REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DR. YAHIA FARES DE MEDEA
FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

Département du Génie Électrique
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***

THEME

SMART PARK1:

Etude, conception et implémentation VHDL
D'un système intelligent de parking de véhicules.
Application au centre-ville de Médéa

Proposé et Dirigé par :

➔***Dr. BOUHEDDA Mounir***

Préparé et Présenté par:

➪ ***TANDJAOUI Ryma*** ➪***ZAHRAOUI Soumia***

Promotion: 2011-2012.


بسم الله الرحمن الرحيم
١٤٢٠

« Toute technologie avancée est magique »

« Arthur C. Clarke »

Dédicace :



Au nom de dieu le tout miséricordieux
Toute la louange, à dieu seul, Pour Dieu seul, qui m'a entouré de sa compassion.
Je voudrais dédier ce modeste travaille :

A mon père feu, mon cher disparus, c'est à toi que je dois tout, ce travail est le fruit de la rigueur de ton éducation.
Que dieu te garde dans sa miséricorde, et que ton âme repose en paix.

A toi ma chère mère qui a veillé à ce que je sois ce que je suis maintenant, et qui m'a tant soutenue et encouragé dans tous les domaines et surtout pour réaliser ce mémoire

A vous, les gens qui me sont très chers.

A vous les membres de ma petite famille.

À toi ma tendre sœur et deuxième mère (Sara), tu n'es qu'un prolongement de moi (un deuxième moi dans ce monde), et à ton mari (Mohamed) mon 5eme frère que j'aime tant,
Que dieu vous protège l'un pour l'autre.

A vous mes chers frères (Ahmed, Bilal, Slimane, Mohamed), et à vos femme (Merieme, Sara Ryma, Aicha, Fatima), Pour tant de confiance, d'amour, et de patience, sans oublier mes inestimables nièces et neveux bien chouchoutés (Mohamed,Lina,Imene,Abd el samie, Abd el momine),(Rymel et Mohamed), (et les jumelles qu'on attend)), Meilleurs vœux de succès dans vos études.
À toi ma chère soeur Ratiba, et à ton marie (Muhamed), et vos enfants (Mahdia, Wisale, Zine el dine).

Que dieux vous protège pour moi et me protège pour vous.

A vous tous mes chères copines, qu'on a partagé ensemble le bon et le mauvais, (Zhore, Nawal, Amina, Fayza, Souhila, Zahia, Soumia, Rachda, ma cousine Salima, Hadjer).
Que vous trouviez dans ce modeste travail, l'expression de ma reconnaissance, mon amour, mon amitié, et mon estime.

Ryma

Dédicace :



Elle est mon âme, elle est mon cœur,
Elle est ma flamme, mon bonheur,
C'est celle qui m'a donné la vie,
je dédie ce travail d'abord à elle.
(Ma mère).

« Alleh Yarhamha »

La lumière de mes yeux, mon très cher père pour son soutien et leur patience et je le remercie beaucoup pour tous ce qu'il a sacrifié pour mon éducation, et pour leur soutien
« Alleh Yahfadho »

A mes grands-mères.
A ma très chers frères : Benaissa, Nasser Eddine, Saleh Eddine.
A petite sœur : Ibtessem.
A mon oncle : Boukourtbi Othmen.
A mes amies que je ne les oublier jamais : Khadîdja , Hassina,rima,faiza ,Sihem, Amina,meriem,amouna.

A tous collègues de la promotion 2011/2012

Soumia

Avant-Propos

Ce travail a été réalisé dans le cadre de notre projet de fin d'études à la Faculté des Sciences et de la Technologie de l'Université de Dr.Yahia Fares de Médéa. Pour l'obtention du diplôme de Master en Génie Electrique, option Systèmes intelligents et communicants.

Ce projet étant parvenu à terme, nous nous apercevons que le temps a vite passé, Heureusement que les connaissances, les bons souvenirs, font toujours durer la réjouissance.

Au nom de Dieu le tout miséricordieux
Toute la louange, à dieu seul qui nous a aidés

Nous souhaitons manifester notre reconnaissance particulière à Monsieur, Docteur Bouhedda Mounir, d'une part pour nous avoir donné l'opportunité de participer à ce projet. Ce projet qui a développé en nous une capacité de recherche et d'adaptation. Et d'autre part d'avoir accepté d'être notre encadreur, avec un suivi constant et un intérêt démontré tout au long de notre travail.

Nos plus sincères remerciements vont également aux membres du jury et tout le comité pédagogique de la formation Master Système intelligent et communicant (SIC).

Nos remerciements également pour tous les enseignants du département de Génie Electrique et de l'Informatique qui ont contribué à notre formation.

Aussi nous remercions nos collègues, qui ont été nos compagnons durant cette riche période d'étude, nous avons traversé des étapes que nous croyons au départ insurmontables.

Que tous ceux qui nous ont aidés de près ou de loin trouvent ici l'expression de notre sincère gratitude.

ملخص

الهدف من هذا العمل هو وضع نظام لتسيير ذكي لوقوف السيارات، بهدف تخفيف حدة الازدحام الناتج عن المركبات التي هي في صدد البحث عن مكان شاغر للوقوف ، النظام يهتم بالسيارة منذ اللحظة التي يقوم السائق بطلب مكان شاغر لركن سيارته، حيث يوجهه إلى مكان فارغ في أقرب موقف سيارات،مع الحرص على احترام قوانين وسياسات السير المختلفة، ، وضمان توجهه إلى مكان فارغ و ذالك عن طريق خاصية نظام التتبع الذي يقدمه النظام على طول الطريق، وهذه الخاصية تسمح بالكشف عن الاتجاه الخطأ و/ استعمال المكان من قبل سيارة أخرى قبل الوصول إليه، يتم البحث عن مكان آخر لهذه السيارة، مع الأخذ في الاعتبار الموقع الجديد للسيارة (في حالة الخطأ في الاتجاه)، والحالة الجديدة لاستمرارية توفر المكان(إذ اتم أخذ مكان من قبل سيارة أخرى). تم تطبيق النظام على جزء من وسط مدينة المدية، الذي تمت نمذجته رياضيا،كما تمت محاكاة هذا النموذج عن طريق Matlab ومن ثم تنفيذها في VHDL. و تظهرت نتائج الاختبارات و التحليل أداء النظام المقترح.

كلمات مفتاحية: VHDL ، أنظمة ذكية، مسار، اتجاه، نمذجة، موقف سيارات.

Résumé

Le but de ce travail est de concevoir un système de gestion intelligente de parking, dans le but de réduire la congestion due aux véhicules recherchant des places de stationnement. Le système prend en charge le véhicule dès que son conducteur souhaite se garer, et s'occupe de lui trouver une place vide dans le parking le plus proche, en respectant les différentes lois et politiques de circulation, et en assurant la disponibilité de la place grâce à la propriété de poursuite que le système offre tout le long du chemin à parcourir. Cette propriété permet de détecter la faute de direction et/ou la prise de place par un autre véhicule avant d'y arriver, et une nouvelle place est envisagée pour ce véhicule en tenant compte de la nouvelle position (en cas de faute de direction) et le nouvel état de disponibilité (en cas que la place a été prise par un autre véhicule).

Le système a été appliquée sur une partie du centre-ville de Médéa, qui a été modélisé mathématiquement, cette modélisation a été simulée en Matlab et implémenté en VHDL. Les résultats des tests et analyses ont montré les performances du système proposé.

Mots clés : chemin, direction, intelligent, modélisation, parking, VHDL.

Abstract

The aim of this work is to design an intelligent system for parking management, in the purpose to reduce congestion caused by vehicles that are looking for parking spaces, the system takes the vehicle in charge from the moment that the driver wishes to park his car, and takes care finding an empty place in the nearest parking lot, respecting the different traffic laws and policies, and ensuring the availability of space with the property of tracking, that the system offers all the way along the path, this property allows the detection of the false direction and / or whether or not the place was taken by another vehicle before arriving to it, and another place is being considered for this vehicle, taking into account the new position (in case of a false direction) and the new state of availability (if the place was taken by another vehicle).

The system was applied to a part of the city of Medea, which was modeled mathematically; this model has been simulated in Matlab then implemented using VHDL. Test results and analyzes showed the performance of the proposed system.

Key words: VHDL, intelligent, path, direction, modeling, parking.

Table des matières

Introduction générale 2

Chapitre I : contexte d'étude de la structure du parking et de l'algorithme utilisé.

I.1 Introduction 6

I.2 Les types des parkings 6

I.2.1 Parking de surface 6

I.2.2 Parking fermé ou en souterrain 6

I.2.3 Parking à étages 7

I.2.4 Parking-relais 7

I.3 Structure d'un parking 7

I.3.1 Les détecteurs 8

I.3.1.1 Les systèmes de détection à induction 9

I.3.1.2 Le système de traitement d'image vidéo (capteurs vidéo-traitement d'image) 10

I.3.1.3 Les systèmes de détection basés sur les radars à micro-onde 10

I.3.1.4 Capteur Neavia 11

I.3.2 Le système de payement 11

I.3.3 Le GPS 13

I.4 Le système de gestion des parkings 14

I.5 L'algorithme de Floyd 15

I.5.1 Principe de Floyd 15

I.5.2 Exemple d'application de l'algorithme de Floyd [8] 16

I.6 Conclusion 17

Chapitre II: modélisation mathématique et simulation.

II.1 Introduction……19

II.2 Réseau routier de Médéa……19

II.3 Modélisation mathématique……...19

II.3.1 La matrice des distances initiales (notée : MDI)……...20

II.3.2 La matrice des distances minimales (notée : MDM)……..22

II.3.3 La matrice de marquage :MMrq……22

II.3.4 La matrice des arcs (notée : MAC)……23

II.3.5 La matrice des parkings (notée : MPK)……24

II.3.6 La matrice des espaces de stationnement (notée : MES)……24

II.3.7 La matrice des directions (notée : MDR)……...25

II.4 Algorithme utilisé 26
II.5 Simulation avec Matlab 27
II.5.1 Principe général de fonctionnement : 27
II.5.1.1 La procédure Floyd 28
II.5.1.2 La procédure distances 28
II.5.1.3 La procédure dispo 30
II.5.1.4 La procédure « direction » 30
II.5.1.5 La procédure « vériff » 31
II.5.2 Principe de collaboration : 32
II.5.3 Exemples de traitement 34
Exemple1 34
Exemple2 : 35
Exemple3 : 37
Exemple 4 : 37
II.6 Conclusion :……..38

chapire III:implementation VHDL

III.1 Introduction 40
III.2 Principe général 40
III.2.1 Description de l'algorithme 40
III.2.2 Pseudo code de fonctionnement du système 43
III.3 Réseau routier 44
III.4 Description détaillé du code VHDL 45
III.4.1 Les entrées 45
III.4.2 Les sorties 45
III.4.3 Les données internes 45
III.5 Implémentation en VHDL 46
III.5.1 Le module examinateur de disponibilité (disponible) 46
III.5.2 le module identificateur du point d'arrivé (term) 47
III.5.3 Le module identificateur de l'arc correspondant au parking vide le plus proche (distance) 48
III.5.4 Le module de restauration du chemin 49
III.5.5 Le module superviseur de collaboration (app) 50
III.5.6 Machine d'état ou module contrôleur 51
III.6 Tests et analyses 55

III.7 Conclusion…….67

Bibliographies 71

Annexe A: 73

Annexe B..... 83

La liste des acronymes

VHDL : **V**HSIC **H**ardware **D**escription **L**angage.

VHSIC : **V**ery **H**igh **S**peed **I**ntegrated **C**ircuit.

FPGA : **F**ield **P**rogrammable **G**ate **A**rrays ou "réseaux logiques programmables".

VLSI : **V**ery **L**arge **S**cale **I**ntegration: circuits intégrés à très haut niveau d'intégration.

MDI : la **Matrice** des **D**istances **I**nitiales.

MDM : la **M**atrice de **D**istance **M**inimale.

MMrq : la **M**atrice de **M**a**rq**uage.

MAC : la **M**atrice des **A**r**c**s.

MPK : la **M**atrice des **P**ar**k**ing.

MES : la **M**atrice des **E**spaces de **S**tationnement.

MDR : la **M**atrice de **D**i**r**ection.

La liste des figures

Figure I.1 :parking de surface……..06
Figure I.2 :un exemple d'un parking fermé……..07
Figure I.3 :un parking à étages…….07
Figure I.4 :parking relais……..……..07
Figure I.5 :structure d'un parking……..……08
Figure I.6 :l'emplacement d'un detecteur……..……09
Figure I.7 :boucles éléctroniques collées……..……09
Figure I.8 :utilisation du traitement d'images…..……..10
Figure I.9 :système Neavia……..……...11
Figure I.10 :la caisse de payement automatique...…… 12
Figure I.11 :système de payement individuel dans un parking...……...12
Figure I.12 :système de payement collectif dans un parking…….13
Figure I.13 :un système de GPS dans un voiture……...14
Figure I.14 :un parking automatique……..15
Figure II.1: réseau routier du centre-ville de Médéa……..20
Figure II.2 : exemple de deux nouds adjacents……..20
Figure II.3 : exemple de deux nœuds non-adjacents…….21
Figure II.4 : exemple chemin à distance minimale entre les nœuds 1 et 6. ……..22
Figure II.5 : exemple du marquage du chemin……...23
Figure II.6 : exemple d'arc reliant entre deux nœuds……...23
Figure II.7 : exemple d'arc contenant un parking ……24
Figure II.8 : exemple d'un parking……25
Figure II.9 : exemple de direction entre deux arcs adjacents……26
Figure II.10 : principe de l'algorithme de Floyd……...27
Figure II.11 : entrées/sorties de la procédure Floyd……..28
Figure II.12 : les entrées/sorties de la procédure distance…….29
Figure II.13 : fonctionnement de la procédure distance…….29
Figure II.14 : entrées/sorties de la procédure dispo……30
Figure II.15 : principe de la fonction dispo……30
Figure II.16 : entrées/sorties de la procédure direction……..31
Figure II.17 : principe de la procédure direction……31

Figure II.18: les entrées/sorties de la procédure veriff……...31
Figure II.19 : le principe de fonctionnement de la procédure veriff……..32
Figure II.20 : principe de collaboration……..33
Figure II.21 : exemple de position……..34
Figure II.22 : exemple de place de parking prise……35
Figure II.23 : exemple de faute de direction……...36
Figure II.24 : exemple de véhicule situé dans un sens opposé d'un parking…….37
Figure II.25 : exemple d'utilisation de rond-point pour faire le demi-tour…….38
Figure III.1 : schéma globale de fonctionnement de système de stationnement**……..….**42
Figure III.2 : schéma globale de stationnement dans un parking**……..…**41
Figure III.3 : réseau routier d'une partie du centre-ville de Médéa **……...**44
Figure III.4 : fonctionnement du module disponible……...…...46
Figure III.5 : la simulation de disponible en modelsim **……**47
Figure III.6 : le module qui déterminer le point d'arrivé**……..** 47
Figure III.7 : exemple de simulation de fonction term**…….... …**48
Figure III.8 : la simulation de module distance……..……..……49
Figure III.9 : la simulation de module de restauration du chemin.……...…….50
Figure III.10 : la simulation de la fonction app**…….……..**……..51
Figure III.11 : schéma globale de la machine d'état…….……52
Figure III.12 : ensemble de directions renvoyé pour l'exemple 1…...……...……..57
Figure III.13: détail du scenario1……58
Figure III.14 : ensemble des directions fournis pour l'exemple2……60
Figure III.15 : détail du scenario2……...60
Figure III.16 :l'ensemble des directions fournis pour l'exemple3…….62
Figure III.17 : détail du scenario3……..63
Figure III.18 : l'ensemble des directions fournis pour l'exemple 4……...64
Figure III.19 : détail du scenario 4…….65
Figure III.20: l'ensemble des directions fournis pour l'exemple5……..66
Figure III.21 : détail du scenario5……...67

La liste des tableaux

Table III.1 : palette correspondante à l'état0……..….53
Table III.2: palette correspondante à l'état1……....53
Table III.3 : palette correspondante à l'état2 ……...…53
Table III.4 : palette correspondante à l'état3……...……..…..54
Table III.5 : palette correspondante à l'état4……...54
Table III.6 : palette correspondante à l'état5……...54
Table III.7 : palette correspondante à l'état6……...55
Table III.8 : palette correspondante à l'état7……...55

La liste des annexes

Annexe A……73

Annexe B…….83

Introduction générale

Introduction générale :

Le parking est une composante essentielle des politiques de mobilité, les véhicules Doivent se garer à chaque destination. La facilité de stationnement engendre des coups important pour la société, concepteur, exploitants, et autorités publiques qui sont confrontés quotidiennement aux conflits liés au stationnement.

Ces problèmes peuvent être définis soit en termes de congestion, soit en termes de systèmes de gestion(les systèmes de gestion classique ne prennent pas le véhicule en charge avant qu'il y arrive).Beaucoup de villes , voir toutes les villes sont confrontées à des problèmes de stationnement.la recherche de place de parking prend un temps accru ce qui amplifie la saturation du réseau routier, la production d'émission et de bruit, et surtout engendre une surconsommation du sol, l'accessibilité du centre de la ville est réduite.

Les conducteurs gaspillent leur temps, leur argent et leur énergie en pure perte, tandis Que la collectivité subit des émissions de CO2 inutiles et un manque à gagner considérable. Le manque d'innovations technologiques dans le secteur du stationnement sur voirie, dont la dernière remonte à 1935 [1] avec l'invention du premier parcmètre, explique en grande partie l'immobilité de cette situation depuis des années.

Un besoin à des solutions radicales pour cet ennui quotidien est nécessaire. Les parkings intelligent font partie des technologies qui sont susceptibles de révolutionner notre vie, effectivement, un système de guidage et de poursuite qui prend en charge le véhicule dès qu'il souhaite stationner, peut aider à éviter un temps perdu et à économiser une quantité importante de carburant, et de diminuer un taux d'émission potentiel, pour un grand nombre de véhicules , ceci pourra certainement produire une différence.

Des solutions sont envisagées afin d'arriver à combler les besoins de places de Stationnement en évitent de causer la congestion, comme les parkings multi-étages, les parkings sous-sol,…etc.

Et des politiques de stationnement sont adoptées en vue de réduire les stationnements aléatoires, comme les zones bleues, la favorisation de l'arrêt au stationnement …etc.

Et des grand projets sont mis en place par des équipe en collaboration, comme celui « ParkSense » appliqué en France et aux états unies, la solution « ParkSense » vise à optimiser la gestion du stationnement en parc et sur voirie. Fondées sur l'utilisation de détecteurs de véhicules sans fil installés au sol sur les places, Chaque jour, « ParkSense » aide

plus de 100’000 automobilistes à trouver une place de stationnement disponible.le concept des « grains intelligents » (« smart grains ») capables de localiser les objets en temps réel, a été développé et perfectionné par la société « SmartGrains » en partenariat avec « L’INRIA »,le « CNAM » et le « CNRS » afin de donner naissance aux capteurs « ParkSense » des détecteurs ultra-basse consommation capables de repérer des véhicules et de communiquer en sans-fil avec une autonomie de 5 ans.

Presque 50% des véhiculent qui circulent en ville sont à la recherche d’un parking ce taux élevé a pour effet de créer l’embouteillage en ville, ce qui nous amène à un phénomène qui devient de plus en plus l’un des problèmes majeurs des grandes villes en particulier et même des villages, le trafic n’est pas le seul affecté par cet embouteillage, en effet il existe des victimes oubliés, parmi lesquels on cite l’atmosphère qui requiert des solutions radicales pour sa pollution. L’accroissement du trafic automobile et le problème de stationnement, sont devenus une préoccupation inévitable, effectivement à certain moment de la journée, il devient problématique, voire impossible de stationner au centre-ville, au abord de la gare, à la marie, à la poste, au Masdjid …etc.

Cela n’ira pas en s’améliorant avec les systèmes de gestion de parkings classiques, le conducteur qui a parcouru une distance importante pour arriver à un parking plein, ou en recherchant une place pour se garer est certainement soucié du temps perdu, du gaspillage de carburant consommé et de la pollution générée. Dans notre projet on se propose de concevoir et implémenter un système intelligent de gestion des parkings en ville en prenant comme exemple une partie du centre-ville de Médéa.

Durant la réalisation de ce projet on a été amené à utiliser essentiellement le logiciel Matlab de MathWorks pour la vérification du modèle mathématique proposé. Le logiciel Modelsim de Modeltech pour l’implémentation VHDL.

Le présent document est organisé en 3 chapitre dont :

- Le premier chapitre présente une vue panoramique sur le concept de la gestion intelligente des parkings en détaillant les outils nécessaires pour la réussite de sa réalisation (détecteurs, GPS), et celui de l’utilisation des circuits FPGA, ainsi que les algorithmes utilisés (algorithme de Floyd).
- Le deuxième chapitre traite la problématique en appliquant la solution proposée par ce travail sur une zone géographique bien précisée, en se servant du logiciel Matlab en vue de vérifier les résultats de modélisation.

- Le troisième chapitre sera consacré à l'implémentation VHDL, et en s'arrêtant sur les différences entre les résultats de simulation en Matlab et ceux obtenus en VHDL.

chapitre 1

Contexte d'étude de la structure du parking et de l'algorithme utilisé

I.1 Introduction

Un parking, ou parc de stationnement est un espace ou un bâtiment spécifiquement aménagé pour le stationnement des véhicules. On en trouve le plus souvent à côté des bâtiments publics (gare, aéroport), des lieux de travail, des centres commerciaux ou devant les grandes surfaces pour accueillir les usagers [2]. Gérer des parkings s'avère très important car il permet de réduire la congestion de trafic routier, gain de temps,….

I.2 Les types des parkings

Il existe différents types des parkings en fonction de leur destination et du nombre de places :

I.2.1 Parking de surface

Se situe de plain-pied, à l'extérieur, sur l'espace public ou privé. Ce type de parking comprend le stationnement en voirie (places le long d'une rue, d'un quai, etc.) et les espaces dégagés à cette fin entre des bâtiments, ou établis sur des anciens champs, des anciens terrains vagues, comme le montre la figure I.1[3].



Figure I.1 : parking de surface.[3]

I.2.2 Parking fermé ou en souterrain

C'est un parking fermé créée sur plusieurs niveaux qui peut être sous les bâtiments ou sous les aéroports ou en ville [3]. La figure I.2 représente un exemple d'un parking fermé.



Figure I.2 : un exemple d'un parking fermé.[3]

I.2.3 Parking à étages

C'est un parking à étages construit en extérieur (figure I.3).[3]



Figure I.3 : un parking à étages.[3]

I.2.4 Parking-relais

Ce sont des parking subventionnés par les collectivités et situé près des aéroports , port ou gare afin d'inciter l'utilisation de moyen de transport collectif [3]. La figure I.4 représente un parking-relais (dans un aéroport).



Figure I.4: parking-relais. [3]

I.3 Structure d'un parking

En général, les parkings sont composés d'un système de payement, des détecteurs, et autres dispositifs:

- Le système guide le conducteur à une place qui a été déterminée à priori.

- Après le stationnement du véhicule, le conducteur paye le tarif dans le système de payement et reçoit un ticket.
- Le système de payement incrémente les places pleines, et informe le système.
- Le détecteur détecte l'arrivé de véhicule et indique les places vides au système.

La figure I.5 donne les différentes composantes d'un parking ainsi la relation entre eux.

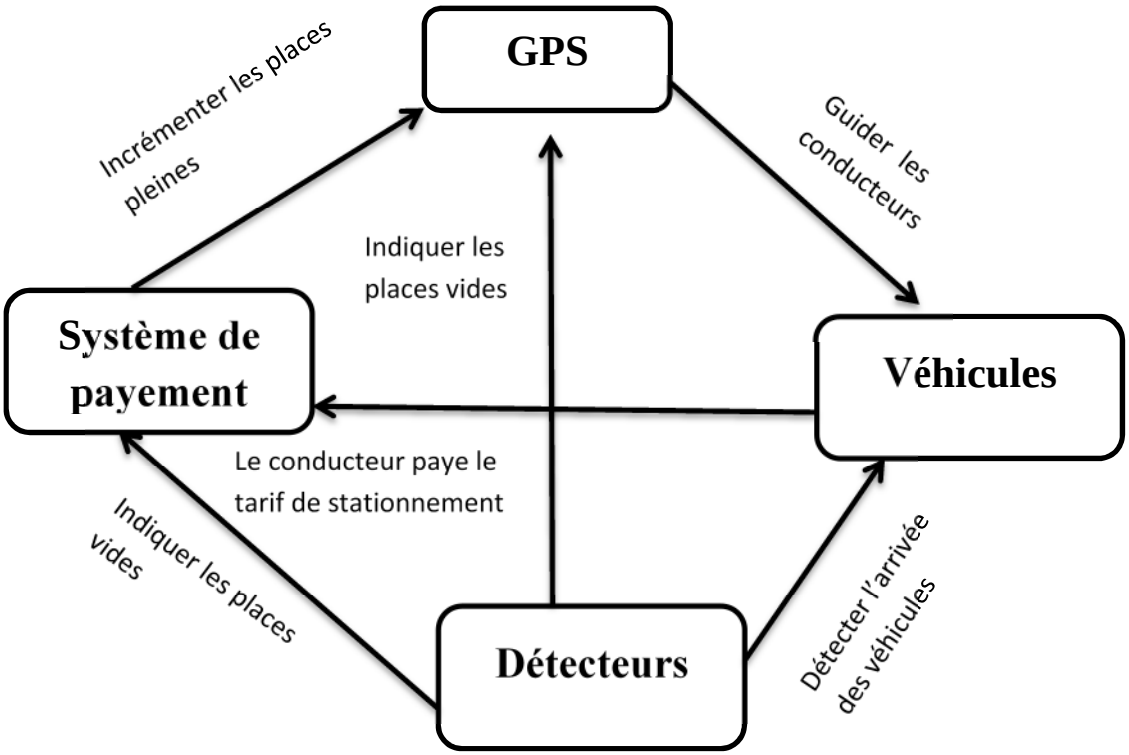


Figure I.5 : structure d'un parking

I.3.1 Les détecteurs

Un détecteur est un dispositif technique (instrument, substance, matière) qui change d'état en présence de l'élément ou de la situation pour lequel il a été spécifiquement conçu.
Des fonctions supplémentaires peuvent apporter des précisions qualitatives ou quantitatives sur la nature du phénomène observé [3].

Dans notre projet, il faut placer un détecteur dans chaque zone de stationnement de véhicule pour reconnaitre les places vides comme il est montré dans la figure I.6. Ces détecteurs sont reliés au système de GPS.

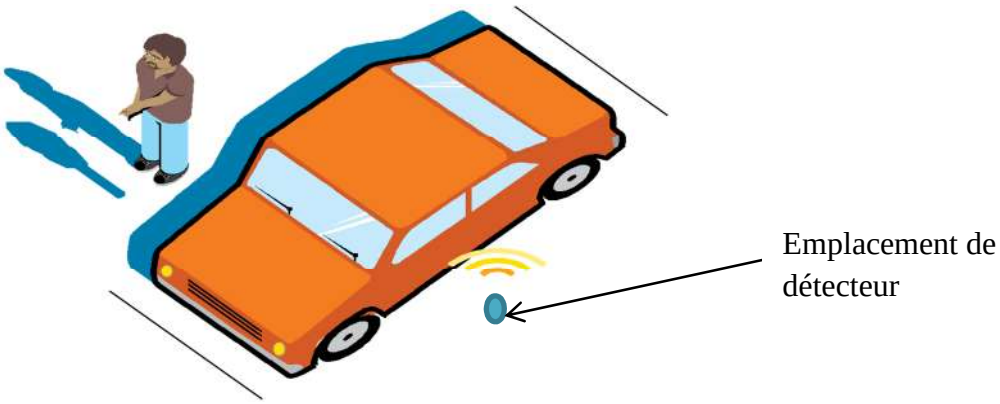


Figure I.6 : l'emplacement d'un détecteur.[4]

Ils existent plusieurs types des détecteurs sont :

I.3.1.1 Les systèmes de détection à induction

Le détecteur électromagnétique est le dispositif le plus utilisé actuellement dans le monde. Constitué d'une boucle inductive enfouie à 10 centimètres de profondeur dans le revêtement de la chaussée, il est susceptible de réagir à des phénomènes d'auto-induction.

Le passage d'un véhicule au-dessus de la boucle engendre des courants de Foucault dans la masse métallique du véhicule.

Cette variation électromagnétique peut être détectée et est ensuite transformée en un signal électrique de type tout ou rien correspondant à la présence d'un véhicule sur la boucle. Ils peuvent aussi mesurer des taux d'occupation, des intervalles inter-véhiculaires, des vitesses instantanées, ainsi que la vitesse moyenne d'un flot de véhicules [4].

La figure I.7 représente les boucles électroniques collées.



Figure I.7 : boucles électroniques collées [4].

I.3.1.2 Le système de traitement d'image vidéo (capteurs vidéo-traitement d'image)

Ils ont deux applications principales : l'analyse du trafic et la lecture de plaques minéralogiques. Pour le recueil des données de trafic, les technologies de capteurs existantes reposent sur deux familles : CCD (Charged Coupled Device) et CMOS (Complementary Metal Oxide Semi-Conductor).

Les caméras utilisées comme des boucles, sur une zone limitée de la chaussée, permettent obtenir des mesures de vitesse, de taux d'occupation et d'intervalles inter-véhiculaires.

Elles peuvent également mesurer des densités (ou concentration) du trafic sur une longueur donnée de chaussée, des mouvements tournants ou changements de voie, des longueurs de files d'attente ainsi que la détection automatique d'incidents (et notamment de véhicules arrêtés) [4].

La figure I.8 donne l'utilisation du traitement d'images.

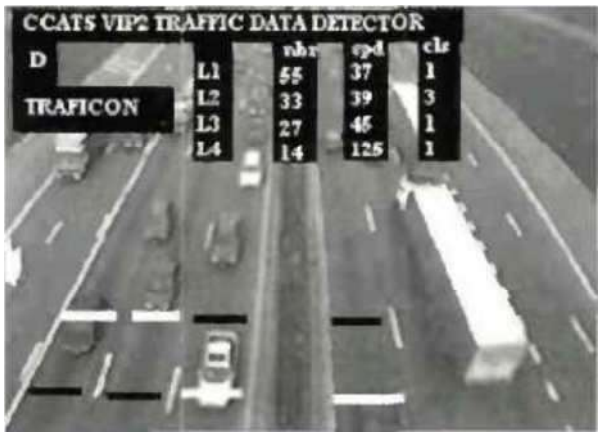


Figure I.8 : utilisation du traitement d'images.[4]

I.3.1.3 Les systèmes de détection basés sur les radars à micro-onde

Les détecteurs radar sont utilisés dans les tunnels en France et en Allemagne pour détecter la vitesse des véhicules et la distance entre véhicules. Des micro-ondes sont envoyées vers les véhicules du tunnel et retransmises après réflexion vers le récepteur du système de détection radar. Le détecteur mesure le temps mis par le signal réfléchi et calcule la vitesse du véhicule en appliquant le principe de Doppler. Les détecteurs radar mesurent également la présence des véhicules et les comptent. En général, les détecteurs radar sont installés au-dessus des voies de circulation dans le tunnel. Les systèmes de détection ultra-son sont comparables aux systèmes de détection radar, à l'exception que des ondes sonores sont utilisées, au lieu de la micro-onde. Les ondes sonores étant affectées par le bruit ambiant des tunnels, ce procédé n'est donc pas utilisé. Des problèmes surviennent quand les systèmes ne sont pas entretenus. La sensibilité des détecteurs radar doit être adaptée au profil en travers

pour éviter des fausses alarmes [4].

I.3.1.4 Capteur Neavia

C'est un capteur non intrusif développé par la société Neavia qui est basé sur le traitement des mesures sonores. Le système est équipé de 8 microphones qui sont alignés et placés perpendiculairement à l'axe de la chaussée. Un traitement des mesures permet ensuite des mesures de trafic par sens (débit, vitesse, taux d'occupation).

Les balises sont également équipées d'une caméra qui permet d'affiner les mesures, en offrant une distinction véhicules légers/poids lourds ainsi que de différencier les mesures par voie.

Le traitement du signal sonore prend en fait une place secondaire, et sert à affiner les mesures par traitement d'images et à combler certaines lacunes de la vidéo (nuit, brouillard…).

Le système utilise deux capteurs acoustiques qui mesurent les débits tous véhicules et les vitesses moyennes par période de 6 minutes.

Les balises sont alimentées par des panneaux solaires donc faiblement consommatrices en énergie. Le système permet en outre de détecter automatiquement des incidents et des congestions comme il est montré dans la figure I.9.[4]

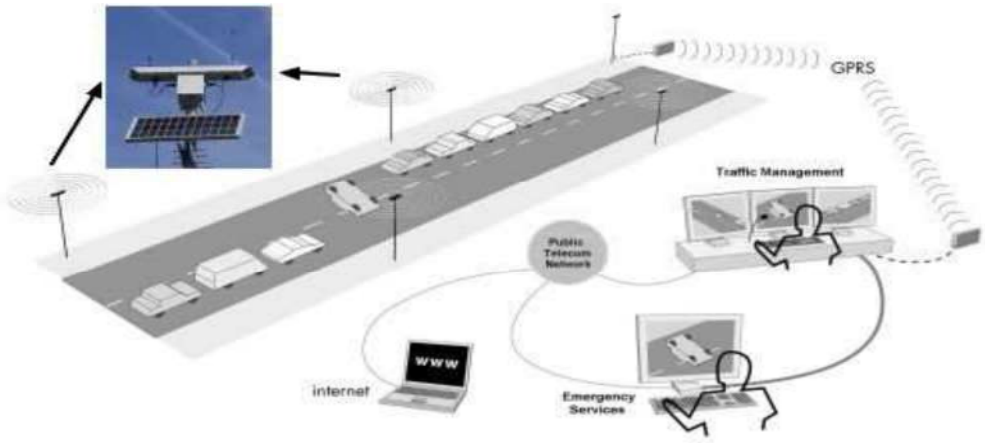


Figure I.9 : système Neavia.[4]

I.3.2 Le système de payement

Ces jours, pour payer un tarif de stationnement dans un parking, on trouve des appareils de paiement (des caisses automatiques).

La caisse automatique permet le payement de la redevance due par les usagers, avant Que ceux-ci ne rejoignent leur véhicule.

Une signalétique claire et conviviale indique aux clients les étapes successives pour effectuer leur payement. La figure I.10 est un exemplaire d'une caisse de payement automatique.



Figure I.10 : la caisse de payement automatique.[5]

Ces caisses sont à deux types :

- Individuel : à chaque zone de stationnement des véhicules, on trouve l'une fixé.

  La figure I.11 est représenté un système de payement dans un parking.

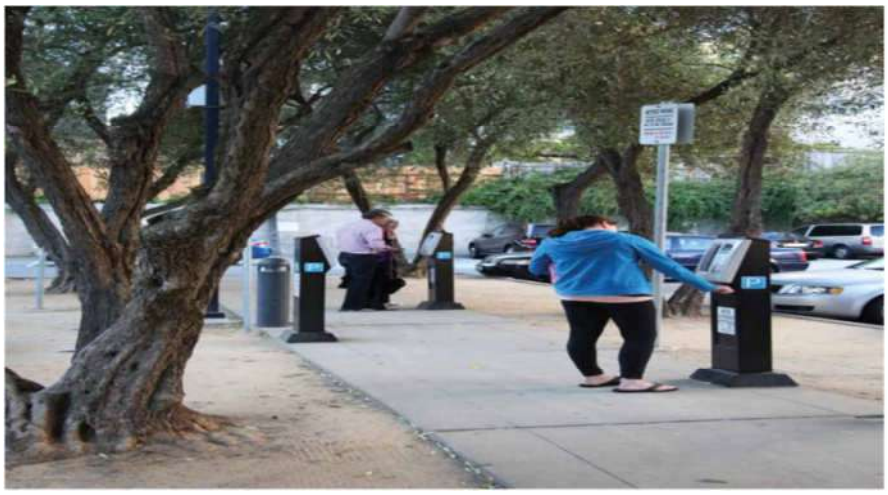


Figure I.11 : système de payement individuel dans un parking [5].

- Ou bien collectif : pour chaque ligne des zones de stationnement, il y a une seule caisse, comme il est indiqué sur la figure I.12.



Figure I.12: système de payement collectif dans un parking.[5]

L'opération de paiement déroule soit en espèce **(**des pièces de monnaie ou des billets de banque) soit par carte magnétique (carte bancaire).

I.3.3 Le GPS

Aujourd'hui comme jadis, les hommes se servent du ciel pour s'orienter. Mais à la différence des anciennes méthodes, aujourd'hui la navigation est assurée par satellites, offrant une précision et une vitesse d'exécution sans commune mesure avec l'observation visuelle du soleil et des étoiles.

La navigation par satellites et en particulier par GPS (Global Positioning System) est devenue un outil indispensable pour explorer une région qui n'est pas signalisée ou inconnue.

Depuis quelques années, le prix des GPS suivant celui de l'informatique, cet accessoire s'est démocratisé et a également trouvé sa place sur le tableau de bord des voitures, dans les avions privés et même sur les bateaux. Ce système de positionnement par satellites est tellement pratique pour ne pas s'égarer que les randonneurs l'utilisent pour faire le point à pied ou en véhicule.

Pour certains utilisateurs le GPS a depuis longtemps remplacé la carte routière et les panneaux de signalisation car ce système peut tirer avantage de toute la puissance d'un ordinateur pour mémoriser des points de référence ("waypoints") au milieu de nulle part et tenir compte des éventuels embouteillages pour trouver une route alternative en milieu urbain.

Un GPS reste toutefois un système sophistiqué. Il y a encore quelques années, dans sa version baroudeur associée à un calculateur, il requérait un apprentissage si vous vouliez arriver rapidement à destination ou si vous l'utilisiez à des fins scientifiques. La figure I.13 est une figure sur un système GPS utilisé dans une voiture [6].



La figure I.13 : un système GPS dans une voiture

I.4 Le système de gestion des parkings

Avant une dizaine d'année, les parkings ont été gérés par des hommes tels que le payement a été manuellement, le nombre de places vides et pleines non connus par les conducteurs.

Ce dernier pose un problème de tourner en rond des véhicules qui a causé la circulation surtout dans les parkings situé dans les centres villes.

Et après le développement de la technologie et des logiciels, et l'apparition de GPS et satellites, plusieurs équipes de recherche pense à résoudre ce problème par l'automatisation des stationnements (réservation par net, guidage par GPS, payement automatique,….).

Ce dernier permet de gagner du temps, aider les personnes qui ne connaissent pas le lieu exact et diminuer l'encombrement. La figure I.14 présente l'entrée un exemple d'un parking géré automatiquement.



Figure I.14 : un parking automatique.

I.5 L'algorithme de Floyd

Les problèmes de cheminement sont des problèmes classiques de la théorie des graphes. L'objectif est de calculer une route entre des sommets d'un graphe qui minimise ou maximise une certaine fonction économique. Le problème le plus classique consiste à chercher le chemin qui minimise la somme des valuations des arêtes traversées. Il existe plusieurs algorithmes qui résolvent ce problème comme algorithme de Djikstra, Kruskal, Ford-Bellman et Floyd-Warshall [7].

Nous avons choisis l'algorithme de Floyd car il est l'un des plus simples à mettre en œuvre dans notre travail d'implémentation et capable de travailler avec les arcs de poids

I.5.1 Principe de Floyd

Cet algorithme permet de trouver tous les plus courts chemins entre toutes les paires de sommets dans un graphe orienté sans circuit absorbant, la valeur des arcs peut être négative.
Pour utiliser cet algorithme, il nous faut une matrice des sommets précédents P et une matrice des chemins les plus courts L. Le principe de l'algorithme est le suivant:

1. Initialiser la matrice L en copiant la matrice d'adjacence.
2. Initialiser la matrice P: P(i,j):= i.
3. Pour chacun des chemins, chercher un chemin plus court passant par le sommet 1, si cela est le cas remplacé les valeurs ad hoc dans P et L.
4. Comme pour le point 3, mais pour des chemins passant par les sommets 1 ou 2.
5. Continuer ainsi jusqu'au dernier sommet [8].

I.5.2 Exemple d'application de l'algorithme de Floyd [8]

Soit un graphe illustré par la figure I.15 suivante:

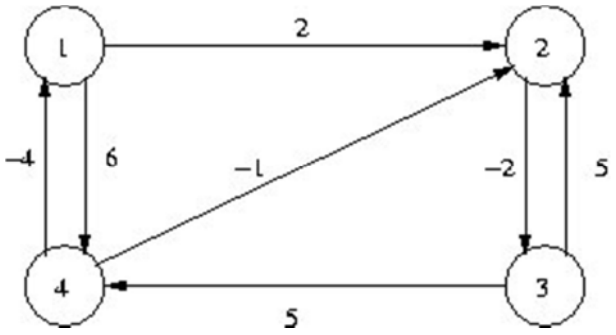


Figure I.15 : Exemple d'un graphe à 4 nœuds.

Départ, k=0:

$$L = \begin{pmatrix} 0 & 2 & \infty & 6 \\ \infty & 0 & -2 & \infty \\ \infty & 5 & 0 & 5 \\ -4 & -1 & \infty & 0 \end{pmatrix} \qquad P = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}$$

k=1:

$$L = \begin{pmatrix} 0 & 2 & \infty & 6 \\ \infty & 0 & -2 & \infty \\ \infty & 5 & 0 & 5 \\ -4 & -2 & \infty & 0 \end{pmatrix} \qquad P = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 1 & 4 & 4 \end{pmatrix}$$

k=2:

$$L = \begin{pmatrix} 0 & 2 & 0 & 6 \\ \infty & 0 & -2 & \infty \\ \infty & 5 & 0 & 5 \\ -4 & -2 & -4 & 0 \end{pmatrix} \qquad P = \begin{pmatrix} 1 & 1 & 2 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 1 & 2 & 4 \end{pmatrix}$$

k=3:

$$L = \begin{pmatrix} 0 & 2 & 0 & 5 \\ \infty & 0 & -2 & 3 \\ \infty & 5 & 0 & 5 \\ -4 & -2 & -4 & 0 \end{pmatrix} \qquad P = \begin{pmatrix} 1 & 1 & 2 & 3 \\ 2 & 2 & 2 & 3 \\ 3 & 3 & 3 & 3 \\ 4 & 1 & 2 & 4 \end{pmatrix}$$

k=4:

$$L = \begin{pmatrix} 0 & 2 & 0 & 5 \\ -1 & 0 & -2 & 3 \\ 1 & 3 & 0 & 5 \\ -4 & -2 & -4 & 0 \end{pmatrix} \qquad P = \begin{pmatrix} 1 & 1 & 2 & 3 \\ 4 & 2 & 2 & 3 \\ 4 & 1 & 3 & 3 \\ 4 & 1 & 2 & 4 \end{pmatrix}$$

Le chemin le plus court entre 1 et 4 vaut 5, il passe par les sommets 2 et 3, car: P(1,4)=3, P(1,3)=2 et P(1,2)=1.

Le chemin le plus court entre 3 et 2 vaut 3, il passe par les sommets 4 et 1, car: P(3,2)=1, P(3,1)=4 et P(3,4)=3.

Dans notre projet, on utilise cet algorithme pour trouver les distances minimales reliant chaque deux sommet dans un graphe.

Le principe ce résume dans l'insertion de la description d'une zone géographique sous forme d'une matrice contenant les distances initiales entre chaque deux sommets adjacents.

L'algorithme de Floyd utilise cette matrice comme entrée pour calculer les distances minimales reliant chaque sommet avec tous les autres sommets et ainsi fournir à la sortie deux nouvelles matrices : la première contenant les distances minimales et la deuxième contenant le marquage du chemin entre deux sommets. Et ainsi, un bon marquage pour le chemin le plus court est garanti.

I.6 Conclusion

Les parkings sont des emplacements importants dans une ville à lequel il faut penser très sérieusement. Actuellement avec l'augmentation du nombre des véhicules, les systèmes de parking intelligents s'avèrent une solution prometteuse et la solution FPGA peut être bien adaptée.

Chapitre II

Modélisation Mathématique et Simulation

II.1 Introduction

Dans le chapitre I on a abordé les différents outils hard/soft nécessaires pour la réalisation de la solution proposée, ainsi que les différentes contraintes à tenir en compte.

Dans ce chapitre on se propose d'approfondir dans ce sens en détaillant la vue globale de la problématique ainsi que la solution proposée, et la modélisation des différents concepts, tout en touchant l'utilisation du logiciel « Matlab » afin d'arriver à simuler la solution au niveau soft.

II.2 Réseau routier de Médéa

La problématique en cour de résolution nécessite son application sur une région géographique contenant toutes les contraintes qu'on doit prendre en compte en dirigeant un véhicule vers un espace de stationnement , comme la garantie de la bonne direction, le respect des lois de circulation, la disponibilité,..etc , toutes ces situation sont fortement présentes dans la zone géographique choisie pour l'application de la solution proposée, cette zone est caractérisée par un taux de circulation élevé , et des segments de route monodirectionnels,..etc, ces caractéristiques rendent cette région typique pour un tel traitement.

On présente la zone du centre-ville de Médéa montré dans la figure(II.1) comme suit :

→Les flèches décrivent les sens de circulation autorisés dans chaque segment de route.

→Les flèches en rouge définissent les espaces de stationnement légales dans chaque segment de route.

II.3 Modélisation mathématique

Afin d'arriver à diriger le conducteur vers la place de parking disponible la plus proche, la solution proposé nécessite une connaissance préalable et une description bien détaillée de la région contenant le parking, cette description peut être décrite sous forme de matrices.

Plusieurs matrices sont mises en place en vue de décrire les différentes caractéristiques de la région (chemin entre un emplacement et l'autre, sens interdit, espaces de parking, rond-point possibilité de faire un demi-tour).

Pour arriver à manipuler la région géographique, on décompose cette dernière en nœuds et arcs, ou les nœuds décrivent les extrémités des chemins directes ainsi que les carrefours, un arc orienté représente un chemin d'un seul sens d'un nœud vers un autre.

- Une remarque très importante pour notre projet est que tous les arcs sont orientés.

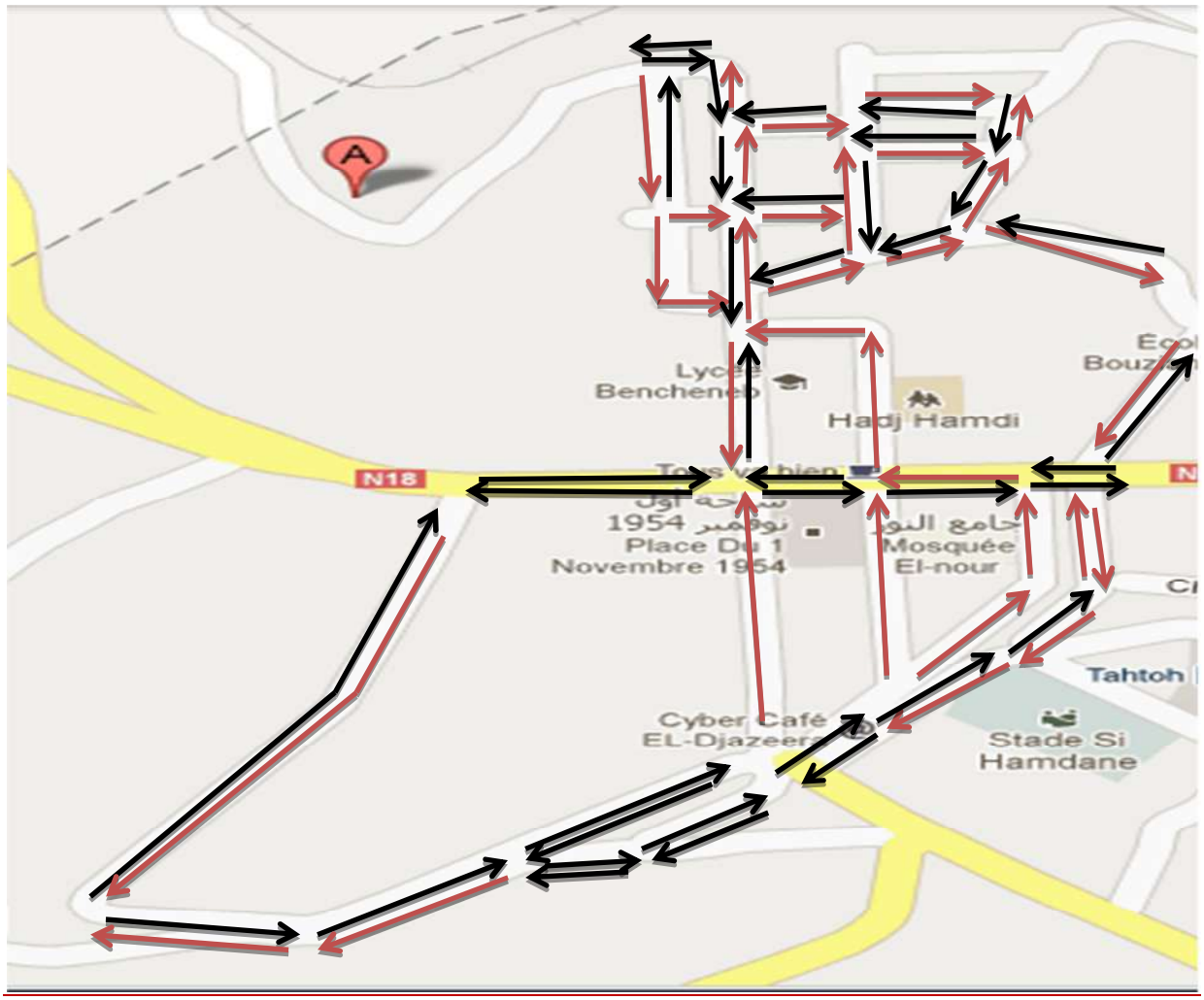


Figure II.1: réseau routier du centre-ville de Médéa

II.3.1 La matrice des distances initiales (notée : MDI)

Chaque case de la MDI contient la distance entre deux nœuds adjacents, ou la valeur infinie (représenté par inf) qui peut être soit entre deux nœuds non-adjacents, ou entre deux nœuds adjacents ou le déplacement est dans un sens interdit.

La matrice MDI est inséré dans l'annexe A.

Exemple 1:la figure (II.2) montre un exemple de deux nœuds adjacents 1 et 2

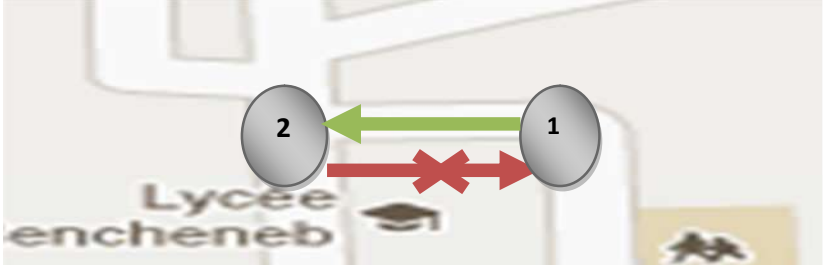


Figure II.2 : exemple de deux nouds adjacents

Alors on trouve dans la case (1,2) la distance entre ces derniers, tandis que dans la case (2,1) on trouve la valeur inf car le sens de 2 vers 1 est interdit. Pour bien exprimer ceci, on montre dans la MDI :

```
[0   100   inf   inf   inf   inf…
 inf   0    inf    200   inf   inf…].
```

Exemple 2: les nœuds 1 et 6 sont non-adjacents, car il n'existe aucun chemin direct du point 1 vers le point 6, cela est illustré dans la figure (2.3), ceci est montré dans la figure (II.3)

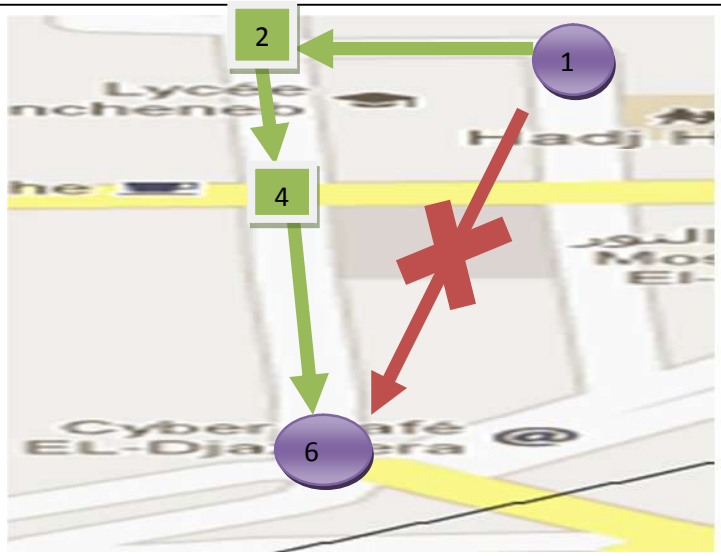


Figure II.3 : exemple de deux nœuds non-adjacents.

Et cela est exprimé dans la MDI comme suit :

```
[0   100 inf inf inf inf   inf   inf…
inf   0 inf 200 inf   inf   inf   inf…
200   inf 0 100 inf   inf   inf    100…
inf   200 100 0 inf   400   inf    inf…
inf   inf 250 inf 0 150    250    inf…
inf inf   inf 400 150 0   inf   inf…]
```

La MDI est fournie à la fonction Floyd afin d'obtenir deux autres matrices, la matrice des distances minimales et la matrice de marquage, ces matrices seront décrites par la suite.

II.3.2 La matrice des distances minimales (notée : MDM)

C'est une matrice dont chaque case contient la distance minimale entre deux nouds adjacents ou non-adjacents, cette matrice est obtenue à partir de la matrice des distances initiales en appliquant la fonction Floyd sur cette dernière (voir chapitre I), elle est inséré dans l'annexe A.

Exemple : la case (1,6) contient la distance minimale entre les nœuds 1 et 6, et voici la partie de la MDM qui contient cette valeur [0 100 400 300 850 700 1100…]

La figure (II.4) indique le chemin correspondant à cette distance.

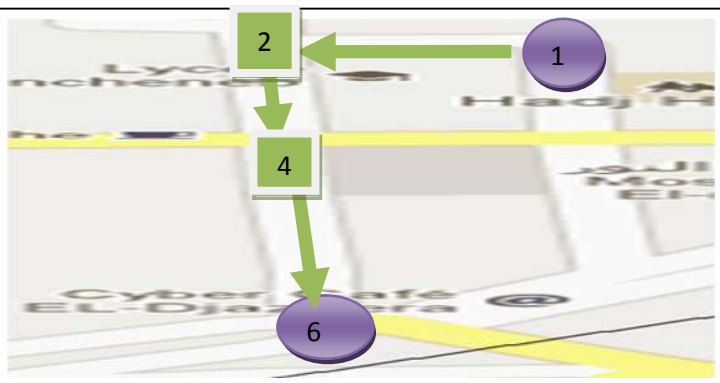


Figure II.4 : exemple chemin à distance minimale entre les nœuds 1 et 6.

II.3.3 La matrice de marquage :MMrq

C'est une matrice dont chaque case contient le premier point à visiter pour partir du point (indice ligne) vers le point (indice colonne), cette matrice est obtenue à partir de la matrice des distances initiales en appliquant la fonction Floyd sur cette dernière.

MMrq est inséré dans l'annexe A.

Exemple : pour partir du point 1 vers le point 3 et puisque de 1 vers 3 directement c'est un sens interdit on doit parcourir le chemin 1 2 4 3 , alors on trouve dans la case (1,3) la valeur 4 qui correspond point qui précède le point de destination, puis on trouve dans la case (1,4) la valeur 2 qui correspond au premier point visité pour aller de 1 vers 3, comme on trouve dans la case (1,2) la valeur 1 qui correspond au point de départ ici on s'arrête . Cette opération est plus claire en jetant un œil sur la partie de la MMrq suivante.

[1 **1** **4** **2** 6 4 5 3 4 9 12 6 6…], ceci est démontré dans La figure (II.5).

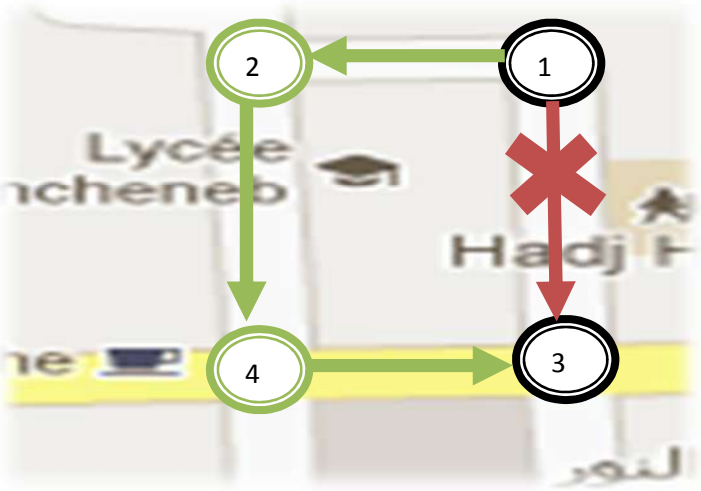


Figure II.5 : exemple du marquage du chemin

II.3.4 La matrice des arcs (notée : MAC)

C'est une matrice dont Chaque case contient le numéro d'arc reliant le point (indice ligne) avec le point (indice colonne) dans cet ordre s'il existe, ou la valeur « nan » qui indique qu'il n'existe aucun arc reliant ces deux nœuds ou si le sens et interdit entre ces derniers.

MAC est inséré dans l'annexe A.

Exemple : la figure (II.6) décrit l'arc 1 qui relie entre les nœuds 1 et 2.

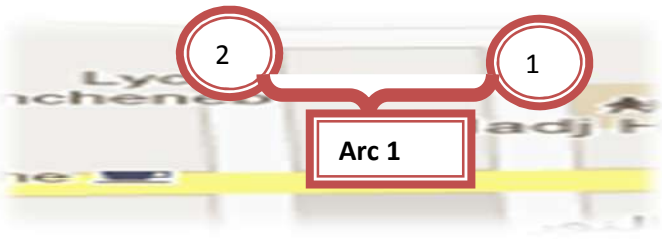


Figure II.6 : exemple d'arc reliant entre deux nœuds.

Evidemment, on trouve dans la case (1,2) le numéro 1, ceci est plus claire en jetant un œil sur la partie de la MAC suivante :

```
[nan 1 nan nan nan nan…
 nan nan nan 2 nan nan…].
```

II.3.5 La matrice des parkings (notée : MPK)

C'est une matrice inséré dans l'annexe A dont chaque case contient le numéro d'arc reliant le point (indice ligne) avec le point (indice colonne) dans cet ordre s'il existe et si le stationnement est

légale dans ce dernier, ou la valeur « nan » s'il n'existe aucun arc reliant ces deux nœuds ou si le sens et interdit entre ces derniers, ou s'il existe un arc entre eux mais le stationnement est illégal dans ce dernier.

Exemple : l'arc 1 reliant les nœuds 1 et 2 contient un espace de stationnement légal, alors la case (1,2) contient la valeur 1 correspondant au numéro d'arc, tandis que celui reliant entre les nœuds 4 et 3 ne contient aucun espace de stationnement, donc la case (3,4) contient la valeur « Nan » indiquant (not a number) en Matlab. La figure (II.7) explique cet exemple.

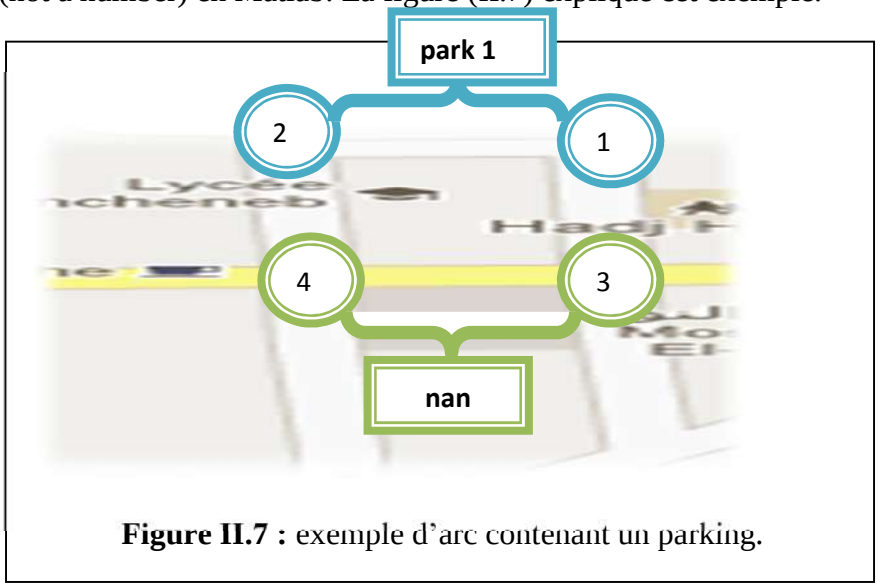


Figure II.7 : exemple d'arc contenant un parking.

Montrant une partie de la MPK afin de clarifier ceci :

```
[nan 1 nan nan nan nan…
 nan nan nan 2 nan  nan…
 3 nan nan nan nan nan…].
```

II.3.6 La matrice des espaces de stationnement (notée : MES)

La matrice des espaces de stationnement ou la MES contient autant de ligne que d'arcs dans cette région, ou chaque ligne décrit un arc correspondant à un espace stationnement autorisé, le numéro de ligne correspond à celui d'arc (parking), la matrice contient ainsi autant de colonne que de places de stationnement dans chaque parking.

La matrice MES est inséré dans l'annexe A.

Exemple : le parking 1 situé entre les points 1 et 2 contient 50 places de stationnement, ce qui correspond à la ligne 1 de la matrice « Z », ou une case contient la valeur 0 si la place correspondante est vide, et la valeur 1 dans le cas contraire. Prenant comme exemple une partie de Cette ligne afin de mieux éclairer:

[1 1 1 1 1 1 1 1 1 0 0 0 0…].

La figure(II.8) en montre un exemple.

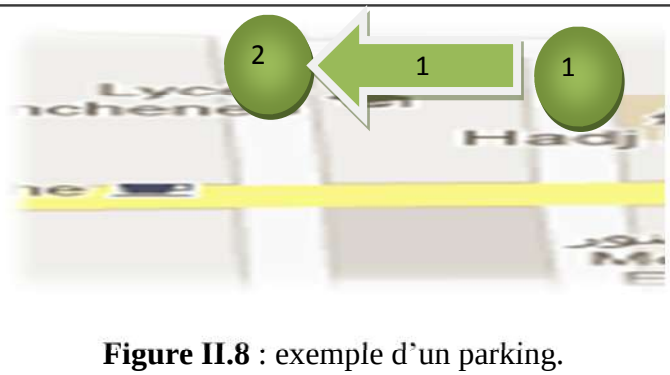


Figure II.8 : exemple d'un parking.

- les numéros de lignes qui correspondent à des arcs dont le stationnement est interdit , ne sont jamais visité par le module de recherche de la place de stationnement.

II.3.7 La matrice des directions (notée : MDR)

Le but de la conception de cette matrice est de minimiser le nombre de tests qui s'avère très élevée, la MDR permet de définir la direction entre chaque deux arcs de la zone géographique traité, il ne reste que de consulter la case reliant ces deux arcs, En effet, on trouve la valeur « 1, 2, 3…6 » qui détermine la direction à choisir, dont la signification est la suivante

1 : roulez directement.

2 : tournez au premier virage à droite.

3 : tournez au premier virage à gauche.

4 : tournez au deuxième virage à droite.

5 : tournez au deuxième virage à gauche.

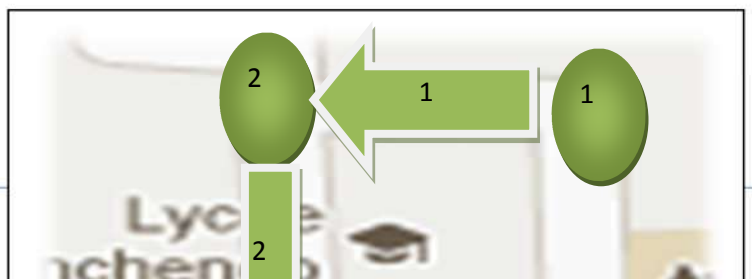
6 : vous pouvez vous garer ici.

Une case de la MDR contenant la valeur « nan » (not a number) signifie que les deux arcs (indice ligne, indice colonne) de cette dernière ne possèdent aucune relation entre eux (non-adjacents).

La matrice MDR est inséré dans l'annexe A.

Exemple : pour déplacer à partir de l'arc 1 vers l'arc 2 , on doit tourner à gauche, ceci est définit dans la MDR comme suit

[nan **3** nan nan nan nan…], ceci est montré dans La figure (II.9).



II.4 Algorithme utilisé

Trouver une place de parking à partir d'une position quelconque peut se transformer en un problème de recherche du plus court chemin entre deux points sous la contrainte de place disponible et autorisée pour le parking.

Pour résoudre ce problème, on a pensé à utiliser l'algorithme de Floyd (voir chapitre 1), La figure (II.10) illustre l'organigramme mis en place dans Matlab.

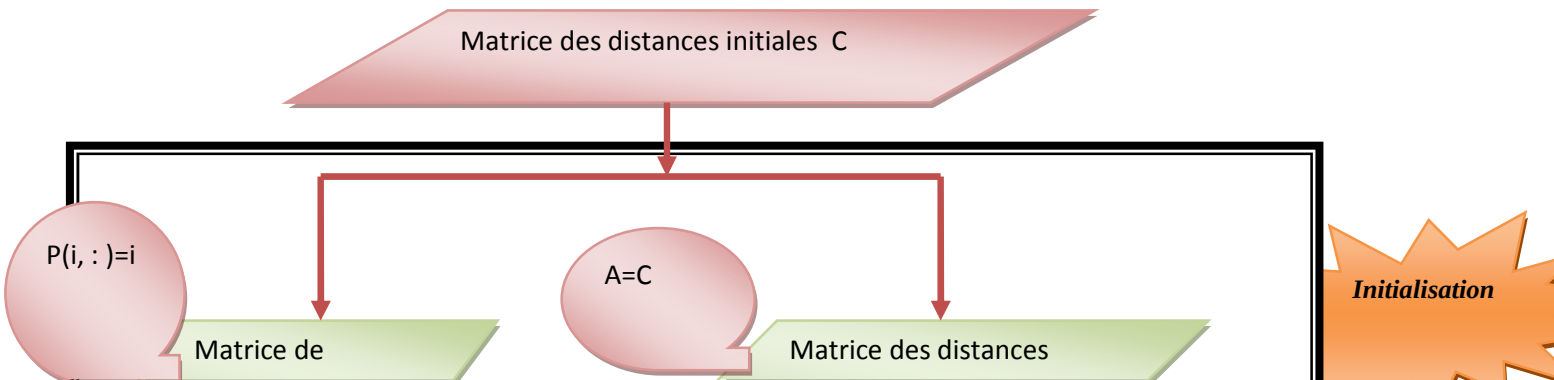


Figure II.10 : principe de l'algorithme de Floyd.

II.5 Simulation avec Matlab

II.5.1 Principe général de fonctionnement :

En Matlab on se propose de réaliser une application qui reçoit la position d'un véhicule cherchant une place pour se garer, et qui se sert d'une matrice de description d'une région géographique (matrice des distances initiales) pour donner la direction vers une place de parking vide et la plus proche possible.

Pour cela plusieurs procédures sont utilisées, chacune s'occupe de fournir une solution pour une partie du problématique, citant quelque une parmi celles les plus importantes tout en exposant le fonctionnement globale du programme :

II.5.1.1 La procédure Floyd

Le fonctionnement de cette procédure est identique à celui de l'algorithme de floyd expliqué précédemment dans ce chapitre.la figure (II.11) illustre l'ensemble des entrées/sorties de cette procédure

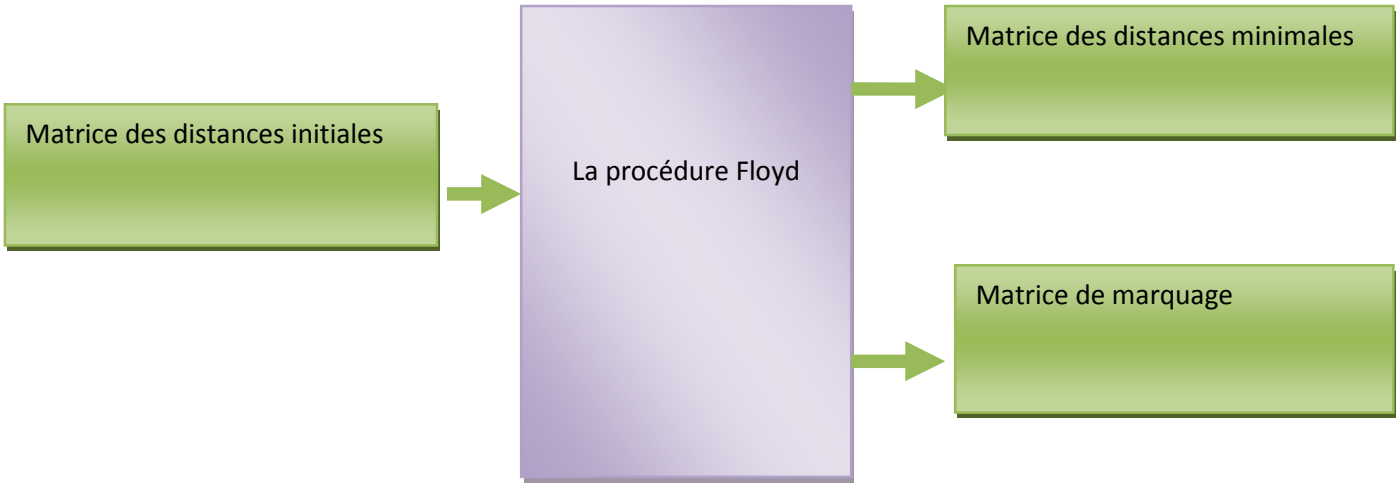


Figure II.11 : entrées/sorties de la procédure Floyd.

Floyd nous donne en sortie la matrice des distances minimales ainsi que celle de marquage, avec ces deux matrices on pourra continuer le traitement.

- Après avoir une description détaillée de la zone géographique notre solution pourra utiliser les différentes procédures et fonctions nécessaires pour répondre au différents besoins de programme réalisé.
- On utilise la matrice des espaces de parking pour vérifier les espaces de stationnement libres à chaque fois qu'un véhicule demande une place pour se garer.

II.5.1.2 La procédure distances

Cette procédure permet d'avoir en sortie le parking le plus proche tout en respectant la position du demandeur, et en évitant qu'il emprunte un sens interdit ou qu'il fasse un demi-tour interdit.
La figure (II.12) démontre les entrées/sorties de cette procédure. Le fonctionnement de la procédure distance est illustrée par la figure (II.13).

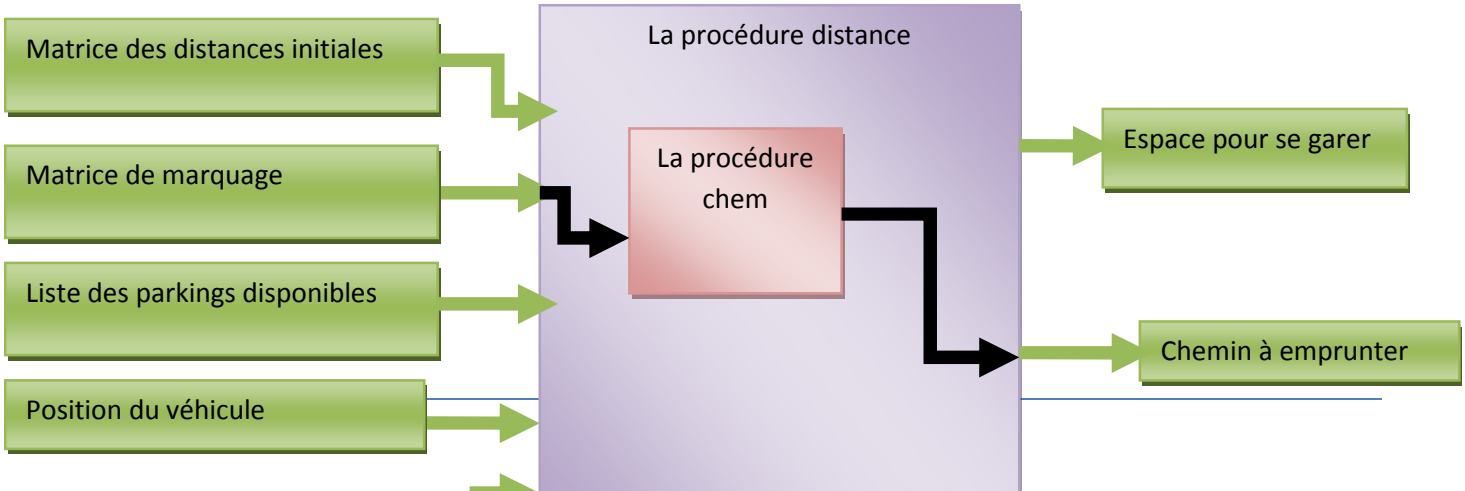


Figure II.12 : les entrées/sorties de la procédure distance.

La procédure distance effectue un appel à la procédure « chem », cette dernière renvoie le chemin à emprunter à partir de la matrice de marquage tout en fournissant comme entrée le point de départ et celui d'arrivé.

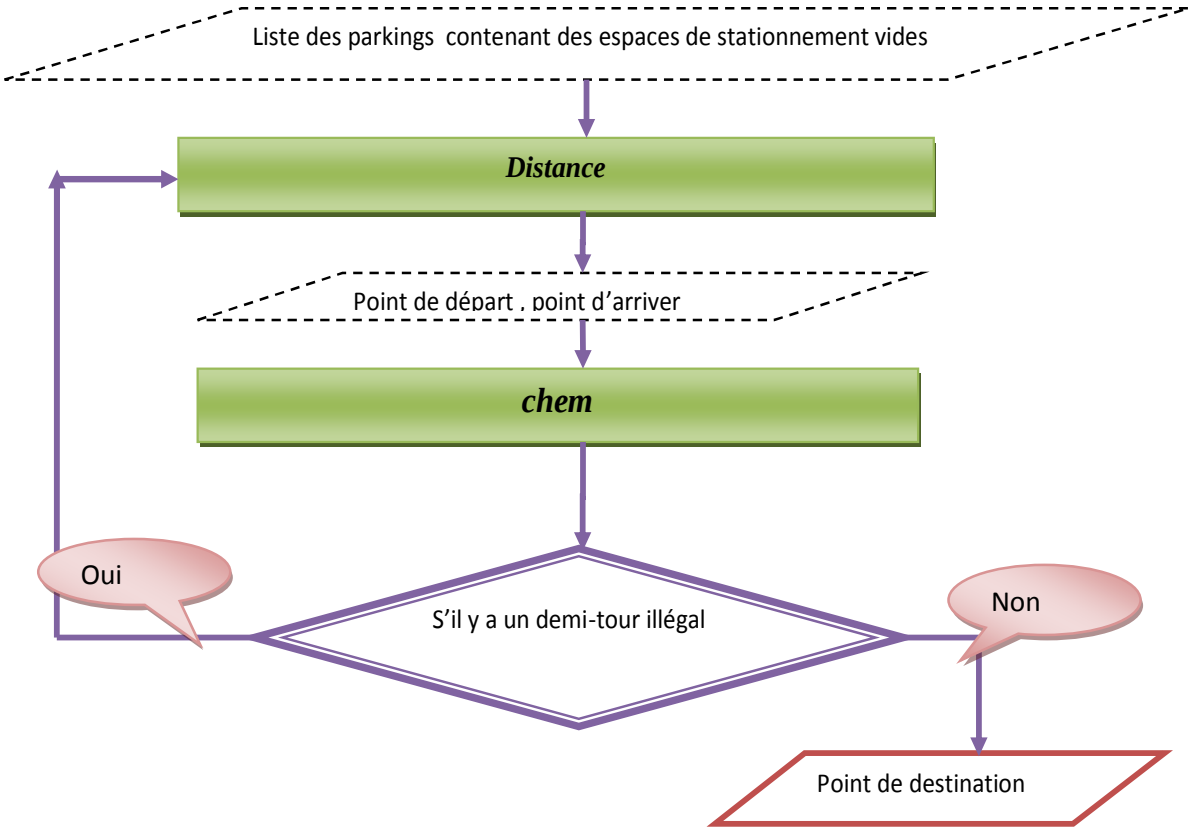


Figure II.13 : fonctionnement de la procédure distance.

II.5.1.3 La procédure dispo

Cette procédure est appelée afin de vérifier la disponibilité de places de stationnement dans un parking. La figure (II.14) montre les entrées/sorties de cette procédure.

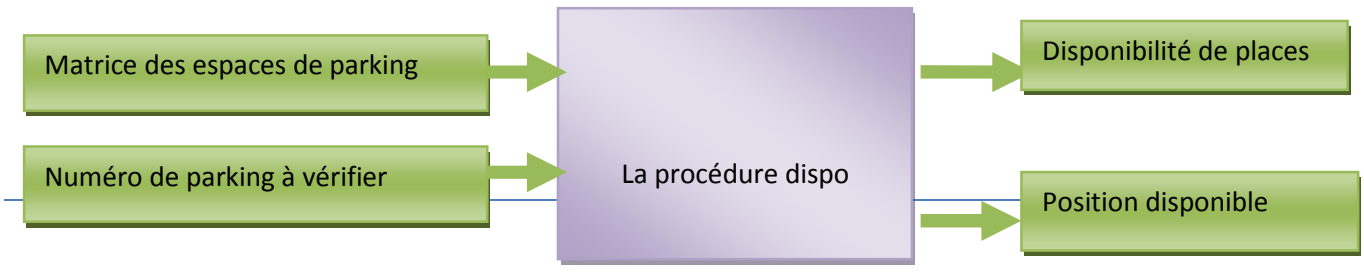


Figure II.14 : entrées/sorties de la procédure dispo.

La figure (II.15) illustre le principe de fonctionnement de la procédure dispo.

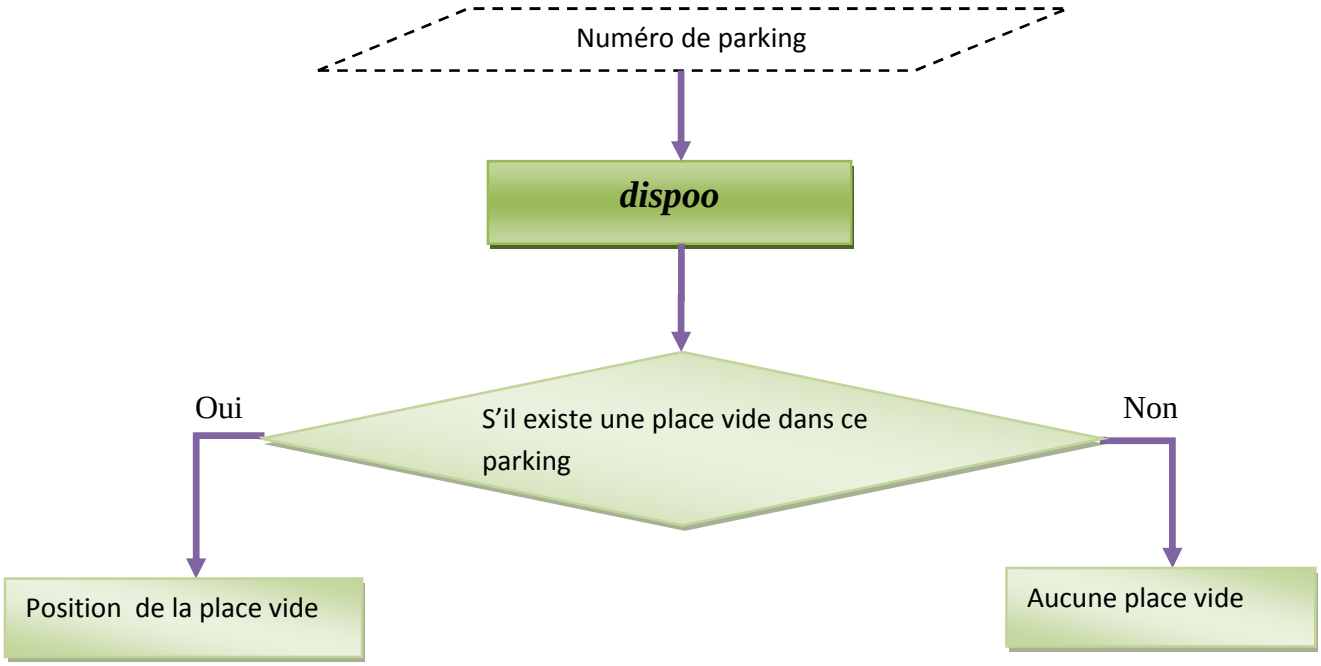


Figure II.15 : principe de la fonction dispo.

II.5.1.4 La procédure « direction »

Cette procédure s'occupe de fournir les directions, afin de guider le conducteur vers sa place de stationnement. En vue de diminuer les tests dans le programme, et à cause du nombre de possibilités très élevés, notamment, on peut atteindre 4 condition pour chaque arc ce qui fait (4*80 arcs), ceci est évité grâce à l'utilisation d'une matrice dite (matrice des arcs) qui permet de déterminer la direction à choisir pour déplacer entre n'importe quel deux arcs.

La figure (II.16) montre les entrées/sorties, et la figure (II.17) illustre le principe de fonctionnement de cette procédure.



Figure II.16 : entrées/sorties de la procédure direction.

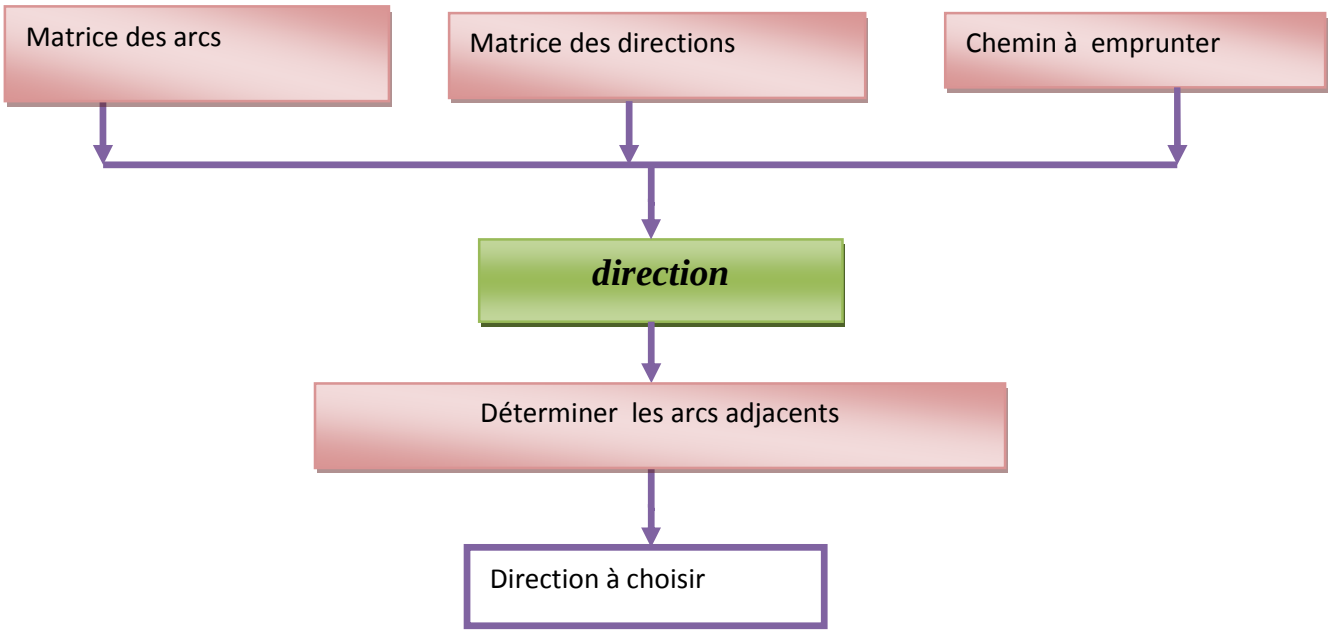


Figure II.17 : principe de la procédure direction.

II.5.1.5 La procédure « vériff »

A ce stade, Le chemin à emprunter est prêt il ne reste que vérifier le bon acheminement du véhicule vers sa place de stationnement et la disponibilité de celle-ci avant qu'il y arrive, cela est garantie à travers la procédure « veriff ». La figure (II.18) permet de clarifier les entrées/sorties de cette dernière.

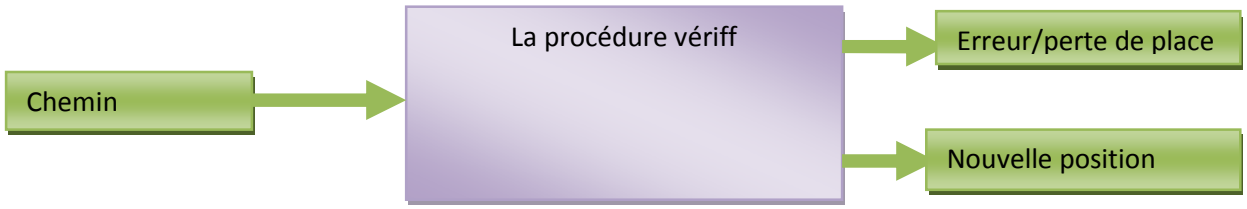
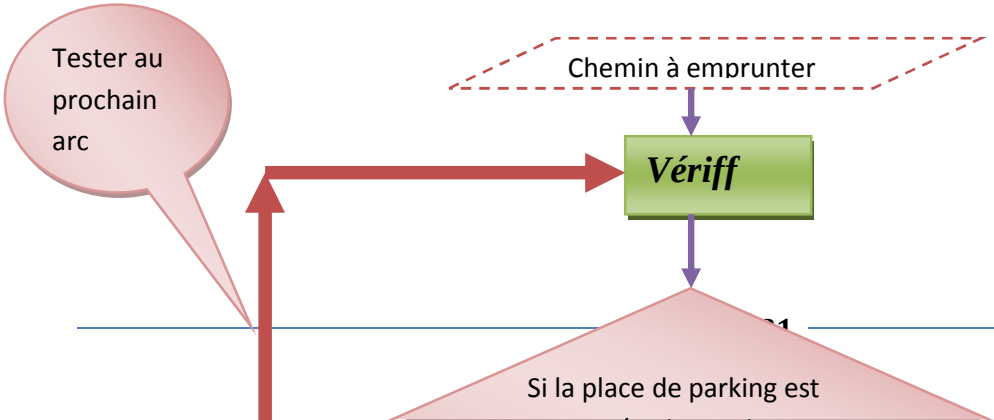


Figure II.18: les entrées/sorties de la procédure veriff.

- La figure (II.19) illustre le principe général de la procédure veriff.



Non Oui

Figure II.19 : le principe de fonctionnement de la procédure veriff.

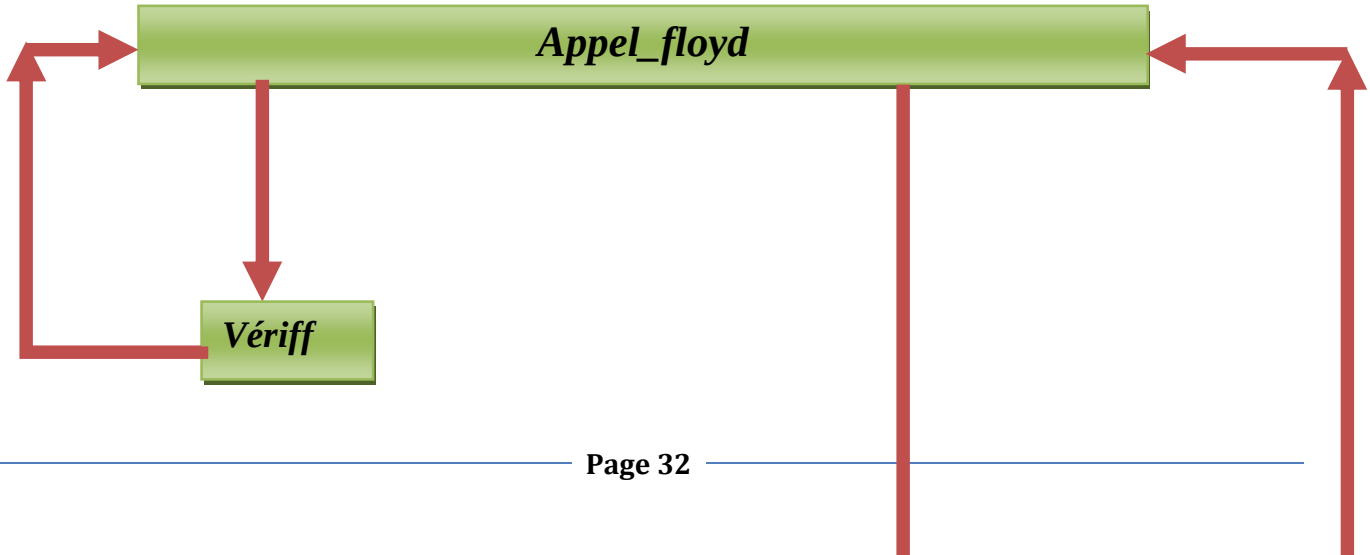
II.5.2 Principe de collaboration :

L'organigramme figuré dans la figure (II.20), explique la collaboration entre ces différentes procédures afin de générer un model complet dont :

- On symbolise les procédures en vert.

- **La procédure appel_Floyd**

  Cette procédure est considérée comme procédure principale, elle reçoit toutes les données nécessaires pour le fonctionnement du modèle et s'occupe de faire toutes les appels aux autres procédures et fonctions.



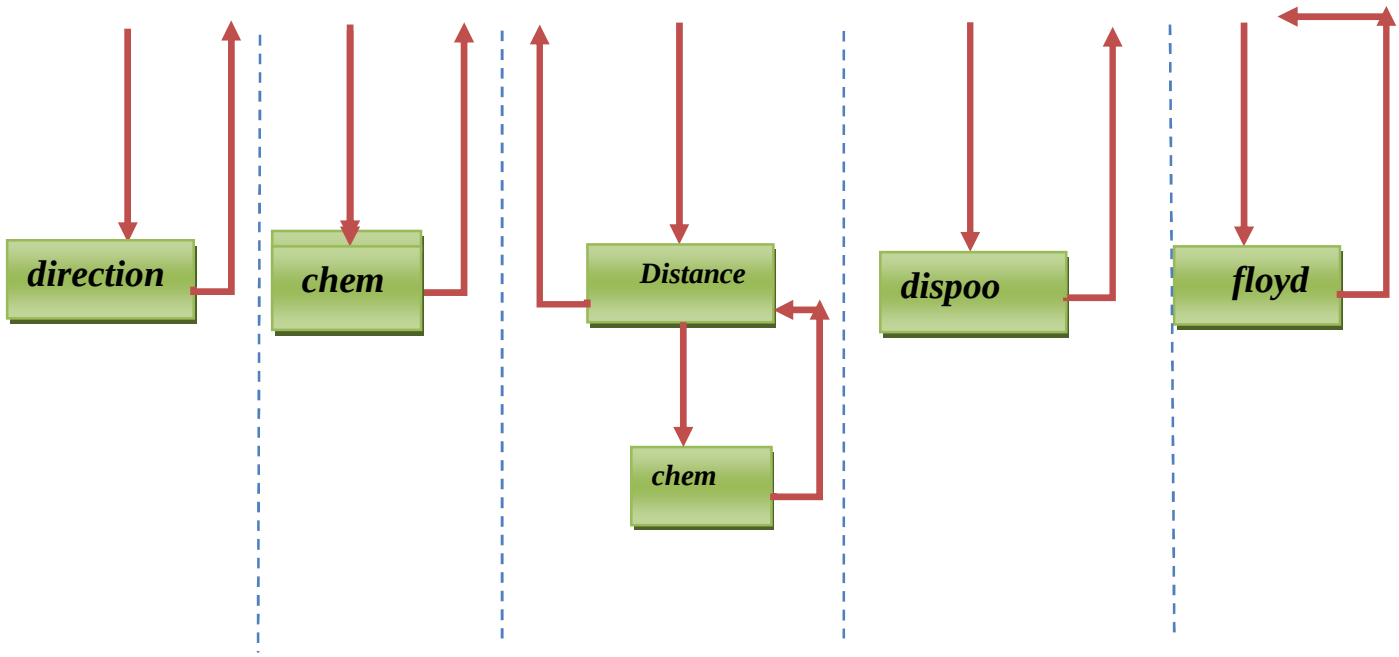


Figure II.20 : principe de collaboration.

II.5.3 Exemples de traitement

- Le modèle est en exécution tant qu'il existe une place de stationnement vide dans toute la région, en attendant une requête d'un conducteur pour se garer :

Exemple1

Voulez-vous vous garer? : 1 pour oui 0 pour non1

segarre = 1 (cette variable décrit la réponse de l'utilisateur ici c'est oui)

Demande de position

Prenant comme exemple un véhicule qui se trouve entre les pts 1 et 2 cela signifie qu'on est dans la position et le sens montré par la figure (II.21)

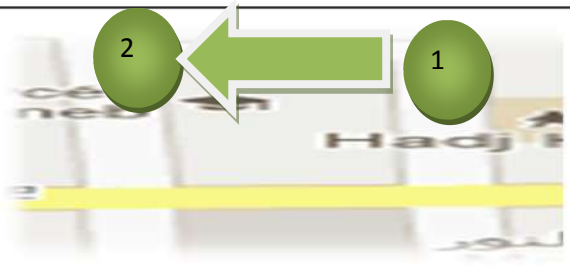


Figure II.21 : exemple de position.

Le modèle renvoie le chemin à emprunter :

[1 2 20 19 24] (Les nœuds à visiter en vue d'arriver à la place de parking vide)

En allant du point de départ vers le point d'arrivé, l'utilisateur est suivi par le model afin de vérifier régulièrement la disponibilité de sa place avant qu'il y arrive, ainsi s'il emprunte la bonne direction :

Le parking situé entre les points (19→24) contient 2 places vides, jetant un œil sur la partie de la MES qui détermine ceci, les deux dernières valeurs nulles correspondent aux places vides.

`[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1…0 0]`

Supposant que l'une de ces deux places est devenues pleine avant que ce véhicule y arrive, les détecteurs placés dans chaque espace de stationnement permettent de transmettre une information est envoyé au modèle indiquant que la place est prise.

Et le chemin à emprunter est : [1 2 20 19 24] malgré que la places est devenue pleine et puisque le parking contient une autre place de stationnement vide, ce véhicule est guidé vers le même parking, et cela continuera jusqu'au remplissage de tous les espaces de ce parking :

Supposant que la deuxième place est devenue pleine avant que ce véhicule y arrive, une information est envoyée au modèle indiquant que la place est prise. Après que le parking soit complètement plein une autre place est envisagée. Et le chemin à emprunter est :

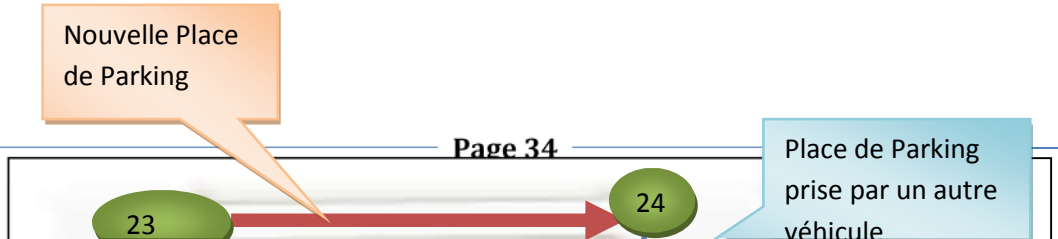
[1 2 20 23 24]

→Et cette vérification continuera tout le long du chemin i.e. dans ce cas la procédure de vérification se lancera 4 fois (en parcourant chaque arc). Et le chemin à emprunter est :

[1 2 20 23 24] le conducteur reçoit l'ensemble des directions tout le long du chemin.

Direction =Tournez au premier virage à droite ---->Roulez directement ---->Tournez au premier virage à droite ----> vous pouvez vous garer ici.

Ceci est illustré dans la figure (II.22) :



Exemple2 :

Le modèle vérifie à tout instant que le conducteur a pris le bon chemin, et s'il se trompe (information donnée par le GPS), le système l'oriente vers un autre emplacement pour se garer, selon sa nouvelle position.

Prenant le même exemple pour le cas d'erreur de direction : le conducteur est supposé d'emprunter le chemin suivant

[1 2 20 23 24]

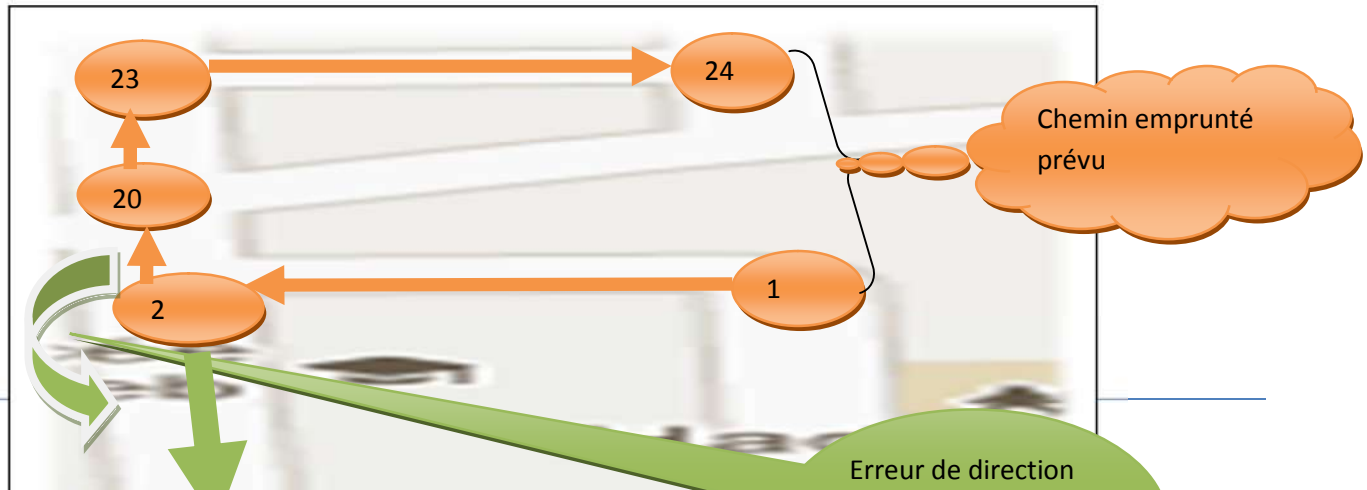
Au lieu de prendre l'arc (2→20) le conducteur à pris l'arc (2→4) , la nouvelle position est prise du GPS (2→4)

Ainsi une nouvelle place de stationnement est envisagée pour ce véhicule suivant sa nouvelle position, et le chemin à emprunter est le suivant :

[2 4 6]

Direction =Roulez directement ----> vous pouvez vous garer ici.

la figure (II.23) permet de mieux expliquer ce cas :



Exemple3 :

Le modèle prend en compte l'état légal de la circulation, en évitant par exemple le demi-tour interdit. Prenant un exemple concret en vue d'illustre ceci :

Un véhicule se trouve dans la position 17→18 (dans ce sens), demande un espace de stationnement, sachant que cette position contient un parking légal et disponible dans le sens opposé i.e. dans le sens (18→17). Le modèle l'oriente vers le parking situé dans (18→26), en le dirigeant comme suit

[17 18 26]

Direction = Tournez au premier virage à droite ----> vous pouvez vous garer ici.

La figure (II.24) aide à clarifier ce cas :

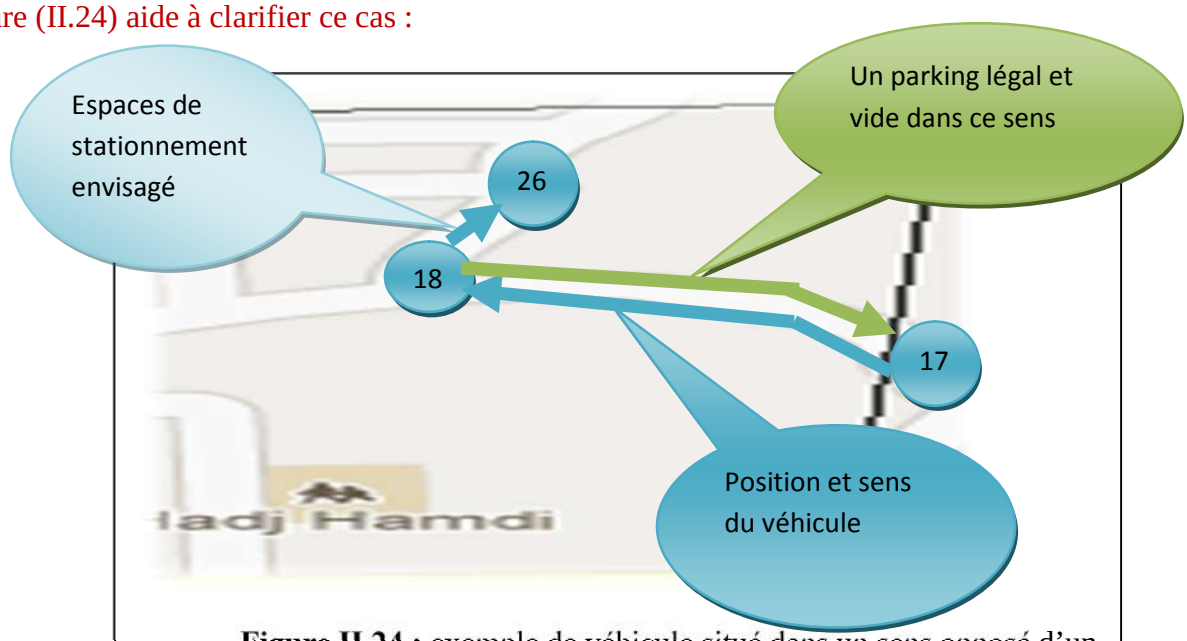


Figure II.24 : exemple de véhicule situé dans un sens opposé d'un parking

Exemple 4 :

- Le demi-tour peut être réalisé d'une manière légale grâce aux ronds-points, l'exemple suivant traite un cas pareil.

  (Un véhicule qui se trouve dans la position 4→9, sachant que le point 9 est un rond-point) ,

  Et l'arc (4→6) contient des espaces de stationnement vides.

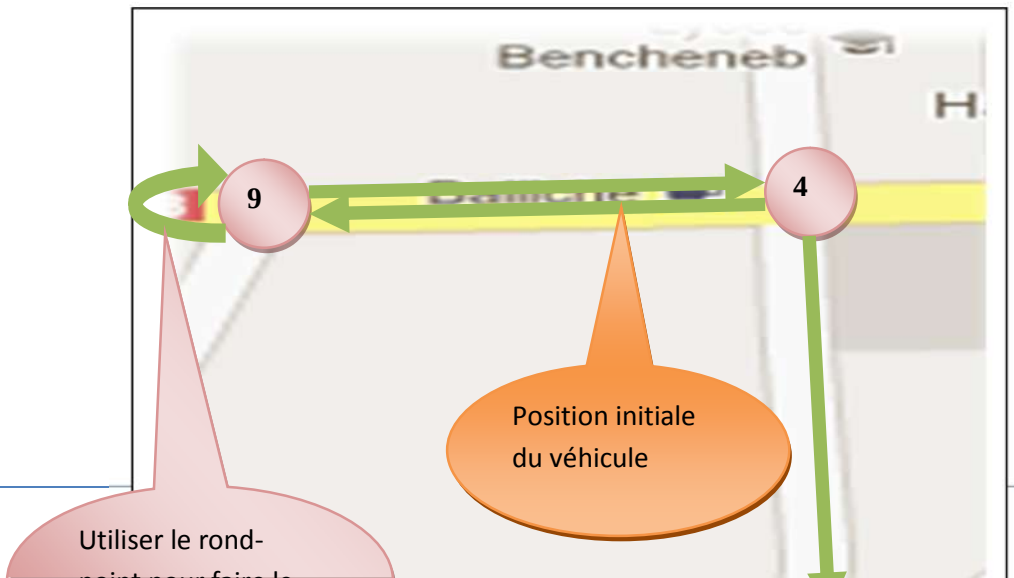
  Le chemin renvoyé par le modèle est donné par :

  [4 9 4 6]

  L'ensemble des directions à prendre est le suivant :

  Direction = utiliser le rond-point pour tourner au sens inverse--->Tournez au premier virage à droite ---->vous pouvez vous garer ici.

  Ceci est mieux expliqué par la figure (II.25).



II.6 Conclusion :

En réalisant le programme de simulation sous Matlab, on s'est trouvé face à plusieurs contraintes notamment :

- La prise en compte des sens interdits.
- La possibilité de perdre la place de parking avant d'y arriver.
- La possibilité de se tromper de direction.

Cette réalisation repose sur l'utilisation de quelques procédures dont chacune s'occupe de garantir la prise en compte d'une certaine contrainte imposée soit par la région géographique soit par le bute de développement de cette application, cela assure l'aspect dynamique de cette dernière, en effet pour changer de région géographique il suffit de changer la matrice des distances initiales(MDI).

Chapitre III

Implémentation VHDL

III.1 Introduction

Après avoir passé par la structure du parking dans la partie théorique, ainsi que la modélisation mathématique de la problématique, nous passons dans ce chapitre à l'implémentation de notre solution.

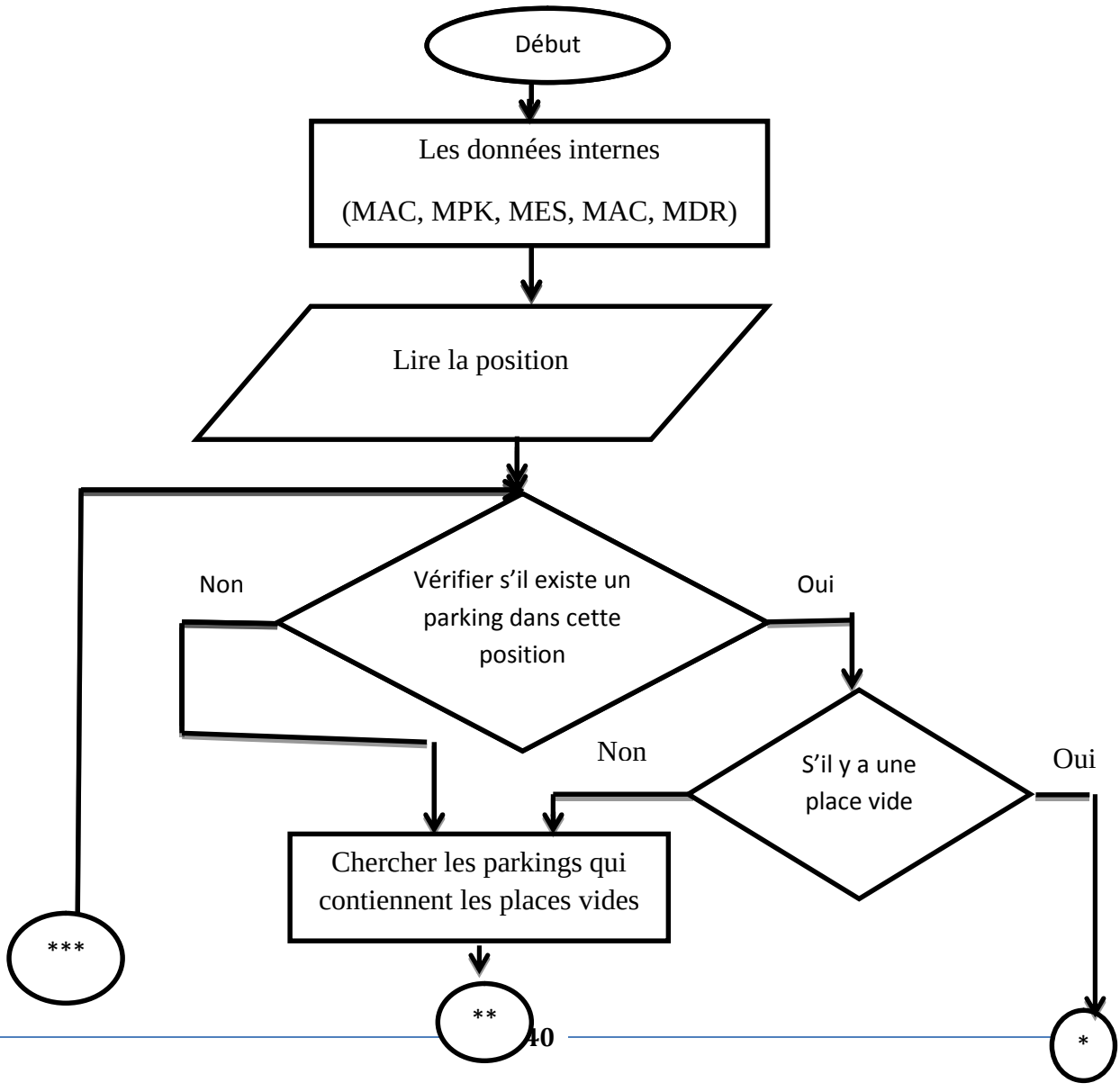
Nous commençons tout d'abord par la description détaillé d'algorithme utilisé ensuit quelques détails sur le code VHDL, puis des tests et des analyses de notre système.

III.2 Principe général

Le principe général de guidage du conducteur vers le plus proche parking vide, ainsi que celui de poursuite tout le long du chemin parcouru, est expliqué dans la figure (III.1)

III.2.1 Description de l'algorithme

La figure III.2 représente le schéma global de stationnement dans un parking.



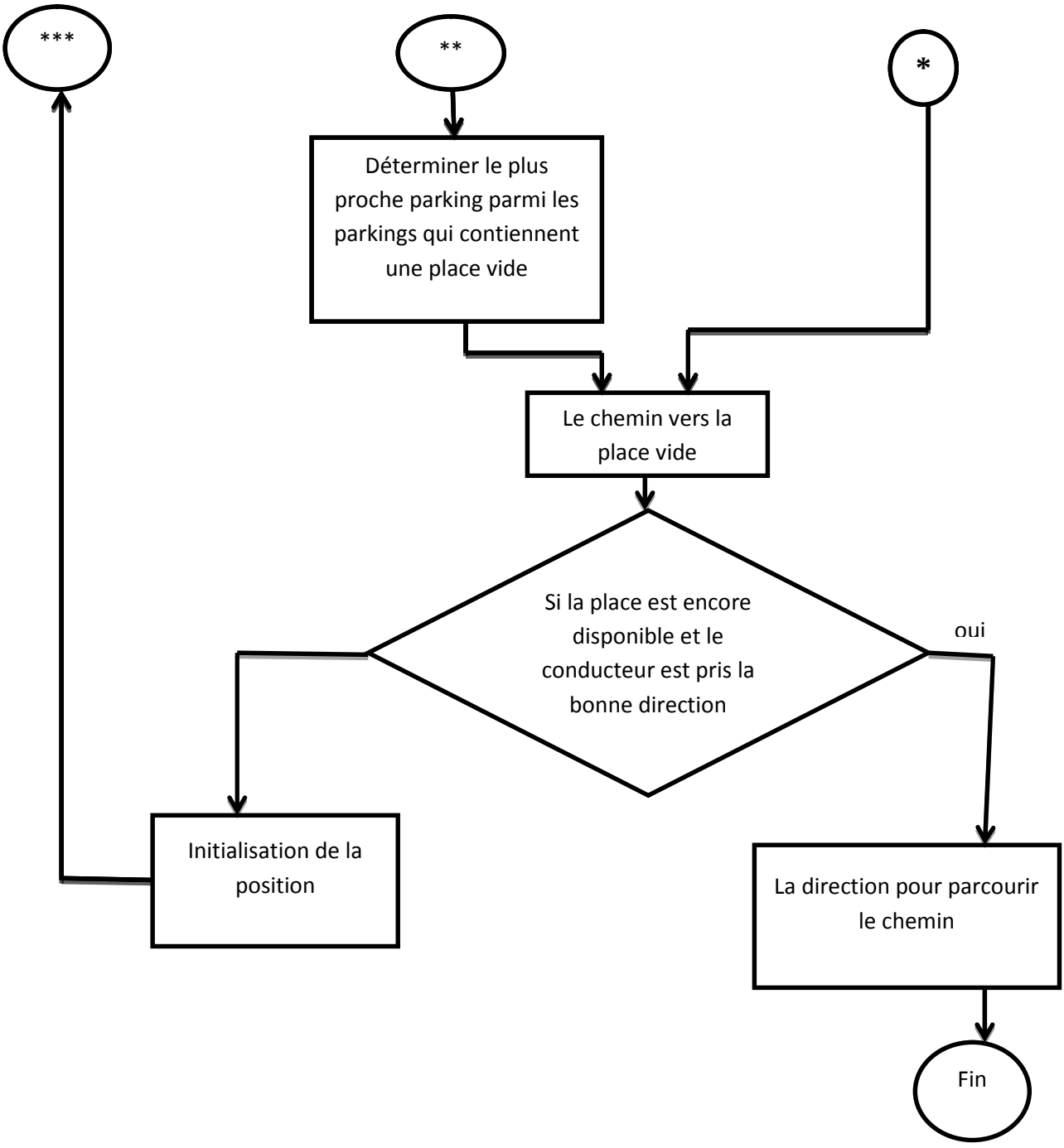


Figure III.2 : schéma globale de stationnement dans un parking.

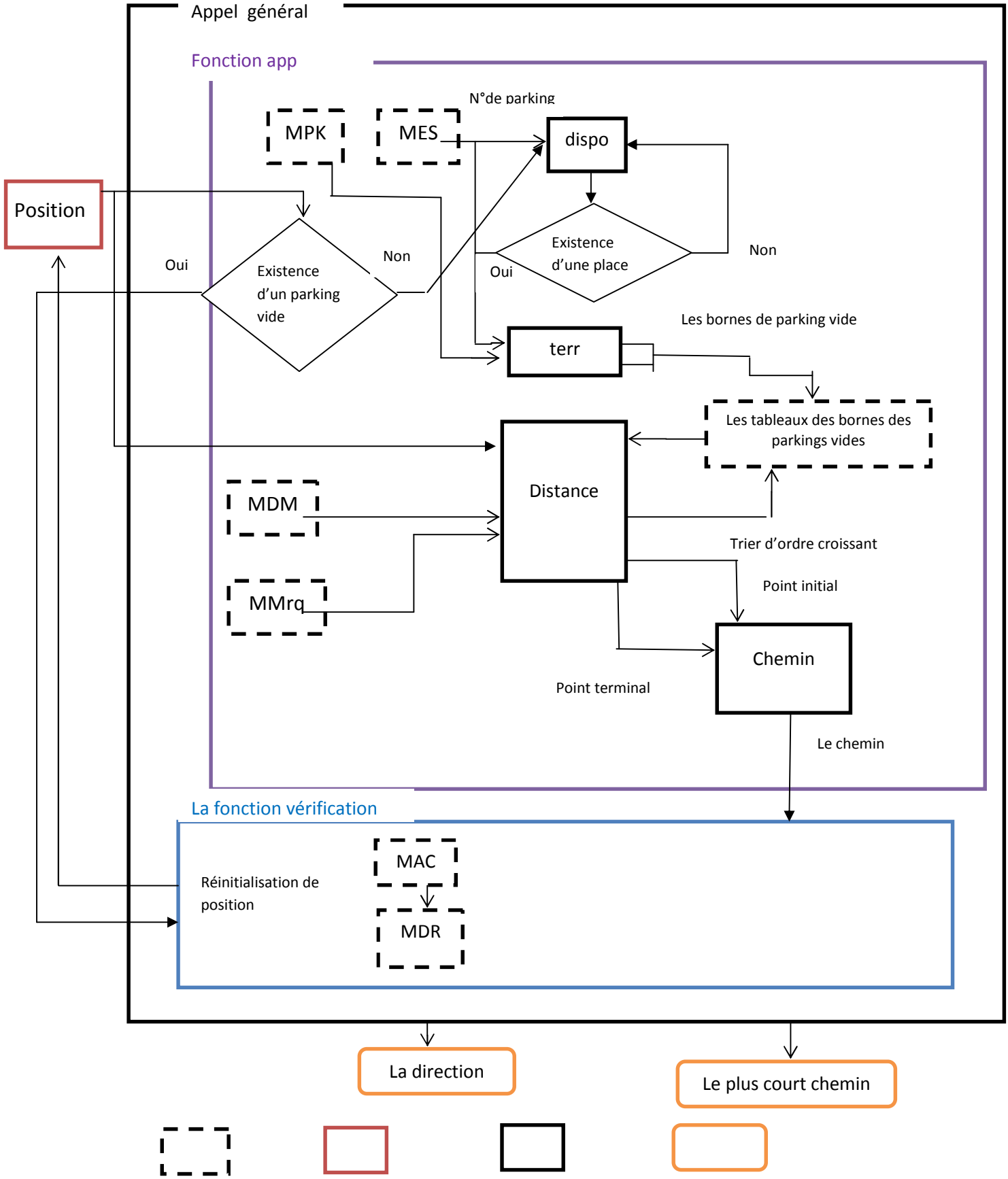


Figure III.1 : schéma globale de fonctionnement de système de stationnement.

III.2.2 Pseudo code de fonctionnement du système

Le déroulement des taches de notre projet peut être résumé par le pseudo code suivant :

1-les données : C (la matrice des distances initiales).

Park (la matrice de numérotation des parkings).

Z (la matrice de disponibilité des places vides dans les parkings).

Matt (la matrice de numérotation des arcs).

M (la matrice de direction).

2-Entrées : la position de véhicule.

3-s'il existe un parking dans l'arc enter Dp et Ar alors

4-s'il existe une place vide alors

-aller à étape 8.

5-sinon à étape 7.

finsi.

6-sinon aller à étape **7.**

7-chercher les parkings qui contiennent une place vide.

-chercher le plus proche parking contient la place vide (park vide).

-aller à étape **8.**

8-acheminement vers parking vide.

9-si la place est encore disponible et le conducteur a pris la bonne direction alors

-diriger le conducteur vers cette place vide (direction).

Sinon

-initialisation de la position.

-aller à étape **3.**

Finsi.

Fin.

III.3 Réseau routier

La figure III.3 représente le réseau routier d'une partie du centre de Médéa.



Rond-point

figureIII.3 : réseau routier d'une partie du centre-ville de Médéa.

III.4 Description détaillé du code VHDL

III.4.1 Les entrées

- ✓ **La position :** c'est la position de véhicule donné sous forme de deux points dp et ar par exemple si le véhicule est situé dans l'arc 0 qui est borné par le point 0 et 1, on prend comme entrée dp=0 et ar=1 ,ou dp et ar sont de type entier.

III.4.2 Les sorties

- ✓ **Le chemin (ch) :** le chemin dont le véhicule doit emprunter afin d'arriver au plus proche parking contenant une place vide (le chemin contient tous les points visités pour arriver à la place vide dans le plus proche parking), le chemin de type vecteur d'entiers.
- ✓ **La direction (direct):** ensemble de directions à prendre pour parcourir le chemin et arriver au plus proche parking vide.

III.4.3 Les données internes

- ✓ **Matt :** MAC de taille 31 × 31 d'entiers correspondant au nombre des nœuds.
- ✓ **C :** MDI de taille 31 × 31 d'entiers correspondant au nombre des nœuds.
- ✓ **Park :** MPK de taille 31 × 31 d'entiers correspondant au nombre des nœuds.
- ✓ **Z :** MES de taille 80×50 de bit, 80 arcs contenant chacun 50 places.
- ✓ **M :** MDR de taille 80 × 80 (80 arcs) d'état tel que l'état correspond à direction entre les arcs :

  et1 : roulez directement.

  et2 : tournez au premier virage à droite.

  et3 : tournez au premier virage à gauche.

  et4 : tournez au deuxième virage à droite.

  et5 : tournez au deuxième virage à gauche.

  et6 : vous pouvez vous garer ici.).

- ✓ **L :** MDM de taille 31 × 31 d'entiers (31 nœuds).
- ✓ **P :** MMrq de taille 31 × 31 d'entiers (31 nœuds).

III.5 Implémentation en VHDL

Notre système est composé d'un ensemble de modules (procédures/fonctions) dont chacune assure une fonction parmi lesquelles les plus important on cite :

- Le module examinateur de disponibilité.
- le module identificateur du point d'arrivé
- Le module identificateur de l'arc correspondant au plus proche parking vide.
- Le module de restauration du chemin.
- Le module superviseur de collaboration.
- Machine d'état ou le module contrôleur.

III.5.1 Le module examinateur de disponibilité (disponible)

Ce module se charge de tester s'il existe une place vide dans un parking, Il prend comme entrée le numéro de parking et renvoie 2 sorties : l'une binaire égal à 1 s'il y a une place vide sinon égal à 0, et l'autre correspond à la position de la place vide. La figure III.4 explique le fonctionnement de ce module.

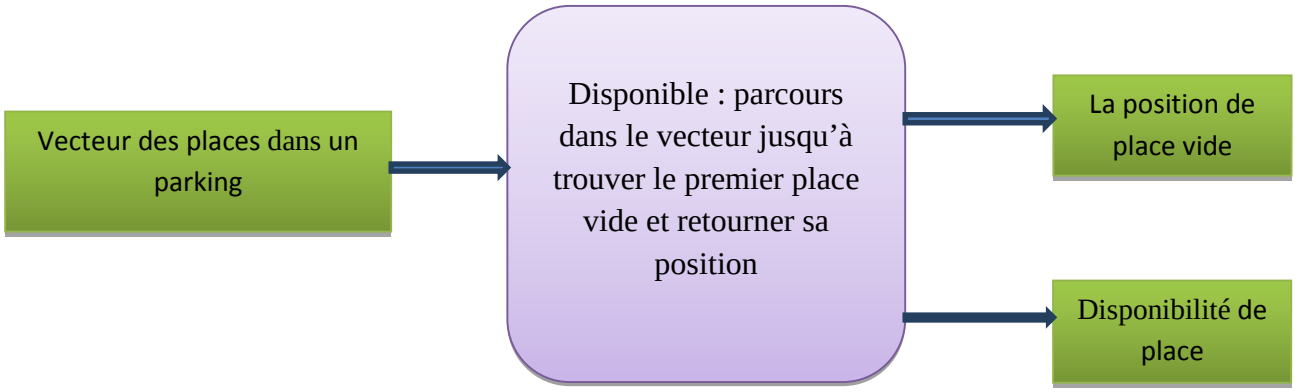
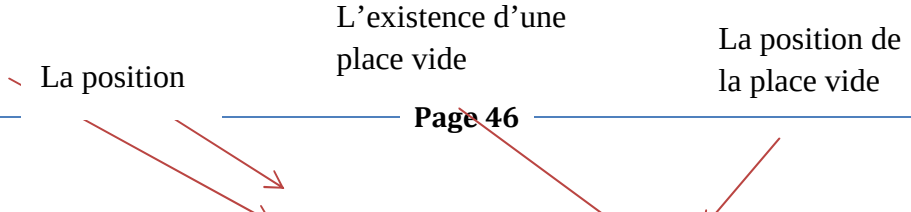


Figure III.4 : fonctionnement du module disponible.

- **Simulation**

Pour expliquer le fonctionnement du module disponible, on a choisi un exemple prenant la position (6→7). Le résultat de simulation est donné par la figure III.5



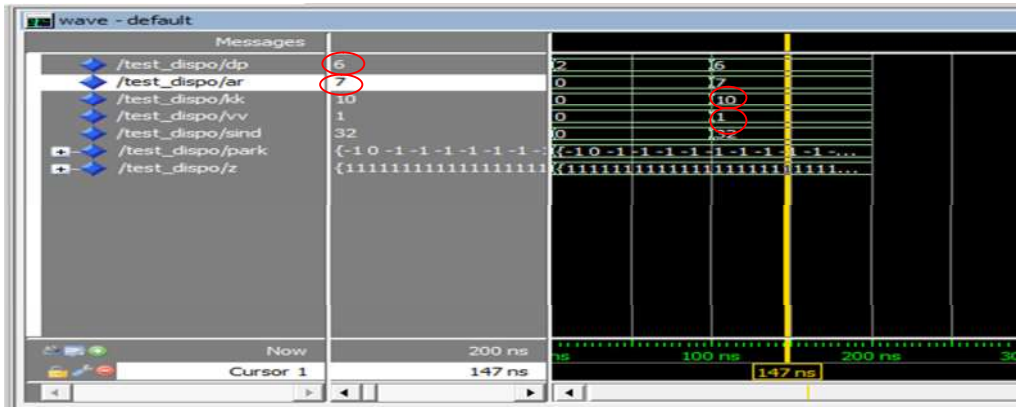


Figure III.5 : la simulation du module disponible en modelsim.

On trouve dans le résultat de simulation vv égal à 1 (vv : variable indiquant la disponibilité d'une place), (c.-à-d.) il y a une place vide dans le parking qui situé dans (6→ 7) , et kk= 10 (kk : variable indiquant la position de la place vide) .

III.5.2 le module identificateur du point d'arrivé (term)

Ce module s'occupe de déterminer le point d'arrivé en déterminant le point d'accès au parking donné comme paramètre.

La figure III.6 représente les entrées /sorties du module « term ».

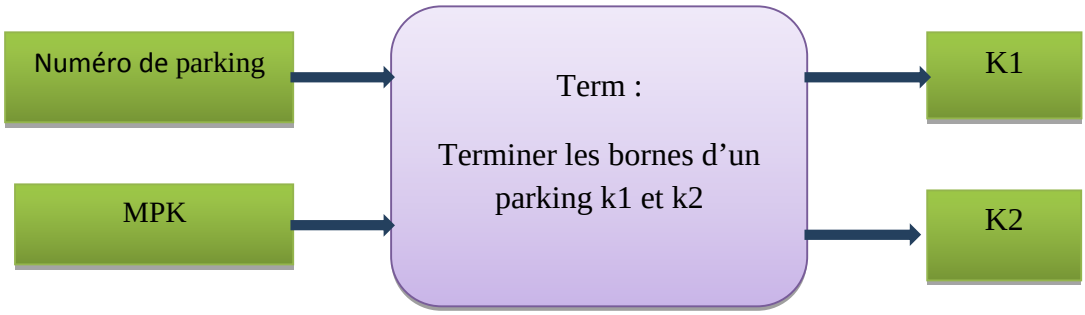
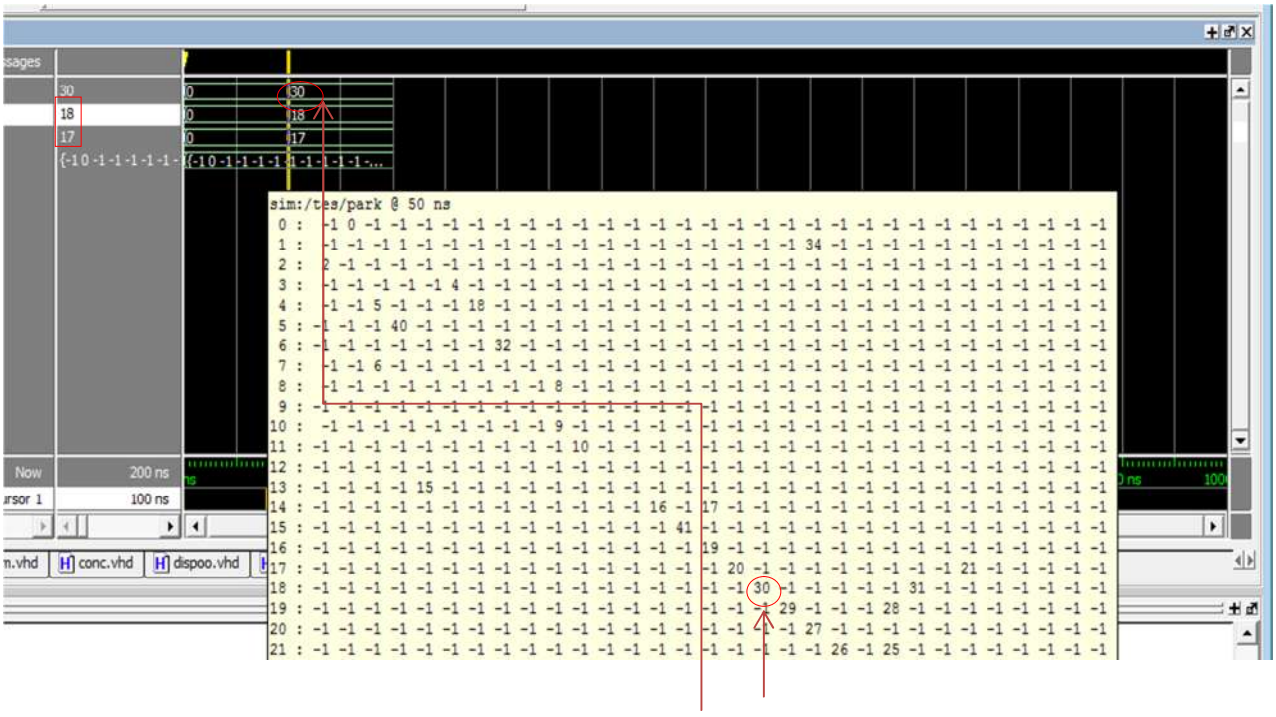


Figure III.6 : le module qui déterminer le point d'arrivé.

- **Simulation**

La figure (III.7) est une simulation de la fonction term avec entrée n=30 (le numéro de parking) qui donne les bornes de ce parking (b1=18 et b2=17).

Les bornes de parking 30



La figure III.7 : exemple de simulation de fonction term.

III.5.3 Le module identificateur de l'arc correspondant au parking vide le plus proche (distance)

Ce module se charge de trier les listes des parkings contenants des espaces vides (donné comme paramètre), selon leur distance par rapport au point de départ (entré comme paramètre) en utilisant la MDM (mesure de distances entre les points).

Son fonctionnement est expliqué dans le chapitre II aves une illustration par la figure II.13.

- **Simulation**

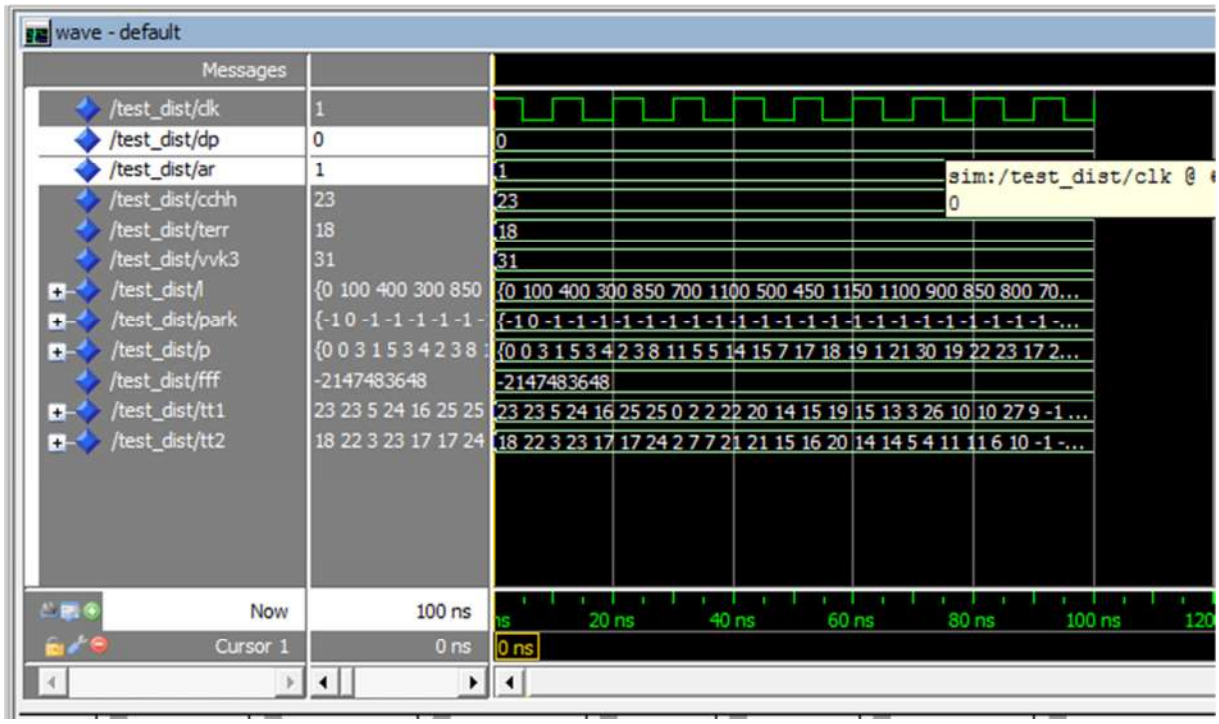
  En prend comme exemple la position (0→1) et le test contient deux tableaux qui correspondent aux bornes des parkings.

  Le résultat donne le parking le plus proche par rapport à la position donnée.

  Cet exemple est montré dans la figure III.8

La position

Les bornes de parking proche à la position



Les bornes des parkings proposés

Figure III.8 : la simulation de module distance.

III.5.4 Le module de restauration du chemin

Ce module s'occupe de la restauration du chemin à partir de la matrice MMrq, il reçoit en entré le point de départ et celui d'arrivé et renvoi le chemin à emprunter.

- **Simulation**

  La figure III.9 représente la simulation de la procédure chemin :on propose comme entrée init(point de départ)=0 et term(point d'arrivé)=5 , et le résultat de cet procédure, est le plus court chemin entre init et term .

Point initial (départ)

Point terminal

Le plus court chemin entre 0 et 5

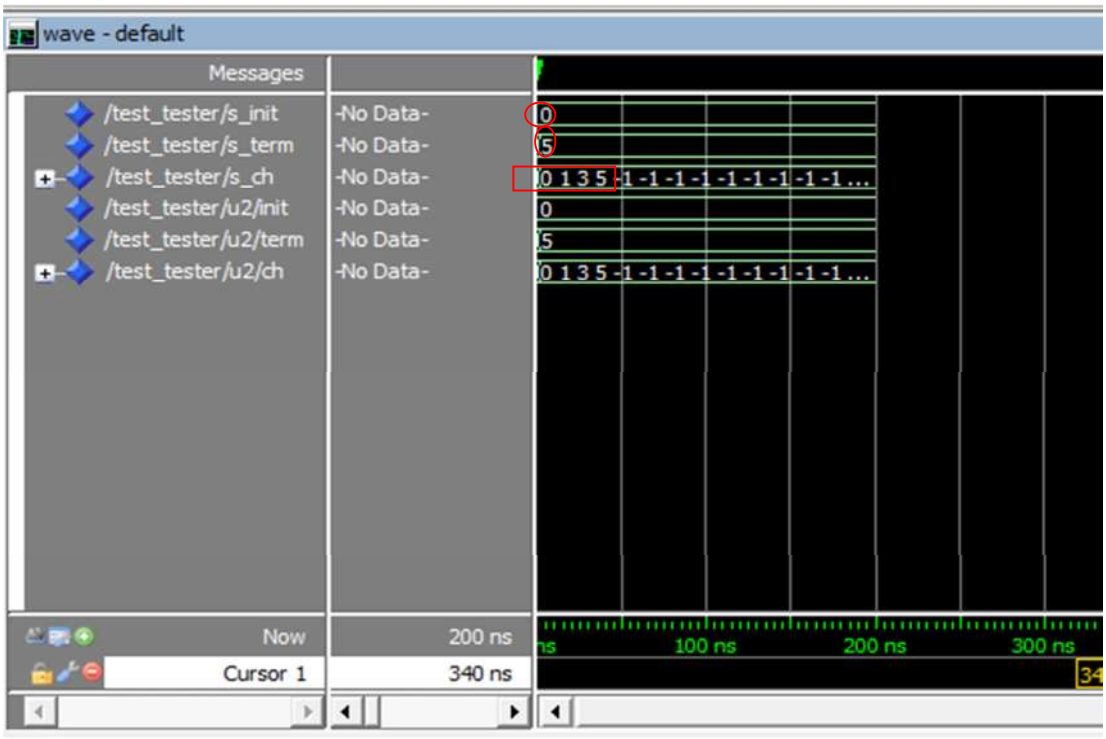


Figure III.9 : la simulation de module de restauration du chemin.

III.5.5 Le module superviseur de collaboration (app)

Cette procédure s'occupe d'effectuer les appels aux autres procédures en assurant pour chacune les paramètres nécessaires pour son fonctionnement et qui peuvent être le résultat d'une autre, elle prend comme entrée la position du véhicule extraire du GPS, et renvoie comme résultat le chemin le plus court ainsi que le numéro du parking vide le plus proche.

Le fonctionnement d'app est expliqué dans le chapitre II dans la figure II.20.

- **Simulation**

On prend comme exemple pour simuler la fonction app, la position (3→8), le résultat est:

Chemin= [3, 8, 3,5].

Et k3 =4 qui indique le numéro du parking le plus proche.

Cet exemple est montré dans la figure III.10.

La position de véhicule

Le chemin vers le parking n°4

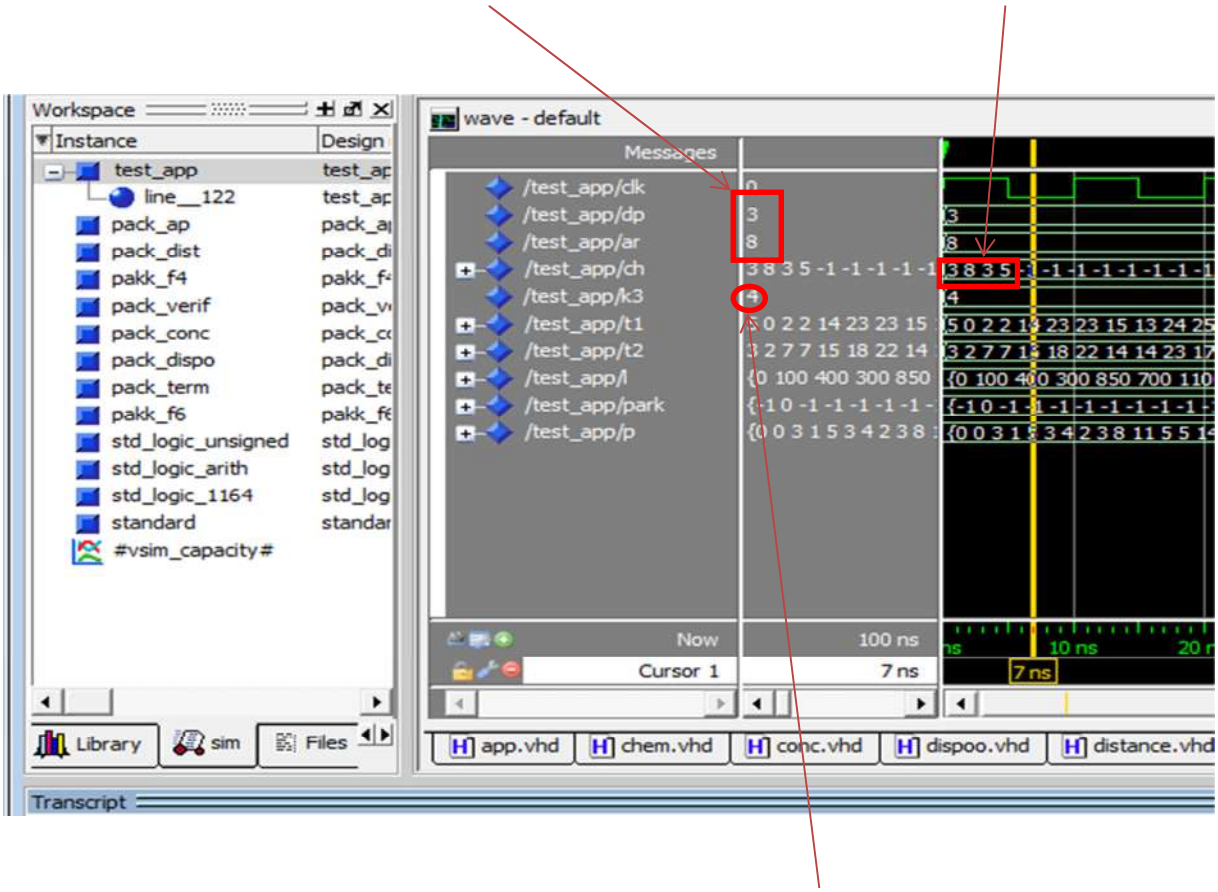
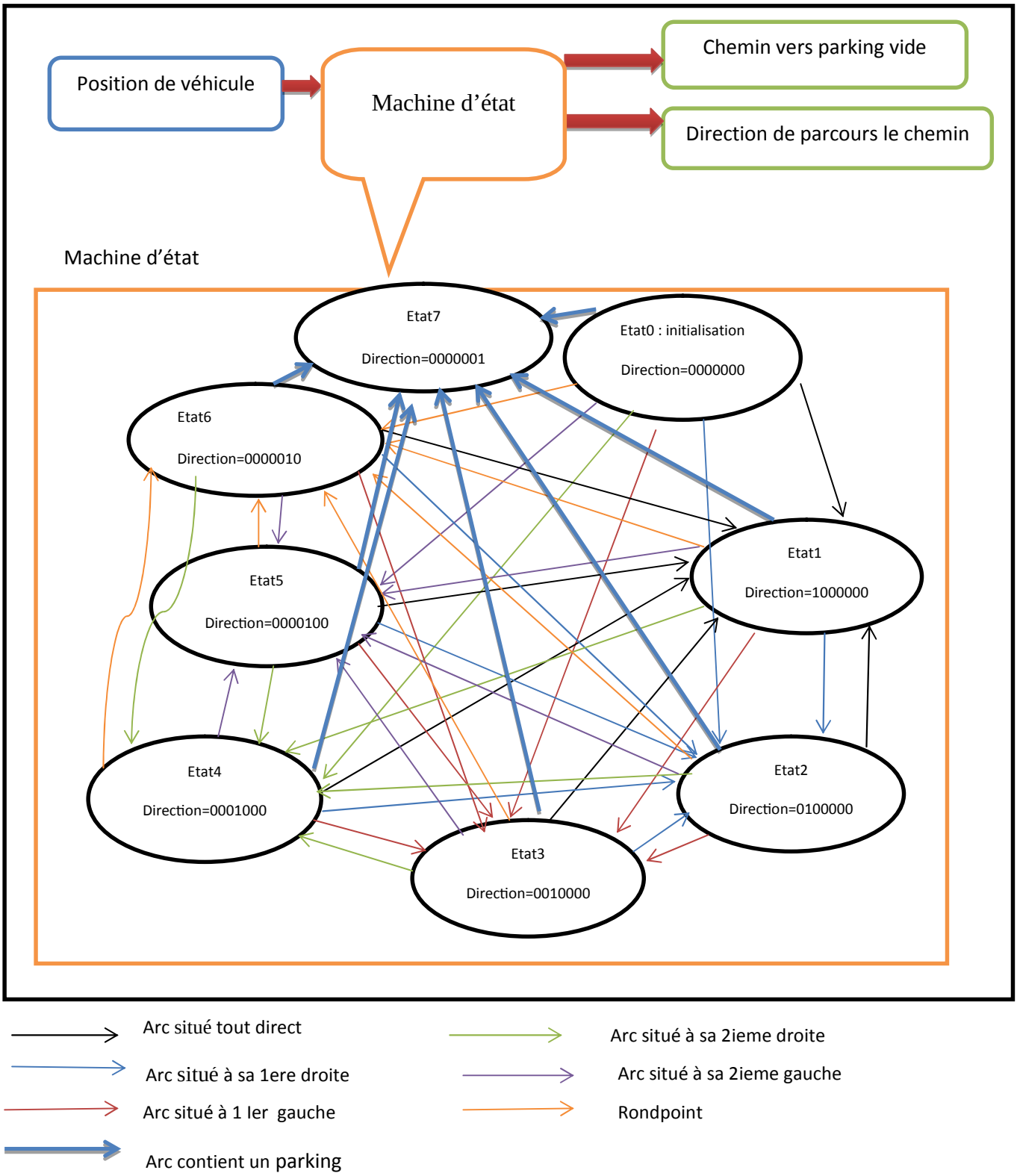


Figure III.10 : la simulation de la fonction app.

III.5.6 Machine d'état ou module contrôleur

Le déroulement de notre programme est illustré dans l'organigramme de machine d'état suivant :



figureIII.11 : schéma globale de la machine d'état.

Ou la palette supposé de diriger l'utilisateur est la suivante :

L'état 0 : pour l'initialisation et restauration du chemin.

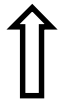
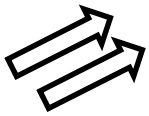
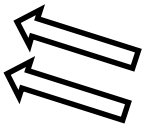
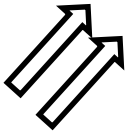
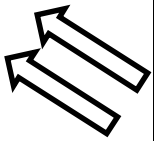
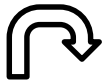
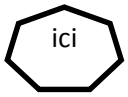
Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Table III.1 : palette correspondante à l'état0.

L'état 1 : rouler directement

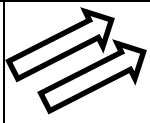
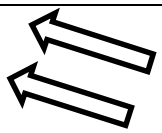
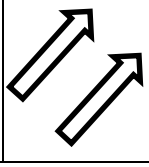
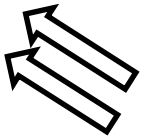
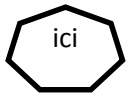
Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Table III.2 : palette correspondante à l'état1.

L'état 2 : tourner au 1ier virage à droite.

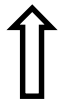
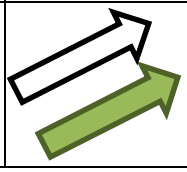
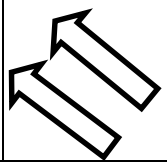
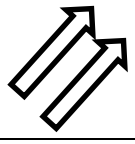
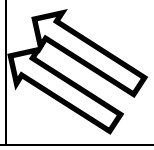
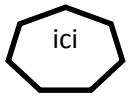
Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Table III.3 : palette correspondante à l'état2.

L'état 3 : tourner au 1ier virage à gauche.

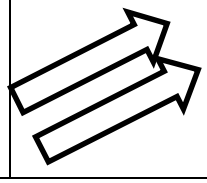
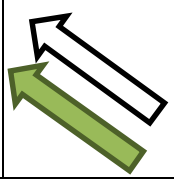
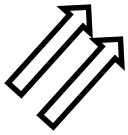
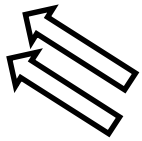
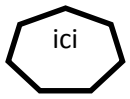
Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Table III.4 : palette correspondante à l'état3.

L'état 4 : tourner au 2ieme virage à droite.

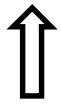
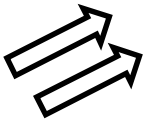
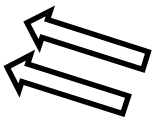
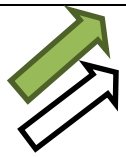
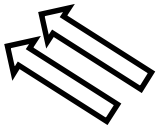
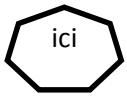
Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Table III.5 : palette correspondante à l'état4.

L'état 5 : tourner au 2ieme virage à gauche.

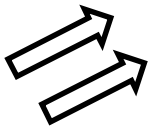
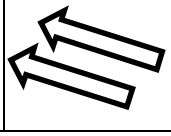
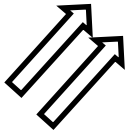
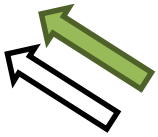
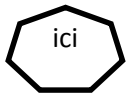
Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Table III.6 : palette correspondante à l'état5.

L'état 6 : faire le rond-point.

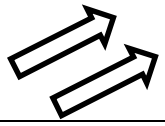
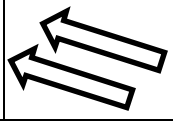
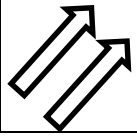
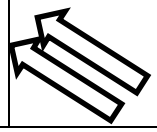
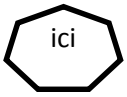
Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Table III.7 : palette correspondante à l'état6.

L'état 7 : Garer ici.

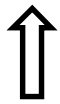
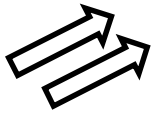
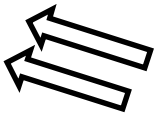
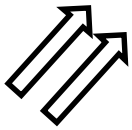
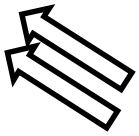
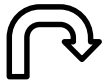
Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Table III.8 : palette correspondante à l'état7.

III.6 Tests et analyses

Scenario 1 :

-On propose comme position (9→10), dans ce cas le plus proche parking vide c'est 4→2 (le parking numéro 5) le chemin renvoyé par le système sera [9, 10, 11, 5,4 ,2].

-Mais quand le véhicule est dans la position (11→5) et avant l'arrivée au parking prévu, la place a été prise (information entrante indiquant le remplissage de la place) alors notre système propose un autre chemin vers un autre parking (parking N° 10) : chemin=[11,5,11,10].mais pendant que le véhicule parcours ce chemin, une information entrante du GPS indique la faute de direction au lieu de prendre l'arc (5→11) le conducteur a pris (5→3) , donc il est orienté vers un nouveau emplacement (parking N°2) :chemin=[5, 3, 2, 0].

La figure III.12 illustre les résultats de simulation de ce scénario.

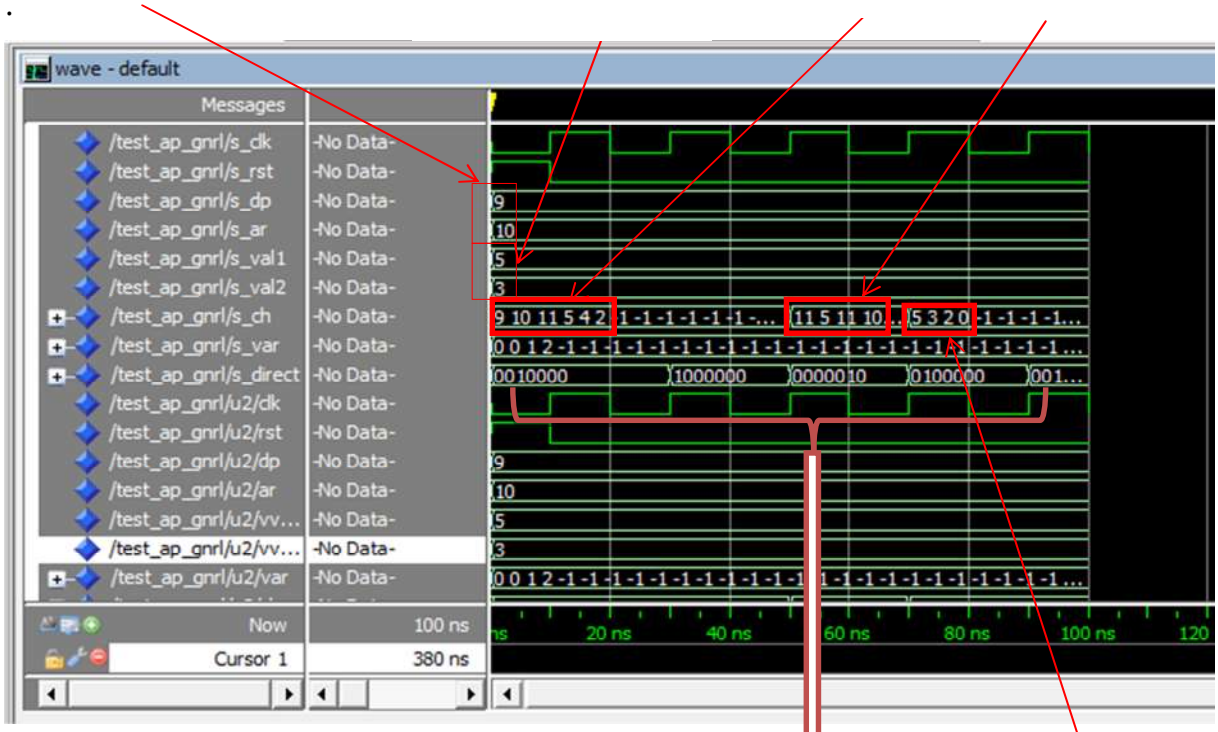
La position initiale du véhicule

Le chemin après le remplissage de la place

Les bornes de parking proposé

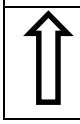
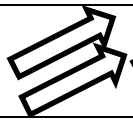
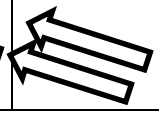
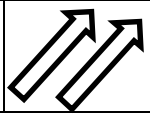
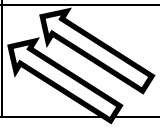
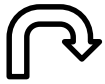
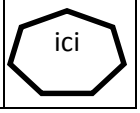
Le chemin à parcourir

.

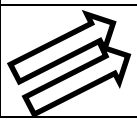
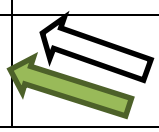
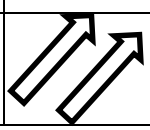
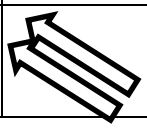
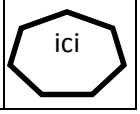


L'ensemble des directions indiquées par la palette est :

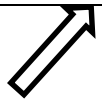
Le chemin envisagé après la faute de direction

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						 ici

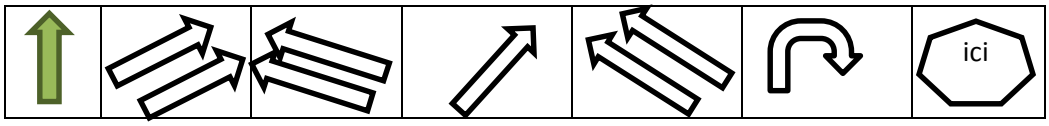
Etat0

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						 ici

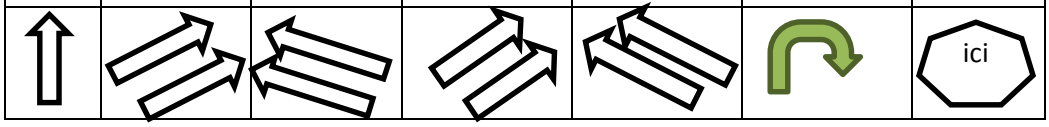
Etat3

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Etat1

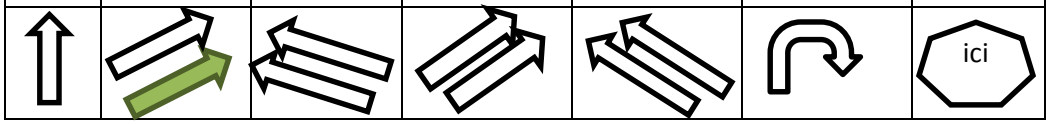


Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici



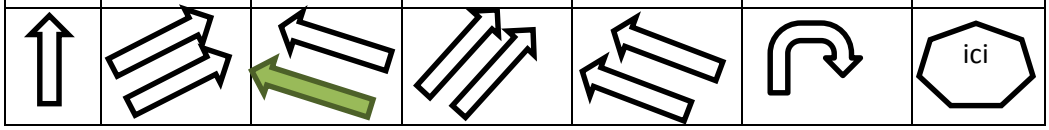
Etat6

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici



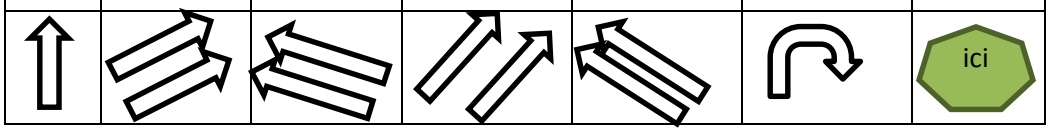
Etat2

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici



Etat3

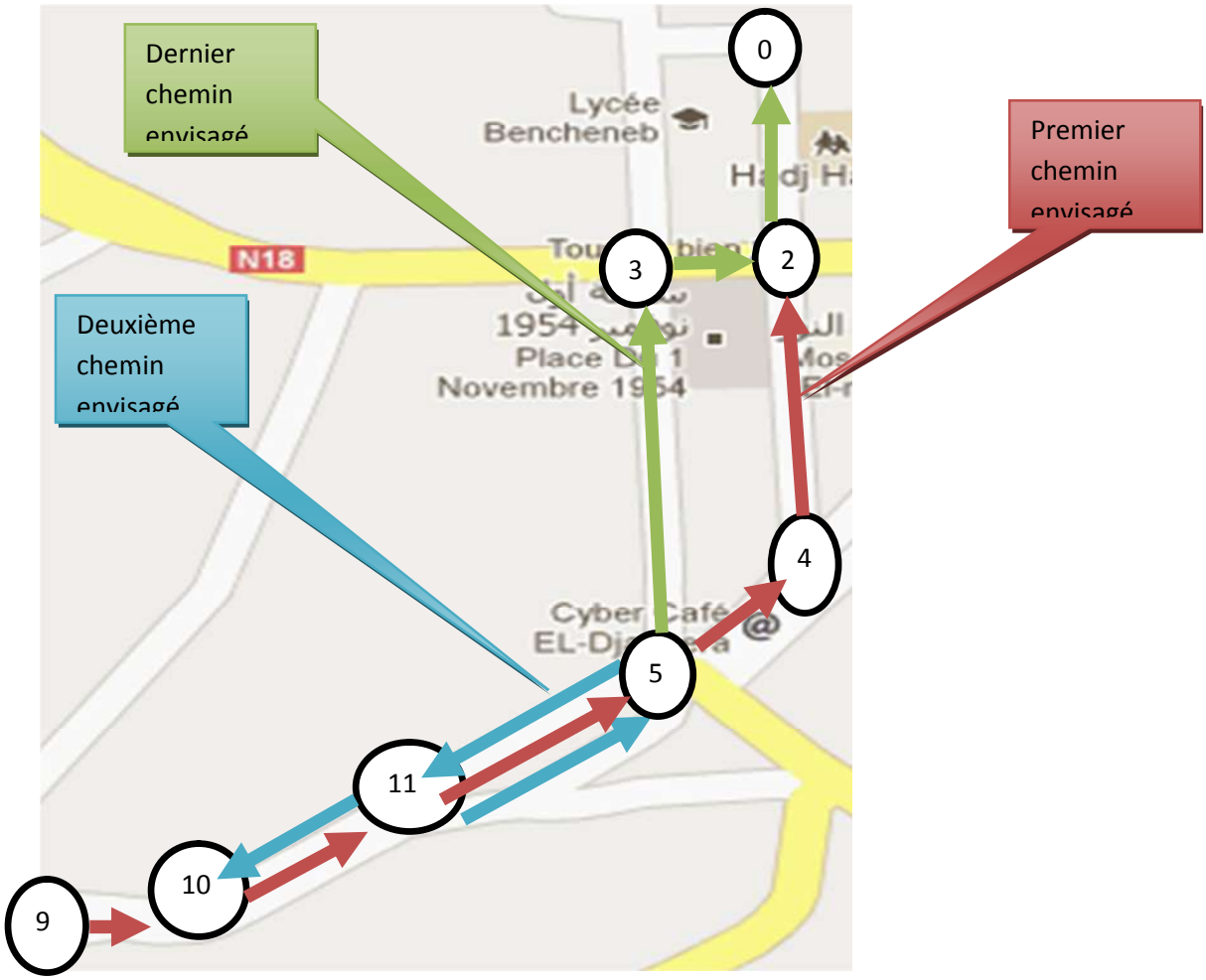
Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici



Etat7

La figureIII.12 : ensemble de directions renvoyé pour l'exemple 1.

La figure (III.13) offre plus de détails sur ce Scénario.



Figure(III.13) : détail du scenario1.

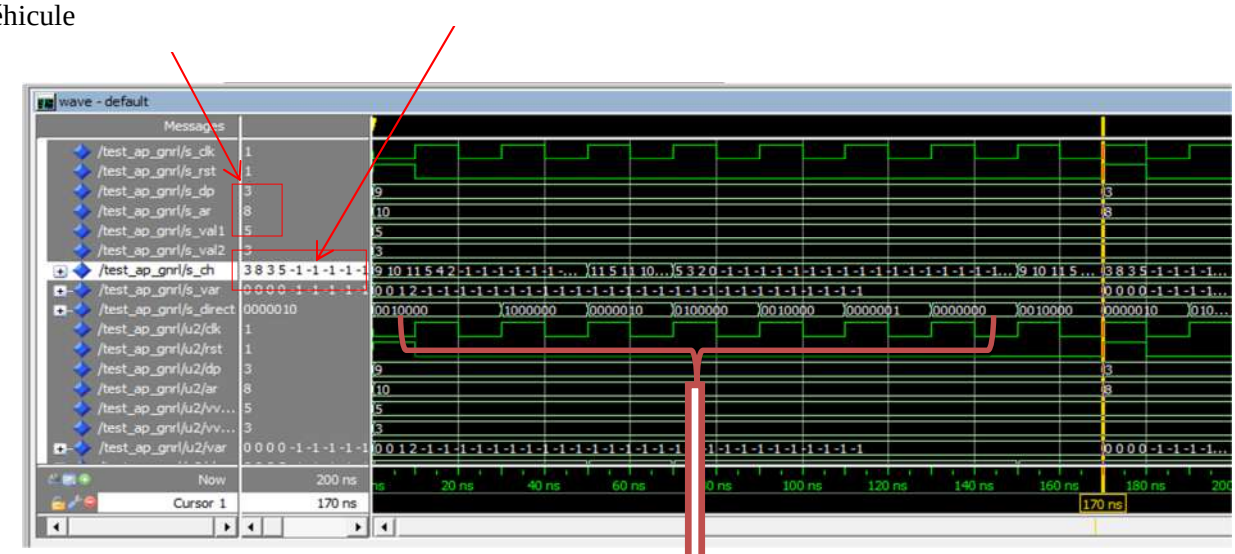
Scenario 2 :

Quand le véhicule est dans la position (3→ 8) et dans cet arc il n'existe aucun parking et le plus proche parking de cette position est celui numéro 4(situé dans l'arc 3→ 5), le véhicule fait un demi-tour à travers le rond-point correspondant au nœud 8, pour arriver à ce parking.

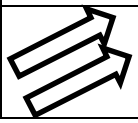
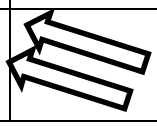
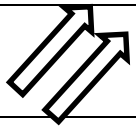
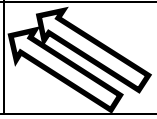
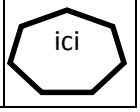
La figure III.14 montre le résultat de simulation de ce scenario.

La position initiale du véhicule

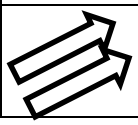
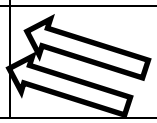
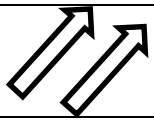
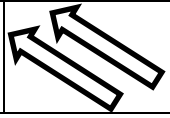
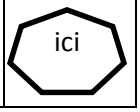
Le chemin à emprunter



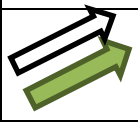
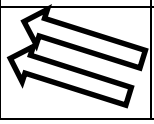
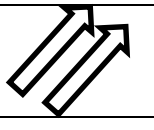
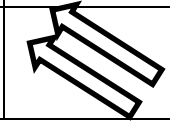
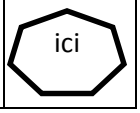
L'ensemble des directions à prendre fournies par la palette est :

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

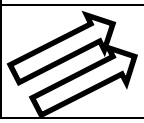
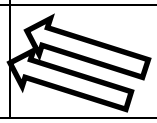
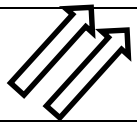
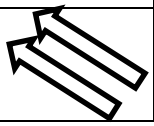
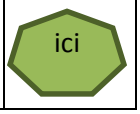
Etat0

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Etat6

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

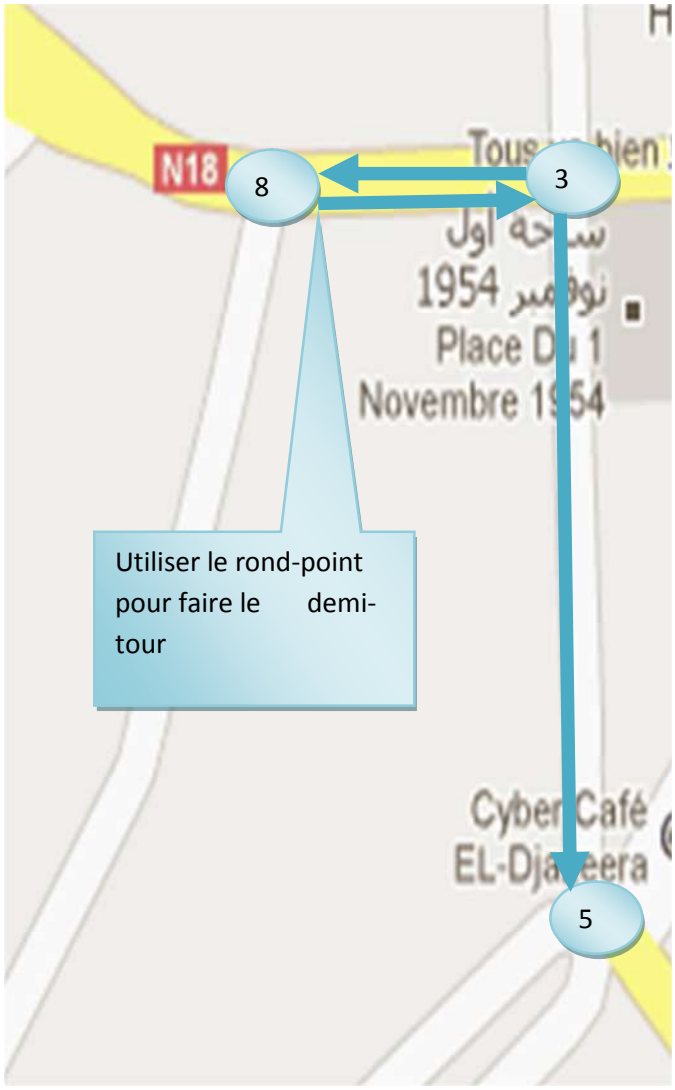
Etat2

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Etat7

La figure III.14 : ensemble des directions fournis pour l'exemple2.

La figure (III.15) offre plus de détails sur ce Scénario.



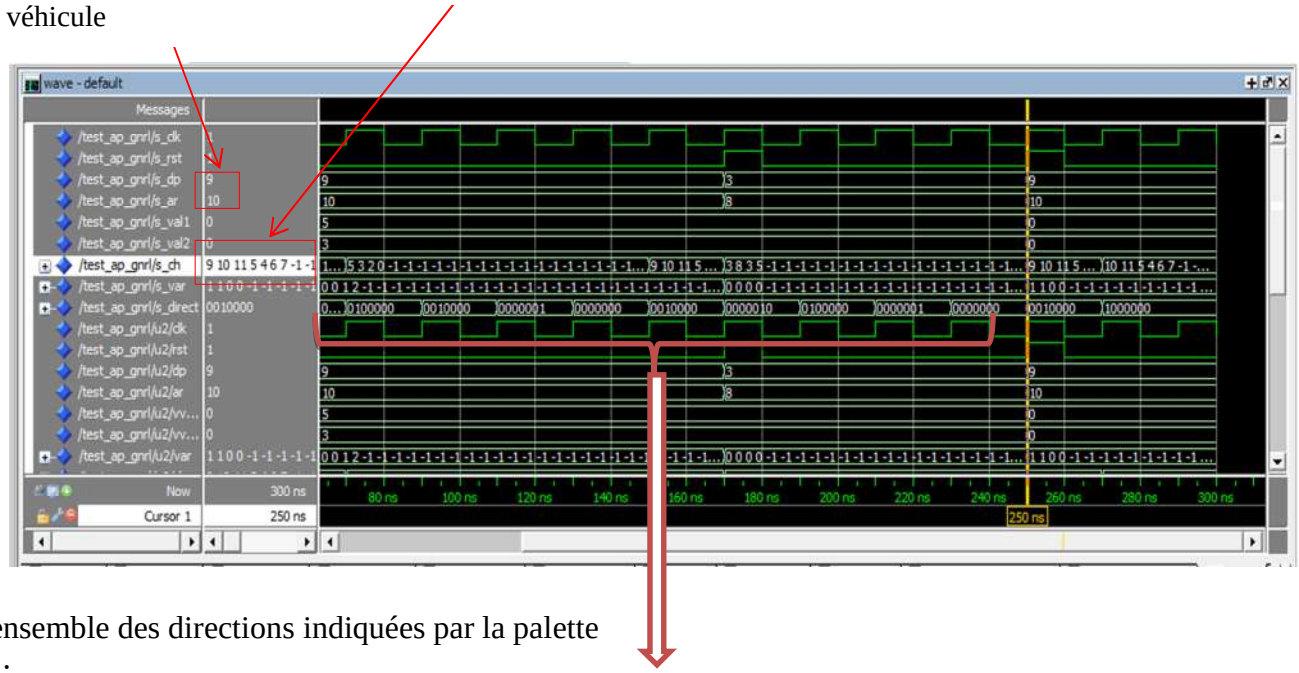
Figure(III.15) : détail du scenario2.

Scenario 3 :

Prenant comme exemple, un véhicule demandant une place pour se garer, ce dernier est situé dans la position (9→ 10),le système renvoie le chemin[9,10,11,5,4,6,7], et pendant le parcours de ce chemin et exactement quand cet automobile est situé à l'arc (9→10), une information entre au système indique le remplissage de la place vide, mais comme il y a plus d'une seule place vide l'acheminement ne change pas et ce véhicule est dirigé vers le même parking.

Le chemin proposé

La position de véhicule



L'ensemble des directions indiquées par la palette est :

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
↑	↱	↰	↱	↰	↻	ici

Etat0

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
↑	↱	↰	↱	↰	↻	ici

Etat3

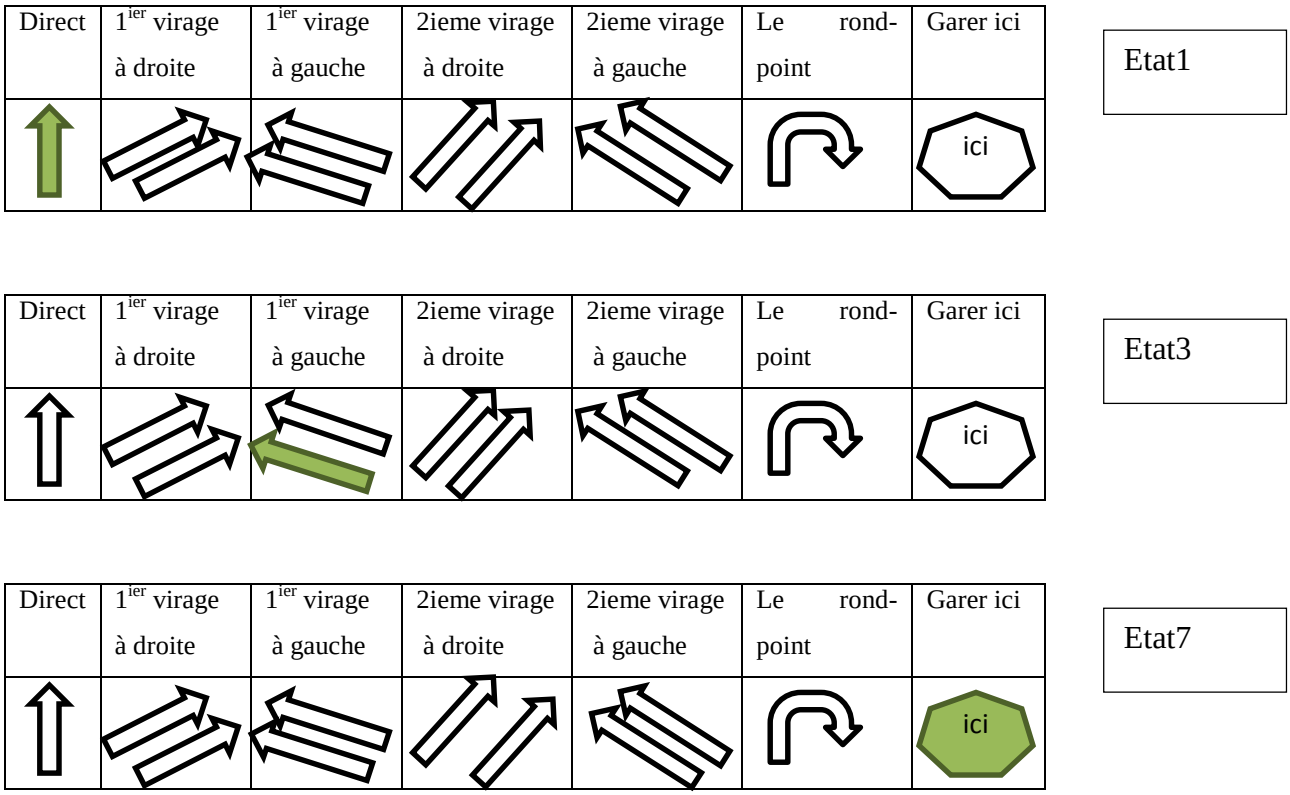
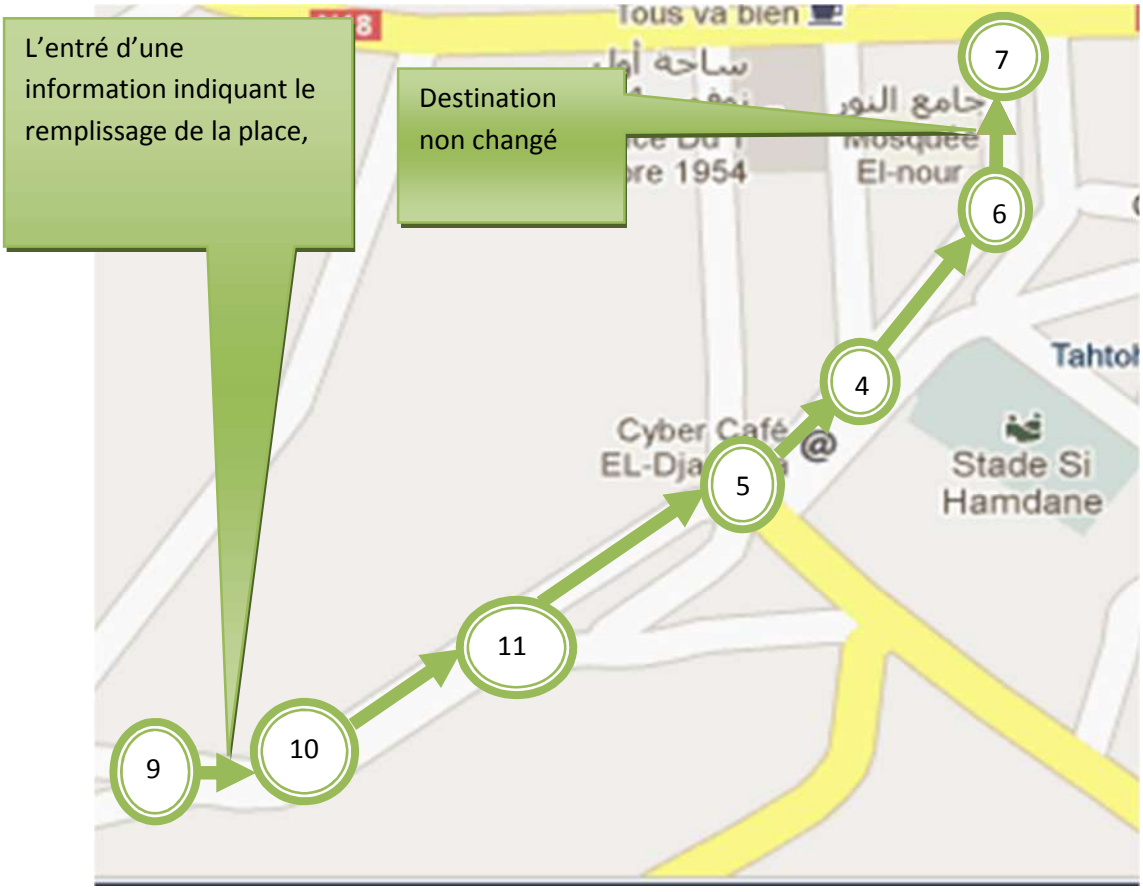


Figure III.16 :l'ensemble des directions fournis pour l'exemple3.

La figure (III.17) offre plus de détails sur ce Scénario



Figure(III.17) : détail du scenario3.

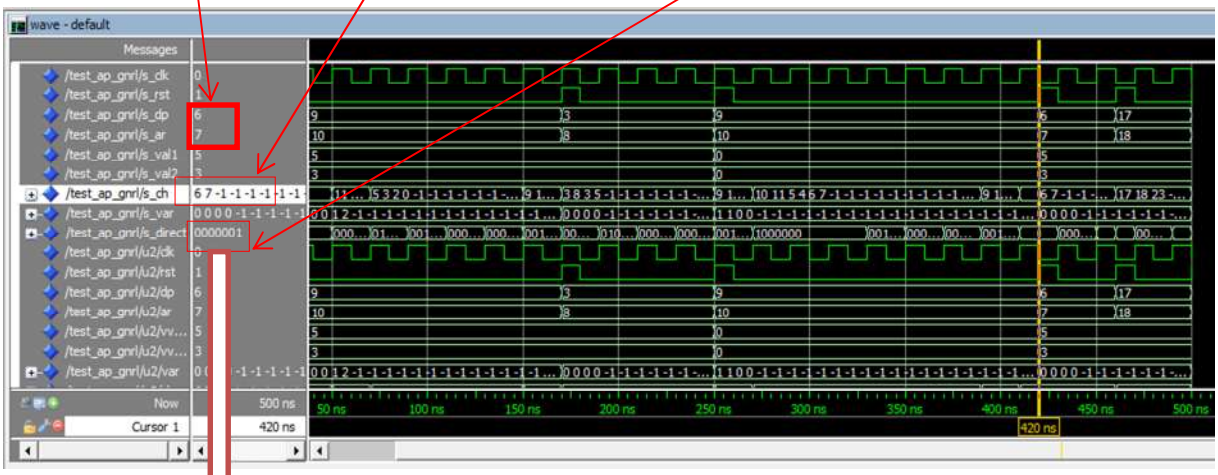
Scenario 4 :

La figure III.18 montre le résultat de simulation pour un exemple d'un véhicule situé dans un arc contenant un parking vide .Prenant la position dans l'arc 32 (6→7), le système indique l'existence d'une place vide.

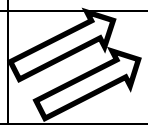
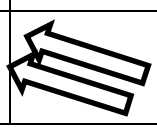
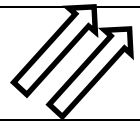
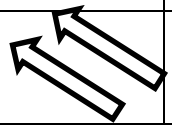
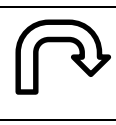
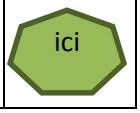
Le chemin à emprunter

La position initiale du véhicule

garez ici



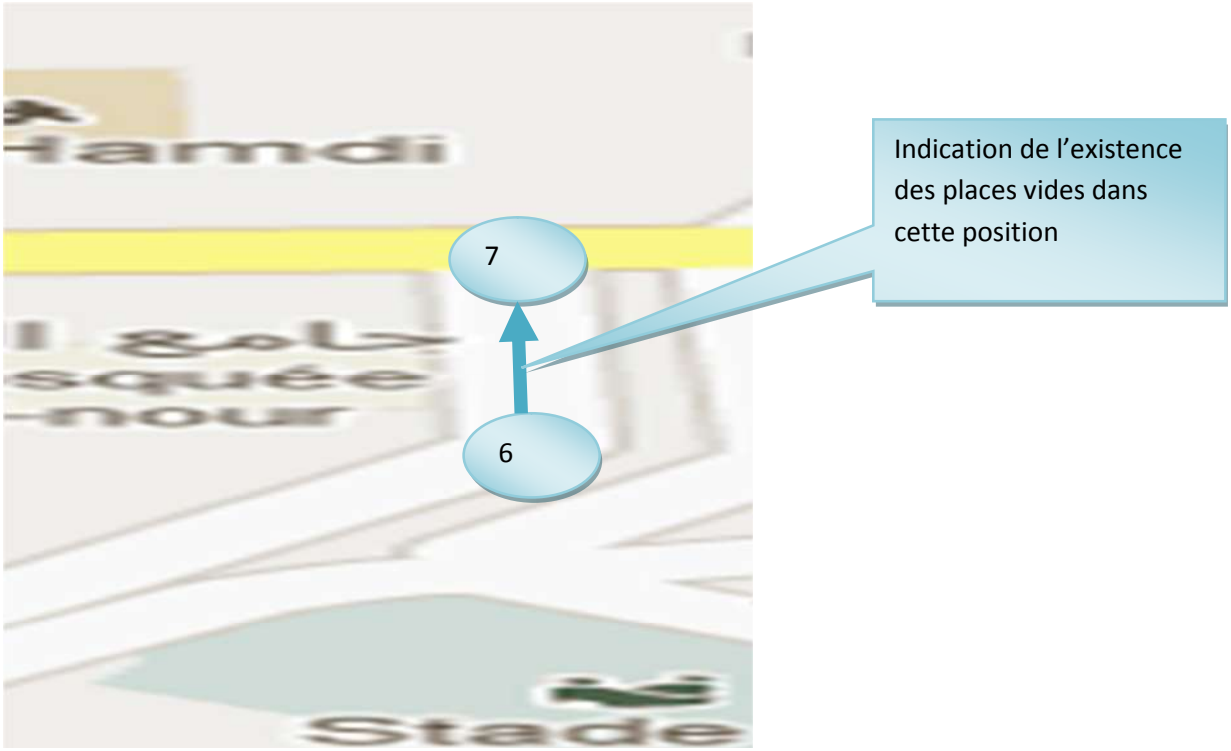
L'ensemble des directions fournis par la palette est :

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
						

Etat7

Figure III.18 : l'ensemble des directions fournis pour l'exemple 4.

La figure (III.19) offre plus de détails sur ce Scénario.



Figure(III.19) : détail du scenario 4.

Scenario 5

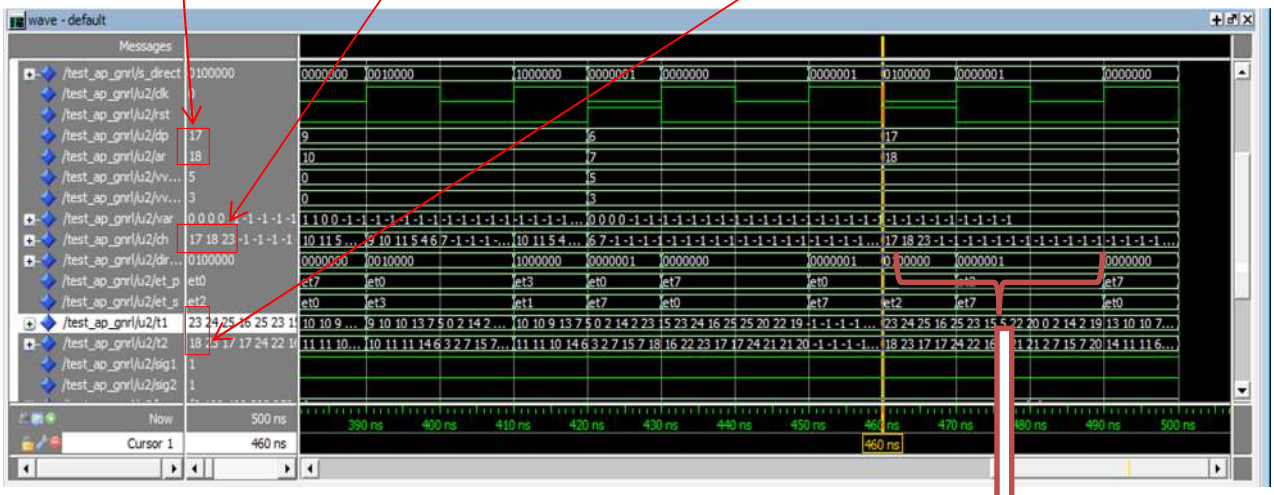
Prenant comme exemple un véhicule qui se situe dans la position (17→18), ou cette position contient un espace de stationnement légal dans le sens opposé (18→17), le système l'oriente vers une autre place vide afin d'éviter un demi-tour interdit.

Ce cas est simulé et expliqué par la figure III.20.

La position initiale du véhicule

Le chemin à emprunter

Le plus proche parking



L'ensemble des directions fournis par la palette est :

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
↑	↱	↰	↱	↰	↻	ici

Etat0

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
↑	↱	↰	↱	↰	↻	ici

Etat2

Direct	1ier virage à droite	1ier virage à gauche	2ieme virage à droite	2ieme virage à gauche	Le rond-point	Garer ici
↑	↱	↰	↱	↰	↻	ici

Etat7

Figure III.20: l'ensemble des directions fournis pour l'exemple5.

La figure (III.21) offre plus de détails sur ce Scénario.

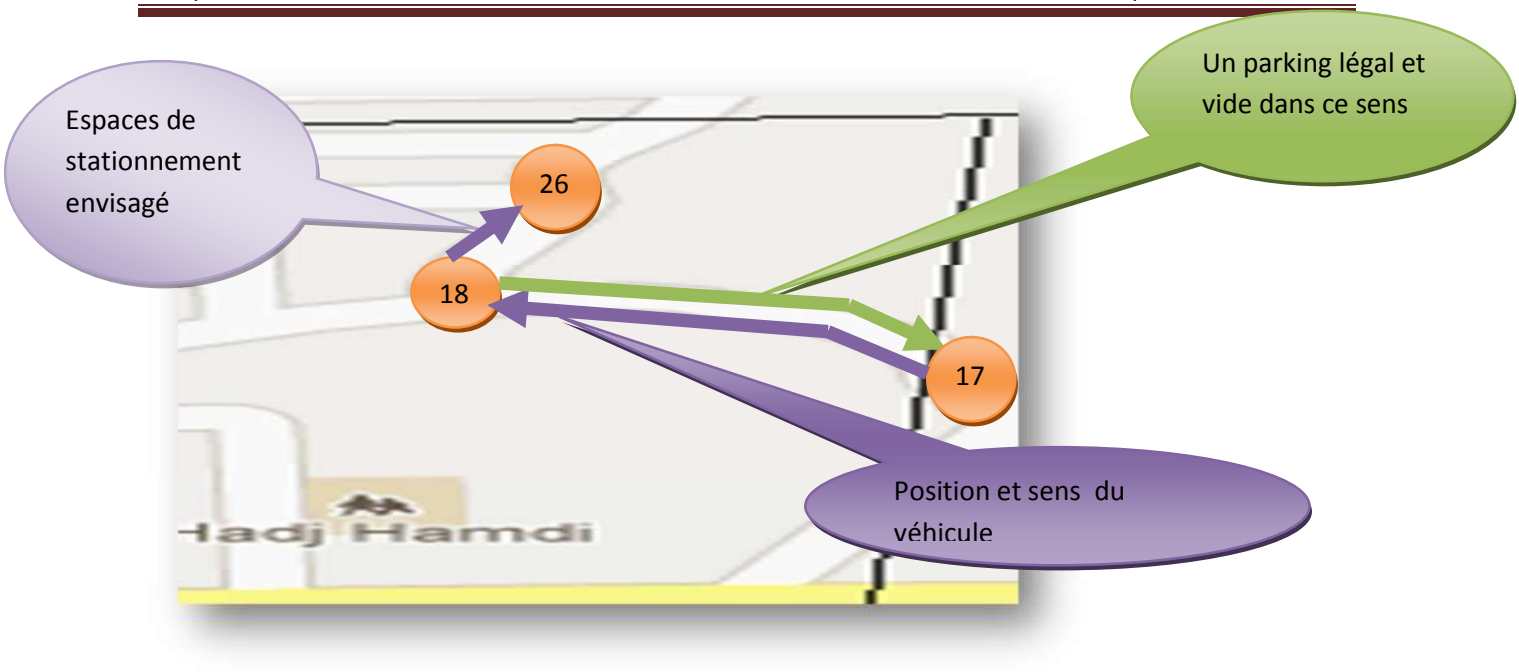


Figure III.21 : détail du scenario5.

III.7 Conclusion

La modélisation mathématique du centre-ville de Médéa, a été adaptée pour une implémentation VHDL, et traité à travers un ensemble de modules, chacun assurant un ensemble de taches bien précises ou une architecture digitale a peut-être développée qui présente plusieurs avantages :

- Modulable : adaptable à n'importe qu'elle parking en ville il suffit de changer les mémoires.
- Intelligent : prend en compte l'erreur du conducteur dans la prise du chemin ainsi que la mise à jour des listes des places de parking vide et orienter ainsi le conducteur vers un autre emplacement
- Assure le respect le code de la route : prise en compte des sens interdits, rond-point et code de la route.

Conclusion générale

Conclusion générale

La solution proposée ou « SmartPark » vise à optimiser la gestion du stationnement en parc et sur voirie. Fondées sur l'utilisation des détecteurs de véhicules sans fil installés au sol sur les places, et sur le système de guidage à la place, ainsi que de poursuite en temps réel. Des capteurs intelligents détectent l'état de disponibilité de chaque place, et communiquent cette information au serveur en temps réel. Les afficheurs renseignent alors les automobilistes sur la nouvelle direction à prendre en tenant compte du nouvel état de la disponibilité de la place de parking.

Le modèle offre aussi un système poursuite du véhicule dès qu'il demande une place pour se garer et en cas de faute de direction, ce système reçoit une information sur sa nouvelle position, et le modèle s'occupe de le guider vers une place de parking en tenant compte de sa nouvelle position.

La solution proposée revisite le stationnement et révolutionne cette composante essentielle des politiques de mobilité en offrant aux conducteurs, une palette d'outils de guidage et de poursuite, d'analyse et de contrôle. Des capteurs autonomes installés sur chaque place détectent leur état de disponibilité, et fournissent un panorama temps réel de l'occupation des places.

Un système de guidage vers les places libres et réduction du trafic inutile et de manœuvre illégale et dangereux, optimisation du contrôle et augmentation de la rotation, observateur du stationnement permanent.

Le travail contient des imperfections, dus au temps limités. Ces imperfections se résument dans finalisation du travail, comme l'implémentation matérielle, l'application expérimentale, le traitement des autres types de parking (parking multi-étages, parking sous-sol), et la mise en place d'un système de payement.

bibliographie

Bibliographies

[1] : Encyclopédie des inventions et inventeurs, url:http://www.invention-europe.com /CMpro-v-p-225.htm, dernière date de consultation: 21/06/2012.

[2] : les routes, les ponts et les parcs de stationnement, parcs de stationnement couverts .techniques d'ingénieur. Auteur : Jean Claude, responsable sécurité incendie au centre d'assistance technique et de documentation (CATED).

[3]: site web: parking-Paris. Lien: http://parking-paris.net/parking-paris/les-differents-types-de-parking-paris.html. Dernière date de consultation :10/06/2012.

[4] : utilisation d'internet dans la gestion du trafic. Auteur : Jean Christophe Roy VA transport. Promotion 54, 29/06/1990.

[5] : Site du parking de la ville de San Francisco, url:http://sfpark.org/, dernière date de consultation: 24/06/2012.

[6]: site web: www.astrosurf.com/luxorion/gps.html. Dernière consultation:15/03/2012.

[7] : Site web des cours : cours des graphes. lien : http://www.infeig.unige.ch/~albuquer/Portfolio_07/SiteAlgo/Cours/Graphes/graphes/node16.html

Auteur : ALBUQUERQUE PAUL EIG prof ,17/07/2002.

[8] : site web :http://cermics.enpc.fr/polys/oap/node77.html. R.Lalement, Octobre 2000.

[9] : Architectures-reconfigurable-FPGA, technique d'ingénieur. Dernière date de consultation: 13/06/2012.

[10] : UTILISATION D'INTERNET DANS LA GESTION DU TRAFIC. Auteur : Jean-Christophe ROY, 29 Juin 2009.

Les annexes

Annexe A:

L'ensemble des matrices précédement citées :

1. La matrice des distances initiales (notée : MDI)

La (MDI) est donnée par :

```
[0 100 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf 0 inf 200 inf inf inf inf inf inf inf inf inf inf inf inf inf inf 50 inf inf inf inf inf inf inf inf inf inf inf
200 inf 0 100 inf inf inf 100 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf 200 100 0 inf 400 inf inf 150 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf 250 inf 0 150 250 inf inf inf inf inf inf 100 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf 400 150 0 inf inf inf inf inf 200 150 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf 0 150 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf 100 inf inf inf inf 0 inf inf inf inf inf inf inf 50 inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf 150 inf inf inf inf 1 700 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf 700 0 200 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf 200 0 200 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf 200 inf inf inf inf 200 0 100 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf 150 inf inf inf inf inf 100 0 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf 150 inf inf inf inf inf inf inf inf 0 100 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf 100 0 150 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf 50 inf inf inf inf inf inf 150 0 250 inf inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 200 0 250 inf inf inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 250 0 100 inf inf inf inf inf inf 100 inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 100 0 100 inf inf inf 100 inf inf inf inf inf inf
inf 50 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 100 0 inf inf 110 inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 60 0 inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 140 0 60 inf inf inf inf inf inf 110
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 110 inf inf 0 60 inf inf inf inf 60 inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 100 inf inf inf 60 0 50 inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 50 0 100 inf 60 inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 100 inf inf inf inf inf inf 100 0 50 inf inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 50 0 100 inf inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 50 inf 100 0 100 inf inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 60 inf inf inf inf 100 0 50 inf
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 50 0 50
inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 110 inf inf inf inf inf inf 50 0];
```

2. La matrice des distances minimales (notée : MDM)

La (MDM) est donnée par

```
[0 100 400 300 850 700 1100 500 450 1150 1100 900 850 800 700 550 600 350 250 150 670 530 260 320 370 450 500 420 320 370 420
500 0 300 200 750 600 1000 400 350 1050 1000 800 750 700 600 450 500 250 150 50 570 430 160 220 270 350 400 320 220 270 320
200 300 0 100 550 500 800 100 250 950 900 700 650 400 300 150 400 550 450 350 870 730 460 520 570 650 700 620 520 570 620
300 200 100 0 550 400 800 200 150 850 800 600 550 500 400 250 500 450 350 250 770 630 360 420 470 550 600 520 420 470 520
450 550 250 350 0 150 250 350 500 750 550 350 300 100 200 350 600 800 700 600 1120 980 710 770 820 900 950 870 770 820 870
600 600 400 400 150 0 400 500 550 600 400 200 150 250 350 500 750 850 750 650 1170 1030 760 820 870 950 1000 920 820 870 920
450 550 250 350 600 750 0 150 500 1200 1150 950 900 450 350 200 450 700 700 600 1120 980 710 770 820 800 850 870 770 820 870
300 400 100 200 450 600 700 0 350 1050 1000 800 750 300 200 50 300 550 550 450 970 830 560 620 670 650 700 720 620 670 720
450 350 250 150 700 550 950 350 1 700 900 750 700 650 550 400 650 600 500 400 920 780 510 570 620 700 750 670 570 620 670
1150 1050 950 850 750 600 1000 1050 700 0 200 400 500 850 950 1100 1350 1300 1200 1100 1620 1480 1210 1270 1320 1400 1450 1370 1270 1320 1370
1000 1000 800 800 550 400 800 900 900 200 0 200 300 650 750 900 1150 1250 1150 1050 1570 1430 1160 1220 1270 1350 1400 1320 1220 1270 1320
800 800 600 600 350 200 600 700 750 400 200 0 100 450 550 700 950 1050 950 850 1370 1230 960 1020 1070 1150 1200 1120 1020 1070 1120
750 750 550 550 300 150 550 650 700 500 300 100 0 400 500 650 900 1000 900 800 1320 1180 910 970 1020 1100 1150 1070 970 1020 1070
600 700 400 500 150 300 400 300 650 900 700 500 450 0 100 250 500 750 850 750 1270 1130 860 920 950 850 900 1000 920 970 1020
500 600 300 400 250 400 500 200 550 1000 800 600 550 100 0 150 400 650 750 650 1170 1030 760 820 850 750 800 900 820 870 920
350 450 150 250 400 550 650 50 400 1100 950 750 700 250 150 0 250 500 600 500 1020 880 610 670 700 600 650 750 670 720 770
550 500 350 450 600 750 850 250 600 1300 1150 950 900 450 350 200 0 250 350 450 920 780 510 450 450 350 400 500  570 620 670
750 250 550 450 850 850 1100 500 600 1300 1250 1050 1000 700 600 450 250 0 100 200 670 530 260 200 200 100 150 250 320 370 420
650 150 450 350 900 750 1150 550 500 1200 1150 950 900 800 700 550 350 100 0 100 570 430 160 100 150 200 250 210 220 270 320
550 50 350 250 800 650 1050 450 400 1100 1050 850 800 750 650 500 450 200 100 0 520 380 110 170 220 300 350 270 170 220 270
```

610 110 410 310 860 710 1110 510 460 1160 1110 910 860 810 710 560 510 260 160 60 0 440 170 230 280 360 410 330 230 280 330
720 220 520 420 970 820 1220 620 570 1270 1220 1020 970 920 820 670 570 320 220 170 140 0 60 120 170 270 320 220 120 160 110
660 160 460 360 910 760 1160 560 510 1210 1160 960 910 860 760 610 510 260 160 110 410 270 0 60 110 210 260 160 60 110 160
720 220 520 420 970 820 1220 620 570 1270 1220 1020 970 900 800 650 450 200 100 170 470 330 60 0 50 150 200 110 120 170 220
770 270 570 470 1020 870 1270 670 620 1320 1270 1070 1020 900 800 650 450 200 150 220 510 370 110 50 0 100 150 60 160 210 260
850 350 650 550 950 950 1200 600 700 1400 1350 1150 1100 800 700 550 350 100 200 300 600 460 210 150 100 0 50 150 250 300 350
900 400 700 600 1000 1000 1250 650 750 1450 1400 1200 1150 850 750 600 400 150 250 350 550 410 260 200 150 50 0 100 200 250 300
820 320 620 520 1070 920 1320 720 670 1370 1320 1120 1070 950 850 700 500 250 200 270 450 310 160 100 50 150 100 0 100 150 200
720 220 520 420 970 820 1220 620 570 1270 1220 1020 970 920 820 670 570 320 220 170 350 210 60 120 150 250 200 100 0 50 100
770 270 570 470 1020 870 1270 670 620 1320 1270 1070 1020 970 870 720 620 370 270 220 300 160 110 170 200 300 250 150 50 0 50
820 320 620 520 1070 920 1320 720 670 1370 1320 1120 1070 1020 920 770 670 420 320 270 250 110 160 220 250 350 300 200 100 50 0];

3. La matrice de marquage :MMrq

La MMrq est donnée par :

[0 0 3 1 5 3 4 2 3 8 11 5 5 14 15 7 17 18 19 1 21 30 19 22 23 17 25 28 22 28 30
2 1 3 1 5 3 4 2 3 8 11 5 5 14 15 7 17 18 19 1 21 30 19 22 23 17 25 28 22 28 30
2 0 2 2 13 3 4 2 3 8 11 5 5 14 15 7 15 18 19 1 21 30 19 22 23 17 25 28 22 28 30
2 3 3 3 5 3 4 2 3 8 11 5 5 14 15 7 15 18 19 1 21 30 19 22 23 17 25 28 22 28 30
2 0 4 2 4 4 4 2 3 10 11 5 5 4 13 14 15 18 19 1 21 30 19 22 23 17 25 28 22 28 30
2 3 4 5 5 5 4 2 3 10 11 5 5 4 13 14 15 18 19 1 21 30 19 22 23 17 25 28 22 28 30
2 0 7 2 13 3 6 6 3 8 11 5 5 14 15 7 15 16 19 1 21 30 19 22 23 17 25 28 22 28 30
2 0 7 2 13 3 4 7 3 8 11 5 5 14 15 7 15 16 19 1 21 30 19 22 23 17 25 28 22 28 30
2 3 3 8 5 3 4 2 8 8 9 5 5 14 15 7 15 18 19 1 21 30 19 22 23 17 25 28 22 28 30
2 3 3 8 5 11 4 2 9 9 9 10 11 4 13 7 15 18 19 1 21 30 19 22 23 17 25 28 22 28 30
2 3 4 5 5 11 4 2 9 10 10 10 11 4 13 14 15 18 19 1 21 30 19 22 23 17 25 28 22 28 30
2 3 4 5 5 11 4 2 3 10 11 11 11 4 13 14 15 18 19 1 21 30 19 22 23 17 25 28 22 28 30
2 3 4 5 5 12 4 2 3 10 11 5 12 4 13 14 15 18 19 1 21 30 19 22 23 17 25 28 22 28 30
2 0 4 2 13 4 4 15 3 10 11 5 5 13 13 14 15 16 17 1 21 30 19 22 25 17 25 26 22 28 29
2 0 7 2 13 4 4 15 3 10 11 5 5 14 14 14 15 16 17 1 21 30 19 22 25 17 25 26 22 28 30
2 0 7 2 13 4 4 15 3 8 11 5 5 14 15 15 15 16 17 1 21 30 19 22 25 17 25 26 22 28 30
2 19 7 2 13 4 4 15 3 8 11 5 5 14 15 16 16 16 17 18 21 30 23 18 25 17 25 26 22 28 30
2 19 3 1 13 3 4 15 3 8 11 5 5 14 15 16 17 17 17 18 21 30 23 18 25 17 25 26 22 28 30
2 19 3 1 5 3 4 2 3 8 11 5 5 14 15 16 17 18 18 18 21 30 23 18 23 17 25 24 22 28 30
2 19 3 1 5 3 4 2 3 8 11 5 5 14 15 7 17 18 19 19 21 30 19 22 23 17 25 28 22 28 30
2 19 3 1 5 3 4 2 3 8 11 5 5 14 15 7 17 18 19 20 20 30 19 22 23 17 25 28 22 28 30
2 19 3 1 5 3 4 2 3 8 11 5 5 14 15 7 17 18 23 22 21 21 21 22 23 24 25 28 22 30 22
2 19 3 1 5 3 4 2 3 8 11 5 5 14 15 7 17 18 23 22 21 30 22 22 23 24 25 28 22 28 30
2 19 3 1 5 3 4 2 3 8 11 5 5 14 15 16 17 18 23 22 21 30 23 23 23 24 25 24 22 28 30
2 19 3 1 5 3 4 2 3 8 11 5 5 14 15 16 17 25 23 22 21 30 23 24 24 24 25 24 27 28 30
2 19 3 1 13 3 4 15 3 8 11 5 5 14 15 16 17 25 17 18 21 30 23 24 25 25 25 26 27 28 30
2 19 3 1 13 3 4 15 3 8 11 5 5 14 15 16 17 25 17 18 21 30 23 24 25 26 26 26 27 28 30

2 19 3 1 5 3 4 2 3 8 11 5 5 14 15 16 17 25 23 22 21 30 23 24 27 24 27 27 27 28 30

2 19 3 1 5 3 4 2 3 8 11 5 5 14 15 7 17 18 23 22 21 30 28 22 27 24 27 28 28 28 30

2 19 3 1 5 3 4 2 3 8 11 5 5 14 15 7 17 18 23 22 21 30 28 22 27 24 27 28 29 29 30

2 19 3 1 5 3 4 2 3 8 11 5 5 14 15 7 17 18 23 22 21 30 28 22 27 24 27 28 29 30 31]

La matrice des arcs (notée : MAC)

La (MAC) est donnée par :

[nan 0 nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan 1 nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan 34 nan nan nan nan nan nan nan nan nan nan nan 2 nan nan 39 nan nan nan 52 nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan

nan nan 6 nan nan nan

'1'1'1'1'1'1'1'1'1'1'0'0'0

[(et0,et3,et0,et0,et0

t0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et2,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0),
(et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et3,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0

,et0,et0,et0,et0,et0,et0,et0),
(et0,et0,et0,et0,et0,et0,

(et0,et0,et0,

t0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et1,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0),

**(et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,et0,

Annexe B

On trouve dans cette annexe, les entetes des circuits, procédures et fonctions utilsés

1. Le module examinateur de disponibilité (disponible) :

Procedure dispo (variable vect:in std_logic_vector; variable kk, vv:out integer range 0 to 100);

Ou :

Vect : correspond à une ligne de la matrice MES.

vv : variable qui prend la valeur 1 s'il existe une place vide dans ce parking, et 0 sinon.

Kk : variable indiquand la position de la place vide.

2. Le module identificateur du point d'arrivé (term) :

procedure term (variable ii:in integer range 0 to 100;signal park:in matrice;variable k1,k2:out integer range 0 to 100);

Ou :

ii : correspond au numéro du parking.

Park : la matrice MPK.

K1,k2 : les deux nœuds correspondant aux éxtrémités du parking.

3. Le module identificateur de l'arc correspondant au plus proche parking vide (distance)

procedure dist(signal P:in matrice; variable dp,ar:in integer range 0 to 100; variable tab2,tabb:inout integer_vector; signal l:in matrice;variable init:in integer;signal park:in matrice ;variable chh,ter:out integer range 0 to 100;variable vk3:out integer range 0 to 100);

Ou :

p : est la MMrq.

Dp, ar : variable indiquant la position et le sens du véhicule.

Tab2, tabb : tableau contenant les nœuds correspondants aux éxtrémités des parkings contenants des espaces vides.

L : est la MDM.

Init : variable qui désigne le point de départ.

Park : est la MPK.

Chh,ter : les nœuds correspondants aux éxtrémités du parking choisi comme déstination.

Vk3 : numéro du parking choisi.

4. Le module de restauration du chemin :

procedure chem(variable init,term:in integer range 0 to 100;signal p:in matrice;variable ch:out integer_vector);

ou :

init : variable qui indique le nœud de départ.

term : variable qui indique le nœud d'arrivé.

p :la MMrq.

Ch : ensemble des nœuds à parcourir en vue de partir du (init) et arriver à (term).

5. le module superviseur de collaboration (app) :

procedure ap(variable dp,ar: in integer range 0 to 100;signal park:in matrice ; variable z:in matrice1;signal l :in matrice;signal p:in matrice ; variable ch:out integer_vector; variable k3: out integer range 0 to 100);

Ou :

Dp, ar : variable indiquant la position et le sens du véhicule.

Park : est la MPK.

Z : est la MES.

L : est la MDM.

p : est la MMrq.

Ch : l'ensemble des nœuds à parcourir afin d'arriver à l'espace de stationnement choisi.

k3 : numéro du parking choisi.

6. machine d'état ou module contrôleur

Ce module correspond au circuit principal, dont l'entête est le suivant :

```
entity ap_general is
   port (clk,rst:in std_logic;
        dp,ar,vval1,vval2:in integer range 0 to 100;
        var:in integer_vector;
         ch:out integer_vector;
         direct:out std_logic_vector(0 to 6)
           );
```

end ap_general;

Ou :

Clk : l’entré d’orloge.

Rst : signale de redimarage.

Dp, ar : la position et le sens du véhicule.

Var : ensemble d’information sur la disponibilité de la place et la prise de la bonne direction (venant du GPS)

Ch : l’ensemble des nœuds à parcourir afin d’arriver à l’espace de stationnement choisi.

Direct : ensemble des directions à prendre.