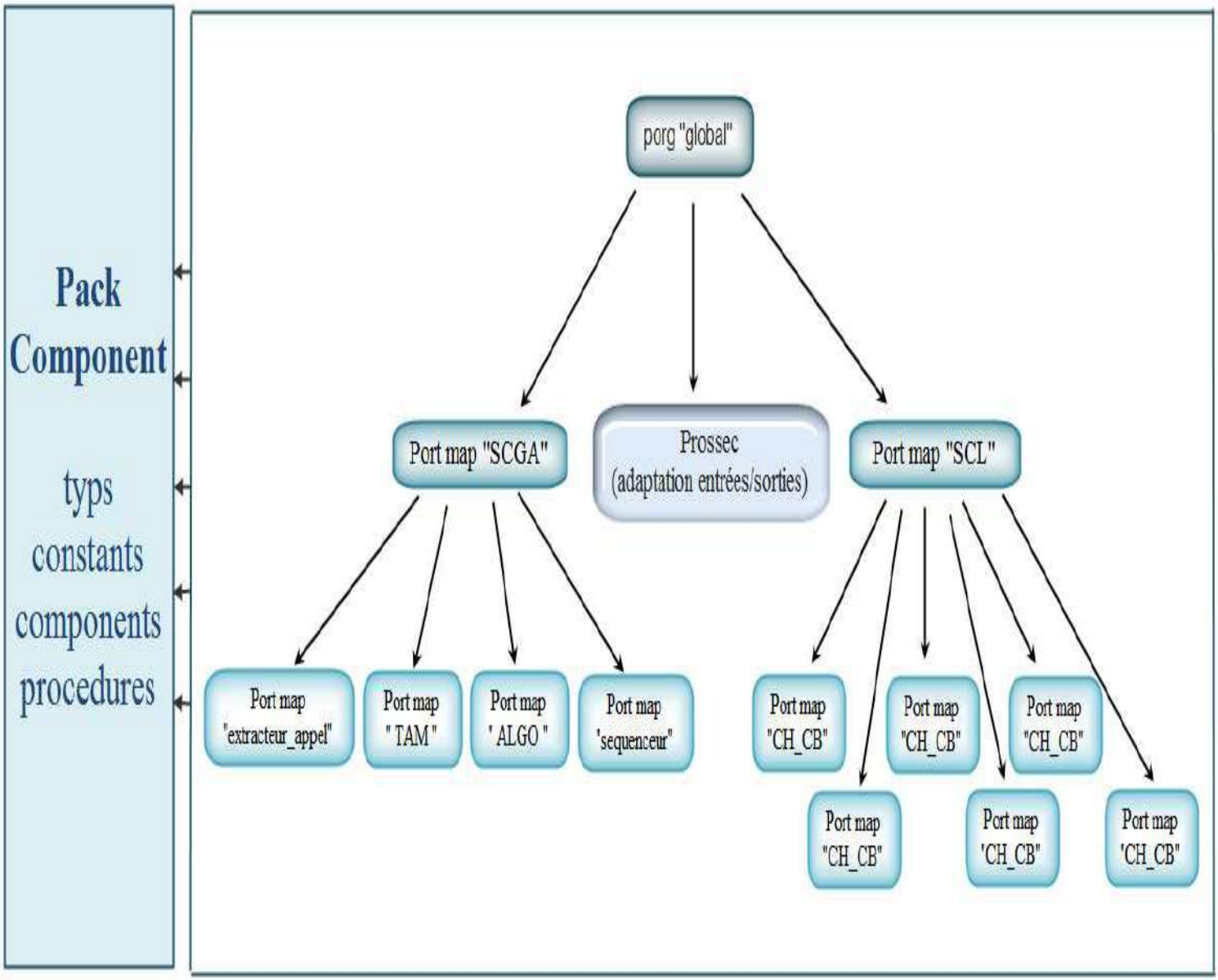


Fig AX.56 : Structure des programmes VHDL